Generative deep learning foster pdf online test free online

Gradually, each unit becomes skilled at iden- tifying a particular feature that ultimately helps the network to make better predictions. Some of these photos are shown in Figure 4-1. Next we flatten this input into a vector, using a Flatten layer. Dense layer of one unit, with a sigmoid activation function that transforms the output from the dense layer to the range [0, 1]. If we apply the two generators one after the other (in both direc- tions), do we return to the original image? 34 | Chapter 2: Deep Learning Within the context of deep learning, tensors are simply multidimensional arrays that store the data as it flows through the network. A Question and Answer Generator We're now going to put everything together and build a model that can generate question and answer pairs from a block of text. The self-attention mechanism within the SAGAN model10 The problem with GAN-based models that do not incorporate attention is that con- volutional feature maps are only able to process information locally. However, they are still used in many of the most impressive GANs in the literature and have proven to be a powerful tool in the deep learning practitioner's toolbox—again, I suggest you experiment with both methods and see which works best for you. Fetch the corresponding values of mu and log_sigma for this distribution. The greater the learn- ing rate, the larger the change in weights at each training step. The field of generative modeling is diverse and the problem definition can take a great variety of forms. Simply comparing the pixel values of the two images won't do, because even in two distinct images of the same scene, we wouldn't expect individual pixel values to be similar. Notice that this is not a supervised learning prob- lem—we are not trying to predict the correct action. The key to understanding GANs is in understanding the training process. These line drawings need not be realistic human forms; the model will find a way to make the drawing look as human as possible, as this is what it has been trained to do. To clone the repository for this book, navigate to the folder where you would like to store the files and type the following into your terminal: git clone Always make sure that you have the most up-to-date version of the codebase by run- ning the following command: git pull You should now be able to see the files in a folder on your machine. Figure 2-9. In Keras, the code in Example 5-6 describes the training loop. While we will be using a 2D computer simulation as our environment, the same technique could also be applied to real-world scenarios where testing strategies in the live environment is expensive or infeasible. Action A feasible move that an agent can make. To see this in action, let's start building our first GAN in Keras, to generate pictures of nocturnal ganimals. Keras is not the library that performs the low-level array operations required to train neural networks. It doesn't matter what these values are as we shall be using an embedding layer to transform the integer lookup values into vectors. 172 | Chapter 6: Write Figure 6-4. First, let's take a look at some uncurated example outputs from the generator, after 3,000 training batches (Figure 4-17). It would be far too time-consuming and computationally intensive to use the entire dataset to calculate the gradient at each training step, so generally a batch size between 32 and 256 is Your First Deep Neural Network | 43 used. This is an excellent resource for many Keras examples of important GANs from the literature. In this chapter, we shall start by taking a trip to planet Wrodl, where our first generative modeling assignment awaits… Hello Wrodl! The year is 2047 and you are delighted to have been appointed as the new Chief Fash- ion Officer (CFO) of Planet Wrodl. Variational Autoencoders. Instead, only small (117M parameter) and medium (345M parame- ter) versions of GPT-2 have been released officially.5 MuseNet MuseNet is a model also released by OpenAI that applies the Transformer architec- ture to music generation. The normal distribution in one dimension3 We can sample a point z from a normal distribution with mean μ and standard devia- tion σ using the following equation: z = μ + σϵ where ϵ is sampled from a standard normal distribution. The environment file is stored in the custom_envs folder. GloVe vectors allocate exactly the same representation to the word water regardless of the context, whereas BERT incorporates surrounding infor- mation to create a bespoke representation for the word in context. Her love of literature is one of the reasons I first decided that writing a book would be an exciting thing to do. To comment or ask technical questions about this book, send email to bookques- For more information about our books, courses, conferences, and news, see our web- site at . Figure 5-13. Specifically, it is Wasserstein GAN | 117 required that the critic is a 1-Lipschitz continuous function. Therefore for the WGAN critic, minimizing the loss function can be written as follows: 116 | Chapter 4: Generative Adversarial Networks WGAN critic loss minimization minD − $\mathbb{E}_{x \sim p_D}[D(x)] - \mathbb{E}_{z \sim p_D}[D(G(z))]$ X Z In other words, the WGAN critic tries to maximise the difference between its predic- tions for real images and generated images, with real images scoring higher. However, this was shown to suffer from the vanishing gradient problem and didn't scale well to long sequences of data. 272 | Chapter 8: Play Summary In this chapter we have seen how a generative model (a VAE) can be utilized within a reinforcement learning setting to enable an agent to learn an effective strategy by test- ing policies within its own generated dreams, rather than within the real environment. The hidden state is a vec- tor with length equal to the number of units in the cell—it can be thought of as the cell's current understanding of the sequence. Figure 9-11 shows some of the images generated by BigGAN, trained on the ImageNet dataset. Other games have rewards constantly awar- ded throughout the episode (e.g., points in a game of Space Invaders). Plot of the latent space, colored by digit There are a few interesting points to note: 1. The requirements.txt file contains the names and version numbers of all the packages that you will need to run the examples. For example, we could use reinforcement learning to train a robot to walk across a given terrain. To recap, Figure 7-11 shows how a standard encoder–decoder model for music gen- eration might look, without attention—the kind that we introduced in Chapter 6. The training loop of the WGAN is shown in Example 4-8. 5 Or integral if the sample space is continuous. The 128 players in the orchestra are divided into 4 equal sections of 32 players. Analysis of the MuseGAN We can perform some experiments with our MuseGAN by generating a score, then tweaking some of the input noise parameters to see the effect on the output. We can summarize the responsibili- ties of each section as shown in Table 7-2. The input data to the decoder is the question matching the given answer range. A generative model can be broadly defined as follows: A generative model describes how a dataset is generated, in terms of a probabilistic model. Figure 1-2. encoder_mu_log_var This accepts a 64 × 64 × 3 observation as input and outputs the mu and log_var vectors corresponding to this input. This idea was introduced by Geoffrey Hinton in 2012 and presented in a 2014 paper by Srivastava et al.7 Dropout layers are very simple. Why does this addition to the loss function help? Let's pick this apart to understand what it means in more detail. This is then passed to a bar generator, which ultimately outputs the specified bar for the specified track. As we shall see, there are many different types of layer, but one of the most common is the dense layer that connects all units in the layer directly to every unit in the previous layer. These answer phrases are then fed to the question generator and the output of this model is shown on the lefthand side of the diagram ("Predicted Question"). We also set the covariance matrix of the new normal distribution to be the covariance matrix of the points in P, but use the existing mean in the covariance calculation rather than the cur- rent mean of the points in P. We have already seen an example of the output from the CycleGAN model in Figure 5-3. The darker the square, the greater the attention placed on the hidden state corre- sponding to this point in the sequence. Generative modeling estimates p x —the probability of observing observation x. After a brief discussion, they decide to change the way that new paintings are marked on the wall. We'll work through the notebook 07_01_notation_compose.ipynb in the book reposi- tory. 136 | Chapter 5: Paint Your First CycleGAN Much of the following code has been inspired by and adapted from the amazing Keras-GAN repository maintained by Erik Linder-Norén. You can train your own variational autoencoder on the digits dataset by running the notebook 03_03_vae_digits_train.ipynb in the book repository. Improving the Model | 53 Dropout Layers When studying for an exam, it is common practice for students to use past papers and sample questions to improve their knowledge of the subject material. The Generative Modeling Framework Let's start by playing a generative modeling game in just two dimensions. • How does the model find one of the tiny proportion of satisfying possible gener- ated observations among a high-dimensional sample space? • You will need to either tokenize the punctuation, or remove it altogether. If this switch happens early on (4 × 4 or 8 × 8 resolution), coarse styles such as pose, face shape, and glasses from source B are carried across onto source A. The function returns the squared distance between the L2 norm and 1. Simply put, a GAN is a battle between two adversaries, the generator and the discrim- inator. The Input layer is an entry point into the network. They allow the network to blend high-level abstract information captured during the downsam- pling process (i.e., the image style) with the specific spatial information that is being fed back in from previous layers in the network (i.e., the image content). Thus domain A is the space of apple images and domain B is the space of orange images. 6. The output is an image of the same size as an image in the original training data. Example 5-4. We can use a Reshape layer to squash the output to a single vector, of length equal to the length of the input sequence (seq_length). The VAE doesn't suffer from this problem, since the convo- lutional layers of the encoder are designed to translate low-level pixels into high-level features and the decoder is trained to perform the opposite task of translating the high-level features in the latent space back to raw pixels. Model architecture for the encoder–decoder network that formulates a question given an answer encoder_input_mask = Input(shape=(ANSWER_SIZE, DOC_SIZE) , name="encoder_input_mask") encoder_inputs = Lambda(lambda x: K.batch_dot(x[0], x[1]) , name="encoder_inputs")([encoder_input_mask, answer_outputs]) encoder_cell = GRU(2 * GRU_UNITS, name = 'encoder_cell')(encoder_inputs) decoder_inputs = Input(shape=(Q_SIZE,), name="decoder_inputs") decoder_emb = embedding(decoder_inputs) decoder_emb.trainable = False decoder_cell = GRU(2 * GRU_UNITS, return_sequences = True, name = 'decoder_cell') A Question and Answer Generator | 195 decoder_states = decoder_cell(decoder_emb, initial_state = [encoder_cell]) decoder_projection = Dense(VOCAB_SIZE, name = 'decoder_projection' , activation = 'softmax', use_bias = False) decoder_outputs = decoder_projection(decoder_states) total_model = Model([document_tokens, decoder_inputs, encoder_input_mask] , [answer_tags, decoder_outputs]) answer_model = Model(document_tokens, [answer_tags]) decoder_initial_state_model = Model([document_tokens, encoder_input_mask] , [encoder_cell]) The answer mask is passed as an input to the model—this allows us to pass the hidden states from a single answer range through to the encoder–decoder. For example, the Keras code corresponding to the diagram in Figure 2-11 is: input_layer = Input(shape=(64,64,1)) conv_layer_1 = Conv2D(filters = 2 , kernel_size = (3,3) , strides = 1 , padding = "same")(input_layer) Improving the Model | 47 Figure 2-11. Let's now reveal the true data-generating distribution, pdata, and see how the frame- work applies to this example. A residual block in Keras from keras.layers.merge import add def residual(layer_input, filters): shortcut = layer_input y = Conv2D(filters, kernel_size=(3, 3), strides=1, padding='same')(layer_input) y = InstanceNormalization(axis = -1, center = False, scale = False)(y) y = Activation('relu')(y) 10 Kaiming He et al., "Deep Residual Learning for Image Recognition," 10 December 2015, 1512.03385. One facet of Scott's work centers around using pix2pix models to create art of the human form. Some examples of passages generated by the model when seeded only with a sequence of tokens; here we use a temperature of 0.5 for the note names and durations Second, we can analyze the distribution of note pitches over time by plotting the pre- dicted distribution at each timestep as a heatmap. Your First Deep Neural Network | 35 By default the image data consists of integers between 0 and 255 for each pixel channel. 224 | Chapter 7: Compose Figure 7-14. For example, browse, browsing, browses, and browsed would all be stemmed to brows. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For the reward loss, we simply use the binary cross entropy between the true reward and the predicted log odds from the network. To start collecting data, run the following command from your terminal: python 03_generate_rnn_data.py Example 8-2 contains an excerpt from the 03_generate_data.py file that shows how the MDN-RNN training data is generated. Early on in the generation process (the lefthand side of the dia- gram) the key is not yet firmly established and therefore there is more uncertainty in how to choose the next note. Copyright 2019 Applied Data Science Partners Ltd., 978-1-492-04194-8." xii | Preface If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at O'Reilly Online Learning For almost 40 years, O'Reilly Media has provided technology and business training, knowledge, and insight to help compa- nies succeed. Let's now take a look at the model architecture that is able to generate questionanswer pairs from a given block of text. For example, the vector between embeddings for the words man and woman is approximately the same as the vector between the words king and queen. Each process calls the Python file 01_generate_data.py. Instead, we judge the generators simultaneously on three criteria: 1. KL divergence is a way of measuring how much one probability distribution differs from another. Conceptually, for image generation this is an incredibly dif- ficult task, considering the vast number of ways that individual pixel values can be assigned and the relatively tiny number of such arrangements that constitute an image of the entity we are trying to simulate. The VAE is trained to learn a latent representation of the environment, which is then used as input to a recurrent neural network that forecasts future trajectories within the latent space. The error in the prediction is then propagated backward through the network, adjusting each set of weights a small amount in the direction that improves the prediction most significantly. The parameters to the model are given in Example 7-4. The batch normalization process6 We can place batch normalization layers after dense or convolutional layers to nor- malize the output from those layers. Style The style input is also a vector of length z_dim. Here, we use the vari- able DOC_SIZE to describe the size of this input, but the variable is actually set to None. 270 | Chapter 8: Play Figure 8-21. Even though this is a supervised learning problem, when we come to building generative models in future chapters many of the core ideas from this network (such as loss functions, activation functions, and under- standing layer shapes) will still be extremely important. This is achieved by setting the appropriate elements of the input to the softmax to –∞. Lastly, each hidden state vector hj is multiplied by its respective weight αj, and the results are then summed to give the context vector c (thus c has the same length as a hidden state vector). They grow to about 9 inches in length and live for 13 to 15 years. Thank you for taking the time to read to the end of this book—I hope you have enjoyed reading it as much as I have enjoyed generating it. First, let's take a set of new images that the model hasn't seen, pass them through the encoder, and plot the 2D representations in a scatter plot. It is also the reason why you get a performance increase when training deep neural networks on GPUs instead of CPUs: GPUs are optimized for large tensor multiplications since these calculations are also necessary for complex graphics manipulation. 252 | Chapter 8: Play Figure 8-9. Sample a value for each dimension of z from the normal distribution parameter- ized by the chosen parameters of mu and log_sigma for this dimension. The VAE encoder for the CelebA dataset Using VAEs to Generate Faces | 89 Figure 3-17. The output of the training process is shown in Figure 8-12. Previously we have trained a deep neural network such as a VAE or GAN by backpropagating the error through the entire network to learn from a training set of data and generalize the information learned to generate new images. Generating the dataset import numpy as np from keras.utils import np_utils def generate_sequences(token_list, step): X = [] y = [] Your First LSTM Network | 171 for i in range(0, len(token_list) - seq_length, step): X.append(token_list[i: i + seq_length]) y.append(token_list[i + seq_length]) y = np_utils.to_categorical(y, num_classes = total_words) num_seq = len(X) print('Number of sequences:', num_seq, "") return X, y, num_seq step = 1 seq_length = 20 X, y, num_seq = generate_sequences(token_list, step) X = np.array(X) y = np.array(y) The LSTM Architecture The architecture of the overall model is shown in Figure 6-3. This is a super- vised image classification problem, so we can use the same network architecture as in Chapter 2: stacked convolutional layers, followed by a dense output layer. Figure 1-2 shows the discriminative modeling process—note how it dif- fers from the generative modeling process shown in Figure 1-1. Using VAEs to Generate Faces | 87 3. Coder. At each training step, one batch of images is passed through the network and the errors are backpropagated to update the weights. For a machine to compose music that is pleasing to our ear, it must master many of the same technical challenges that we saw in the previous chapter in relation to text. TensorFlow is an open-source Python library for machine learning, developed by Google. Even if you have no musical training you may still be able to guess. Let's call this rule pdata. This way, you can be sure that your system version of Python isn't affected by any of the libraries that we will be using. We are not yet at the stage where text and music can be convincingly combined to produce pop songs with long-term structure, but given the impressive progress in both text and music generation in recent years it surely won't be long before this is a reality. Next, you need to set up a virtual environment. This way, the layer can learn from information both preceding and succeeding the given timestep. This Lambda layer samples a point z in the latent space from the normal distribu- tion defined by the parameters mu and log_var. What is remarkable is that the model hasn't explicitly decided to set the music in a certain key at the beginning, but instead is literally making it up as it goes along, trying to choose the note that best fits with those it has chosen previously. 113 114 114 115 115 115 117 118 119 120 121 121 125 127 Teaching Machines to Paint, Write, Compose, and Play 5. We save each episode as an individual file inside the ./data/rollout/ directory. Another point to remember is that it is the layers in a deep neural network that are convolutional, rather than the network itself. The orange box, pmodel, is an estimate of the true data-generating distribution pdata (the gray area) What Is Generative Modeling? This is because we calculate the overall style loss as a weighted sum across 158 | Chapter 5: Paint several layers, all of which have different sizes. Both of these methods—Upsampling + Conv2D and Conv2DTranspose—are acceptable ways to transform back to the original image domain. This is similar to predictive text on your mobile phone, where you are given the choice of a few words that might fol- low on from what you have already typed. 198 | Chapter 6: Write Figure 6-16. For every channel in the preceding layer, two weights need to be learned: the scale (gamma) and shift (beta). The Keras model that defines the discriminator—a model that takes an input image and outputs a single number between 0 and 1. SAGAN solves this problem by incorporating the attention mechanism that we explored earlier in this chapter into the GAN. Jupyter notebook You are now ready to start building generative deep neural networks. The decoder then transforms this representation into sequen- tial output, such as the translation of the text into another language, or a question relating to the document. However, there is a problem. The WGAN-GP critic training process The gradient penalty loss measures the squared difference between the norm of the gradient of the predictions with respect to the input images and 1. Hyperparameters As we have seen, even with simple GANs, there are a large number of hyperparame- ters to tune. Each image is classified into exactly one of 10 classes, as shown in Figure 2-3. The larger the batch size, the more sta- ble the gradient calculation, but the slower each training step. However, with an autoencoder this is not guaranteed. Initially, this may be true, but as the network trains and the weights move further away from their random initial values, this assumption can start to break down. This is just the start of a gener- ative revolution that will leave us with no choice but to find answers to some of the biggest questions about the mechanics of creativity, and ultimately, what it means to be human. Generally, in a business setting, we don't care how the data was generated, but instead want to know how a new example should be categorized or valued. One of the advantages of using representation learning is that we can perform opera- tions within the more manageable latent space that affect high-level properties of the image. The smaller the truncation threshold, the greater the believability of generated samples, at the expense of reduced variability. Figure 2-15 shows the whole process. 18 | Chapter 1: Generative Modeling accessoriesType n Blank Round Sunglasses Grand Total 11 22 17 50 ^ θ 0.22 0.44 0.34 1.00 clothingType n Hoodie Overall ShirtScoopNeck ShirtVNeck Grand Total 7 18 19 6 50 ^ θ 0.14 0.36 0.38 0.12 1.00 To calculate the probability of the model generating some observation x, we simply multiply together the individual feature probabilities. A deep neural network consists of a series of stacked layers. They take Gene's generated photos to the auction and the experts cannot believe the quality of the new sightings—they are just as convincing as the originals. AI Art I recently attended a meetup in London entitled "Form, Figures and BigGAN," featur- ing presentations by artist Scott Eaton and BigGAN creator Andrew Brock, organized by AI art curator Luba Elliott. The solution consists of three distinct parts, as shown in Figure 8-3, that are trained separately: V M C A variational autoencoder. The Wrodlers are known to be quite particular when it comes to fashion, so your task is to generate new styles that are similar to those that already exist on the planet, but not identical. It achieves this through a masked lan- guage model: during training, 15% of words are randomly masked out and the model must try to re-create the original sentence, given the masked input. The authors show that this results in the latent vectors $\mathbb{E}$ being significantly more disentangled than the original z vectors. For now, it is enough to understand at a high level what repre- sentation learning is trying to achieve. Since Gene has never seen a ganimal, he doesn't know how to produce good photos of the creature, and also, since Di has only ever sold the photos she found, she cannot even tell the difference between a good photo of a gani- mal and a photo of nothing at all. Optimizing the generated output via gradient descent is different to how we have tackled generative modeling problems thus far. Figure 9-11. At test time, the dropout layer doesn't drop any units, so that the full network is used to make predictions. Figure 4-4. He goes over to ask if he can help. Figure 2-3. The new shapes of y_train and y_test are therefore [50000, 10] and [10000, 10] respectively. Pixels or characters are really just the dimples of the canvas into which higher-level informative features, such as an image of a chimney or the word striker, are embed- ded. Stack convolutional layers on top of each other. The relevant code is included in the 03_02_autoencoder_analysis.ipynb notebook in the book repository. Encoder–Decoder Models So far, we have looked at using LSTM networks for generating the continuation of an existing text sequence. Table 1-2. The U-Net architecture diagram In a similar manner to a variational autoencoder, a U-Net consists of two halves: the downsampling half, where input images are compressed spatially but expanded channel-wise, and an upsampling half, where representations are expanded spatially while the number of channels is reduced. MDN for handwriting generation In the car racing example, we allow for each element of the next observed latent state to be drawn from any one of five normal distributions. It is defined as: yi = xi e xj ∑Jj = 1 e Here, J is the total number of units in the layer. This follows the standard GAN practice of alternating the training of the discrim- inators with the training of the generators (through the combined model). Additional evidence that the model has started to understand the concept of an octave can be seen inside the green box below and to the right. In the original MuseGAN paper the authors use a threshold of 0, as each track can contain multiple notes; however, in this setting we can simply take the maximum, to guarantee exactly one note per timestep per track, as is the case for the Bach chorales. Some digits are represented over a very small area and others over a much larger area. Let's take a look at some mathematical notation to describe the difference between generative and discriminative modeling. We will explore some of the most common problems encoun- tered when training GANs in this section, then we will look at some adjustments to the GAN framework that remedy many of these problems. Note that the patches are predicted simultaneously as we pass an image through the network—we do not divide up the image manually and pass each patch through the network one by one. This is the mean of the squared difference between the ground truth yi and predicted value pi of each output unit, where the mean is taken over all n output units: MSE = 1 n yi − pi ni∑ =1 2 If you are working on a classification problem where each observation only belongs to one class, then categorical cross-entropy is the correct loss function. However, we may also wish to build attention into encoder–decoder networks, where we predict a sequence of future notes by using an RNN decoder, rather than building up sequen- ces one note at a time. To start training the controller, run the following command from your terminal (all on one line): xvfb-run -a -s "-screen 0 1400x900x24" python 05_train_controller.py car_racing -n 16 -t 2 -e 4 --max_length 1000 where the parameters are as follows: n The number of workers that will test solutions in parallel (this should be no greater than the number of cores on your machine) t The number of solutions that each worker will be given to test at each generation Training the Controller | 261 e The number of episodes that each solution will be tested against to calculate the average reward max_length The maximum number of timeframes in each episode eval_steps The number of generations between evaluations of the current best parameter set The above command uses a virtual frame buffer (xvfb) to render the frames, so that the code can run on a Linux machine without a physical screen. This makes 5 × 3 × 32 = 480 parameters. Training the controller in the OpenAI Gym environment Notice how in the dream architecture, the training of the controller is performed entirely in z-space without the need to ever decode the z vectors back into recogniza- ble track images. 40 | Chapter 2: Deep Learning Figure 2-6. This is the core idea that I believe will propel generative modeling into the spotlight in the next decade, as one of the keys to unlocking artificial general intelligence. 132 | Chapter 5: Paint put them through as oranges due to Florida's fluorescent spray, and customers who took oranges wrongly pay for apples due to their clever disguise, courtesy of Granny Smith. Keras and TensorFlow Keras is a high-level Python library for building neural networks and is the core library that we shall be using in this book. Imagine you're carrying a tall pile of books, and you get hit by a gust of wind. In Keras, the Conv2D layer applies convolutions to an input tensor with two spatial dimensions (such as an image). First, the weights of the network are initialized to small random values. If we take the average position of encoded images in the latent space with the attribute smiling and subtract the average position of encoded images that do not have the attribute smiling, we will obtain the vector that points from not smiling to smiling, which is exactly what we need. This is a very nice property that ensures that even when we choose a point in the latent space that has never been seen by the decoder, it is likely to decode to an image that is well formed. question A question that could be asked about the story text (e.g., "How much did the striker cost?"). We need a model that can infer relevant structure from the data, rather than being told which assumptions to make in advance. answer_masks A binary mask matrix having shape [max_answer_length, max_docu ment_length]. 1 David Ha and Jürgen Schmidhuber, "World Models," 27 March 2018, . Tokenization is the process of split- ting the text up into individual units, such as words or characters. Initialize the decoder with this initial state and feed in the token to gen- erate the first word of the question. 5. If we compare this to the original discriminator training process from Figure 4-7, we can see that the key addition is the gradient penalty loss included as part of the overall loss function, alongside the Wasserstein loss from the real and fake images. In summary, on Planet Wrodl individual features are independent and the sample space is relatively small, so Naive Bayes works well. While the underlying har- monization is still not as perfect or varied as Bach, it is a good attempt at what is an extremely difficult problem to master and highlights the power of GANs to tackle a wide variety of problems. The resulting vector, ft, has a length equal to the number of units in the cell and contains values between 0 and 1 that determine how much of the previous cell state, Ct–1, should be retained. c. After the final Conv2D layer, we use a tanh activation to transform the output to the range [–1, 1], to match the original image domain. ft and Ct–1 are multiplied element-wise and added to the element-wise multiplica- tion of it and Ct. This represents forgetting parts of the previous cell state and then adding new relevant information to produce the updated cell state, Ct. 5. Most chapters in this book will contain a short story that helps to describe a particu- lar technique. Instead, you would make the reasonable assumption that the other person has a general idea of what an average human looks like, then amend this baseline with features that describe groups of pix- els, such as I have very blonde hair or I wear glasses. • A generative model pmodel tries to mimic pdata. Create an independent bar generator network for every track. The Encoder In an autoencoder, each image is mapped directly to one point in the latent space. Your First Music-Generating RNN | 217 218 | Chapter 7: Compose Figure 7-11. He updates them on their accuracy, and gradually they begin to learn how to write stories in his own unique style. If the reward log odds value is greater than 0, predict 1 for the reward; otherwise, predict 0. To build in the skip connections, we will need to introduce a new type of layer: `Concatenate`. Stacked Recurrent Networks The network we just looked at contained a single LSTM layer, but we can also train networks with stacked LSTM layers, so that deeper features can be learned from the text. Moreover, by performing vector arithmetic within the latent space, we can ach- ieve some amazing effects, such as face morphing and feature manipulation. The car is able to drive accurately around the track and can handle most corners, except those that are especially sharp. The Keras model that outputs the values of mu and log_var for a given input image. This is an amazing achievement—remember, when the controller was evaluated after 10 generations it had never attempted the task of driving fast around the track in the real environment. Figure 9-14 shows this in action. Here we create an instance of the VGG19 model, specifying the input tensor and the weights that we would like to preload. The MuseGAN players tour the world giving concerts in any style where there is suf- ficient existing music to train the MuseGAN. Before we start building the model, however, we need to take a closer look at the con- cept of reinforcement learning and the OpenAI Gym platform. In other words, it can generate the likely sequence of input data that would follow from receiving the cue of an egg-shaped region of light falling through the visual field to the sound of a splat as the egg-shaped region stops moving abruptly. 1 Ashish Vaswani et al., "Attention Is All You Need," 12 June 2017, . Two examples of Scott Eaton's work, generated through a pix2pix model trained on photographs of dancers19 AI Music As well as producing aesthetically interesting and evocative images, generative mod- eling has practical application in the field of music generation, especially for video games and films. Generated examples from a GAN trained progressively on the LSUN dataset at 256 × 256 resolution8 Self-Attention GAN (SAGAN) The Self-Attention GAN (SAGAN)9 is a key development for GANs as it shows how the attention mechanism that powers sequential models such as the Transformer can also be incorporated into GAN-based models for image generation. There are several key differences between text and image data that mean that many of the methods that work well for image data are not so readily applicable to text data. This function calculates the next state and reward, given the current state of the environment and chosen action. A controller. However, over time they become more adept at using the technology, and learn how to identify which fruit has been tampered with. We will introduce the transformer architecture in more detail in Chapter 9. As the discriminator is frozen, its weights will not be affected; instead, the generator weights will move in the direction that allows it to better generate images that are more likely to fool the discriminator (i.e., make the discriminator predict values close to 1). The input to the model is a sequence of integer tokens and the output is the probability of each word in the vocabulary appearing next in the sequence. A dropout layer 7 Nitish Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research 15 (2014): 1929–1958, tava14a.pdf. This makes sampling from our latent space extremely problematic. Note that while this is labeled chords_input, it really could control anything about the music that changes per bar, such as general rhythmic style, without being specific to any particular track. Figure 4-8. It is intractable to calculate this gradient everywhere during the training process, so instead the WGAN-GP evaluates the gradient at only a handful of points. We shall explore some of the most important techniques in later chapters of this book. ot is multiplied element-wise with the updated cell state Ct after a tanh activation has been applied to produce the new hidden state, ht. Generative Deep Learning Teaching Machines to Paint, Write, Compose and Play David Foster Generative Deep Learning Teaching Machines to Paint, Write, Compose, and Play David Foster Beijing Boston Farnham Sebastopol Tokyo Generative Deep Learning by David Foster Copyright © 2019 Applied Data Science Partners Ltd. Example 3-2. Collecting Random Rollout Data To start collecting data, run the following command from your terminal: Training Process Overview | 245 bash 01_generate_data.sh \ where the parameters are as follows: The name of the environment used by the make_env function (e.g., car_racing). Split the 481-dimensional output vector into the 3 variables (log_pi, mu, log_sigma) and the reward value. We would ideally like our generative model to upweight areas of the sample space that it believes are more likely, due to some inherent structure learned from the data, rather than just placing all probabilistic weight on the points that are present in the dataset. To download the data, from inside the book repository run the following command: bash ./scripts/download_gutenburg_data.sh 11339 aesop Let's now take a look at the steps we need to take in order to get the data in the right shape to train an LSTM network. To train the WGAN generator, we calculate the loss when comparing predictions for generated images pi=D(G(zi)) to the response yi=1. Lambda layer A Lambda layer simple wraps any function into Keras layer. What Is Generative Modeling? The number of timesteps to freeze the current action for before changing. The pitch and duration of each note, stored as lists Duration 0.25 1.0 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 Pitch B3 G2.D3.B3 B3 A3 G3 F#3 G3 D3 E3 F#3 G3 A3 Your First Music-Generating RNN To create the dataset that will train the model, we first need to give each pitch and duration an integer value (Figure 7-2), exactly as we have done previously for each word in a text corpus. Analysis of WGAN-GP Running the Jupyter notebook 04_03_wgangp_faces_train.ipynb in the book reposi- tory will train a WGAN-GP model on the CelebA dataset of celebrity faces. Like a child with a set of building blocks, the design of your neural network is only limited by your own imagination—and, crucially, your understanding of how the various layers fit together. If you are just getting started with deep learning, I highly recommend that you choose Keras with a TensorFlow backend as your toolkit. 142 | Chapter 5: Paint The reason for this is that the CycleGAN inherits its discriminator architecture from a model known as a PatchGAN, where the discriminator divides the image into square overlapping "patches" and guesses if each patch is real or fake, rather than pre- dicting for the image as a whole. Let's suppose we have a training set consisting of grayscale images of biscuit tins (Figure 1-11). Example 5-9. The architecture for generating question–answer pairs; input data is shown in green bordered boxes Let's start by taking a look at the Keras code that builds the part of the model at the top of the diagram, which predicts if each word in the document is part of an answer or not. The fact that deep learning can form its own features in a lower-dimensional space means that it is a form of representation learning. With recent advances in methodology and technology, we are now able to build machines that can paint origi- nal artwork in a given style, write coherent paragraphs with long-term structure, compose music that is pleasant to listen to, and develop winning strategies for com- plex games by generating imaginary future scenarios. Batch Normalization in WGAN-GP One last consideration we should note before building a WGAN-GP is that batch normalization shouldn't be used in the critic. A bidirectional layer takes advantage of this by storing two sets of hidden states: one that is produced as a result of the sequence being processed in the usual forward direction and another that is pro- duced when the sequence is processed backwards. Building the temporal network def conv_t(self, x, f, k, s, a, p, bn): x = Conv2DTranspose(filters = f , kernel_size = k , padding = p , strides = s , kernel_initializer = self.weight_init)(x) if bn: The MuseGAN Generator | 227 x = BatchNormalization(momentum = 0.9)(x) if a == 'relu': x = Activation(a)(x) elif a == 'lrelu': x = LeakyReLU()(x) return x def TemporalNetwork(self): input_layer = Input(shape=(self.z_dim,), name='temporal_input') x = Reshape([1,1,self.z_dim])(input_layer) x = self.conv_t(x, f=1024, k=(2,1), s=(1,1), a= 'relu', p = 'valid', bn = True) x = self.conv_t(x, f=self.z_dim, k=(self.n_bars - 1,1) , s=(1,1), a= 'relu', p = 'valid', bn = True) output_layer = Reshape([self.n_bars, self.z_dim])(x) return Model(input_layer, output_layer) The input to the temporal network is a vector of length 32 (z_dim). Coder's wall from Figure 3-1. Adding and subtracting features to and from faces It is quite remarkable that even though we are moving the point a significantly large distance in the latent space, the core image barely changes, except for the one feature that we want to manipulate. Moreover, when the generators are applied one after the other, the difference between the input image and the reconstructed image is minimal. Notice how only 12 of the 32 dimensions differ significantly from the standard normal distribution (mu = 0, log_var = 0). Our model then learns how to discriminate between these two groups and outputs the probability that a new observation has label 1—i.e., that it was painted by Van Gogh. 30 | Chapter 1: Generative Modeling CHAPTER 2 Deep Learning Let's start with a basic definition of deep learning: Deep learning is a class of machine learning algorithm that uses multiple stacked layers of processing units to learn high-level representations from unstructured data. Coder then finds the marker where Epsilon placed it and never hears Mr. N. For example, writing a program that uses several chunks of code from this book does not require permission. The deep learning model had an error rate of 16%—a massive improvement on the next best model, which only achieved a 26.2% error rate. For a binary classification problem such as our artist discriminator, Van Gogh paintings would be labeled 1 and non–Van Gogh paintings labeled 0. We want observations that are scaled between 0 and 1, so this function is simply a division by 255. We will also look to the future and explore how the most cutting-edge techniques available today might change the way in which we view creativity, and whether we will ever be able to create an artificial entity that can produce content that is creatively indistinguishable from works created by the human pioneers of art, literature, and music. The code to build the LSTM network is given in Example 6-3. For example, Figure 9-5 shows how two attention heads of the decoder layer are able to work together to provide the correct German translation for the word the, when used in the context of the street. x | Preface There is a myth that you need a GPU in order to start training deep learning models —while this is of course helpful and will speed up training, it is not essential. This is a parameter that is set when you define the layer and has nothing to do with the length of the sequence. Pixels col- ored blue are normalized by the same mean and variance (calculated over the values of these pixels). However, for the original U-Net application (image segmenta- tion) and also for style transfer, it is critical that when we upsample back to the origi- nal image size, we pass back into each layer the spatial information that was lost during downsampling. Gatys, Alexander S. Coder recruits the help of his daughter, Epsilon. Based on her mistakes, she updates her own understanding of how to discriminate between Gene's attempts and the original photos. The layer that we use to calculate the content loss is the second convolutional layer of the fifth block. In this book we use Ten- sorFlow as it is the most widely adopted and best documented of the three. The players in the melody section have the most exhausting job, because they give dif- ferent instructions to each instrumental track at the start of every bar throughout the piece. Precise positional information is reduced throughout this process in favor of capturing higher-level features, making it computationally inefficient for the model to learn long-range dependencies between distantly connected pixels. From inside the folder where you cloned the book's repository, run the follow- ing command: bash ./scripts/download_cyclegan_data.sh apple2orange This will download the dataset of images of apples and oranges that we will be using. The key difference between the models is that while BERT is bidirectional, GPT-2 is unidirectional. Crucially, paired images from each domain aren't required for a CycleGAN to work, making it an extremely powerful and flexible technique. However, this orchestra isn't about to launch The Musical Organ | 221 into a Beethoven symphony or a Tchaikovsky overture. — Rule 2: It can generate examples that are suitably different from the observa- tions in X. Also, for the InstanceNormalization layers in this network, there are no weights to learn since we do not use scaling (gamma) or shift (beta) parameters. Given the current z vector, action, and reward, the MDN-RNN is then trained to pre- dict the subsequent z vector and reward. The last convolu- tional layer is then flattened and connected to a Dense layer of size 2, which repre- sents our two-dimensional latent space. Let's start by taking a high-level look at how a deep neural network can make a pre- diction about a given input. Printed in the United States of America. The forget and input gates are replaced by reset and update gates. The MDN-RNN Loss Function The loss function for the MDN-RNN is the sum of the z vector reconstruction loss and the reward loss. It is also now recommended practice to increase the batch size as training progresses.4 This continues until all observations in the dataset have been seen once. 1 What Is Generative Modeling? Let's start by paying a visit to a strange art exhibition… The Art Exhibition Two brothers, Mr. N. Figure 8-16 shows several generations of the process. We can achieve this by performing a matrix multiplication of Q with KT and scaling by a factor dk. We now apply the Naive Bayes assumption to simplify the last line: K p x = Π p xk k=1 This is the Naive Bayes model. Neural Style Transfer | 153 iterations. The latent space of the VAE colored by digit So far, all of our work on autoencoders and variational autoencoders has been limited to a latent space with two dimensions. Finally, we use a Lambda layer to perform the summation along the seq_length axis, to give the context vector of length rnn_units. If we use dropout layers, the network cannot rely too much on any one unit and therefore knowledge is more evenly spread across the whole network. Musical Notation We'll be using the Python library music21 to load the MIDI files into Python for pro- cessing. The output from the autoencoder is the output from the encoder passed through the decoder. We shall see an explicit example of this in the next chapter, applied not to biscuit tins but to faces. Let's now build our first nontrivial example of a generative model. It is our goal to build a model that can generate new sets of features that look as if they have been created using the same rules as the original data. The output size from this convolutional layer would also be 5 × 5, as the padding allows the kernel to extend over the edge of the image, so that it fits five times in both directions. World Model Architecture | 243 We will need to train the controller using reinforcement learning as there is no train- ing dataset that will tell us that a certain action is good and another is bad. 258 | Chapter 8: Play Figure 8-14. Defining the CycleGAN gan = CycleGAN(input_dim = (128,128,3) , learning_rate = 0.0002 , lambda_validation = 1 , lambda_reconstr = 10 , lambda_id = 2 , generator_type = 'u-net' , gen_n_filters = 32 , disc_n_filters = 32) Let's first take a look at the architecture of the generators. The tackle this, the authors show a

of key formulae missing from their formula book. Therefore while the image quality isn't yet perfect, we should be encour- aged by the fact that our WGAN is clearly learning the high-level features that make up a color photograph of a horse. Do the images produced by each generator fool the relevant discrimina- tor? Deep learning conceptual diagram For example, in Figure 2-2, layer 1 consists of units that activate more strongly when they detect particular basic properties of the input image, such as edges. One day, he came up with a brilliant idea that would allow him to produce new works of fiction in his style, while also keeping the inmates occupied—he would get the inmates to collectively write the stories for him! He branded the new society the LSTM (Literary Society for Troublesome Miscreants). For example, suppose we want to take an image of somebody who looks sad and give them a smile. We would instead need to train a generative model, which can output sets of pixels that have a high chance of belonging to the original training dataset. 237 Reinforcement Learning OpenAI Gym World Model Architecture The Variational Autoencoder The MDN-RNN The Controller Setup Training Process Overview Collecting Random Rollout Data Training the VAE The VAE Architecture Exploring the VAE Collecting Data to Train the RNN Training the MDN-RNN The MDN-RNN Architecture Sampling the Next z and Reward from the MDN-RNN The MDN-RNN Loss Function Training the Controller The Controller Architecture CMA-ES Parallelizing CMA-ES Output from the Controller Training In-Dream Training In-Dream Training the Controller Challenges of In-Dream Training Summary 238 239 241 242 243 243 244 245 245 248 249 252 255 257 258 259 259 261 262 262 265 267 268 270 272 273 9. In other words, the bar generator should use this vector to establish consistency between bars and tracks. Generating text with an LSTM network def sample_with_temp(preds, temperature=1.0): # helper function to sample an index from a probability array preds = np.asarray(preds).astype('float64') preds = np.log(preds) / temperature exp_preds = np.exp(preds) preds = exp_preds / np.sum(exp_preds) probs = np.random.multinomial(1, preds, 1) return np.argmax(probs) def generate_text(seed_text, next_words, model, max_sequence_len, temp): output_text = seed_text seed_text = start_story + seed_text for _ in range(next_words): token_list = tokenizer.texts_to_sequences([seed_text])[0] token_list = token_list[-max_sequence_len:] token_list = np.reshape(token_list, (1, max_sequence_len)) probs = model.predict(token_list, verbose=0)[0] y_class = sample_with_temp(probs, temperature = temp) output_word = tokenizer.index_word[y_class] if y_class > 0 else '' if output_word == "|": break seed_text += output_word + ' ' output_text += output_word + ' ' return output_text Generating New Text | 179 This function weights the logits with a temperature scaling factor before reap- plying the softmax function. A bar (or measure) is a small unit of music that contains a fixed, small number of beats. (For example, does output from g_BA fool d_A and does output from g_AB fool d_B?) 2. generate_data_action is a custom function that stores the rules for generating random actions. 5 Source: Wikipedia, . First, what do we mean by "unstructured data" and its counterpart, "structured data"? Ten new Planet Pixel styles, generated by the Naive Bayes model The Challenges of Generative Modeling First, since the Naive Bayes model is sampling pixels independently, it has no way of knowing that two adjacent pixels are probably quite similar in shade, as they are part of the same item of clothing, for example. There is no training set of correct actions, as we do not know what the optimal action is for a given state of the environ- ment. If we do not freeze the discriminator's weights, the discriminator will adjust so that it is more likely to predict generated images as real, which is not the desired outcome. Analysis of the CycleGAN Let's see how the CycleGAN performs on our simple dataset of apples and oranges and observe how changing the weighting parameters in the loss function can have dramatic effects on the results. Each square in the matrix indicates the amount of attention given to the hidden state of the network corresponding to the note on the vertical axis, at the point of predicting the note on the horizontal axis; the more red the square, the more attention was given 216 | Chapter 7: Compose We can see that for the second note of the piece (B-3 = B-flat), the network chose to place almost all of its attention on the fact that the first note of the piece was also B-3. Amazingly, the agent can then use this generative model as a pseudoenvironment to iteratively test policies, using an evolutionary methodology, which generalize well to the real environment. All of the environments provide a step method through which you can submit a given action; the environment will return the next state and the reward. 46 | Chapter 2: Deep Learning Figure 2-10. Your First GAN | 107 Figure 4-7. The Dense layer is perhaps the most fundamental layer type in any neural network. The code in Example 7-2 loops through the score and extracts the pitch and duration for each note (and rest) in the piece into two lists. This is shown in Example 5-4. We may wish to build a model that can generate a new image of a horse that has never existed but still looks real because the model has learned the general rules that govern the appearance of a horse. 97 the creatures and is concerned that her business will falter if she can't produce more original photographs soon. The Naive Bayes models that we saw in Chapter 1 wouldn't be able to achieve this level of sophistication since they cannot model the interdependencies between pixels that are crucial to forming these high-level features. Many would now consider the challenge a solved problem. MuseNet can therefore operate with full atten- tion over 4,096 tokens and can learn long-term structure and melodic structure across a range of styles. , xK = p x2, . This orchestra composes original music live during the performance and is powered entirely by a set of players giving instructions to a huge Musical Organ (MuseGAN for short) in the middle of the stage, which converts these instructions into beautiful music for the pleasure of the audience. The effect of this inclusion is shown in Figure 9-10. The plot is not symmetrical about the point (0, 0), or bounded. Individual notes in chords are sep- arated by a dot, so that the whole chord can be stored as a single string. 6 | Chapter 1: Generative Modeling the real world, but instead could learn in its own imaginary environment. 264 | Chapter 8: Play Figure 8-16. How- ever, if the switch happens later, only fine-grained detail is carried across from source B, such as colors and microstructure of the face, while the coarse features from source A are preserved. It is therefore not surprising that humanity has started to ask the ultimate question of creativity: can we create something that is in itself creative? Tokenization The first step is to clean up and tokenize the text. For example, if you can count 1, 2, 1, 2 along to a piece of music, then there are two beats in each bar and you're probably listening to a march. Points A, B, and C show three observations generated by pmodel with varying degrees of success: • Point A breaks Rule 1 of the Generative Modeling Framework—it does not appear to have been generated by pdata as it's in the middle of the sea. The process repeats with the new sequence, and so on, for as many notes as we wish to generate. Example images from the CIFAR-10 dataset2 2 Source: Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009, https:// www.cs.toronto.edu/~kriz/cifar.html. However, when you do so, it is clear that something has gone very wrong. The final layer has a tanh activation applied, as we will be using a WGAN-GP (which requires tanh output activation) to train the network. With WGANs, we can simply train the critic several times between generator updates, to ensure it is close to convergence. This way, you will have the foundations in place to understand all genera- tive models, whether based on deep learning or not, from the same probabilistic standpoint. BigGAN uses a batch size of 2,048—8 times larger than the batch size of 256 used in SAGAN—and a channel size that is increased by 50% in each layer. We clearly cannot just merge the style images with the base image through interpolation, as the content of the style images would show through and the colors would become muddy and blurred. Your First Autoencoder Let's now build an autoencoder in Keras. On arrival, you are presented with a dataset featuring 50 observations of Wrodler fashion (Figure 1-7) and told that you have a day to come up with 10 new styles to present to the Fashion Police for inspection. There are actually only two parts that we need to change: the encoder and the loss function. Notice how the sequence length isn't specified—the attention mechanism does not require a fixed-length input, so we can leave this as variable. We can then multiply this matrix pointwise with the hidden states from the final LSTM layer, which also has shape [seq_length, rnn_units]. The response variable for each sequence is the subsequent word, one-hot encoded into a vector of length 4,169 (the number of distinct words in the vocabulary). How did you know this? For this task you'll need to download the camel numpy file and save it into the ./data/camel/ folder in the book repository.2 The origi- 2 By happy coincidence, ganimals look exactly like camels. There is no difference between this diagram and Figure 6-6; it's just a different way of drawing the mechanics of a recurrent layer. 6 Ishaan Gulrajani et al., "Improved Training of Wasserstein GANs," 31 March 2017, 1704.00028. The LeakyReLU activation function is very similar to ReLU, with one key difference: whereas the ReLU activation function returns zero for input values less than zero, the LeakyReLU function returns a small 38 | Chapter 2: Deep Learning negative number proportional to the input. 12 | Chapter 1: Generative Modeling Maximum Likelihood Estimation Maximum likelihood estimation is the technique that allows us to estimate θ —the set of parameters θ of a density function, pθ x , that are most likely to explain some observed data X. You pay no attention to the background noise as you concen- trate on converting the absence of light on a tiny proportion of your retina into a sequence of abstract concepts that convey almost no meaning individually but, when combined, trigger a wave of parallel representations in your mind's eye—images, 299 emotions, ideas, beliefs, and potential actions all flood your consciousness, awaiting your recognition. We could use a structured table of these features to train a logistic regression, random forest, or XGBoost model to predict the binary response variable —did the person subscribe (1) or not (0)? The Transformer | 279 Figure 9-3. One example of this is shown in Figure 7-3. The prison is particularly strange because it only consists of one large cell, containing 256 prisoners. At the same time, it updates the covariance matrix of the distribution to maximize the likelihood of sam- pling the high-scoring agents, given the previous mean. For example, if we were teaching a reinforcement learning algorithm to play chess, the environment would consist of the rules that govern how a given action (e.g., the move e4) affects the next game state (the new positions of the pieces on the board) and would also specify how to assess if a given position is checkmate and allocate the winning player a reward of 1 after the winning move. I strongly recommend experimenting with the parameters that define the models in this book, to understand how the architecture affects the number of weights in each layer, model performance, and model runtime. In the next chapter we will see how many of the same ideas around RNNs can be applied to another type of sequential data: music. Figure 8-5 shows an excerpt from frames 40 to 59 of one episode, as the car approaches a corner, alongside the randomly chosen action and reward. Since we are connecting each input unit directly to the 3 output action units, the total number of weights to tune is 288 × 3 = 864, plus 3 bias weights, giving 867 in total. Therefore, a Naive Bayes model is able to learn some structure from the data and use this to generate new examples that were not seen in the original dataset. This drives the generator to find new ways to fool the discriminator, and so the cycle continues. These rewards are again averaged to return the overall score for the parameter set. As the piece progresses, the model settles on a key and certain notes become almost certain not to appear. The orchestra can be trained to generate music in a particular style, and no two performances are ever the same. In an encoder–decoder with attention, we instead initial- ize the decoder using the built-in standard initializers for a recurrent layer. For this reason, discriminative modeling is synonymous with supervised learning, or learning a function that maps an input to an output using a labeled dataset. It is worth remembering that we haven't provided the model with any additional features beyond the raw pixels, so it has to work out high-level concepts such as how to draw a hump, legs, or a head entirely by itself. The brothers set about their plan and open their new exhibition to the public. Each prisoner has an opinion on how Edward's current story should continue. ©2019 O'Reilly Media, Inc. Therefore for the GAN generator, minimizing the loss function can be written as follows: GAN generator loss minimization minG − $\mathbb{E}_{z \sim p_Z} \log D(G(z))$ Now let's compare this to the Wasserstein loss function. These models rely on the input features to be informative and not spatially dependent. The embedding layer is initialized with GloVe word vectors (explained in the fol- lowing sidebar). Two of the most widely used are pointer networks4 and attention mechanisms.5 Pointer networks give the model the ability to "point" at specific words in the input text to include in the generated question, rather than only relying on the words in the known vocabulary. log_var The logarithm of the variance of each dimension. The resulting dataset now looks a lot more like the text data that we have dealt with previously. Again, you shouldn't need to adjust the parameters of this optimizer too much, but it is worth reading the Keras documentation to understand the role of each parameter. In all but the last layer, we use batch normalization and ReLU activation (LeakyReLU could also be used). Also, a huge thanks to my colleagues at Applied Data Science Partners, Ross Witeszc- zak, Chris Schon, Daniel Sharp, and Amy Bull. Batch normalization also has been shown to reduce overfitting, and therefore many modern deep learning architectures don't use drop- out at all, and rely solely on batch normalization for regularization. Example 4-8. Loss and accuracy of the discriminator and generator during training By observing images produced by the generator at specific epochs during training (Figure 4-9), it is clear that the generator is becoming increasingly adept at producing images that could have been drawn from the training set. Naive Bayes The Naive Bayes parametric model makes use of a simple assumption that drastically reduces the number of parameters we need to estimate. Preface | xv PART I Introduction to Generative Deep Learning The first four chapters of this book aim to introduce the core techniques that you'll need to start building generative deep learning models. Advances in Image Generation | 291 Figure 9-12. If we see any dimensions that are significantly dif- ferent from a standard normal distribution, we should probably reduce the recon- struction loss factor, since the KL divergence term isn't having enough effect. A diagram of the total network is shown in Figure 2-5. 20 | Chapter 1: Generative Modeling Figure 1-9. The Art Exhibition | 63 Figure 3-3. This is exactly where deep learning excels and is one of the key reasons why the technique has driven the major recent advances in generative modeling. In this case, there are four parameters: the coordinates of the bottomleft θ1, θ2 and top-right θ3, θ4 corners of the box. Note the similarity with the Trans- former attention head architecture in Figure 9-2. • Clip the weights of the critic after each update. The output from the generator is an array of values in the range [–1, 1] (due to the tanh activation function of the final layer). The key part of the script is outlined in Example 8-1. Coder and Mr. D. Notice how we use a stride of 2 in some of the convolutional layers to reduce the size of the tensor as it passes through the network, but increase the number of channels (1 in the grayscale input image, then 64, then 128). It is a densely connected neural net- work with no hidden layer; it connects the input vector directly to the action vector. We've seen this concept before. This is a consequence of the model architecture, so you get the reverse direction for free. This prevents the action from changing too rapidly for the car to make progress. 214 | Chapter 7: Compose Figure 7-9. If you have studied machine learning, most problems you will have faced will have most likely been discriminative in nature. We also choose to map to the logarithm of the variance, as this can take any real number in the range (– ∞, ∞), matching the natural output range from a neural network unit, whereas variance values are always positive. We would ideally like to have full control of the style of the image, and this requires a disentangled separation of high-level features in the latent space. This is known as additive smoothing. The input vec- tor is created through the concatenation of the four relevant chord, style, melody, and groove vectors, each of length z_dim. Loop over the tracks and bars, creating a generated bar for each combination. How can we design an agent to maximize the sum of rewards in a given environ- ment? The solution is surprisingly simple. The conductor and the MuseGAN use exactly the same process as we saw in Chapter 4, when Di and Gene worked together to continuously improve the photos of ganimals taken by Gene. You're allowed to play around with hair- styles, hair color, glasses, clothing type, and clothing color to create your masterpieces. The relationship between the four models is shown in Figure 5-5. Example 4-5. We also create a model that outputs the alpha layer vector, so that we will be able to understand how the network is attributing weights to previous hidden states. The Variational Art Exhibition | 77 Building a Variational Autoencoder The previous story showed how, with a few simple changes, the art exhibition could be transformed into a successful generative process. The blue dot is part of the bird's long tail and so attention falls on other tail pixels, some of which are distant from the blue dot. Armed with this knowledge, we will then hypothesize how the field will continue to develop in the near future, providing a tantalizing view of what might be possible in the years to come. 7 Ronneberger et al., 2015, . GPT-2 GPT-2 is a model developed by OpenAI that is trained to predict the next word in a passage of text. The problem is that we do not know pdata explicitly—all we have to work with is the sample of observations X generated by pdata. Experience books, videos, live online training courses, and more from O'Reilly and our 200+ partners—all in one place. Coder and then recon- structed using just these two numbers by Mr. D. You can also follow along with the accom- panying notebook in the book repository, 06_02_qa_train.ipynb. As we do not have paired images on which to train our generators, we also need to train two discriminators that will determine if the images produced by the generators are convincing. In other words, the model shouldn't simply reproduce things it has already seen. Compose. We saw how vanilla GANs can suffer from several problems, including mode collapse and unstable training, and how the Wasserstein loss function remedied many of these problems and made GAN training more predictable and reliable. We'll be building both architectures in this chapter, start- ing with U-Net.6 4 Source: Zhu et al., 2017, . Second, the text generated at temperature = 0.2 is less adventurous but more coher- ent in its choice of words than the text generated at temperature = 1.0, as lower temperature values result in more deterministic sampling. The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. Stack convolutional transpose layers on top of each other. The style loss function style_loss = 0.0 def gram_matrix(x): features = K.batch_flatten(K.permute_dimensions(x, (2, 0, 1))) gram = K.dot(features, K.transpose(features)) return gram def style_loss(style, combination): S = gram_matrix(style) C = gram_matrix(combination) channels = 3 size = img_nrows * img_ncols return K.sum(K.square(S - C)) / (4.0 * (channels ** 2) * (size ** 2)) feature_layers = ['block1_conv1', 'block2_conv1', 'block3_conv1', 'block4_conv1', 'block5_conv1'] for layer_name in feature_layers: layer_features = outputs_dict[layer_name] style_reference_features = layer_features[1, :, :, :] combination_features = layer_features[2, :, :, :] sl = style_loss(style_reference_features, combination_features) style_loss += (style_weight / len(feature_layers)) * sl The style loss is calculated over five layers—the first convolutional layer in each of the five blocks of the VGG19 model. Generative Versus Discriminative Modeling In order to truly understand what generative modeling aims to achieve and why this is important, it is useful to compare it to its counterpart, discriminative modeling. Given the current sequence (of note names and note durations), the model pre- dicts two distributions, for the next note name and duration. Output after 200 epochs of training11 11 Source: Zhu et al., 2017, . However, now, instead of placing the marker on the wall himself, he passes his opin- ion of where it should go to Epsilon, who decides where the marker will be placed. Feed the document string to the answer generator to produce sample positions for answers in the document. The difference between structured and unstructured data When our data is unstructured, individual pixels, frequencies, or characters are almost entirely uninformative. The output is a Dense layer with a single unit and no activation function, as required by the WGAN-GP framework. Example 3-3. Figure 6-6. The Dropout layer in Keras implements this functionality, with the rate parameter specifying the proportion of units to drop from the preceding layer: Dropout(rate = 0.25) Dropout layers are used most commonly after Dense layers since these are most prone to overfitting due to the higher number of weights, though you can also use them after convolutional layers. Sampling from the MDN-RNN outputs a prediction for the next z and reward, given the current z and chosen action. The Transformer | 283 Figure 9-5. If this sounds confusing, don't worry! We'll start by seeing how to apply an attention mechanism after a simple recurrent layer (i.e., to solve the problem of predicting the next note of Bach's Cello Suite No. 1), before we see how this extends to encoder– decoder networks, where we want to predict a whole sequence of subsequent notes, rather than just one. The relationship between these definitions is shown in Figure 8-1. Example of a MuseGAN predicted score, showing the closest real score in the training data and how the generated score is affected by changing the input noise Let's now play around with the input noise to tweak our generated score. In Keras a model is defined by the input and output layers. Our unique network of experts and innovators share their knowledge and expertise through books, articles, conferences, and our online learning platform. Analysis of the CycleGAN In the original CycleGAN paper, the model was trained for 200 epochs to achieve state-of-the-art results for artist-to-photograph style transfer. Append this word to the existing sequence and repeat. Distribution of word probabilities following various sequences The model is able to generate a suitable distribution for the next most likely word across a range of contexts. A total of 50,416 such sequences can be constructed, so our training dataset X is an array of shape [50416, 20]. Let's put these pieces together into a new deep learning architecture and see how it performs on the CIFAR-10 dataset. The variational autoencoder's encoder ### THE ENCODER encoder_input = Input(shape=self.input_dim, name='encoder_input') x = encoder_input for i in range(self.n_layers_encoder): conv_layer = Conv2D(filters = self.encoder_conv_filters[i] , kernel_size = self.encoder_conv_kernel_size[i] , strides = self.encoder_conv_strides[i] , padding = 'same' , name = 'encoder_conv_' + str(i)) x = conv_layer(x) if self.use_batch_norm: x = BatchNormalization()(x) x = LeakyReLU()(x) if self.use_dropout: x = Dropout(rate = 0.25)(x) shape_before_flattening = K.int_shape(x)[1:] x = Flatten()(x) Building a Variational Autoencoder | 81 self.mu = Dense(self.z_dim, name='mu')(x) self.log_var = Dense(self.z_dim, name='log_var')(x) # encoder_mu_log_var = Model(encoder_input, (self.mu, self.log_var)) def sampling(args): mu, log_var = args epsilon = K.random_normal(shape=K.shape(mu), mean=0., stddev=1.) return mu + K.exp(log_var / 2) * epsilon encoder_output = Lambda(sampling, name='encoder_output')([self.mu, self.log_var]) encoder = Model(encoder_input, encoder_output) Instead of connecting the flattened layer directly to the 2D latent space, we con- nect it to layers mu and log_var. This slightly 220 | Chapter 7: Compose changes the equations to incorporate an extra index (i) to specify the step of the decoder. In addition, we shall see how generative modeling can be used to optimize playing strategy for a game (World Models) and take a look at the most cutting-edge genera- tive architectures available today, such as StyleGAN, BigGAN, BERT, GPT-2, and MuseNet. 165 The Literary Society for Troublesome Miscreants 166 Table of Contents | v Long Short-Term Memory Networks Your First LSTM Network Tokenization Building the Dataset The LSTM Architecture The Embedding Layer The LSTM Layer The LSTM Cell Generating New Text RNN Extensions Stacked Recurrent Networks Gated Recurrent Units Bidirectional Cells Encoder–Decoder Models A Question and Answer Generator A Question-Answer Dataset Model Architecture Inference Model Results Summary 167 168 168 171 172 172 174 176 179 183 183 185 187 187 190 191 192 196 198 200 7. The KL diver- gence loss term ensures that the mu and sigma values never stray too far from a stan- dard normal. The number after each note name indicates the octave that the note is in—since the note names (A to G) repeat, this is needed to uniquely identify the pitch of the note. In the original CycleGAN paper, one of the standout achievements was the ability for the model to learn how to convert a given photo into a painting in the style of a par- ticular artist. To achieve this, we need to choose a different parametric model. Four different normalization methods.9 Here, N is the batch axis, C is the channel axis, and (H, W) represent the spatial axes. This is a free Jupyter Notebook environment that requires no setup and runs entirely in the cloud. Next, we collapse the tensor along the pitch axis. The Keras model that defines the generator—a model that accepts a vector of length 100 and outputs a tensor of shape [28, 28, 1]. 196 | Chapter 6: Write 3. This is true in general; VAEs tend to produce 126 | Chapter 4: Generative Adversarial Networks softer images that blur color boundaries, whereas GANs are known to produce sharper, more well-defined images. The Wasserstein loss function is then defined as follows: Wasserstein loss 1 − n ∑ni = 1 yi pi To train the WGAN critic D, we calculate the loss when comparing predictions for a real images pi=D(xi) to the response yi=1 and predictions for generated images pi=D(G(zi)) to the response yi=-1. Example 5-5. Kingma and Max Welling published a paper that laid the founda- tions for a type of neural network known as a variational autoencoder (VAE).1 This is now one of the most fundamental and well-known deep learning architectures for generative modeling. Information from the start of the source sentence can become diluted by the time it reaches the context vector, especially for long sentences. Figure 2-19. After this transformation, each observation in the training set consists of the following five features: document_tokens The tokenized story text (e.g., [1, 4633, 7, 66, 11, ...]), clipped/padded with zeros to be of length max_document_length (a parameter). CMA-ES is just one form of evolutionary strategy. Like GPT-2, it is unidirectional, trained to predict the next note given a sequence of previous notes. The answer is simple: because nonrandom actions make the stream of environmental data easier to gener- ate. WGAN-GP loss If we compare the WGAN-GP output to the VAE output from the previous chapter, we can see that the GAN images are generally sharper—especially the definition between the hair and the background. First, we can try changing the noise vector—the bottom-left score in Figure 7-16 shows the result. For example, in one concert, each member of the groove section produced five instruc- tions, one for each of the vocal, piano, string, bass, and drum tracks. Conclusion | 301 Index Symbols 1-Lipschitz continuous function, 117 A activation functions, 38 Adam optimizer, 42 AI art, 296 (see also style transfer) AI music, 297 (see also music generation) Anaconda, 27 artifacts, 105 artificial neural networks (ANNs), 33 arXiv, xi attention mechanisms analysis of, 213-217 building in Keras, 208-212 in encoder-decoder networks, 217-221 examples in language translation, 206 generating polyphonic music, 221 multihead attention module, 280 autoencoders analysis of, 72-75 building, 66 decoder architecture, 68-71 encoder architecture, 66 joining encoder to decoder, 71 parts of, 64 process used by, 65 representation vectors in, 65 uses for, 65 B backpropagation, 34 batch normalization, 51-53, 55, 125 BERT (Bidirectional Encoder Representations from Transformers), 285 BigGAN, 291 binary cross-entropy loss, 42, 71, 107 C categorical cross-entropy loss, 42 CelebFaces Attributes (CelebA) dataset, 86 character tokens, 169 CIFAR-10 dataset, 35, 120 CMA-ES (covariance matrix adaptation evolu- tion strategy), 261-268 CNTK, 34 code examples, obtaining and using, x, xii, 27 comments and questions, xiii composition (see music generation) concatenate layer, 140 content loss, 154-156 convolutional layers in neural networks, 46-51, 59 convolutional transpose layers, 68-69 covariate shift, 52 CycleGAN (cycle-consistent adversarial net- work) analysis of, 147 benefits of, 135 compiling, 144-146 CycleGAN versus pix2pix, 135 discriminators, 142 generators (ResNet), 150-151 generators (U-Net), 139-142 Keras-GAN code repository, 137 Monet-style transfer example, 149-153 303 overview of, 137 published paper on, 135 training, 146 training data, 137 D DCGAN (deep convolutional generative adver- sarial network), 101 deep learning deep neural networks, 33, 59 defined, 31 Keras and TensorFlow for, 34 model creation, 35-46 model improvement, 46-58 premise behind, 154 structured versus unstructured data, 31 dense layers, 33 density function, 11 discriminative modeling, 2 dropout layers, 54 E encoder-decoder models, 187-190, 217-221 environment setup, 27-29 evolutionary strategies, 262 exploding gradient problem, 51 F facial image generation dataset used, 86 encoder and decoder architecture, 88 generating new faces, 92 latent space arithmetic, 93 morphing between faces, 94 progress in, 5 VAE analysis, 91 VAE training, 87 features, 2 fit_generator method, 88 Functional API (Keras), 37-41 G gated recurrent units (GRUs), 168, 185 generative adversarial networks (GANs) challenges of, 112-115 defining, 100-106 discriminators, 101, 142 "ganimal" example, 97-99 304 | Index published paper on, 97 theory underlying, 99 training, 107-112 Wasserstein GAN, 115-121 WGAN-GP, 121-127 generative deep learning additional resources, xi advances in generative modeling, 5-7, 277-297 challenges of, 22-27 future of, 299-301 Generative modeling framework, 7-10 history of generative modeling, 275-277 introduction to, 1-5 learning objectives and approach, x learning prerequisites, x probabilistic generative models, 10-21 generators attention-based, 290 bar generator, 229 in GANs, 103-106 MuseGAN generator, 226 question-answer generators, 190-200 ResNet generators, 150-151 StyleGAN generator, 292 U-Net generators, 139 GloVe ("Global Vectors"), 195, 285 Goodfellow, Ian, 97 Google Colaboratory, xi GPT-2 language model, 5, 285 gradient descent, 153 Gram matrices, 158 H Ha, David, 237, 243, 265 Hello Wrodl! example, 13-21 hidden layers, 34 hidden state, 174, 176 Hinton, Geoffrey, 4, 54 Hochreiter, Sepp, 167 Hou, Xianxu, 91 Hull, Jonathan, 118 hyperparameters, 114 I identity, 145 image generation (see also facial image genera- tion; neural style transfer technique) BigGAN, 291 CIFAR-10 dataset for, 35 generative modeling process, 1 generative versus discriminative modeling, 2 key breakthrough in, 4 ProGAN, 287 progress in facial image generation, 5 representation learning for, 23 rise of generative modeling for, 5 Self-Attention GAN (SAGAN), 289 StyleGAN, 292 ImageNet dataset, 154 ImageNet Large Scale Visual Recognition Chal- lenge (ILSVRC), 4 in-dream training, 268-272 inference, 196 instance normalization layers, 140 K Keras attention mechanisms in, 208-212 autoencoder creation in, 67 backends for, 34 benefits of, 34 content loss calculation in, 155 Conv2DTranspose layer, 70, 105 custom loss function creation, 259 CycleGAN creation and training, 137-149 data loading, 35 decoder creation in, 70 documentation, 43 fit_generator method, 88 GAN discriminator creation in, 102 importing, 28 inference model in, 197 LSTM in, 168 model building, 37-41 model compilation, 41 model evaluation, 44 model improvement, 46-58 model training, 43 MuseGAN generator in, 230 PatchGAN discriminators in, 143 residual blocks in, 150 U-Net generators in, 141 VAE creation in, 81 Keras layers Activation, 56 Batch Normalization, 51 Bidirectional, 187 Concatenate, 140 Conv2D, 47 Conv2DTranspose, 70 Conv3D, 232 Dense, 38 Dropout, 54 Embedding, 172 Flatten, 38 GRU, 168 Input, 38 Lambda, 82 LeakyReLU, 56 LSTM, 174 Reshape, 211 Upsampling2D layer, 104 Kingma, Diederik, 61 Kullback–Leibler (KL) divergence, 84 L L-BFGS-B algorithm, 161 labels, 3 language translation, 188, 206 layers, 33 LeakyReLU, 38 likelihood, 12 Lipschitz constraint, 117, 121 loss functions, 41 LSTM (long short-term memory) networks with attention mechanism, 206 dataset used, 168 embedding layer, 172 generating datasets, 171 generating new text, 179-182 history of, 167 LSTM architecture, 172 LSTM cell, 176-178 LSTM layer, 174-176 published paper on, 167 tokenizing the text, 168-170 M machine learning advances in, 4 major branches of, 238 machine painting (see style transfer) machine writing (see text data generation) Machine-Learning-as-a-Service (MLaaS), 5 Maluuba NewsQA dataset, 191 Index | 305 masked language model, 285 maximum likelihood estimation, 13 MDN (mixture density network), 243, 255-261 mean squared error loss, 42 MIDI files, 202 mode collapse, 113 models CycleGAN, 135-152 deep neural networks, 35-58 encoder-decoder models, 187-190 generative adversarial networks (GANs), 97-115 generative modeling, 1-10 generative versus discriminative modeling, 2 improving models, 46-58 LSTM (long short-term memory) networks, 168-178 neural style transfer, 153-162 parametric modeling, 11 probabilistic generative models, 10-21 probabilistic versus deterministic, 2 question-answer generators, 190-200 RNNs (recurrent neural networks), 205-221 variational autoencoders (VAEs), 61-96 Wasserstein GAN, 115-121 WGAN-GP, 121-127 World Model architecture, 241-244 Monet-to-photo dataset, 149 multihead attention module, 280 multilayer RNNs, 183 MuseGAN, 223-235 analysis of, 233 creation, 223-231 MuseNet, 286, 297 MuseScore, 202 music generation challenges of, 201 data extraction, 204 dataset used, 202 generating polyphonic music, 221 importing MIDI files, 202 MuseGAN analysis, 233 MuseGAN creation, 223-231 MuseGAN critic, 232 MuseGAN example, 221 music versus text generation, 201 musical notation, 204 prerequisites to, 202 306 | Index RNN (recurrent neural network) for, 205-221 music21 library, 202 N Naive Bayes parametric model, 17-20 neural style transfer technique analysis of, 161 content loss, 154-156 definition of, 153 premise of, 153 running, 160 style loss, 156-159 total variance loss, 160 nontrainable parameters, 53 parametric modeling, 11 PatchGAN discriminators, 143 pix2pix, 135 positional encoding, 279 probabilistic generative models challenges of, 22 Hello Wrodl! example, 13, 20 model construction, 14-17 Naive Bayes parametric model, 17-20 probabilistic theory behind, 10-13 probability density function, 11, 79 ProGAN, 287 Project Gutenberg, 168 Python, 27 Q qgen-workshop TensorFlow codebase, 190 question-answer generators dataset used, 191 encoder-decoder models, 188 inference, 196 model architecture, 192-196 model parts, 190 model results, 198-200 questions and comments, xiii R reconstruction loss, 84 regularization techniques, 54 reinforcement learning (RL) defined, 238 key terminology, 238 OpenAI Gym toolkit for, 239 process of, 239 ReLU (rectified linear unit), 38 representation learning, 23-27 representation vectors, 65 residual networks (ResNets), 150-151 RMSProp optimizer, 43 RNNs (recurrent neural networks) bidirectional cells, 187 gated recurrent units (GRUs), 185 history of, 167, 167 LSTM (long short-term memory) networks, 167-182 MDN-RNN World Model architecture, 243 for music generation, 205-221 stacked recurrent networks, 183, 206 root mean squared error (RMSE), 71 S sample space, 11 scaled dot-product attention, 282 Schmidhuber, Jurgen, 167, 237 self-attention, 281 Self-Attention GAN (SAGAN), 289 sequence modeling, 277 Sequential models (Keras), 37-41 sigmoid activation, 39 skip connections, 139 softmax activation, 39 stacked recurrent networks, 183 standard deviation, 79 standard normal curves, 79 stemming, 169 stochastic (random) elements, 2 strides parameter (Keras), 48 structured data, 31 style loss, 156-159 style transfer aim of, 131 apples and organges example, 132-134 CycleGAN analysis, 147 CycleGAN creation and training, 137-147 CycleGAN introduction, 135 CycleGAN Monet example, 149-152 neural style transfer technique, 153-162 uses for, 131 StyleGAN, 5, 292 supervised learning, 3 T TensorFlow, 34 text data generation encoder-decoder models, 187-190 LSTM (long short-term memory) networks, 167-182 question-answer generators, 190-200 RNN (recurrent neural network) exten- sions, 183-187 short story generation example, 166 text versus image data, 165 text versus music generation, 201 text summarization, 188 Theano, 34 tokenization, 168-170 total variance loss, 160 trainable parameters, 53 training data, 1 training process, 34 Transformer analysis of, 283 BERT model, 285 decoder layers, 283 GPT-2 language model, 285 history of, 206 models architecture, 277 multihead attention layer, 280 MuseNet model, 286 positional encoding function, 279 published paper on, 277 truncation trick, 291 U U-Net, 139-142 uninformative loss, 114 units, 33, 174 Index | 307 unstructured data, 31 upsampling, 104 V validity, 145 vanishing gradient problem, 151, 167 variance, 79 variational autoencoders (VAEs) autoencoder analysis, 72-75 autoencoder example, 66-72 autoencoder parts, 64 decoders, 68-71 encoders, 66-68 facial image generation using, 86-95 generative art example, 61-64, 75-77 published paper on, 61, 91 VAE analysis, 85 VAE build in Keras, 81 VAE diagram, 82 VAE loss function, 84 VAE parts, 78-85 World Model architecture, 242 World Model training, 248-255 VGG19 network, 154 virtual environments, 27 W Wasserstein GANs (WGANs) 308 | Index analysis of, 120 benefits of, 115 Lipschitz constraint, 117 training, 119 Wasserstein loss, 115-117 weight clipping, 118 Wasserstein GAN–Gradient Penalty (WGANGP) analysis of, 125 converting WGAN to WGAN-GP, 121 gradient penalty loss, 121-125 weight clipping, 118, 121 weights, 33 Welling, Max, 61 World Models collecting random rollout data, 245 collecting RNN training data, 255 model setup, 244 published paper on, 237 training in-dream, 268-272 training overview, 245 training the controller, 261-268 training the MDN-RNN, 257-261 training the VAE, 248-255 World Model architecture, 241-244 World Models paper, 237, 243, 275 Wrodl!, 13 About the Author David Foster is the cofounder of Applied Data Science, a data science consultancy delivering bespoke solutions for clients. Also, it has a higher probability of being sampled than, say, (LongHairStraight, Red, Round, ShirtScoop- Neck, White), because white clothing appears more often than blue clothing in the dataset. The master divides the parameters into the number of slave nodes available. 6 Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 18 May 2015, . Constant width Used for program listings, as well as within paragraphs to refer to program ele- ments such as variable or function names, databases, data types, environment variables, statements, and keywords. Running the notebook 04_01_gan_camel_train.ipynb in the book repository will start training the GAN. In Part II, we will be building on these techniques to tackle several creative tasks, such as painting, writing, and composing music through models such as CycleGAN, encoder–decoder models, and MuseGAN. We can see this more clearly with an example. Exponentiate and scale log_pi so that it can be interpreted as 32 probability dis- tributions over the 5 mixture indices. If they find that there are obvious discrepancies, they'll have to spend their hard-earned profits on learning better techniques. This sampling process is very similar to a VAE, though the faces produced by a GAN are quite different—often sharper, with greater distinction between different parts of the image. 2 Source: Vaswani et al., 2017. Using this loss function instead of binary cross entropy results in a more stable convergence of the GAN. The CycleGAN paper was released only a few months after the pix2pix paper and shows how it is possible to train a model to tackle problems where we do not have pairs of images in the source and target domains. Feed the document and answer mask to the encoder to generate the initial state for the decoder. First, notice how the answer generator is able to accurately identify which words in the document are most likely to be contained in an answer. Examples of the inputs and outputs to the two networks are shown in Figure 4-3. As this is a CycleGAN, the model is also able to translate the other way, converting an artist's paintings into realistic-looking photographs. In Chapter 5, we shall examine two techniques relating to machine painting. Musical notation We use the chordify method to squash all simultaneously played notes into chords within a single part, rather than them being split between many parts. Merging styles between two generated images at different levels of detail18 18 Source: Karras, Laine, and Aila, 2018. The attention mechanism was proposed to solve this problem. Ecker, and Matthias Bethge, "A Neural Algorithm of Artistic Style," 26 August 2015, . Your First CycleGAN | 141 return d def upsample(layer_input, skip_input, filters, f_size=4, dropout_rate=0): u = UpSampling2D(size=2)(layer_input) u = Conv2D(filters, kernel_size=f_size, strides=1, padding='same')(u) u = InstanceNormalization(axis = -1, center = False, scale = False)(u) u = Activation('relu')(u) if dropout_rate: u = Dropout(dropout_rate)(u) u = Concatenate()([u, skip_input]) return u # Image input img = Input(shape=self.img_shape) # Downsampling d1 = downsample(img, self.gen_n_filters) d2 = downsample(d1, self.gen_n_filters*2) d3 = downsample

data, as we can calculate the gradient of our loss func- tion with respect to individual pixels to establish the direction in which pixel col- ors should be changed to minimize the loss. This is exactly why we need the skip connections. Improving the Model | 49 flatten_layer = Flatten()(conv_layer_2) output_layer = Dense(units=10, activation = 'softmax')(flatten_layer) model = Model(input_layer, output_layer) Figure 2-13. Here we run three episodes for each set of parameters. Coder simply chooses a point on the wall to represent the painting, places a marker at this point, then throws the original artwork away. To check that it has installed correctly, navigate in your terminal to the folder where you have cloned the book repository and type: jupyter notebook A window should open in your browser showing a screen similar to Figure 1-15. The model has chosen to ignore this because the note prior to this is also an E-flat, an octave lower (E-2). Example 7-1. 13 Source: Brock et al., 2018. This highlights the importance of ensuring the weightings of the three loss functions are well balanced—too little identity loss and the color shift problem appears; too much identity loss and the CycleGAN isn't sufficiently incentivized to change the input to look like an image from the opposite domain. It defines the set of rules that govern the game state update process and reward allocation, given the agent's previous action and current game state. Therefore, generating music note by note is complex, because we often do not want all the instruments to change note simultaneously. Constant width italic Shows text that should be replaced with user-supplied values or by values deter- mined by context. If we tried to use such a model to generate Picasso paint- ings, it would assign just as much weight to a random collection of colorful pixels as to a replica of a Picasso painting that differs only very slightly from a genuine painting. Don't worry if this looks intimidating! It's only built from elements that we have seen already and we will be walking through the architecture step by step in this section. It had only ever driven around the environment randomly (to train the VAE and MDN-RNN) and then in its own dream environment to train the controller. First, each hidden state hj (a vector of length equal to the number of units in the recurrent layer) is passed through an alignment function a to generate a scalar, ej. One of the reasons for scaling input data into a neural network is to ensure a stable start to training over the first few iterations. 262 | Chapter 8: Play b. I've chosen a rule that has been used to generate the set of points X in Figure 1-4. How- ever, there is a fun example in the middle of this chapter, so be sure not to miss out on that! As a first step, we shall define four key terms: sample space, density function, paramet- ric modeling, and maximum likelihood estimation. However, the final hidden state of the encoder may not be able to sufficiently retain this information, as the word scored appears early in the sentence. The input to the LSTM layer is a vector of length 36—a concatenation of the encoded z vector (length 32) from the VAE, the current action (length 3), and the previous reward (length 1). 222 | Chapter 7: Compose Table 7-2. The same idea can also be applied to the list of durations. As we shall see later, other layer types require multidimensional arrays as input, so you need to be aware of the required input and output shape of each layer type to understand when it is necessary to use Flatten. Summary | 273 CHAPTER 9 The Future of Generative Modeling I started writing this book in May 2018, shortly after the "World Models" paper dis- cussed in Chapter 8 was published. Even when our mother or father points to an apple and says apple, there is no reason for our young brains to associate the two and learn that the way in which light entered our eye at that particular moment is in any way related to the way the sound waves entered our ear. We have already seen how MuseNet is able to generate endless amounts of music in a given style and therefore could be adapted to provide the back- ground mood music for a film or video game. Notice how there is one response per patch, as we are using a PatchGAN discriminator. Convolutional Layers First, we need to understand what is meant by a convolution in the context of deep learning. A ResNet generator It has been shown that ResNet architectures can be trained to hundreds and even thousands of layers deep and not suffer from the vanishing gradient problem, where the gradients at early layers are tiny and therefore train very slowly. The resulting vec- tor, ot, has a length equal to the number of units in the cell and stores values between 0 and 1 that determine how much of the updated cell state, Ct, to output from the cell. Since the inception of this book, significant advancements in GAN and attention-based methodologies have taken us to the point where we can now generate images, text, and music that is practically indistinguishable from human-generated content. Figure 2-12 shows a 3 × 3 kernel being passed over a 5 × 5 input image, with padding = "same" and strides = 1. The general architecture is shown in Figure 9-4. Architecture of the decoder Note that the decoder doesn't have to be a mirror image of the encoder. The only other part we need to change is the loss function. A graphical representation of one game state in the CarRacing environment World Model Architecture We'll now cover a high-level overview of the entire architecture that we will be using to build the agent that learns through reinforcement learning, before we explore the detailed steps required to build each component. The full model If we feed the full_model with an observation, it is able to reconstruct an accurate representation of the image, as shown in Figure 8-8. In short, it works by maintaining a normal distribution from which it can sample the parameters of new agents. Your First GAN | 105 x = Activation(self.generator_activation)(x) x = Reshape(self.generator_initial_dense_layer_size)(x) if self.generator_dropout_rate: x = Dropout(rate = self.generator_dropout_rate)(x) for i in range(self.n_layers_generator): x = UpSampling2D()(x) x = Conv2D(filters = self.generator_conv_filters[i] , kernel_size = self.generator_conv_kernel_size[i] , padding = 'same' , name = 'generator_conv_' + str(i))(x) if i < n_layers_generator - 1: if self.generator_batch_norm_momentum: x = BatchNormalization(momentum = self.generator_batch_norm_momentum))(x) x = Activation('relu')(x) else: x = Activation('tanh')(x) generator_output = x generator = Model(generator_input, generator_output) Define the input to the generator—a vector of length 100. Figure 7-14 shows how two bars of raw data are converted into the trans- formed dataset that will feed the GAN with the corresponding musical notation. 284 | Chapter 9: The Future of Generative Modeling BERT BERT (Bidirectional Encoder Representations from Transformers)3 is a model devel- oped by Google that predicts missing words from a sentence, given context from both before and after the missing word in all layers. Examples from the generator of a WGAN trained on images of horses Clearly, this is a much more difficult task than our previous ganimal example, but the WGAN has done a good job of establishing the key features of horse images (legs, sky, grass, brownness, shadow, etc.). 226 | Chapter 7: Compose Example 7-4. This is because the architecture of the model isn't dependent on the length of the input sequence—the number of cells in the layer will adapt to equal the length of the input sequence, so we don't need to specify it explicitly. Here, log_var is the logarithm of the variance of the distribution. The input vector is a concatenation of the current z vector (length 32) and the current hidden state of the LSTM (length 256), giving a vector of length 288. 114 | Chapter 4: Generative Adversarial Networks This is why it is important to understand the inner workings of the GAN and know how to interpret the loss function—so that you can identify sensible adjustments to the hyperparameters that might improve the stability of the model. If we move the filter across the entire image, from left to right and top to bottom, recording the convolutional output as we go, we obtain a new array that picks out a particular feature of the input, depending on the values in the filter. Samples of Gene's ganimal photography Introduction to GANs The adventures of Gene and Di hunting elusive nocturnal ganimals are a metaphor for one of the most important deep learning advancements of recent years: generative adversarial networks. If an algorithm performs well on the training dataset, but not the test dataset, we say that it is suffering from overfitting. 14 To calculate the dot product between two vectors, multiply the values of the two vectors in each position and sum the results. And in recent times, we have found ourselves rushing to bookshops at midnight to buy stories about a fictional wizard, because the combination of letters creates a narrative that wills us to turn the page and find out what happens to our hero. In the decoder, the query input comes from the previous layer in the decoder and the key–value input comes from the final output from the encoder. Coder (bottom row). The output of a Conv2D layer is another four-dimensional tensor, now of shape (batch_size, height, width, filters), so we can stack Conv2D layers on top of each other to grow the depth of our neural network. Probabilistic Generative Models | 17 where K is the total number of features (i.e., 5 for the Wrodl example). In front of him sits an orchestra. This lack of correlation between the generator loss and image quality sometimes makes GAN training diffi- cult to monitor. Your First GAN | 103 Figure 4-5. Relating this to neural networks, each layer is like a book in the pile. We have already seen one type of encoder–decoder network in the context of image generation: the variational autoencoder. The overall loss is the weighted sum of the loss for each criterion. This is known as the training data, and one such data point is called an observation. This is what distinguishes this as a reinforcement learning problem. While there is only one true density function pdata that is assumed to have generated the observable dataset, there are infinitely many density functions pmodel that we can use to estimate pdata. The output is shown in Figure 3-20. Your First Music-Generating RNN | 209 3. The output from the evaluate method The output from this method is a list of the metrics we are monitoring: categorical cross-entropy and accuracy. We can see that in each case, the decoder chooses the correct "type" of question— who, how much, or when—depending on the type of answer. The sum of these two terms is the total variance loss. Remember, the aim of a mixture density network is to model the fact that our next z could be drawn from one of several possible distributions with a certain probability. In this chapter, we shall start by building a standard autoen- coder and then see how we can extend this framework to develop a variational autoencoder—our first example of a generative deep learning model. This is shown in Figure 1-13. However, musical generation presents additional challenges that are not required for text generation, namely pitch and rhythm. Example 8-3. This is exactly what a convolutional layer is designed to do, but with multiple filters rather than just one. The same technique was applied by one of the authors of the "World Models" paper, David Ha, to a handwriting generation task, as shown in Figure 8-4, to describe the fact that the next pen point could land in any one of the distinct red areas. Example of LSTM-generated passages, at two different temperature values There are a few things to note about these two passages. Figure 6-5. However, there is no right or wrong choice—you should choose whichever works best for your use case. Using VAEs to Generate Faces | 91 Figure 3-19. This means that instead of using recurrent networks we can apply the same convolutional-based techniques that worked so well for image generation problems to music—in particular, GANs. Before we explore this new architecture, there is just enough time to visit the concert hall, where a performance is about to begin… The Musical Organ The conductor taps his baton twice on the podium. There is of course spatial structure to an image, temporal structure to a recording, and both spatial and temporal structure to video data, but since the data does not arrive in columns of features, it is considered unstructured, as shown in Figure 2-1. You should see a Python 3 prompt, with Keras reporting that it is using the TensorFlow backend as shown in Figure 1-14. There are guidelines and best practices but you should feel free to experiment with layers and the order in which they appear. • The text vocabulary (the set of distinct words in the training set) may be very large, with some words appearing very sparsely or perhaps only once. Therefore, if our Naive Bayes model is working directly with the highly correlated pixel values, the chance of it finding a satisfying combination of values is incredibly small. 4 Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly, "Pointer Networks," 9 July 2015, 1506.03134. Next, the softmax function is applied to the vector e1,…,en to produce the vector of weights α1,…,αn. The three output neurons correspond to the three actions (turn, accelerate, brake) and are scaled to fall in the appropriate ranges. The performance is about to begin. This elegant framework for structur- ing a generative modeling problem is the underlying engine behind most state-ofthe-art generative models. Example 3-3 shows how we build the decoder in Keras. Again, it's important to remember that the labels were not used at all Building a Variational Autoencoder | 85 during training—the VAE has learned the various forms of digits by itself in order to help minimize reconstruction loss. Some of the exhibits and corresponding markers are displayed in Figure 3-3. In our previous example, the sample space consists of all points of latitude and longi- tude x = x1, x2 on the world map. Each position thus has its own unique encoding, and since the function can be applied to any value of pos it can be used to encode any position, no matter what the sequence length of the input is. The Keras code to build the critic is given in Example 7-8. Now that we've defined our model, we just need to compile it with a loss function and optimizer, as shown in Example 3-5. In our analogy, Mr. N. Similarly, the reverse is happening in the bottom row, as the generator learns how to convert a photograph into a scene that Monet might have painted himself. In fact, we've already seen this plot: it's just Mr. N. For this reason, the discrimina- tor in a WGAN is usually referred to as a critic. In the car racing example, we choose five normal distributions. The problem with this is that nature does not provide us with rewards; it just provides data. During training, Keras outputs the progress of the procedure, as shown in Figure 2-7. However, since there is nothing to stop any of the activation distributions shifting significantly in a certain direction, this can sometimes lead to runaway weight values and an overall collapse of the network. Figure 7-16 shows a score that has been generated by the model from random nor- mally distributed noise vectors (top left). To convert this to a single note for each track, we choose the note with the maximum value over all 84 pitches for each time- step. The layer returns the set of pixelwise interpolated images that lie along the lines connecting the real images (inputs[0]) to the fake images (inputs[1]), pairwise, weighted by the alpha value for each pair. The CycleGAN methodology allows us to train a model to learn the general style of an artist and transfer this over to a photograph, to generate output that looks as if the artist had painted the scene in the photo. 106 | Chapter 4: Generative Adversarial Networks Training the GAN As we have seen, the architecture of the generator and discriminator in a GAN is very simple and not so different from the models that we looked at earlier. To remain stable, when the network updates the weights, each layer implicitly assumes that the distri- bution of its input from the layer beneath is approximately consistent across itera- tions. Unlike a BatchNormalization layer, it doesn't require mu and sigma parameters to be calculated as a running average during training, since at test time the layer can normalize per instance in the same way as it does at train time. If the word striker appeared slightly earlier or later in the text, the text would still be about football, but different character positions would provide this information. The hidden states of the decoder are passed through a Dense layer to generate a distribution over the entire vocabulary for the next word in the sequence. A single GRU cell The process is as follows: 1. Colophon The animal on the cover of Generative Deep Learning is a painted parakeet (Pyrrhura picta). The Controller Until this point, we haven't mentioned anything about choosing an action. • A customer relations manager would only be interested in knowing if the senti- ment of an incoming email is positive or negative and wouldn't find much use in a generative model that could output examples of customer emails that don't yet exist. This can be thought of as the breeding stage, wherein we only use the best candi- dates to generate a new mean for the distribution. Let's now turn our attention to a more complex dataset and see the amazing things that variational autoencoders can achieve when we increase the dimensionality of the latent space. The architecture using the Functional API from keras.layers import Input, Flatten, Dense from keras.models import Model input_layer = Input(shape=(32,32, 3)) x = Flatten()(input_layer) x = Dense(units=200, activation = 'relu')(x) x = Dense(units=150, activation = 'relu')(x) output_layer = Dense(units=10, activation = 'softmax')(x) model = Model(input_layer, output_layer)> Here, we are using three different types of layer: Input, Flatten, and Dense. 131 Apples and Organges CycleGAN Your First CycleGAN Overview The Generators (U-Net) The Discriminators Compiling the CycleGAN Training the CycleGAN Analysis of the CycleGAN Creating a CycleGAN to Paint Like Monet The Generators (ResNet) Analysis of the CycleGAN Neural Style Transfer Content Loss Style Loss Total Variance Loss Running the Neural Style Transfer Analysis of the Neural Style Transfer Model Summary 132 135 137 137 139 142 144 146 147 149 150 151 153 154 156 160 160 161 162 6. Style X X Groove X ✓ Chords ✓ X Melody ✓ ✓ It is up to the MuseGAN to generate the next bar of music, given the current set of 128 instructions (one from each player). The model must include a stochastic (random) element that influences the individual samples generated by the model. Instead, the encoder will try to use the space around the origin symmetrically and efficiently. Furthermore, there are often long-term sequential dependencies between words that need to be captured by the model: for example, the answer to a question or carrying forward the context of a pronoun. Within its subfamily Arinae are several macaw and parakeet species of the Western Hemisphere. We can therefore sample from the standard normal distribution to generate new points in the space to be decoded (the red dots). Since the weights of the network are ini- tially randomized, unscaled input could potentially create huge activation values that immediately lead to exploding gradients. Figure 4-1. The reported statistics for each generation are as follows: 1. BERT can be built upon by appending output layers that are specific to a given down- stream task. In Figure 7-16 we can see the effect of changing just the melody noise input for the top line of music. The Challenges of Generative Modeling | 25 Figure 1-13. As we can see from Figure 1-6, the data-generating rule is simply a uniform distribu- tion over the land mass of the world, with no chance of finding a point in the sea. The rest of this book presents a set of case studies showing how generative modeling techniques can be applied to particular tasks. Figure 3-21. The network has a double-headed output, one for the next note name and one for the next note length. Positional Encoding The words are first passed through an embedding layer to convert each into a vector of length dmodel = 512. You will need to bear in mind that, like a set of building blocks, some layers will not fit together, simply because the input shape of one does not conform to the output shape of the other. This chapter introduces the recurrent neural network (RNN) architecture that allows us to tackle problems involving sequential data. 2. LSTM cells were first introduced in 1997 in a paper by Sepp Hochreiter and Jürgen Schmidhuber.1 In the paper, the authors describe how LSTMs do not suffer from the same vanishing gradient problem experienced by vanilla RNNs and can be trained on 1 Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," Neural Computation 9 (1997): 1735– 1780, . Closest matches of generated images from the training set GAN Challenges While GANs are a major breakthrough for generative modeling, they are also notori- ously difficult to train. In 2D this issue is subtle; the autoencoder only has a small number of dimensions to work with, so naturally it has to squash digit groups together, resulting in the space between digit groups being relatively small. The probabilities are passed through the sampler to output the next word, para- meterized by temperature. As we shall see, understanding how each layer of a neural network changes the shape of the data as it flows through the network is a key part of truly understanding the mechanics of deep learning. To check everything works as expected, from inside your virtual environment type python into your terminal and then try to import Keras (a deep learning library that we will be using extensively in this book). With this in place, we will then build our first example of a generative model (Naive Bayes) that is probabilistic in nature. How many episodes each process should run (e.g., 125, so 8 processes would cre- ate 1,000 episodes overall). We can see that the training of the network has been successful, because each genera- tor is visibly altering the input picture to look more like a valid image from the oppo- site domain. We could build an agent that contains a set of rules for how to respond to any given game state. The output from the recurrent layer is the final hidden state, a vector with the same length as the number of units. We explored two different types of recurrent layer, long short-term memory and GRU, and saw how these cells can be stacked or made bidirectional to form more complex network architectures. How- ever, like with the Coder brothers' art exhibition, we quickly found that there were some drawbacks to using plain autoencoders as a generative model—sampling from the learned latent space was problematic, for a number of reasons. Figure 2-7. In general, the shape of the output from a convolutional layer with padding="same" is: None, input height input width , , f ilters stride stride In the second convolutional layer, we choose the filters to be 3 × 3 and they now have depth 10, to match the number of channels in the previous layer. Coder can regenerate these points as original artwork. This code is shown in Example 6-6. Note that the weights of these copies are not shared. During training, the output distribution produced by the decoder at each timestep is compared against the true next word, to calculate the loss. The input sequence could be the note embeddings, or the hid- den state sequence from a previous recurrent layer. Finally, we took our first look at representation learning, an important concept that forms the core of many generative models. We apply Conv3D layers through- out the critic as we are working with 4D tensors. The biscuit tin dataset To us, it is obvious that there are two features that can uniquely represent each of these tins: the height and width of the tin. The second generator, G_BA, converts images from domain B into domain A. Your First LSTM Network As usual, you first need to get set up with the data. We follow this with a Dense layer consisting of 3,136 units… …which, after applying batch normalization and a ReLU activation function, is reshaped to a 7 × 7 × 64 tensor. This occurs when the agent finds a strategy that is rewarding in the dream environment, but does not generalize well to the real environment, due to the MDN-RNN not fully capturing how the true environment behaves under cer- tain conditions. Images from the new dataset are shown in Figure 1-9, and a sample of the pixel values for the first 10 observations appears in Table 1-3. He spent his days watching over the prisoners and had no time to follow his true passion of writing short stories. This is the calculation of the z vector reconstruction loss: the negative loglikelihood of observing the true z, under the mixture distribution parameterized by the output from the MDN-RNN. We'll be using the Quick, Draw! data- set from Google. Coder's original opinion. Example 4-3. Ini- tially, this is OK, but with every gust the pile becomes more and more unstable, until eventually the books have shifted so much that the pile collapses. Concatenate Layer The Concatenate layer simply joins a set of layers together along a particular axis (by default, the last axis). answer_labels A binary vector of length max_document_length (e.g., [0, 1, 1, 0, ...]). In the next chapter we shall move away from image-based generative modeling to a domain that presents new challenges: text-based generative modeling. Finally, a useful resource for training deep learning models on accelerated hardware is Google Colaboratory. Note that if the model was guessing randomly, it would achieve approximately 10% accuracy (because there are 10 classes), so 50% is a good result given that we have used a very basic neural network. In Keras, this is implemented as a wrapper around a recurrent layer, as shown here: layer = Bidirectional(GRU(100)) The hidden states in the resulting layer are vectors of length equal to double the num- ber of units in the wrapped cell (a concatenation of the forward and backward hidden states). As you can see from Figure 3-2, it works remark- ably well—customers who come to view the artwork very rarely complain that Mr. D. This is due to the fact that the error gradients can backpropagate freely through the network through the skip connections that are part of the residual blocks. The opening of Bach's Cello Suite No. 1 (Prelude) What note do you think comes next? In total, there are three changes we need to make to our WGAN critic to convert it to a WGAN-GP critic: • Include a gradient penalty term in the critic loss function. Technically, we would be justified in choosing any point in the 2D plane! It's not even guaranteed that points will be centered around (0,0). This is because we have chosen the filter to have height and width of 4 (kernel_size = (4,4)) and there are three channels in the preceding layer (red, green, and blue). In this chapter, we will explore the model in detail and show how it is possible to cre- ate your own version of this amazing cutting-edge technology. Indeed, the last hidden state is the overall output from the layer, and we will be making use of the fact that we can access the hidden state at each individual timestep later in this chapter. 160 | Chapter 5: Paint Example 5-12. The conductor tells the MuseGAN when the music it is producing is clearly distinguishable from real music, and the MuseGAN then adapts its internal wiring to be more likely to fool the conductor the next time around. The means and standard deviations used to normalize each layer are calculated per chan- nel and per observation. The slave nodes return the results to the master, which accumulates the results and then passes the overall result of the generation to the CMA-ES object. The algorithm rewards matrix weight- ings that generate actions that lead to overall high scores on the task, so that future generations are also likely to inherit this desired behavior. There are also examples of where the network has chosen to ignore a certain note nearby, as it doesn't add any additional information to its understanding of the phrase. The Transformer | 285 Figure 9-6. As well as covering core generative modeling theory, we will be building full working examples of some of the key models from the literature and walking through the codebase for each, step by step. The categorical cross-entropy loss has fallen from 1.842 to 1.357, result- ing in an accuracy increase from 33.5% after the first epoch to 51.9% after the tenth epoch. We can see this applied to a Lipschitz continuous 1D function in Figure 4-13 —at no point does the line enter the cone, wherever you place the cone on the line. This way, the decoder can generate a novel sequence of text, having been initialized with a representation of the input data produced by the encoder. Overall, we have seen how the GAN framework is extremely flexible and able to be adapted to many interesting problem domains. 70 | Chapter 3: Variational Autoencoders x = Activation('sigmoid')(x) decoder_output = x self.decoder = Model(decoder_input, decoder_output) Define the input to the decoder (the point in the latent space). To numerically measure how much two feature maps are jointly activated together, we can flatten them and calculate the dot product.14 If the resulting value is high, the fea- ture maps are highly correlated; if the value is low, the feature maps are not correlated. Since the creature is nocturnal, she only has a collection of nighttime photos of the beast that she once found lying on the floor of the jungle, dropped by another gani- mal enthusiast. A generative model must also be probabilistic rather than deterministic. Instead, we only want the image that is generated to fool the discriminator—that is, when the image is fed as input to the discriminator, we want the output to be close to 1. In this section we'll go through the essential notation required to read music and how we can represent this numerically, in order to transform music into the input data required to train a generative model. Figure 8-17. 170 | Chapter 6: Write Figure 6-2. Often when training GANs it is difficult to separate out vectors in the latent space corresponding to high-level attributes—they are frequently entangled, meaning that adjusting an image in the latent space to give a face more freckles, for example, might also inadvertently change the background color. It is able to understand from the context that this is likely to be the surname of a person and therefore likely to form part of an answer. 1 Kaiming He et al., "Deep Residual Learning for Image Recognition," 10 December 2015, 1512.03385. However, rather than creating a U shape by connecting layers from the downsam- pling part of the network to corresponding upsampling layers, a ResNet is built of residual blocks stacked on top of each other, where each block contains a skip con- nection that sums the input and output of the block, before passing this on to the next layer. An encoder–decoder network The final hidden state of the encoder can be thought of as a representation of the entire input document. We need the agent to discover the optimal values for the weights itself by experimenting within the environment and updating its weights based on received feedback. A few examples are shown in Figure 3-15. All rights reserved. To begin with, we shall be using the apples and oranges example from earlier to walk through each part of the CycleGAN and experiment with the architecture. We'll start by generating some music from scratch, by seeding the network with only a sequence of tokens (i.e., we are telling the model to assume it is starting from the beginning of the piece). Click the notebook you wish to run and, from the Kernel → Change kernel dropdown, select the generative virtual environment. In Keras, the BatchNormalization layer implements the batch normalization functionality: BatchNormalization(momentum = 0.9) The momentum parameter is the weight given to the previous value when calculating the moving average and moving standard deviation. The model architecture we shall test is shown here: Improving the Model | 55 input_layer = Input((32,32,3)) x = Conv2D(filters = 32, kernel_size = 3 , strides = 1, padding = 'same')(input_layer) x = BatchNormalization()(x) x = LeakyReLU()(x) x = Conv2D(filters = 32, kernel_size = 3, strides = 2, padding = 'same')(x) x = BatchNormalization()(x) x = LeakyReLU()(x) x = Conv2D(filters = 64, kernel_size = 3, strides = 1, padding = 'same')(x) x = BatchNormalization()(x) x = LeakyReLU()(x) x = Conv2D(filters = 64, kernel_size = 3, strides = 2, padding = 'same')(x) x = BatchNormalization()(x) x = LeakyReLU()(x) x = Flatten()(x) x x x x = = = = Dense(128)(x) BatchNormalization()(x) LeakyReLU()(x) Dropout(rate = 0.5)(x) x = Dense(NUM_CLASSES)(x) output_layer = Activation('softmax')(x) model = Model(input_layer, output_layer) We use four stacked Conv2D layers, each followed by a BatchNormalization and a LeakyReLU layer. Suppose we train the generator over several batches without updating the discriminator in between. This way, the bar gen- erator will be able to use this vector to fine-tune the content of every single bar and track independently. Autoencoders The preceding story is an analogy for an autoencoder, which is a neural network made up of two parts: • An encoder network that compresses high-dimensional input data into a lowerdimensional representation vector • A decoder network that decompresses a given representation vector back to the original domain This process is shown in Figure 3-4. The same multihead attention architecture is used in both the encoder and decoder, with a few small changes. In German, there are three definite articles (der, die, das) depending on the gender of the noun, but the Transformer knows to choose die because one attention head is able to attend to the word street (a feminine word in German), while another attends to the word to translate (the). I would like to thank you for your commitment and for going out of your way to share your knowledge of the subject with me. This technique is quite appropriately called maximum likelihood estimation. 250 | Chapter 8: Play Figure 8-7. For sequential data, the encoder– decoder process works as follows: 1. Each prisoner uses a specific thought process to update their own opinion, which involves balancing information from the new incoming word and other prisoners' opinions with their own prior beliefs. Model Architecture Figure 6-15 shows the overall model architecture that we will be building. This has helped us to visualize the inner work- ings of a VAE on the page and understand why the small tweaks that we made to the architecture of the autoencoder helped transform it into a more powerful class of net- work that can be used for generative modeling. 254 | Chapter 8: Play Figure 8-11. A single sequence as it flows through an embedding layer We embed each integer token into a continuous vector because it enables the model to learn a representation for each word that is able to be updated through backpropa- gation. Training the critic of the WGAN def train_critic(x_train, batch_size, clip_threshold): valid = np.ones((batch_size,1)) fake = -np.ones((batch_size,1)) # TRAIN ON REAL IMAGES idx = np.random.randint(0, x_train.shape[0], batch_size) true_imgs = x_train[idx] self.critic.train_on_batch(true_imgs, valid) # TRAIN ON GENERATED IMAGES noise = np.random.normal(0, 1, (batch_size, self.z_dim)) gen_imgs = self.generator.predict(noise) self.critic.train_on_batch(gen_imgs, fake) for l in critic.layers: weights = l.get_weights() weights = [np.clip(w, -clip_threshold, clip_threshold) for w in weights] l.set_weights(weights) Training the WGAN When using the Wasserstein loss function, we should train the critic to convergence to ensure that the gradients for the generator update are accurate. The Keras code to build the discriminators is provided in Example 5-3. At timestep t, the agent receives the current game state st and uses this to decide its next best action at, which it then performs. To counteract this problem, we use regularization techniques, which ensure that the model is penalized if it starts to overfit. Generative Modeling. There are then two learned param- eters for each channel, the scale (gamma) and shift (beta). Referring back to Figure 3-10, we can see several changes in how the latent space is organized. Download this dataset and place it inside the data folder of the book repository, in a folder called chorales. This presents additional challenges that we shall tackle later on in this chapter. Again, all other parts remain unaf- fected, but the baseline is different. For balance, we can also do the same procedure but shift the image one pixel down. More- over, Keras provides numerous useful building blocks that can be plugged together to create highly complex deep learning architectures through its functional API. Write. Prerequisites This book assumes that you have experience coding in Python. The inputs and outputs for the musical generative model The model we will be building is a stacked LSTM network with an attention mecha- nism. Note that there are no weights to be learned in a Concatenate layer; they are just used to "glue" previous layers together. Equally, if her father professes little confidence in where the markers should go, she The Variational Art Exhibition | 75 feels like she's the one doing all the work! His confidence in the marker placements that he provides has to be just right for her to be happy. Controller: Based on the description from the VAE (z) and the current hidden state from the RNN (h), my neural network outputs [0.34, 0.8, 0] as the next action. However, whether this setup would generate actions that could be classed as intelligent remains to be seen. One update step from the CMA-ES algorithm2 2 Reproduced with permission from David Ha, 2017, . We now have everything we need to build a U-Net generator in Keras, as shown in Example 5-2. The black dots show the mu values of each encoded image. The output from the fit method Evaluating the Model We know the model achieves an accuracy of 51.9% on the training set, but how does it perform on data it has never seen? In-Dream Training | 271 Challenges of In-Dream Training One of the challenges of training agents entirely within the MDN-RNN dream envi- ronment is overfitting. When people talk about "convolutional neural networks," they really mean "neural networks that contain convolutional lay- ers." It is important to make this distinction, because you shouldn't feel constrained to only use the architectures that you have read about in this book or elsewhere; instead, you should see them as examples of how you can piece together the different layer types. In this section we will introduce the attention mechanism1 that now forms an integral part of most sophisticated sequential generative models. However, since the generator is only graded against the current discriminator and the discriminator is constantly improving, we cannot com- pare the loss function evaluated at different points in the training process. Florida checks that if she sprays her oranges orange, they'll look exactly as they did originally. There's no reason to only stop at one attention head! In the paper, the authors choose eight heads that are trained in parallel, each outputting an nQ × dv matrix. It's the whole premise behind deep learning—a neural network trained to recognize the content of an image naturally learns higher-level features at deeper layers of the network by combining simpler features from previous layers. 152 | Chapter 5: Paint Neural Style Transfer So far, we have seen how a CycleGAN can transpose images between two domains, where the images in the training set are not necessarily paired. Notice that all of the examples we have seen so far produce novel observations from scratch—that is, there is no input apart from the random latent vector sampled from the latent space that is used to generate the images. We shall explore attention mechanisms in detail in the next chapter. This knowledge will come with experience and a solid understanding of how each layer changes the tensor shape as data flows through the network. 3 Phillip Isola et al., "Image-to-Image Translation with Conditional Adversarial Networks," 2016, https:// arxiv.org/abs/1611.07004. Tackling the GAN Challenges In recent years, several key advancements have drastically improved the overall sta- bility of GAN models and diminished the likelihood of some of the problems listed earlier, such as mode collapse. Bright green feathers cover most of a painted parakeet, but they are blue above the beak, brown in the face, and reddish in the breast and tail. The gradient penalty loss function def gradient_penalty_loss(y_true, y_pred, interpolated_samples): gradients = K.gradients(y_pred, interpolated_samples)[0] gradient_l2_norm = K.sqrt(K.sum(K.square(gradients), axis=[1:len(gradients.shape)]))) gradient_penalty = K.square(1 - gradient_l2_norm) return K.mean(gradient_penalty) The Keras gradients function calculates the gradients of the predictions for the interpolated images (y_pred) with respect to the input (interpolated_samples). Feel free to try both and observe that they give the same result. This is called a Gram matrix. In general, if example code is offered with this book, you may use it in your programs and documentation. WGAN-GP | 125 Figure 4-17. ReLU units can sometimes die if they always output zero, because of a large bias toward negative values preactivation. All GANs are characterized by a generator versus discriminator (or critic) architec- ture, with the discriminator trying to "spot the difference" between real and fake images and the generator aiming to fool the discriminator. First install the following libraries: sudo apt-get install cmake swig python3-dev \ zlib1g-dev python-opengl mpich xvfb \ xserver-xephyr vnc4server Then clone the following repository: 244 | Chapter 8: Play git clone As the codebase for this project is stored separately from the book repository, I sug- gest creating a separate virtual environment to work in: mkvirtualenv worldmodels cd WorldModels pip install -r requirements.txt Now you're good to go! Training Process Overview Here's an overview of the five-step training process: 1. The hidden state of the decoder RNN at each timestep is connected to a dense layer that outputs a probability distribution over the vocabulary of words. The Transformer architecture has inspired several subsequent models that make use of the multihead attention mechanism. Average reward of the generation (e.g., 136.44) 5. See how the covariance widens as the mean moves in large steps toward the minimum, but narrows as the mean set- tles into the true minimum. Why is this? There are no columns or rows in this dataset; instead, this is a tensor with four dimensions. This helps to solve the prob- lem mentioned earlier. Tokenization import re from keras.preprocessing.text import Tokenizer filename = "./data/aesop/data.txt" with open(filename, encoding='utf-8-sig') as f: text = f.read() seq_length = 20 start_story = '| ' * seq_length # CLEANUP text = text.lower() text = start_story + text text = text.replace('\n', start_story) Your First LSTM Network | 169 text = text.replace('\n', ' ') text = re.sub(' +', '. This is passed on to the next part of the network, the MDN-RNN. . Increasing the stride therefore reduces the size of the output tensor. The original paper represented a significant step forward in the field of style transfer as it showed how it was possible to train a model that could copy the style from a reference set of images onto a different image, without a training set of paired examples.2 Previous style transfer models, such as pix2pix,3 required each image in the training set to exist in both the source and target domain. In Figure 4-14 we show some of the samples generated by the WGAN. First, purely from a theoretical point of view, we should not be content with only being able to excel at categorizing data but should also seek a more complete under- standing of how the data was generated in the first place. It can be any- thing you want, as long as the output from the last layer of the decoder is the same size as the input to the encoder (since our loss function will be comparing these pixel- wise). The following code has been tested on Ubuntu 16.04, so it is spe- cific to a Linux terminal. The one-hot-encoded class labels. To build and run many of the deep learning models that we shall see later in this book, it is not essential to have a deep understanding of statistical theory. That responsibility lies with the controller. The output is simply the normalized input, scaled by gamma and shifted by beta. Example 5-1. In the code, the loss function for a VAE is simply the addition of the reconstruction loss and the KL divergence loss term. The output from three channels (feature maps) for three given input images —the darker orange colors represent larger values We can see that A and B are similar in style—both are grassy. Also, the dataset is automatically split into train, validation, and test sets. The latter is now considered best practice for training the most sophisticated GANs available today. Example 7-7. As babies, we are constantly exploring our surroundings, building up a mental model of possible futures with no apparent aim other than to develop a deeper understand- ing of the world. Suppose you wanted to describe your appearance to someone who was looking for you in a crowd of people and didn't know what you looked like. The views expressed in this work are those of the author, and do not represent the publisher's views. If the cur- rent observation suggests a left turn in the road ahead and you turn the wheel to the left, you expect the next observation to show that you are still in line with the road. However, you can also apply the same analysis to the rhythmic predictions of the model, which may be particularly relevant for other styles of music that you could use to train this model (such as a drum track). Figure 9-9 shows the self-attention mechanism from the paper. A diagram of the neural network trained on CIFAR-10 data The final step is to define the model itself, using the Model class. However, in the latent space, it's simply a case of adding 1 to the height latent dimension, then applying the mapping function to return to the image domain. GANs are highly sensitive to very slight changes in all of these parameters, and finding a set of parameters that works is often a case of educated trial and error, rather than following an established set of guidelines. This weighting term is one of the parameters to tune when you're training your VAE. Keras gives us the flexibility to be able to build a model that can handle the pitch and duration prediction simultaneously. The output of the training process is shown in Figure 8-21. This network takes the four input tensors and con- verts them into a multitrack, multibar score. Here we are trying to find the minimum point of a highly nonlinear function in two dimensions—the value of the function in the red/black areas of the image is greater than the value of the function in the white/yellow parts of the image. Reshape this vector into a tensor that can be fed as input into the first convolu- tional transpose layer. The order in which to use the BatchNormalization and Activa tion layers is a matter of preference. However, unlike in a VAE, there are also skip connections between equivalently shaped layers in the upsampling and downsampling parts of the network. The MDN-RNN can then subse- quently learn how the latent vectors evolve over time. Why 481? The same principle holds for machine learning. The first dimension of this array references the index of the image in the dataset, the sec- ond and third relate to the size of the image, and the last is the channel (i.e., red, green, or blue, since these are RGB images). 3 We are making the assumption here that each chorale in the dataset has four beats in each bar, which is rea- sonable, and even if this were not the case it would not adversely affect the training of the model. If the output word is the start story token, we stop generating any more words as this is the model telling us it wants to end this story and start the next one! Otherwise, we append the new word to the seed text, ready for the next iteration of the generative process. She of course takes her father's opinion into account, so she usually places the marker somewhere near the point that he suggests. The StyleGAN generator architecture16 StyleGAN solves the entanglement problem by borrowing ideas from the style trans- fer literature. For this reason, I generally prefer to use RMSE as the loss function. This is then transformed through a densely connected layer (A) to generate the ⊮b, i and ⊮s, i vectors, both of length n (the number of channels output from the convolutional layer in the synthesis network). I certainly couldn't have completed this project without you, and I'm grateful for the time you have invested in helping me restructure and expand parts of the book that needed more explanation. If you are not familiar with Python, the best place to start is through LearningPython.org. The es object uses this list of rewards to calculate the new normal distribution as per Figure 8-15. The process of finding these weights is what we mean by training the network. The encoder–decoder architecture introduced in this chapter is an important generative tool as it allows sequence data to be condensed into a single vector that can then be decoded into another sequence. The first 50 dimensions in our latent space are shown in Figure 3-19. Due to concerns about how this model could be abused by malicious parties, for example to generate fake news, forged essays, fake accounts on social media,

RNN Extensions | 185 2. Wasserstein Loss Let’s first remind ourselves of binary cross-entropy loss - the function that that we are currently using to train the discriminator and generator of the GAN: 4 Martin Arjovsky et al., “Wasserstein GAN,” 26 January 2017, . How many parame- ters do we need to define these distributions? The activation function is critical to ensure the neural network is able to learn complex functions and doesn’t just output a linear combination of its input. Could it be that before long, we will be able to tune into a radio station that plays music in our favorite style nonstop, so that we never hear the same thing twice? For a full description of the innovations introduced by BigGAN, I recommend read- ing the original paper and accompanying presentation material. The Functional API will give you complete freedom over the design of your deep neural network. On Planet Pixel, the assumption that every pixel value is independent of every other pixel value clearly doesn’t hold. 7 Source: Karras et al., 2017. A good understanding of how the shape of the tensor changes as data flows through the network is also pivotal to building successful networks, and recur- rent layers require particular care in this regard as the time dimension of sequential data adds additional complexity to the transformation process. In Figure 5-12 we show the output from each generator at various stages of the early training process, to show the progression as the model begins to learn how to convert Monet paintings into photographs and vice versa. It’s worth noting that even though the digit labels were never shown to the model during training, the autoencoder has naturally grouped digits that look alike into the same part of the latent space. While both discriminative modeling and generative modeling aim to minimize a loss function over a dataset of observations, reinforcement learning aims to maximize the long-term reward of an agent in a given environment. WGAN-GP CelebA examples Clearly the model has learned the significant high-level attributes of a face, and there is no sign of mode collapse. This new word is then fed back into the cell as usual, and the pro- cess continues until the full story is completed. An example of how one attention head attends to the word “the” and another attends to the word “street” in order to correctly translate the word “the” to the German word “die” as the feminine definite article to “Straße” This gives the Transformer the ability to translate extremely complex and long sen- tences, as it can incorporate information from several places across the input sentence and current translation to form its decision about the next word. This completes the description of the orchestra. Also, text generation can be handled one word at a time. Good progress has been made in text modeling, but solutions to the above problems are still ongoing areas of research. It is a similar idea to convolutional 104 | Chapter 4: Generative Adversarial Networks transpose, but instead of filling the gaps between pixels with zeros, upsampling just repeats the existing pixel values. The cell state is the same length as the hidden state (the number of units in the cell). Training the VAE The network architecture for the faces model is similar to the digits example, with a few slight differences: 1. Imagine if we could build a generative model that doesn’t model possible futures of the environment given an action, as per the world models example, but instead includes its own action-generating process as part of the environment to be modeled. question_input_tokens The tokenized question (e.g., [2, 39, 1, 52, ...]), padded with zeros to be of length max_question_length (another parameter). The LSTM model with attention for predicting the next note in a sequence You can train this LSTM with attention by running the notebook called 07_02_lstm_compose_train.ipynb in the book repository. Coder also provides his daughter with an indication of how sure he is that the marker should be placed at the given point. Figure 8-8. The new process works as follows. The encoder model goes one step further by sampling a particular z vector from this distribution. | 7 Figure 1-4. Example 6-2. The decoder in the righthand half of the diagram then uses this encoding to generate output words one at a time, using previous words as additional input into the model. As mentioned previously, the decoder of a variational autoencoder is identical to the decoder of a plain autoencoder. Take a moment to consider what an incredible gener- ative model you are. Notice how we freeze the weights in the discriminator, as is typical with GANs, so that the combined model only trains the generator weights, even though the dis- criminator is involved in the model. Two images that contain similar-looking scenes (e.g., a photo of a row of buildings and another photo of the same buildings taken in different light from a different angle) should have a smaller loss than two images that contain completely different scenes. You can start training the MuseGAN using the notebook 07_04_musegan_train.ipynb in the book repository. For example, in the first The Transformer | 281 encoder layer, both inputs are the positionally encoded embedding of the input sequence. An LSTM network is a particular type of recurrent neural network (RNN). There is no great difference in style between the two bars, but the whole passage has changed from the original generated score. The brothers carefully monitor the loss of revenue at the ticket office caused by customers asking for money back because of badly reconstructed paintings, and are constantly trying to find a system that minimizes this loss of earnings. Three of the most commonly used are mean squared error, categorical crossentropy, and binary cross-entropy. Collect data to train the MDN-RNN Once we have a trained VAE, we use it to encode each of the collected observations into mu and log_var vectors, which are saved alongside the current action and reward. The discriminative modeling process One key difference is that when performing discriminative modeling, each observa- tion in the training data has a label. The controller uses an evolutionary algorithm, CMA-ES (Covariance Matrix Adapta- tion Evolution Strategy), as its optimizer. Total variance loss We would like the combined image to appear smooth rather than pixelated. Creating a CycleGAN to Paint Like Monet Now that we have explored the fundamental structure of a CycleGAN, we can turn our attention to more interesting and impressive applications of the technique. For example, for image classification, the key breakthrough came in 2012 when a team led by Geoff Hinton at the University of Toronto won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with a deep convolutional neural network. It is an excellent example of how generative modeling can be used to solve practical problems, when applied alongside other machine learning techniques such as reinforcement learning. The correct translation of the sentence is: Ich habe im Fußballspiel gegen England einen Elfmeter erzielt. This results in less vibrant images. The MuseGAN model takes four distinct noise tensors as input and outputs a generated multitrack, multibar score. To address this, we could assign an additional pseudocount of 1 to each possible com- bination of features. For this example, we’ll use lowercase word tokenization, without word stemming. An LSTM cell The hidden state is updated in six steps: 1. The attention mechanism is just a set of layers that converts the previous encoder hidden states and current decoder hidden state into the summation weights that generate the context vector. To achieve this, we could simply assign a probability to each possible combination of features, based on the data we have seen. 86 | Chapter 3: Variational Autoencoders Figure 3-15. If our model is merely a fixed calculation, such as taking the average value of each pixel in the dataset, it is not generative because the model produces the same output every time. The Keras code to build the overall generator is provided in Example 7-7. In some cases, however, this may not be desirable; for example, some proper nouns, such as names or places, may benefit from remaining capi- talized so that they are tokenized independently. The next match is in two weeks time, on July 31st 2005 . If we look at the feature maps, we can see that the green and spikiness channels often fire strongly together at the same spatial point in images A and B, but not in image C. The con- text vector ci is concatenated with the incoming data yi–1 to form an extended vector of data into each cell of the decoder. However, this still fails to be a satisfactory generative model, because the probability of observing a point not in the original dataset is just a constant. The distribution of possible next notes over time (at epoch 20): the darker the square, the more certain the model is that the next note is at this pitch An interesting point to note here is that the model has clearly learned which notes belong to particular keys, as there are gaps in the distribution at notes that do not belong to the key. This is why it is important to add randomness to the population, to ensure we are not too narrow in our search field. 14 | Chapter 1: Generative Modeling hairColor Red SilverGray Blonde Red SilverGray Black SilverGray SilverGray topType ShortHairShortFlat ShortHairFrizzle ShortHairShortFlat LongHairStraight NoHair LongHairStraight LongHairStraight ShortHairShortFlat face_id accessoriesType clothingColor clothingType hairColor topType 8 Round Pink Hoodie SilverGray LongHairStraight 9 Round PastelOrange ShirtScoopNeck Blonde LongHairStraight The possible values for each feature include: • 7 different hairstyles (topType): — NoHair, LongHairBun, LongHairCurly, LongHairStraight, ShortHairShort- Waved, ShortHairShortFlat, ShortHairFrizzle • 6 different hair colors (hairColor): — Black, Blonde, Brown, PastelPink, Red, SilverGray • 3 different kinds of glasses (accessoriesType): — Blank, Round, Sunglasses • 4 different kinds of clothing (clothingType): — Hoodie, Overall, ShirtScoopNeck, ShirtVNeck • 8 different clothing colors (clothingColor): — Black, Blue01, Gray01, PastelGreen, PastelOrange, Pink, Red, White There are 7 × 6 × 3 × 4 × 8 = 4,032 different combinations of these features, so there are 4,032 points in the sample space. pix2pix dataset and domain mapping example While pix2pix only works in one direction (from source to target), CycleGAN trains the model in both directions simultaneously, so that the model learns to translate images from target to source as well as source to target. The architecture using a Sequential model from keras.models import Sequential from keras.layers import Flatten, Dense model = Sequential([Dense(200, activation = 'relu', input_shape=(32, 32, 3)), Flatten(), Dense(150, activation = 'relu'), Dense(10, activation = 'softmax'),]) Your First Deep Neural Network | 37 Example 2-2. Some stu- dents try to memorize the answers to these questions, but then come unstuck in the exam because they haven’t truly understood the subject matter. Training the VAE The VAE Architecture As we have seen previously, the Keras functional API allows us to not only define the full VAE model that will be trained, but also additional models that reference the encoder and decoder parts of the trained network separately. The controller is a densely connected neural network, where the input is a concatena- tion of z (the current latent state sampled from the distribution encoded by the VAE) and the hidden state of the RNN. Using temperature to control dream environment volatility4 4 Source: Ha and Schmidhuber, 2018. We shouldn’t be surprised that the model resorts to using the tag in this context, as many of the niche words in the original corpus would be tokenized this way. The input to the generator is a vector, usually drawn from a multivariate standard normal distribution. If you run the predict method of this model for the same input multiple times, you Training the VAE | 249 will get different output, since even though the mu and log_var values are con- stant, the randomly sampled z vector will be different each time. This makes your model very flexible and able to be easily tested and ported to other projects as necessary. It’s really important to understand how the shape of the tensor changes as data flows through from one convolutional layer to the next. By sampling from this model, we are able to generate new data. answer_token_ranges The token positions of the answer in the story text (e.g., 24:27). Again, this makes sense; there is probably enough information contained in the previous six hidden states to understand how the phrase should continue. Probabilistic Generative Models | 11 Thus, each density function pθ x in this parametric model (i.e., each box) can be uniquely represented by four numbers, θ = θ1, θ2, θ3, θ4 . To demonstrate this, Figure 6-10 shows the top 10 words with the highest probabili- ties to follow various sequences (not from the training set). When customers bring their fruit to the self-checkout tills, they sometimes errone- ously select the wrong option on the machine. Figure 3-2. The players in the chords section only produce one instruction per bar that then applies to every instrumental track. It’s important to remember that all of the cells in this diagram share the same weights (as they are really the same cell). Therefore, using the Wasserstein loss removes one of the key difficulties of training GANs—how to balance the training of the discriminator and generator. • Point C can be deemed a success because it could have been generated by pdata and is suitably different from any point in the original dataset. In a VAE, we want to measure how different our normal distribution with parameters mu and log_var is from the standard normal distribution. Likelihood The likelihood ℒ θ | x of a parameter set θ is a function that measures the plausibil- ity of θ, given some observed point x. OpenAI Gym OpenAI Gym is a toolkit for developing reinforcement learning algorithms that is available as a Python library. Analysis of the Transformer The Tensorflow GitHub repository contains a Colab notebook where you can play around with a trained Transformer model and see how the attention mechanisms of the encoder and decoder impact the translation of a given sentence into German. Example 6-3. If the sole goal of a brain is to minimize the amount of surprise between the actual input stream of data and the model of the future input stream, then the brain must find a way to make its actions create the future that it expects. For predictive classification models, this is all we require, so we could con- nect this to a final Dense layer to output the probability of a particular class being present in the image. Figure 2-4 shows ReLU, LeakyReLU, and sigmoid activation functions side by side for comparison. The loss function is usually chosen to be either the root mean squared error (RMSE) or binary cross-entropy between the individual pixels of the original image and the reconstruction. In this section we will introduce four such developments—Pro- GAN, SAGAN, BigGAN, and StlyeGAN. 296 | Chapter 9: The Future of Generative Modeling Figure 9-15. Evaluate each agent in the environment, returning the average reward over multiple episodes. The output from the layer is the final hidden state of the cell, after each word in the input sequence has been processed. Figure 6-1. Therefore, much emphasis in recent years has been placed on training discriminative models to reach human or superhuman performance in a variety of image or text classification tasks. Training the controller Each line of the output represents one generation of training. Coder decide where to place the markers? First, a trip to the local prison, where the inmates have formed a literary society… The Literary Society for Troublesome Miscreants Edward Sopp hated his job as a prison warden. World Model Architecture | 241 Figure 8-3. The Transformer | 277 Figure 9-2. Headshots of 50 Wrodlers7 As you’re a data scientist at heart, you decide to deploy a generative model to solve the problem. A key component of the architecture is a generative model that can construct a prob- ability distribution for the next possible state, given the current state and action. 134 | Chapter 5: Paint Figure 5-3. Here, the query input comes from the previous layer of the decoder and the key–value input comes from the encoder. 2015, . It consists of a cell that updates its hidden state, ht, as each element of the sequence xt is passed through it, one timestep at a time. Probabilistic Generative Models | 21 Figure 1-10. The best students use the practice material to further their general understanding, so that they are still able to answer correctly when faced with new questions that they haven’t seen before. To understand what note or sequence of notes is likely to follow from a particular given passage of music, it may be crucial to use information from far back in the sequence, not just the most recent information. We saw in Chapter 3 how a VAE can take a high-dimensional input image and con- dense it into a latent random variable that approximately follows a standard multi- variate normal distribution, through minimization of the reconstruction error and KL divergence. Using a neural network to expand the input vector along the time axis means the model has a chance to learn how music flows across bars, rather than treating each bar as completely independent of the last. Clearly, the meaning of the entire sentence would be changed by replacing the word scored with missed. This gate is simply a dense layer with weights matrix Wf, bias bf, and a sigmoid activation function. This process continues for many days. 6 Tero Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” 27 October 2017, Advances in Image Generation | 287 Figure 9-7. Diagram of a multihead attention module, followed by the add & norm layer The multihead attention layer requires two inputs: the query input, xQ, and the key– value input, xKV. It does this by creating representations of the key aspects of the visual and auditory fields and modeling how these latent representations will evolve over time. In our example, as required, the shape of our Input layer matches the shape of x_train and the shape of our Dense output layer matches the shape of y_train. Figure 3-22 shows this process in action. Training the MuseGAN to do this isn’t easy. Binary cross-entropy places Autoencoders | 71 heavier penalties on predictions at the extremes that are badly wrong, so it tends to push pixel predictions to the middle of the range. Current elapsed time in seconds (e.g., 2395) 4. The cube therefore represents the input tensor to the normalization layer. This means we need to change the number of channels in the final convolutional transpose layer of the decoder to 3. The RandomWeightedAverage layer creates the interpolated images, which are then also passed through the critic. There’s much more where this came from. For exam- ple, in a game of chess, checkmating the opponent’s king has a reward of 1 and every other move has a reward of 0. Frames 40 to 59 of one episode Training the VAE We can now build a generative model (a VAE) on this collected data. Hav- ing built up an understanding of the underlying physics of the environment through random movements, the model is then able to train itself from scratch on a new task, entirely within its own internal representation of the environment. Notice how we set return_sequences to True to make each layer pass the full sequence of hidden states to the next layer, rather than just the final hidden state. Sequence modeling has primarily been driven by the invention of the Transformer, an attention-based module that removes the need for recurrent or convolutional neural networks entirely and now powers most state-of-the-art sequential models, such as BERT, GPT-2, and MuseNet. Each section gives instructions to the MuseGAN and has a distinct responsibility within the orchestra. The master groups all the scores together and sends this list to the es object. We can train the discriminator by creating a training set where some of the images are randomly selected real observations from the training set and some are outputs from the generator. Output from the neural style transfer process at 1, 200, and 400 iterations We can see that with each training step, the algorithm becomes stylistically closer to the style image and loses the detail of the base image, while retaining the overall con- tent structure. First, we must place ourselves at the very edge of the generative modeling landscape, among the most radical, innovative, and leading ideas in the field. To pre- vent this from happening, you need to understand the root cause of the exploding gradient problem. Here we have to use Conv3D layers, which are analo- gous to Conv2D layers but accept four-dimensional input tensors (n_bars, n_steps_per_bar, n_pitches, n_tracks). Inference To test the model on an input document sequence that it has never seen before, we need to run the following process: 1. encoder This accepts a 64 × 64 × 3 observation as input and outputs a sampled z vector. The Progressive GAN training mechanism, and some example generated faces7 The earlier layers are not frozen as training progresses, but remain fully trainable. Some- times your network can be happily training for hours when suddenly the loss func- tion returns NaN and your network has exploded. The LSTM layer can accept any length of sequence as input, but the longer the sequence is the more time it will take to generate the next word, so the sequence length should be capped. Once the sequence is finished, the layer outputs the final hidden state of the cell, hn, which is then passed on to the next layer of the network. Example 6-5. To ensure the shop is run as efficiently as possible, they each look after different areas—specifically, Granny Smith takes great pride in her apple selection and Florida spends hours ensuring the oranges are perfectly arranged. Therefore for the GAN discriminator, minimizing the loss function can be written as follows: GAN discriminator loss minimization minD − ℮x ~ p X log D x + ℮z ~ p Z log 1 − D G z To train the GAN generator G, we calculate the loss when comparing predictions for generated images pi=D(G(zi)) to the response yi=1. Based on the original paper, we will be building a reinforcement learning algorithm that learns how to drive a car around a racetrack as fast as possible. We shall be using a latent space with two hundred dimensions instead of two. This object updates the 3 Source: . We use a stride of 2 on some of the layers to reduce the size of the output. However, since then, it has been shown that convolutional layers give greater pre- dictive power to the discriminator. As in previous chapters, you can instantiate a CycleGAN object in the notebook, as shown in Example 5-1, and play around with the parame- ters to see how it affects the model. How- ever, many of the models in this book require that the output from a layer is passed to multiple separate layers beneath it, or conversely, that a layer receives input from multiple layers above it. However, more worryingly, even points that are right in the middle of the domain may not be decoded into wellformed images. Reloads the virtualenvwrapper initialization script To create a virtual environment called generative, simply enter the following into your terminal: mkvirtualenv generative You’ll know that you’re inside the virtual environment because your terminal will show (generative) at the start of the prompt. In order to generate passages that have greater chance of being semanti- cally reasonable, we can build a human-assisted text generator, where the model out- puts the top 10 words with the highest probabilities and it is then ultimately up to a human to choose the next word from among this list. Finally, Chapter 9 summarizes the current landscape of generative modeling and looks back on the techniques that have been presented in this book. The final part of the encoder consists of a feed-forward (densely connected) layer applied to each position separately. This is because the VAE is trying to minimize the KL divergence, so it tries to differ from the standard normal distribution in as few dimensions as possible. In this step, we pass all of the random rollout data through the encoder_mu_log_var model and store the mu and log_var vectors corresponding to each observation. The horizontal axis shows the generated sequence of notes; the vertical axis shows where the attention of the network was aimed when predicting each note along the horizontal axis (i.e., the alpha vector). It’s a really useful and fun dataset for learning the fundamentals of deep learning. Figure 5-17. The bar generator is a neural network that uses convolutional transpose layers to expand the time and pitch dimensions. It is also true that GANs are generally more difficult to train than VAEs and take longer to reach a satisfactory quality. A diagram of a convolutional neural network We can use the model.summary() method to see the shape of the tensor as it passes through the network (Figure 2-14). | 175 Learn more at oreilly.com/online-learning A deep learning model, on the other hand, can learn how to build high-level informative features by itself, directly from the unstructured data. Figure 1-8 shows 10 observations sampled from the model. A brief history of generative modeling: green marks represent ideas that are covered in this book and red marks are ideas that we shall explore in this chapter 276 | Chapter 9: The Future of Generative Modeling This is by no means an exhaustive list; there are dozens of GAN flavors that are groundbreaking in their own fields (e.g., video generation or text-to-image genera- tion). Therefore, the number of parameters (or weights) in the layer is (4 × 4 × 3 + 1) × 10 = 490, where the + 1 is due to the inclusion of a bias term attached to each of the filters. If we sample from this distribu- tion, we know that we’re very likely to get a point that lies within the limits of what the VAE is used to seeing. Sopp’s Fables. Incorporating a signifi- cant amount of example code from this book into your product’s documentation does require permission. This is defined as follows: − n ∑ yi log i=1 pi Finally, if you are working on a binary classification problem with one output unit, or a multilabel problem where each observation can belong to multiple classes simulta- neously, you should use binary cross-entropy: − 1 n yi log pi + 1 − yi log 1 − pi ni∑ =1 The optimizer is the algorithm that will be used to update the weights in the neural network based on the gradient of the loss function. She also doesn’t like it when he is too strict about where the markers should be placed, as then she feels she doesn’t have enough responsibility. We need to output a different vec- tor for every bar, as its job is to control the general dynamic nature of the music over time. Then the network performs a series of training steps. The resulting output is a 128 × 128 × 3 image that we can view. With style transfer, our aim isn’t to model the underlying distribution of the style images, but instead to extract only the stylistic components from these images and embed these into the base image. It also has length equal to the number of units in the cell and contains values between –1 and 1. We will be working through the Jupyter notebook in the book repository called 02_01_deep_learning_deep_neural_network.ipynb. In-Dream Training So far, the controller training has been conducted using the OpenAI Gym CarRacing environment to implement the step method that moves the simulation from one state to the next. Generative Adversarial Networks. Increasing this parameter magnifies the variance when sampling z through the MDN-RNN, leading to more volatile rollouts when training in the dream environment. If you already have a good understanding of probability, that’s great and much of the next section may already be familiar to you. The overall variety is poor and some pieces of artwork do not really resemble single-digit numbers. 97 Ganimals Introduction to GANs Your First GAN The Discriminator The Generator Training the GAN GAN Challenges Oscillating Loss iv | Table of Contents 97 99 100 101 103 107 112 112 Mode Collapse Uninformative Loss Hyperparameters Tackling the GAN Challenges Wasserstein GAN Wasserstein Loss The Lipschitz Constraint Weight Clipping Training the WGAN Analysis of the WGAN WGAN-GP The Gradient Penalty Loss Analysis of WGAN-GP Summary Part II. The Bar Generator The bar generator converts a vector of length 4 * z_dim to a single bar for a single track—i.e., a tensor of shape [1, n_steps_per_bar, n_pitches, 1]. You may see this type of GAN called a DCGAN (deep convolutional generative adversarial network) in the literature, but now essen- tially all GAN architectures contain convolutional layers, so the “DC” is implied when we talk about GANs. It is also common to see batch normalization layers in the discriminator for vanilla GANs, though we choose not to use them here for simplicity. 94 | Chapter 3: Variational Autoencoders Mathematically, we are traversing a straight line, which can be described by the fol- lowing equation: z_new = z_A * (1- alpha) + z_B * alpha Here, alpha is a number between 0 and 1 that determines how far along the line we are, away from point A. 182 | Chapter 6: Write RNN Extensions The network in the preceding section is a simple example of how an LSTM network can be trained to learn how to generate text in a given style. Your First CycleGAN | 139 The intuition here is that with each subsequent layer in the downsampling part of the network, the model increasingly captures the what of the images and loses informa- tion on the where. 228 | Chapter 7: Compose Melody The melody input is an array of shape [n_tracks, z_dim]—that is, we provide the model with a random noise vector of length z_dim for each track. We can compile the two discriminators directly, as we have the inputs (images from each domain) and outputs (binary responses: 1 if the image was from the domain or 0 if it was a generated fake). This process repeats until the specified number of epochs have elapsed. This way, the loss gradually decreases with each iteration and we end up with an image that merges the content of one image with the style of another. Deep Learning. Again, relating this back to our story, this term represents Epsilon’s annoyance at hav- ing to move the ladder away from the middle of the wall (mu different from 0) and also if Mr. N. Attention The attention mechanism was originally applied to text translation problems—in par- ticular, translating English sentences into French. This has the effect of giving each bar a distinct musical character, for example, through a change of chord. In other words, there is a limit on the rate at which the line can rise or fall at any point. One way we can achieve this is by defining an additional Keras model (question_model) that accepts the current word token and current decoder hidden state as input, and outputs the predicted next word distribution and updated decoder hidden state. Some poorly generated images Partly, this is because of the large spaces at the edge of the domain where there are few points—the autoencoder has no reason to ensure that points here are decoded to legible digits as very few images are encoded here. To recap, the length of ht is equal to the number of units in the LSTM. For example, the word water can be used either as a verb (I need to water the plant) or as a noun (The ocean is full of water). Your First Music-Generating RNN | 211 Figure 7-7. The weights are calculated through applying a softmax activation to the align- ment values. The model also gives us the reverse process for free, converting paintings into realistic photographs. You are free to choose whichever you are most comfortable with, or whichever library works fastest for a particular network archi- tecture. 5 Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” 1 September 2014, . However, this process can be parallelized, since there are no dependencies between individual simulations. Figure 7-15. This is immediately followed by a Dropout layer for regularization, and the net- work is concluded with an output Dense layer of size 10. Compiling the WGAN-GP critic from functools import partial ### COMPILE CRITIC MODEL self.generator.trainable = False real_img = Input(shape=self.input_dim) z_disc = Input(shape=(self.z_dim,)) fake_img = self.generator(z_disc) fake = self.critic(fake_img) valid = self.critic(real_img) interpolated_img = RandomWeightedAverage(self.batch_size)([real_img, fake_img]) validity_interpolated = self.critic(interpolated_img) partial_gp_loss = partial(self.gradient_penalty_loss, interpolated_samples = interpolated_img) partial_gp_loss.__name__ = 'gradient_penalty' self.critic_model = Model(inputs=[real_img, z_disc], outputs=[valid, fake, validity_interpolated]) self.critic_model.compile(loss=[self.wasserstein,self.wasserstein, partial_gp_loss] ,optimizer=Adam(lr=self.critic_learning_rate, beta_1=0.5) ,loss_weights=[1, 1, self.grad_weight]) Freeze the weights of the generator. Example 6-7. Figure 4-18. The moving average and standard deviation also need to be calculated for each channel but since they are derived from the data passing through the layer rather than trained through backpropagation, they are called nontrainable parameters. The overall process is shown in Figure 5-2. For example, we could feed the model the tokens for the greedy cat and the and would expect the model to output a suitable next word (e.g., dog, rather than in). To do this we first need to find a vector in the latent space that points in the direction of increasing smile. If we achieve this goal, we can sam- ple from pmodel to generate observations that appear to have been drawn from pdata. Next we’ll look at ways of improving this model, by introducing a few new layer types. An example of how GPT-2 can extend a given system prompt4 If you are slightly terrified at how realistic this appears, you are not alone. Your challenge is to choose a different point x = x1, x2 in the space that looks like it has been generated by the same rule. These word vectors have been cre- ated as part of the Stanford GloVe (“Global Vectors”) project, which uses unsuper- vised learning to obtain representative vectors for a large set of words. Because the input is scaled, it’s natural to expect the activations from all future layers to be relatively well scaled as well. Standard deviation of the rewards (e.g., 62.78) 8. To encode an image into a specific point z in the latent space, we can sample from this distribution, using the following equation: z = mu + sigma * epsilon where4 sigma = exp(log_var / 2) 4 σ = exp 80 | log σ = exp 2 log σ /2 = exp log σ 2 /2 Chapter 3: Variational Autoencoders epsilon is a point sampled from the standard normal distribution. WGAN-GP One of the most recent extensions to the WGAN literature is the Wasserstein GAN– Gradient Penalty (WGAN-GP) framework.6 The WGAN-GP generator is defined and compiled in exactly the same way as the WGAN generator. For example, the area of 1’s is far bigger than the area for 8’s, so when we pick points randomly in the space, we’re more likely to sample something that decodes to look like a 1 than an 8. If we treat this as a supervised learning problem, we can train the dis- criminator to learn how to tell the difference between the original and generated images, outputting values near 1 for the true images and values near 0 for the fake images. The Transformer The Transformer was first introduced in the 2017 paper “Attention is All You Need,”1 where the authors show how it is possible to create powerful neural networks for sequential modeling that do not require complex recurrent or convolutional architec- tures but instead only rely on attention mechanisms. With no more than 10 or so of these statements, the person would be able to map the description back into pixels to generate an image of you in their head. Now let’s see what happens when this assumption breaks down Hello Wrodl! Continued You feel a certain sense of pride as you look upon the 10 new creations generated by your Naive Bayes model. This time, instead of one input channel (grayscale) we have three (red, green, and blue). We also show one of the paintings that has been marked on the wall at the point [–3.5, –0.5] by Mr. N. While it is possible to manufacture this kind of dataset for some style problem settings (e.g., black and white to color photos, maps to satellite images), for others it is impossible. Update the population pool by adding the newly created agents and removing poorly performing agents. Summary | 235 CHAPTER 8 Play In March 2018, David Ha and Jürgen Schmidhuber published their “World Models” paper.1 The paper showed how it is possible to train a model that can learn how to perform a particular task through experimentation within its own generative halluci- nated dreams, rather than inside the environment itself. This will be the predicted probability that the image is real. This is known as scaled dot-product attention. Overview A CycleGAN is actually composed of four models, two generators and two discrimi- nators. This makes Di’s task harder, since now Gene’s photos aren’t quite as easy to distinguish from the real photos, so she must again learn how to improve. This com- pletes the first epoch. Instance Normalization Layer The generator of this CycleGAN uses InstanceNormalization layers rather than BatchNormalization layers, which in style transfer problems can lead to more satis- fying results.8 8 Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, “Instance Normalization: The Missing Ingredient for Fast Stylization,” 27 July 2016, . He was running low on inspiration and needed to find a way to generate new content. For these kinds of problems, we can use a type of network known as an encoder– decoder. Collecting Random Rollout Data | 247 Figure 8-5. This is already quite impressive given that it has never seen this text before and also may not have seen certain words from the document that are included in the answer, such as Bloggs. Figure 8-6. He therefore needs to be careful that he doesn’t end up paying his daughter too much while also monitoring the loss of revenue at the ticket office. This makes it 165 very difficult to train a model to generate coherent text, as every word is vital to the overall meaning of the passage. The overall loss is the sum of these three losses, with the gradient loss weighted by a factor of 10, in line with the recommendations from the original paper. We now have all the necessary terminology to start describing how we can build a probabilistic generative model. If the weighting term is too small, the KL divergence loss will dominate and the reconstructed images will be poor. The training loop for the neural style transfer model from scipy.optimize import fmin_l_bfgs_b iterations = 1000 x = preprocess_image(base_image_path) for i in range(iterations): x, min_val, info = fmin_l_bfgs_b(evaluator.loss , x.flatten() , fprime=evaluator.grads , maxfun=20) The process is initialized with the base image as the starting combined image. By repeatedly Reinforcement Learning | 239 calling the step method with the actions chosen by the agent, you can play out an episode in the environment. Coder, run an art gallery. Probability Density Function A probability density function (or simply density function), p x , is a function that maps a point x in the sample space to a number between 0 and 1. The key to GANs lies in how we alternate the training of the two networks, so that as the generator becomes more adept at fooling the discriminator, the discriminator must adapt in order to maintain its ability to correctly identify which observations are fake. This network can be built in Keras as shown in Example 7-3. 280 | Chapter 9: The Future of Generative Modeling Figure 9-4. The latent space of biscuit tins and the function f that maps a point in the latent space to the original image domain Representation learning comes so naturally to us as humans that you may never have stopped to think just how amazing it is that we can do it so effortlessly. Table 1-1. Another excellent resource for getting started with music generation using Python is Sigurður Skúli’s blog post and accompanying GitHub repository. With discrete text data, we can’t apply backpropagation in the usual manner, so we need to find a way around this problem. We will make use of the CarRacing environment within OpenAI Gym. Moreover, they must also make sure that the technique they use doesn’t affect the appearance of their own fruit. While ProGAN generates fantastically realistic images, it is no exception to this general rule. Each observation can be described by five features, (accessoriesType, clothingColor, clothingType, hairColor, topType), as shown in Table 1-1. This inserted zero values in between pixels before performing the convolution operations. Each layer contains units, that are connected to the previous layer’s units through a set of weights. 72 | Chapter 3: Variational Autoencoders Figure 3-8. We want to measure the similarity of these representations across each position in the query input and key–value input. The style loss is scaled by a weighting parameter and the number of layers that it is calculated over. In the lefthand half of Figure 9-2, a set of N = 6 stacked attention layers encodes the input sentence ℮ = x1, . To work with the GloVe word vectors within this project, download the file glove.6B. This is exactly what we require; we would rather our discriminator is good at identifying when two images differ in style than content. Then, to generate a particular bar for a particular track, the relevant vectors from the chords, style, melody, and groove parts of the network are concatenated to form a longer vector. I knew at the time that I wanted this paper to be the focus of the final core chapter of the book, as it is the first practical example of how generative models can facilitate a deeper form of learning that takes place inside the agent’s own world model of the environment. Choose one of these answer blocks to carry forward to the encoder–decoder question generator (i.e., create the appropriate answer mask). Each encoder layer consists of several sublayers, starting with the multihead attention layer. 54 | Chapter 2: Deep Learning Incredibly, this simple addition drastically reduces overfitting, by ensuring that the network doesn’t become overdependent on certain units or groups of units that, in effect, just remember observations from the training set. This allows the car to excel at driving around the track without ever having attempted the task, as it has already imagined how to do this successfully in its own imagined world model. The tensor is reshaped to add two extra dimensions of size 1, to prepare it for concatenation with other bars and tracks. We’ll then see how variational autoencoders are a natural extension that fixes the issues faced by autoencoders. This is beneficial for model training speed as there are fewer weights to learn in the final output layer. Example 3-4. Figure 5-16. In fact, the MDN-RNN can be thought of as an environment in its own right, but operating in z-space rather than in the original image space. The same noisy stream of data that was essentially meaningless to your infant brain is not so noisy any more. If you want to re-create this technique and experiment with the parameters, I suggest work- ing from this repository as a starting point. The first step of the layer is to create three matrices, the query Q, key K, and value V, through multiplication of the input with three weight matrices, WQ, WK, and WV, as follows: Q = xQW Q K = xKV W K V = xKV W V Q and K are representations of the query input and key–value input, respectively. You can play around with the VAE architecture by editing the ./vae/arch.py file. Your First GAN First, you’ll need to download the training data. In particular, our model must be able to learn from and re-create the sequential struc- ture of music and must also be able to choose from a discrete set of possibilities for subsequent notes. The Encoder In an autoencoder, the encoder’s job is to take the input image and map it to a point in the latent space. There is no cell state or output gate, only a hidden state that is output from the cell. This shows that the generator has understood these high-level features and can gen- erate examples that are distinct from those it has already seen. xiv | Preface Finally, I would like to thank my fiancée (and soon to be wife) Lorna Barclay. There are many ways to regularize a machine learning algorithm, but for deep learn- ing, one of the most common is by using dropout layers. Autoencoders can also be used to clean noisy images, since the encoder learns that it is not useful to capture the position of the random noise inside the latent space. Given this action, the environment then calculates the next state st + 1 and reward rt + 1 and passes these back to the agent, for the cycle to begin again. Second, it is highly likely that generative modeling will be central to driving future developments in other fields of machine learning, such as reinforcement learning (the study of teaching agents to optimize a goal in an environment through trial and error). CHAPTER 1 Generative Modeling This chapter is a general introduction to the field of generative modeling. It has deci- ded that 12 dimensions are enough to capture sufficient information about the obser- vations for accurate reconstruction. Let’s now try to understand mathematically what we need to do to our autoencoder to convert it into a variational autoencoder and thus make it a truly generative model. Ideally, we would like to design a network that can accept multiple channels of music as individual streams and learn how these streams should interact with each other to generate beautiful-sounding music, rather than disharmonious noise. Dense layer that connects this vector to the 2D latent space. Figure 5-10. If we look at the correct German translation, we can see that the word for scored (erzielt) actually appears right at the end of the sentence! So not only would the model have to retain the fact that the penalty was scored rather than missed through the encoder, but also all the way through the decoder as well. Attention in Encoder–Decoder Networks The attention mechanism is a powerful tool that helps the network decide which pre- vious states of the recurrent layer are important for predicting the continuation of a sequence. To ensure consistency over time, they also test their techniques out on their own fruit. Stack convolutional layers sequentially on top of each other. There are pros and cons to using both word and character tokens, and your choice will affect how you need to clean the text prior to modeling and the output from your model. You see structure everywhere. Conclusion. Moreover, they show how, by dividing up the responsibilities of the noise vectors that feed the generator, they are able to maintain fine-grained control over the high-level temporal and trackbased features of the music. Summary | 127 PART II Teaching Machines to Paint, Write, Compose, and Play Part I introduced the field of generative deep learning and analyzed two of the most important advancements in recent years, variational autoencoders and generative adversarial networks. The way this is achieved is with a neural network consisting of convolutional trans- pose layers that we call the temporal network. Output at various stages of the training process Figure 5-13 shows some of the results from the original paper achieved by the model after it was trained for 200 epochs. The VAE decoder for the CelebA dataset To train the VAE on the CelebA dataset, run the Jupyter notebook 03_05_vae_faces_train.ipynb from the book repository. Example 6-6. Other Resources Two books I highly recommend as a general introduction to machine learning and deep learning are as follows: • Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron (O’Reilly) • Deep Learning with Python by Francois Chollet (Manning) Most of the papers in this book are sourced through arXiv, a free repository of scien- tific research papers. Now, there is another source of feedback. The generator First though, we need to introduce a new layer type: the upsampling layer. Training the GAN def train_discriminator(x_train, batch_size): valid = np.ones((batch_size,1)) fake = np.zeros((batch_size,1)) # TRAIN ON REAL IMAGES idx = np.random.randint(0, x_train.shape[0], batch_size) true_imgs = x_train[idx] self.discriminator.train_on_batch(true_imgs, valid) # TRAIN ON GENERATED IMAGES noise = np.random.normal(0, 1, (batch_size, z_dim)) gen_imgs = generator.predict(noise) self.discriminator.train_on_batch(gen_imgs, fake) def train_generator(batch_size): valid = np.ones((batch_size,1)) Your First GAN | 109 noise = np.random.normal(0, 1, (batch_size, z_dim)) self.model.train_on_batch(noise, valid) epochs = 2000 batch_size = 64 for epoch in range(epochs): train_discriminator(x_train, batch_size) train_generator(batch_size) One batch update of the discriminator involves first training on a batch of true images with the response 1… …then on a batch of generated images with the response 0

this point through the decoder, since the decoder has learned how to convert points in the latent space into viable images. Since this piece is performed by one instrument (a cello), we are justified in doing this, though some- times we may wish to keep the parts separate to generate music that is polyphonic in nature. There aren't any distributions that stand out as being significantly different from the standard normal, so we can move on to generating some faces! 6 Xianxu Hou et al., "Deep Feature Consistent Variational Autoencoder," 2 October 2016, 1610.00291. Upsampling In the decoder of the variational autoencoder that we built in the previous chapter, we doubled the width and height of the tensor at each layer using Conv2DTranspose lay- ers with stride 2. The Wasserstein loss can therefore be very large, which is unset- tling—usually, large numbers in neural networks are to be avoided! In fact, the authors of the WGAN paper show that for the Wasserstein loss function to work, we also need to place an additional constraint on the critic. • The vocabulary is usually much smaller when using character tokenization. We can sample from this distribution to produce a latent vector z that represents the current state. This chapter presents one of the most excit- ing papers published in recent years, in which the authors show how a generative model can be used as the environment in which the agent trains, thus essentially allowing the agent to dream of possible future scenarios and imagine what might hap- pen if it were to take certain actions, entirely within its own conceptual model of the environment. Music and text generation share a lot of features in common, and often similar techniques can be used for both. The output is more positive when the portion of the image closely matches the filter and more negative when the portion of the image is the inverse of the filter. It is as if the gender of the word is encoded into the latent space in which the word vectors exist. Constant width bold Shows commands or other text that should be typed literally by the user. A file storing the weights of the trained network is saved every eval_steps generations. The decoder ### THE DECODER decoder_input = Input(shape=(self.z_dim,), name='decoder_input') x = Dense(np.prod(shape_before_flattening))(decoder_input) x = Reshape(shape_before_flattening)(x) for i in range(self.n_layers_decoder): conv_t_layer = Conv2DTranspose(filters = self.decoder_conv_t_filters[i] , kernel_size = self.decoder_conv_t_kernel_size[i] , strides = self.decoder_conv_t_strides[i] , padding = 'same' , name = 'decoder_conv_t_' + str(i)) x = conv_t_layer(x) if i < self.n_layers_decoder - 1: x = LeakyReLU()(x) else: 2 Source: Vincent Dumoulin and Francesco Visin, "A Guide to Convolution Arithmetic for Deep Learning," 12 January 2018, . A clutch of painted parakeet chicks is usually around five eggs, which are less than one inch wide at hatching. A good measure for dis- tance is the L1 distance, defined as: def l1_compare_images(img1, img2): return np.mean(np.abs(img1 - img2)) Figure 4-10 shows the closest observations in the training set for a selection of gener- ated images. How a single sequence flows through a recurrent layer Here, we represent the recurrent process by drawing a copy of the cell at each time- step and show how the hidden state is constantly being updated as it flows through the cells. It really is a case of testing both methods in your own problem setting and seeing which produces better results. Figure 1-15. How many parameters are contained within a batch normalization layer? The competition involves classifying images into one of a thousand categories and is used as a benchmark to compare the latest state-of-the-art techniques. This is useful for debugging and finding areas where your agent could improve. One day, Mr. N. Example 5-8. Parallelizing CMA-ES—here there is a population size of 8 and 4 slave nodes (so t = 2, the number of trials that each slave is responsible for) The master asks the CMA-ES object (es) for a set of parameters to trial. Now that we have the RandomWeightedAverage layer that can interpolate between two images and the gradient_penalty_loss function that can calculate the gradient loss for the interpolated images, we can use both of these in the model compilation of the critic. For this reason, deep learning has now almost become synonymous with deep neural networks. Customers are thoroughly confused and now can't tell the difference between real and fake apples and real and fake oranges. In both cases we have seen how it is important to understand how to transform unstructured text data to a structured format that can be used with recurrent neural network layers. Flatten the last convolutional layer to a vector. Figure 5-17 shows the Gram matrices for the three features, for each of the images. 206 | Chapter 7: Compose context vector, then passing this vector through the decoder network to output the translated text. ix Part I. Loading the Data For this example we will be using the CIFAR-10 dataset, a collection of 60,000 32 × 32–pixel color images that comes bundled with Keras out of the box. Occasionally, when feeding below the canopy, painted parakeets will eat algae from forest pools. We will start this chapter by simplifying the problem to focus on music generation for a single (monophonic) line of music. An example of neural style transfer13 The idea works on the premise that we want to minimize a loss function that is a weighted sum of three distinct parts: Content loss We would like the combined image to contain the same content as the base image. We'll then apply the same tech- nique to build a model that can apply a given artist's style to a photo of your choice. It's attached to a neck and a set of legs that can put its core input sensors in any myriad of positions relative to the source of the input data. • Text data is highly sensitive to small changes in the individual units (words or characters). I'll now introduce OpenAI Gym, home of the CarRacing environment that we will use to simulate a car driving around a track. 200 | Chapter 6: Write CHAPTER 7 Compose Alongside visual art and creative writing, musical composition is another core act of creativity that we consider to be uniquely human. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights. It is our job to build a model that mimics this distribution as closely as possible and then sample from it to generate new, distinct observations that look as if they could have been included in the original training set. The model can generate the facial color and mouth, as all of these pixels in the training set are roughly the same shade in each observation; however for the T-shirt pixels, each pixel is sampled at random from a variety of different colors in the training set, with no regard to the colors that have been sampled in neighboring pixels. Specifically, the MDN-RNN is an LSTM layer with 256 hidden units followed by a mixture density network (MDN) output layer that allows for the fact that the next latent state could actually be drawn from any one of several normal distributions. 3 Diederik Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," 22 December 2014, https:// arxiv.org/abs/1412.6980v8. When it comes to predic- tion, we may only want to predict a single observation, so there is no batch over which to take averages. Let's take a look at the definition of the Wasserstein loss function. Training the CycleGAN batch_size = 1 patch = int(self.img_rows / 2**4) self.disc_patch = (patch, patch, 1) valid = np.ones((batch_size,) + self.disc_patch) fake = np.zeros((batch_size,) + self.disc_patch) for epoch in range(self.epoch, epochs): for batch_i, (imgs_A, imgs_B) in enumerate(data_loader.load_batch(batch_size)): fake_B = self.g_AB.predict(imgs_A) fake_A = self.g_BA.predict(imgs_B) dA_loss_real = self.d_A.train_on_batch(imgs_A, valid) dA_loss_fake = self.d_A.train_on_batch(fake_A, fake) dA_loss = 0.5 * np.add(dA_loss_real, dA_loss_fake) dB_loss_real = self.d_B.train_on_batch(imgs_B, valid) dB_loss_fake = self.d_B.train_on_batch(fake_B, fake) dB_loss = 0.5 * np.add(dB_loss_real, dB_loss_fake) 146 | Chapter 5: Paint d_loss = 0.5 * np.add(dA_loss, dB_loss) g_loss = self.combined.train_on_batch([imgs_A, imgs_B], [valid, valid, imgs_A, imgs_B, imgs_A, imgs_B]) We use a response of 1 for real images and 0 for generated images. Again, we use strides = 2 and padding = "same", so the width and height both halve. A huge thank you goes to my mum, Gillian Foster, for checking every single line of text for typos and for teaching me how to add up in the first place! Your attention to detail has been extremely helpful while proofreading this book, and I'm really grateful for all the opportunities that both you and dad have given me. The Gradient Penalty Loss Figure 4-15 is a diagram of the training process for the critic. The new beliefs of the cell ht and the current hidden state, ht–1, are blended in a proportion determined by the update gate, zt, to produce the updated hidden state, ht, that is output from the cell. It can then perform well at the task first time, having never attempted the task in reality. Probabilistic Generative Models Firstly, if you have never studied probability, don't worry. Dropout layers are also used. Git is an open source version control system and will allow you to copy the code locally so that you can run the notebooks on your own machine, or perhaps in a cloud-based environment. For exam- ple, when strides = 2, the height and width of the output tensor will be half the size of the input tensor. Figure 2-5. The difference here is that instead of using a recurrent layer such as an LSTM inside the encoder and decoder, the Trans- former uses stacked attention layers. • The WGAN is trained using labels of 1 for real and –1 for fake. To understand deep learning fully, and particularly why it is so useful within genera- tive modeling, we need to delve into this definition a bit further. This chapter will also introduce the atten- tion mechanism that will allow us to build RNNs that are able to choose which 201 previous notes to focus on in order to predict which notes will appear next. Applications of Generative Modeling | 297 CHAPTER 10 Conclusion In this book, we have taken a journey through the last half-decade of generative mod- eling research, starting out with the basic ideas behind variational autoencoders, GANs, and recurrent neural networks and building upon these foundations to under- stand how state-of-the-art models such as the Transformer, advanced GAN architec- tures, and world models are now pushing the boundaries of what generative models are capable of achieving, across a variety of tasks. We use the Adam optimizer, which is generally regarded to be the best optimizer for WGAN-GP models. To achieve this, we use the follow- ing positional encoding function that converts the position pos of the word in the sentence into a vector of length dmodel: PE pos, 2i = sin PE pos, 2i + 1 = cos pos 2i/dmodel 10000 pos 10000 2i + 1 /dmodel For small i, the wavelength of this function is short and therefore the function value changes rapidly along the position axis. The wall at the art exhibition 62 | Chapter 3: Variational Autoencoders In Figure 3-2 you can see examples of other original paintings (top row), the coordi- nates of the point on the wall given by Mr. N. This is only possible because the deeper layers of the network are capturing high-level features that we can work with directly. In their earlier pix2pix paper,5 the authors used a U-Net architecture, but they switched to a ResNet architecture for CycleGAN. 1005 Gravenstein Highway North Sebastopol, CA 95472 800-998-9938 (in the United States or Canada) 707-829-0515 (international or local) 707-829-0104 (fax) We have a web page for this book, where we list errata, examples, and any additional information. The problem is reduced to estimating the parameters θkl = p xk = l for each feature separately and multiplying these to find the probability for any possible combination. To train the inmates and the guard, Edward feeds short sequences of words that he has written previously into the cell and monitors if the inmates' chosen next word is correct. Excerpt from rnn/arch.py def get_responses(self, y_true): z_true = y_true[:,:,:Z_DIM] rew_true = y_true[:,:,-1] Training the MDN-RNN | 259 return z_true, rew_true def get_mixture_coef(self, z_pred): log_pi, mu, log_sigma = tf.split(z_pred, 3, 1) log_pi = log_pi - K.log(K.sum(K.exp(log_pi), axis = 1, keepdims = True)) return log_pi, mu, log_sigma def tf_lognormal(self, z_true, mu, log_sigma): logSqrtTwoPI = np.log(np.sqrt(2.0 * np.pi)) return -0.5 * ((z_true - mu) / K.exp(log_sigma)) ** 2 - log_sigma - logSqrtTwoPI def rnn_z_loss(y_true, y_pred): z_true, rew_true = self.get_responses(y_true) d = normal_distribution_MIXTURES * Z_DIM z_pred = y_pred[:,:,:(3*d)] z_pred = K.reshape(z_pred, [-1, normal_distribution_MIXTURES * 3]) log_pi, mu, log_sigma = self.get_mixture_coef(z_pred) flat_z_true = K.reshape(z_true,[-1, 1]) z_loss = log_pi + self.tf_lognormal(flat_z_true, mu, log_sigma) z_loss = -K.log(K.sum(K.exp(z_loss), 1, keepdims=True)) z_loss = K.mean(z_loss) return z_loss def rnn_rew_loss(y_true, y_pred): z_true, rew_true = self.get_responses(y_true) #, done_true d = normal_distribution_MIXTURES * Z_DIM reward_pred = y_pred[:,:,-1] rew_loss = K.binary_crossentropy(rew_true, reward_pred, from_logits = True) rew_loss = K.mean(rew_loss) return rew_loss def rnn_loss(y_true, y_pred): z_loss = rnn_z_loss(y_true, y_pred) rew_loss = rnn_rew_loss(y_true, y_pred) 260 | Chapter 8: Play return Z_FACTOR * z_loss + REWARD_FACTOR * rew_loss opti = Adam(lr=LEARNING_RATE) model.compile(loss=rnn_loss, optimizer=opti, metrics = [rnn_z_loss, rnn_rew_loss]) Split the 481-dimensional output vector into the 3 variables (log_pi, mu, log_sigma) and the reward value. Let's think about how this might occur. The Keras code to build this is shown in Example 7-5. The granularity of the data combined with the high degree of spatial dependence destroys the concept of the pixel or character as an informative feature in its own right. 168 | Chapter 6: Write • Words can be stemmed, meaning that they are reduced to their simplest form, so that different tenses of a verb remained tokenized together. 240 | Chapter 8: Play Figure 8-2. Setting up your environment Finally, you will need to ensure you are set up to access your virtual environment through Jupyter notebooks on your machine. We can also alter tracks individually, through the melody and groove inputs. This recurrent process continues until the end of the sequence. Once our VAE is trained, we can sample from the latent space to gener- ate new examples of celebrity faces. The Decoder The decoder is a mirror image of the encoder, except instead of convolutional layers, we use convolutional transpose layers, as shown in Figure 3-6. batch_size The number of episodes fed to the MDN-RNN in each training iteration. This makes sense; if you know that the first note is a B-flat, you will probably use this information to inform your decision about the next note. A set of points in two dimensions, generated by an unknown rule pdata Where did you choose? Here, I show a selection of the most recent developments that have pushed the boundaries of generative modeling in general. Without padding, the kernel could only fit three times along each direction, giving an output size of 3 × 3. Convolutional neural network (CNN) for CIFAR-10 Improving the Model | 57 Before moving on, make sure you are able to calculate the output shape and number of parameters for each layer by hand. The representation vector is a compression of the original image into a lowerdimensional, latent space. Secondly, since this term tries to force all encoded distribu- tions toward the standard normal distribution, there is less chance that large gaps will form between point clusters. This is passed to the MDN, which is just a densely connected layer that transforms the vector of length 256 into a vector of length 481. Style transfer examples1 In this chapter you'll learn how to build two different kinds of style transfer model (CycleGAN and Neural Style Transfer) and apply the techniques to your own photos and artwork. It has ultimately given rise to the transformer, a type of model based entirely on attention that doesn't even require recurrent or convolutional layers. Training the autoencoder self.model.fit(x = x_train , y = x_train , batch_size = batch_size , shuffle = True , epochs = 10 , callbacks = callbacks_list) Analysis of the Autoencoder Now that our autoencoder is trained, we can start to investigate how it is representing images in the latent space. Every OpenAI Gym environment includes a step method. If the dataset is labeled, we can also build a generative model that estimates the distri- bution p x y . Figure 3-14. Figure 5-7 shows the difference between batch normalization and instance normal- ization, as well as two other normalization methods (layer and group normalization). The overall architecture of the ResNet is shown in Figure 5-11. Some examples from the CelebA dataset5 Of course, we don't need the labels to train the VAE, but these will be useful later when we start exploring how these features are captured in the multidimensional latent space. Di then looks at this set of pho- tos and tries to guess which were taken by Gene and which are originals. The Keras model that defines the full autoencoder—a model that takes an image, and passes it through the encoder and back out through the decoder to generate a reconstruction of the original image. After flattening the resulting tensor, we pass the data through a Dense layer of size 128, again followed by a BatchNormalization and a LeakyReLU layer. The goal of generative modeling is to use these observations to build a pmodel that can accurately mimic the observations pro- duced by pdata. The Transformer model architecture2 The authors apply the Transformer to English–German and English–French transla- tion datasets. This process is shown in Figure 6-14, in the context of translation between English and German. 208 | Chapter 7: Compose Figure 7-6. Why would the network willingly choose to ignore this note when deciding how to continue the phrase? For example, classification tasks such as sentiment analysis can be con- structed by adding a classification layer on top of the Transformer output, and question answering tasks can be tackled by marking the answer in the input sequence using a pointer network as the output layer to BERT. Figure 8-1. We also saw how these kinds of basic models can fail as the complexity of the generative task grows, and analyzed the general challenges associated with generative modeling. 154 | Chapter 5: Paint Figure 5-15. If we are working with color images, then each filter would have three channels rather than one (i.e. each having shape 3 × 3 × 3) to match the three channels (red, green, blue) of the image. What if he randomly placed markers on parts of the wall that currently do not have a marker? Suppose that the brain is a near-perfect generative model of the input stream of data that it is subjected to. This note is highly unlikely to appear in a piece of music in the key of E-flat major. Mathematically speaking, it tries to find the highly nonlinear manifold on which the data lies and then establish the dimensions required to fully describe this space. Example 4-10. The original concatenated vector is also passed through an output gate: a dense layer with weights matrix Wo, bias bo, and a sigmoid activation. It would be difficult to maintain this long-range dependency for pixels without attention, especially for long, thin struc- tures in the image (such as the tail in this case). 10 | Chapter 1: Generative Modeling Sample Space The sample space is the complete set of all values an observation x can take. For now, let's focus on incorporating attention into a stacked LSTM network to try to predict the next note, given a sequence of previous notes. An encoder–decoder model with attention The attention mechanism works in exactly the same way as we have seen previously, with one alteration: the hidden state of the decoder is also rolled into the mechanism so that the model is able to decide where to focus its attention not only through the previous encoder hidden states, but also from the current decoder hidden state. Your First Deep Neural Network Let's start by seeing how easy it is to build a deep neural network in Keras. , 04031 . It may be wise to replace sparse words with a token for unknown word, rather than includ- ing them as separate tokens, to reduce the number of weights the neural network needs to learn. This makes the layer output the hidden state from every timestep, rather than just the final timestep. The model that trains the critic is defined to have two inputs: the batch of real images and the random input that will generate the batch of fake images. Summary In this chapter we have seen how recurrent neural networks can be applied to gener- ate text sequences that mimic a particular style of writing, and also generate plausible question–answer pairs from a given document. Starting from this state of ignorance, how can they work together to ensure Gene is eventually able to produce impressive ganimal photographs? Notice that the number of parameters is actually fewer in our new model than the previous model, even though the number of layers is far greater. Let's look more closely at a single cell and how the hidden state is updated (Figure 6-8). We then apply a softmax function to ensure all rows sum to 1. Example 3-8 shows how we include the KL divergence term in our loss function. If the brothers saw that particular paintings weren't being re-created accurately, they would adjust their understanding of marker placement and image regeneration to ensure revenue loss was minimized. The VAE architecture for the "World Models" paper Training the VAE | 251 Exploring the VAE We'll now take a look at the output from the predict methods of the different models built on the VAE to see how they differ, and then see how the VAE can be used to generate completely new track observations. This way, you can instantiate an Autoencoder object with parameters that define a particular model architecture in the notebook, as shown in Example 3-1. Here, we cannot take this approach as we only have two images to work with, the base image and the style image. Figure 2-10 shows a 3 × 3 × 1 portion of a grayscale image being convoluted with a 3 × 3 × 1 filter (or kernel). The resulting vector, rt, has a length equal to the number of units in the cell and stores values between 0 and 1 that determine how much of the previous hidden state, ht–1, should be carried forward into the calculation for the new beliefs of the cell. The red curve is the standard normal—the normal distri- bution with mean equal to 0 and variance equal to 1. While training is ini- tially faster with a large learning rate, the downside is that it may result in less stable training and may not find the minima of the loss function. However, this quickly becomes infeasible as the environment becomes more complex and doesn't ever allow us to build an agent that has superhu- man ability in a particular task, as we are hardcoding the rules. Style loss We would like the combined image to have the same general style as the style image. As in the previous chapter on VAEs, you can instantiate a GAN object in the notebook, as shown in Example 4-1, and play around with the parame- ters to see how it affects the model. The original input sequence is summarized into a single vector by the encoder RNN. For most purposes, it doesn't matter which you choose as you usually won't be coding directly using the underlying backend framework. We define a new model whose input is a 100-dimensional latent vector; this is passed through the generator and frozen discriminator to produce the output probability. This combination of colors camouflages the birds in the rainforest. Figure 4-6. Initially, Gene doesn't get any useful feedback from Di, since she is randomly guessing which photos are genuine. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Figure 4-13. We'll start by looking at one of the most utilized and established models for generating sequential data such as text, the recurrent neu- ral network (RNN), and in particular, the long short-term memory (LSTM) layer. After passing through a Dense layer, we reshape the tensor to prepare it for the convolutional transpose operations. Most of our analysis in this section will focus on the note pitch predictions, rather than rhythms, as for Bach's Cello Suites the harmonic intricacies are more difficult to capture and therefore more worthy of investigation. Now that you are familiar with the CycleGAN architecture, you might recognize that this image represents the three criteria through which the combined model is judged: validity, reconstruction, and identity. 275 Figure 9-1. Previously, we saw how there was no requirement for the latent space to be continu- ous—even if the point (–2, 2) decodes to a well-formed image of a 4, there was no requirement for (–2.1, 2.1) to look similar. Representation Learning The core idea behind representation learning is that instead of trying to model the high-dimensional sample space directly, we should instead describe each observation in the training set using some low-dimensional latent space and then learn a mapping function that can take a point in the latent space and map it to a point in the original domain. In other words, we can imagine that there is some unknown probabilistic distribution that explains why some images are likely to be found in the training dataset and other images are not. Figure 7-13 shows the inner workings of an attention module within an encoder– decoder framework. Analysis of the Neural Style Transfer Model Figure 5-18 shows the output of the neural style transfer process at three different stages in the learning process, with the following parameters: • content_weight: 1 • style_weight: 100 • total_variation_weight: 20 Neural Style Transfer | 161 Figure 5-18. Finally, we collapse the tensor along the timesteps axis. The difference between the encoder in an autoencoder and a variational autoencoder 78 | Chapter 3: Variational Autoencoders The Normal Distribution A normal distribution is a probability distribution characterized by a distinctive bell curve shape. The layers are joined together along the channels dimension so the number of channels increases from k to 2k, while the number of spatial dimensions remains the same. In the original paper, the authors show this technique successfully applied to a differ- ent environment: DoomTakeCover, based around the computer game Doom. d. It consists of N = 50 observations of fash- ions currently seen on the planet. Figure 1-3. The out- put from a given unit is the weighted sum of the input it receives from the previous layer, which is then passed through a nonlinear activation function before being sent to the following layer. Each image in the CelebA dataset is labeled with attributes, one of which is smiling. This explains why Naive Bayes models cannot be expected to work well on raw image data. 150 | Chapter 5: Paint y = Conv2D(filters, kernel_size=(3, 3), strides=1, padding='same')(y) y = InstanceNormalization(axis = -1, center = False, scale = False)(y) return add([shortcut, y]) On either side of the residual blocks, our ResNet generator also contains downsam- pling and upsampling layers. This is understandable, as the decoder only has the final encoder state to work with and cannot reference the original document for extra information. Mode collapse Uninformative Loss Since the deep learning model is compiled to minimize the loss function, it would be natural to think that the smaller the loss function of the generator, the better the qual- ity of the images produced. While exciting, this also raises ethical 1 Tero Karras, Samuli Laine, and Timo Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," 12 December 2018, . One weekend, they host an exhibition focused on monochrome studies of single-digit numbers. That's because tensor operations are conducted across all observations simultaneously using linear algebra—this is the part handled by TensorFlow. So why does this small change to the encoder help? How to Contact Us Please address comments and questions concerning this book to the publisher: O'Reilly Media, Inc. The Generative Modeling Framework • We have a dataset of observations X. As we shall see later in the chapter, the crux of the "World Models" paper is that it demonstrates how this reinforcement learning can take place within the agent's own generative model of the environment, rather than the OpenAI Gym environment. Discriminative modeling estimates p y x —the probability of a label y given observa- tion x. It is an excellent resource for anyone wanting to quickly understand which techniques are currently achieving the highest scores in a range of tasks and has certainly helped me to decide which techniques to cover in this book. 24 | Chapter 1: Generative Modeling Figure 1-12. The weights are shared across positions, but not between layers of the encoder–decoder. The output is therefore a vector of length z_dim for every track of every bar. Only the last max_sequence_len tokens are kept. 8 Source: Karras et al., 2017. Figure 1-6. Since mid-2018 there has been a flurry of remarkable developments in both sequence and image-based generative modeling. This is distinct from the hidden state, ht, which is ultimately output by the cell after the final timestep. Figure 7-10. Chapter 7 looks at music generation, which, while also a sequential generation prob- lem, presents additional challenges such as modeling musical pitch and rhythm. Indeed, in Figure 4-8, the loss function of the generator actually increases over time, even though the quality of the images is clearly improving. The identity term helps regu- late the generator to ensure that it only adjust parts of the image that are necessary to complete the transformation and no more. The generator forms part of the model that we are using to train the critic, as the interpolated images are now actively involved in the loss function, so this is required. The batch_size determines how many observations will be passed to the net- work at each training step. We can also see how the loss functions of the model evolve over time (Figure 4-18)— the loss functions of both the discriminator and generator are highly stable and con- vergent. Content Loss The content loss measures how different two images are in terms of the subject mat- ter and overall placement of their content. In our case, we have one input layer that we defined earlier, and the output layer is the final Dense layer of 10 units. Figure 6-3. The alignment function is just a Dense layer with one output unit and tanh acti- vation. The word representations learned by BERT are superior to counterparts such as GloVe because they change depending on the context of the word. For example, there is a gray gap along the row for note 54 (corre- sponding to Gb/F#). It is defined as follows: ℒ θ x = pθ x That is, the likelihood of θ given some observed point x is defined to be the value of the density function parameterized by θ, at the point x. Figure 2-8. The image wouldn't be perfect, but it would be a close enough likeness to your actual appearance for them to find you among pos- sibly hundreds of other people, even if they've never seen you before. Use of the information and instructions contained in this work is at your own risk. First, we downsample the image, using Conv2D layers with stride 2. Each bar consists of 16 timesteps and there are a potential 84 pitches across the 4 tracks. Generative mod- els are often more difficult to evaluate, especially when the quality of the output is largely subjective. Example 5-2. Importing a MIDI file from music21 import converter dataset_name = 'cello' filename = 'cs1-2all' file = "./data/{}/{}.mid".format(dataset_name, filename) original_score = converter.parse(file).chordify() 202 | Chapter 7: Compose Preliminaries | 203 Figure 7-1. They both open with the familiar statement of the characters in the story, and generally the text within speech marks is more dialogue-like, using personal pronouns and prepared by the occurrence of the word said. The words are the pitches, and we should try to build a model that pre- dicts the next pitch, given a sequence of previous pitches. Add randomness to the parameters of the new members. Before we see how this compares in accuracy to our densely connected neural network, I'm going to introduce two more layer types that can also improve performance: BatchNormalization and Dropout. 5 Isola et al., 2016, . For tasks such as this, a 2D latent space is probably too small to encode sufficient relevant Autoencoders | 65 information from the input. The idea here is to build up a dataset of how the physics of the environment works, which the VAE can then learn from to capture the states efficiently as latent vectors. Customers who took apples sometimes 1 Jun-Yan Zhu et al., "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 30 March 2017, . The second LSTM layer can then use the hidden states from the first layer as its input data. Reinforcement learn- ing involves creating an agent that can learn optimal strategies by itself in complex environments through repeated play—this is what we will be using in this chapter to build our agent. The adaptive instance normalization layers ensure that the style vectors that are injected into each layer only affect features at that layer, by pre- venting any style information from leaking through between layers. Deep learning is the key to solving both of these challenges. Each night, Gene takes 64 photographs, each in a different location with different random moonlight readings, and mixes them with 64 ganimal photos from the original collection. As errors are propagated backward through the Improving the Model | 51 network, the calculation of the gradient in the earlier layers can sometimes grow exponentially large, causing wild fluctuations in the weight values. This way of training encoder–decoder networks is known as teacher forcing. You can change the number of convolutional layers in the encoder simply by adding elements to the lists that define the model architecture in the notebook. We also tend to use smaller learning rates for WGANs. Example 4-6. This is a parameter that you may want to tune or adjust during training. Compilation ### COMPILATION optimizer = Adam(lr=learning_rate) def r_loss(y_true, y_pred): return K.mean(K.square(y_true - y_pred), axis = [1,2,3]) self.model.compile(optimizer=optimizer, loss = r_loss) We can now train the autoencoder by passing in the input images as both the input and output, as shown in Example 3-6. A huge thank you goes to the staff at O'Reilly for guiding me through the process of writing this book. For example, take the opening passage of the Prelude to Bach's Cello Suite No. 1 (Figure 7-4). To understand the difference, let's look at an example. To construct the input into the first encoder layer, the matrix of positional encodings is added to the word embedding matrix, as shown in Figure 9-3. By manipulating each of these inputs independently we can change high-level properties of the generated music. Mean squared error is used for the validity criterion—checking the output from the discriminator against the real (1) or fake (0) response—and mean absolute error is used for the image-toimage-based criteria (reconstruction and identity). This process is shown in Figure 9-7. This is carried across to the bar gener- ator without any change, as it is independent of the track and bar. Figure 3-11. Reward The value given back to the agent by the environment after an action has been taken. The most time-consuming part of the algorithm is calculating the score for a given set of parameters, since it needs to simulate an agent with these parameters in the environment. The Discriminators The discriminators that we have seen so far have output a single number: the predic- ted probability that the input image is "real." The discriminators in the CycleGAN that we will be building output an 8 × 8 single-channel tensor rather than a single number. This is a great resource for sourcing data that can be used to train text-based deep learning models. The output from the encoder layer is also incorporated into each layer of the decoder, after the initial self-attention mechanism. The question word tokens are passed through the same embedding layer used in the answer identification model. The agent must set all three values at each timestep. Figure 3-22. The generated sequence of possible futures is not only dependent on its understanding of the phys- ics of the environment, but also on its understanding of itself and how it acts. The training process works as follows. Since each individual element of the discriminator predic- tion is based only on a small square of the image, it must use the style of the patch, rather than its content, to make its decision. Training the controller in the MDN-RNN dream environment In-Dream Training the Controller To train the controller using the dream environment, run the following command from your terminal (on one line): xvfb-run -a -s "-screen 0 1400x900x24" python 05_train_controller.py car_racing -n 16 -t 2 -e 4 --max_length 1000 --dream_mode 1 This is the same command used to train the controller in the real environment, but with the added flag --dream_mode 1. Therefore for the WGAN genera- tor, minimizing the loss function can be written as follows: WGAN generator loss minimization minG − ℒz ∼ p D G z Z When we compile the models that train the WGAN critic and generator, we can spec- ify that we want to use the Wasserstein loss instead of the binary cross-entropy, as shown in Example 4-6. As CFO, it is your sole responsibility to create new and exciting fashion trends for the inhabitants of the planet to follow. For example: "Generative Deep Learning by David Foster (O'Reilly). We compile the critic with three loss functions: two Wasserstein losses for the real and fake images, and the gradient penalty loss. O'Reilly books may be purchased for educational, business, or sales promotional use. Introduction to Generative Deep Learning 1. Generative Versus Discriminative Modeling Advances in Machine Learning The Rise of Generative Modeling The Generative Modeling Framework Probabilistic Generative Models Hello Wrodl! Your First Probabilistic Generative Model Naive Bayes Hello Wrodl! Continued The Challenges of Generative Modeling Representation Learning Setting Up Your Environment Summary 1 2 4 5 7 10 13 14 17 20 22 23 27 29 2. It is also worth pointing out that the model has learned Bach's characteristic style of dropping to a low note on the cello to end a phrase and bouncing back up again to start the next. In the next section we'll see how we can build a MuseGAN using Keras, to learn how to generate realistic polyphonic music. MuseScore is a great tool for this purpose and can be downloaded for free. The code is contained in the 06_01_lstm_text_train.ipynb notebook in the book repository. 201 Preliminaries Musical Notation Your First Music-Generating RNN Attention Building an Attention Mechanism in Keras Analysis of the RNN with Attention Attention in Encoder–Decoder Networks Generating Polyphonic Music The Musical Organ Your First MuseGAN The MuseGAN Generator Chords, Style, Melody, and Groove The Bar Generator Putting It All Together The Critic Analysis of the MuseGAN Summary vi | Table of Contents 202 202 205 206 208 213 217 221 221 223 226 227 229 230 232 233 235 8. Example 3-9. The recurrent layer is a bidirectional GRU that returns the hidden state at each timestep. Just an endless stream of extremely noisy data. These will be useful when we want to encode a specific image, or decode a given z vector, for example. Figure 9-10. Neural Style Transfer | 157 We can define a matrix that contains the dot product between all possible pairs of fea- tures in the layer. The input tensor to the VGG19 model is a concatenation of the three images. If you look across to see which note this corresponds to, you can see that it is the first of three E-3 (E-flat) notes. We can find the closest score in the dataset (by Euclidean distance) and check that our generated score isn't a copy of a piece of music that already exists in the dataset—the closest score is shown just below it, and we can see that it does not resemble our generated score. One batch update of the generator involves training on a batch of generated images with the response 1. To begin with, the model plays it safe by sticking to the same group of notes and rhythms. This returns the next observation, reward, and done flag, given an action. This gate is a dense layer with weights matrix Wz and a sigmoid activation. (See, for example, OpenAI's Chopin and Mozart recordings on SoundCloud.) Advances in Image Generation In recent years, image-based generative modeling has been revolutionized by several significant advancements in the architecture and training of GAN-based models, in the same way that the Transformer has been pivotal to the progression of sequential generative modeling. Figure 8-12. Output from the CycleGAN when the identity loss weighting is set to zero The CycleGAN has still managed to translate the oranges into apples but the color of the tray holding the oranges has flipped from black to white, as there is now no iden- tity loss term to prevent this shift in background colors. For some tasks, the goal isn't to predict the single next word in the existing sequence; instead we wish to predict a completely different sequence of words that is in some way related to the input sequence. There are no labels on the data that we receive—a seemingly ran- dom stream of light and sound waves that bombard our senses from the moment we are born. My brother, Rob Foster, is the most modest genius you will ever find, particularly within linguistics—chatting with him about AI and the future of text-based machine learning has been amazingly helpful. The shape of the filters in the first convolutional layer is 4 × 4 × 3. Thus, we treat the context vectors as additional data to be fed into the decoder. Generating New Text | 181 Figure 6-10. It would also take 2 Jun-Yan Zhu et al., "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 30 March 2017, . After around five epochs of training your VAE should be able to produce novel images of celebrity faces! 90 | Chapter 3: Variational Autoencoders Analysis of the VAE You can replicate the analysis that follows by running the notebook 03_06_vae_faces_analysis.ipynb, once you have trained the VAE. This back-and-forth process continues, over many days and weeks. Example 7-3. You can try changing the weighting parameters in the loss function or the layer that is used to determine the content similarity, to see how this affects the combined output image and training speed. CNN predictions This improvement has been achieved simply by changing the architecture of the model to include convolutional, batch normalization, and dropout layers. We increase the reconstruction loss factor to ten thousand. There are many ways to experiment with this architecture. Therefore to calculate the style loss, all we need to do is calculate the Gram matrix (GM) for a set of layers throughout the network for both the base image and the com- bined image and compare their similarity using sum of squared errors. Rather than only using the final hidden state of the encoder RNN as the context vector, the attention mecha- nism allows the model to create the context vector as a weighted sum of the hidden states of the encoder RNN at each previous timestep. In our neural network, we use a soft- max activation in the final layer to ensure that the output is a set of 10 probabilities that sum to 1, which can be interpreted as the chance that the image belongs to each class. Typically, there is some small oscillation of the loss between batches, but in the long term you should be looking for loss that stabilizes or gradually increases or decreases (see Figure 4-8), rather than erratically fluctuating, to ensure your GAN converges and improves over time. Initially the instrument only produces horrendous noise, as it has no way to under- stand how it should interpret the instructions to produce bars that are indistinguisha- ble from genuine music. Training the Controller | 265 mean and covariance matrix of the normal distribution as per Figure 8-15 and pro- vides the master with a new population to test. Concatenate everything together to form a single output tensor. 61 The exhibition wall is shown in Figure 3-1, where each black dot is a marker placed by Mr. N. In order to structure our approach to finding a suitable pmodel(X) we can use a technique known as parametric modeling. In many ways, it has the easiest job of all the sections as each player simply has to generate a single instruction at the start of the concert that is then continually fed to the MuseGAN throughout the performance. Exactly the same principle is true in music. Therefore, to train the generator, we must first connect it to the discriminator to cre- ate a Keras model that we can train. In this sense, your brain is an extremely sophisticated generative model, equipped with the ability to attend to particular parts of the input data, form representations of concepts within a latent space of neural pathways, and process sequential data over time. By connecting the final hidden state to a dense layer, the network can output a probability distribution for the next word. We then add or subtract multiples of a certain vector (e.g., smile, blonde, male, eyeglasses) to obtain different versions of the image, with only the rele- vant feature changed. Finally, the StyleGAN architecture adds noise after each convolution to account for stochastic details such as the placement of individual hairs, or the background behind the face. This chapter will first lay out the theoretical underpinning of generative adversarial networks (GANs). It is a glimpse into a future where agents learn not only through maximizing a single reward in an environment of our choice, but by generat- ing their own internal representation of an environment and therefore having the capability to create their own reward functions to optimize. 6 If the sample space is discrete, p(x) is simply the probability assigned to observing point x. 176 | Chapter 6: Write Figure 6-8. LeakyReLU activations fix the issue by always ensuring the gradient is nonzero. Notice that to train the MDN-RNN, we do not need to sample specific z vectors from the MDN output, but instead calculate the loss directly using the 481-dimensional output vector. One of the main criticisms of the WGAN is that since we are clipping the weights in the critic, its capacity to learn is greatly diminished. Oscillating Loss The loss of the discriminator and generator can start to oscillate wildly, rather than exhibiting long-term stability. Your First LSTM Network | 175 The fact that the output from the cell is called a hidden state is an unfortunate naming convention—it's not really hidden, and you shouldn't think of it as such. The decoder doesn't need to sample from these distributions to generate words during the training process, as the subsequent cell is fed with the ground-truth next word, rather than a word sam- pled from the previous output distribution. This is where the conductor comes in. In Figure 8-11 we linearly interpolate two of the dimensions of z to show how each dimension appears to encode a particular aspect of the track—for example, z[4] con- trols the immediate left/right direction of the track nearest the car and z[7] controls the sharpness of the approaching left turn. P

Wang, Xingdi Yuan, and Adam Trischler, “A Joint Model for Question Answering and Question Genera- tion,” 5 July 2017, . The division of the image into patches arises naturally as a result of the discriminator’s convolutional architecture. To explain, let’s revisit the Coder brothers’ art exhibition, where a few changes have taken place since our last visit... The Variational Art Exhibition Determined to make the generative art exhibition work, Mr. N. Figure 2-16. So, what went wrong and how can the brothers improve their scheme? It remains to be seen whether we will one day read news articles or novels written by a generative model, but the recent progress in this area is staggering and it is certainly not outrageous to suggest that this one day may be the case. Perhaps you decided that pmodel should look like Figure 1-5—a rectangular box where points may be found, and an area outside of the box where there is no chance of finding any points. Compiling the GAN ### COMPILE MODEL THAT TRAINS THE DISCRIMINATOR self.discriminator.compile(optimizer= RMSprop(lr=0.0008) , loss = 'binary_crossentropy' , metrics = ['accuracy']) ### COMPILE MODEL THAT TRAINS THE GENERATOR self.discriminator.trainable = False model_input = Input(shape=(self.z_dim,), name='model_input') model_output = discriminator(self.generator(model_input)) 108 | Chapter 4: Generative Adversarial Networks self.model = Model(model_input, model_output) self.model.compile(optimizer=RMSprop(lr=0.0004) , loss='binary_crossentropy' , metrics=['accuracy']) The discriminator is compiled with binary cross-entropy loss, as the response is binary and we have one output unit with sigmoid activation. The MDN-RNN architecture It consists of an LSTM layer (the RNN), followed by a densely connected layer (the MDN) that transforms the hidden state of the LSTM into the parameters of mixture distribution. The context vector can then be passed to a Dense layer with softmax output as usual, to output a distribution for the potential next note. If you can count 1, 2, 3, 1, 2, 3, then there are three beats to each bar and you may be listening to a waltz. We can imagine that our dataset has been generated by some distribution pdata that favors some feature values over others. Figure 8-4. Note how the reward changes to 3.22 as the car rolls over new track tiles but is otherwise –0.1. Also, the action changes every five frames as the action_refresh_rate is 5. When we change the style vector (top right), both bars change in a similar way. By balancing how these two adversaries are trained, the GAN generator can gradually learn how to produce similar observations to those in the training set. In the previous chapter, we saw how we are able to stack LSTM layers by pass- ing the hidden states of the previous layer as input to the next LSTM layer. A diagram of the training process for the discriminator and generator is shown in Figure 4-7. 2 Alec Radford et al., “Language Models Are Unsupervised Multitask Learners,” 2019, code.com/paper/language-models-are-unsupervised-multitask. In this example, we define four different models on the VAE: full_model This is the full end-to-end model that is trained. We can see that model accuracy is still 49.0% even on images that it has never seen before. This movement has already started to gather momentum, particularly among artists and musicians. See how around note 20, the phrase ends on a low E-flat—it is com- mon in the Bach Cello Suites to then return to a higher, more sonorous range of the instrument for the start of next phrase, which is exactly what the model predicts. RNN Extensions | 183 184 | Chapter 6: Write Figure 6-11. As with all reinforcement learning solu- tions, there is a balance to be found between greedily searching for locally optimal solutions and exploring unknown areas of the parameter space for potentially better solutions. For further information on the model, there is also an excellent interactive explana- tion available online. Music is often polyphonic—that is, there are several streams of notes played simultaneously on different instruments, which combine to create harmonies that are either dissonant (clashing) or consonant (har- monious). If we weight the reconstruction loss too heavily, the KL loss will not have the desired 84 | Chapter 3: Variational Autoencoders regulatory effect and we will see the same problems that we experienced with the plain autoencoder. The individ- ual steps of the theory are clearer within this context because they involve people, actions, and emotions, all of which are well understood, rather than neural networks, backpropagation, and loss functions, which are abstract constructs. We can view some of the predictions on the test set using the predict method: CLASSES = np.array(['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog' , 'frog', 'horse', 'ship', 'truck']) preds = model.predict(x_test) preds_single = CLASSES[np.argmax(preds, axis = -1)] actual_single = CLASSES[np.argmax(y_test, axis = -1)] preds is an array of shape [10000, 10]—i.e., a vector of 10 class probabilities for each observation. This ensures that the latent space is continuous and that we are able to easily sample from it to to generate meaningful new observations. GAN Challenges | 113 Figure 4-12. The input shape is (None, 32, 32, 3)—Keras uses None to represent the fact that we can pass any number of images through the network simultaneously. The Generator Now let’s build the generator. Both are only minor adjustments to the framework we have explored thus far. y_train and y_test are numpy arrays with shape [50000, 1] and [10000, 1], respectively, containing the integer labels in the range 0 to 9 for the class of each image. Typically, CycleGAN gener- ators take one of two forms: U-Net or ResNet (residual network). If actions are random to begin with, why would the model learn anything except to predict random actions from the body in which it resides? When building generative models, it becomes even more important to understand the inner workings of your model since it is the middle layers of your network that capture the high-level fea- tures that you are most interested in. This demonstrates the impor- tance of being experimental with your model design and being comfortable with how the different layer types can be used to your advantage. If we do this multiple times, we would also ideally like to get a roughly equal mixture of different kinds of digit (i.e., it shouldn’t always produce the same digit). Coder chooses a point on the wall where he would like to place the marker to represent the artwork, as before. We saw that with a few minor adjustments, we can transform our autoencoder into a variational autoencoder, thus giving it the power to be a generative model. If you are already familiar with a particular technique, then have fun finding the parallels of each model element within the story! In Part I of this book I shall introduce the key techniques that we will be using to build generative models, including an overview of deep learning, variational autoen- coders, and generative adversarial networks. In particular: • Text data is composed of discrete chunks (either characters or words), whereas pixels in an image are points in a continuous color spectrum. 204 | Chapter 7: Compose Table 7-1. For example, on an 8-core machine, you could run: bash 01_generate_data.sh car_racing 8 125 300 0 5 This would start 8 processes running in parallel, each simulating 125 episodes, with a maximum of 300 timesteps each and an action refresh rate of 5 timesteps. You can find more about his artistic process on YouTube. In summary, the KL divergence term penalizes the network for encoding observa- tions to mu and log_var variables that differ significantly from the parameters of a standard normal distribution, namely mu = 0 and log_var = 0. Figure 8-14 explains the composition of the output from the MDN-RNN. The generator generator_input = Input(shape=(self.z_dim,), name='generator_input') x = generator_input x = Dense(np.prod(self.generator_initial_dense_layer_size))(x) if self.generator_batch_norm_momentum: x = BatchNormalization(momentum = self.generator_batch_norm_momentum)(x) 3 Source: Augustus Odena et al., “Deconvolution and Checkerboard Artifacts, 17 October 2016, 31MgHUQ. Keras requires the function to be named. The batch_size determines how many images are in each training step batch. In short, there has never been a better time to learn about generative modeling—so let’s get started! ix Objective and Approach This book covers the key techniques that have dominated the generative modeling landscape in recent years and have allowed us to make impressive progress in creative tasks. The Challenges of Generative Modeling | 23 Figure 1-11. Now that we are not using a recurrent layer, we also need to encode the position of each word in the sentence. On the balance sheet, this expense is listed as his kittyloss (KL) divulgence. Figure 2-14. Each position in the output from the final decoder layer is fed through one final dense layer with a softmax activation function to give next word probabilities. Summary | 59 CHAPTER 3 Variational Autoencoders In 2013, Diederik P. This is shown in Figure 6-11, and the overall model architecture is shown in Figure 6-12. , xK x1, x2 p x2 | x1 p x1 K = Π p xk x1, . We can of course do so, in order to visually inspect the performance of the agent, but it is not required for training. Even though their content may be very different, the Gram matrix—a measure of correlation between all pairs of features in the layer—is similar. Imagine two points in the latent space, A and B, that represent two images. For example, a per- son’s age, income, and number of website visits in the last month are all features that could help to predict if the person will subscribe to a particular online service in the coming month. 275 Five Years of Progress The Transformer Positional Encoding Multihead Attention The Decoder Analysis of the Transformer BERT GPT-2 MuseNet Advances in Image Generation ProGAN Self-Attention GAN (SAGAN) 275 277 279 280 283 283 285 285 286 287 287 289 Table of Contents | vii BigGAN StyleGAN Applications of Generative Modeling AI Art AI Music 291 292 296 296 297 10. To answer this question we can use the evaluate method provided by Keras: model.evaluate(x_test, y_test) 4 Samuel L. The concatenated vector is also passed through an input gate which, like the for- get gate, is a dense layer with weights matrix Wi, bias bi, and a sigmoid activation function. In the codebase, we use a master/slave setup, where there is a master process that sends out parameter sets to be tested to many slave processes in parallel. For this GAN we rescale the data to the range [–1, 1]. Your patience with me while I have taken time to finish the book is hugely appreciated, and I am greatly looking forward to all the machine learning projects we will complete together in the future! Particular thanks to Ross—had we not decided to start a business together, this book might never have taken shape, so thank you for believing in me as your business partner! I also want to thank anyone who has ever taught me anything mathematical—I was extremely fortunate to have fantastic math teachers at school, who developed my interest in the subject and encouraged me to pursue it further at university. Ideally, we’d like to obtain a roughly equal spread of digits when sampling randomly from our latent space. The ith element of the vector is 1 if the ith word of the document could be consid- ered part of an answer and 0 otherwise. This book is here to help you get your job done. To demonstrate the difference between the two methods, Examples 2-1 and 2-2 show the same network coded using a Sequential model and the Functional API. The seed text is a string of words that you would like to give the model to start the generation process (it can be blank). This means the encoder only needs to map each input to a mean vector and a var- iance vector and does not need to worry about covariance between dimensions. The maximum number of timesteps per episode (e.g., 300). We tell the network the shape of each data element to expect as a tuple. The generator also contains another new layer type, InstanceNormalization. Stacking layers in this way gives the model freedom to learn more sophisticated features from the data. Some the- ories even suggest that the output from this generative model is what we directly per- ceive as reality. The RNN framework is certainly flexible enough to conceive of an architecture whereby multiple lines of music are generated simultaneously, through a recurrent mechanism. To achieve this, we set the return_sequences parameter within the first LSTM layer to True. The encoder uses self-attention—that is, the query input and key–value input are the same (the output from the previous layer in the encoder). Coder’s confidence in the marker position isn’t just right (log_var dif- ferent from 0), both of which incur a cost. We also need to change the integer labeling of the images to one-hot-encoded vectors. However, as we shall see, many of the same techniques that have driven development in discriminative modeling, such as deep learning, can be utilized by generative models too. Moreover, it may be the style image set as a whole rather than one single image that captures the artist’s style, so we need to find a way to allow the model to learn about style across a whole collec- tion of images. Collect random rollout data Here, the agent does not care about the given task, but instead simply explores the environment at random. Building the MuseGAN generator chords_input = Input(shape=(self.z_dim,), name='chords_input') style_input = Input(shape=(self.z_dim,), name='style_input') melody_input = Input(shape=(self.n_tracks, self.z_dim), name='melody_input') groove_input = Input(shape=(self.n_tracks, self.z_dim), name='groove_input') # CHORDS -> TEMPORAL NETWORK self.chords_tempNetwork = self.TemporalNetwork() self.chords_tempNetwork.name = 'temporal_network' chords_over_time = self.chords_tempNetwork(chords_input) # [n_bars, z_dim] # MELODY -> TEMPORAL NETWORK melody_over_time = [None] * self.n_tracks # list of n_tracks [n_bars, z_dim] tensors self.melody_tempNetwork = [None] * self.n_tracks for track in range(self.n_tracks): self.melody_tempNetwork[track] = self.TemporalNetwork() melody_track = Lambda(lambda x: x[:,track,:])(melody_input) melody_over_time[track] = self.melody_tempNetwork[track](melody_track) 230 | Chapter 7: Compose # CREATE BAR GENERATOR FOR EACH TRACK self.barGen = [None] * self.n_tracks for track in range(self.n_tracks): self.barGen[track] = self.BarGenerator() # CREATE OUTPUT FOR EVERY TRACK AND BAR bars_output = [None] * self.n_bars for bar in range(self.n_bars): track_output = [None] * self.n_tracks c = Lambda(lambda x: x[:,bar,:] , name = 'chords_input_bar_' + str(bar))(chords_over_time) s = style_input for track in range(self.n_tracks): m = Lambda(lambda x: x[:,bar,:])(melody_over_time[track]) g = Lambda(lambda x: x[:,track,:])(groove_input) z_input = Concatenate(axis = 1 , name = 'total_input_bar_{}_track_{}'.format(bar, track))([c,s,m,g]) track_output[track] = self.barGen[track](z_input) bars_output[bar] = Concatenate(axis = -1)(track_output) generator_output = Concatenate(axis = 1, name = 'concat_bars')(bars_output) self.generator = Model([chords_input, style_input, melody_input, groove_input] , generator_output) The inputs to the generator are defined. A SAGAN-generated image of a bird (leftmost cell) and the attention maps of the final attention-based generator layer for the pixels covered by the three colored dots (rightmost cells)11 The red dot is a pixel that is part of the bird’s body, and so attention naturally falls on surrounding body cells. To download the Monet-to-photo dataset, run the following command from inside the book repository: bash ./scripts/download_cyclegan_data.sh monet2photo This time we will use the parameter set shown in Example 5-7 to build the model: Example 5-7. We convert this array of probabilities back into a single prediction using numpy’s argmax function. Probabilistic Generative Models | 13 Figure 1-7. 68 | Chapter 3: Variational Autoencoders Figure 3-6. After a suitable number of epochs, the discriminator and generator will have found an equilibrium that allows the generator to learn meaningful information from the discriminator and the quality of the images will start to improve (Figure 4-8). Advances in Image Generation | 295 Applications of Generative Modeling As is clear from the preceding examples, generative modeling has come a long way in the last five years. To summarize, the encoder will take each input image and encode it to two vectors, mu and log_var which together define a multivariate normal distribution in the latent space: mu The mean point of the distribution. A file storing the weights of the trained network is saved to ./rnn/rnn.json every 10 steps. Under this model, our MLE for the parameters would be: θj = 16 | nj + 1 N+d Chapter 1: Generative Modeling Now, every single combination has a nonzero probability of being sampled, including those that were not in the original dataset. The model has three outputs: 1 for the real images, –1 for the fake images, and a dummy 0 vector, which isn’t actually used but is required by Keras as every loss function must map to an output. We can see that while there is some degree of similarity between the gen- erated images and the training set, they are not identical and the GAN is also able to complete some of the unfinished drawings by, for example, adding legs or a head. GloVe Word Vectors The embedding layer is initialized with a set of pretrained word embeddings, rather than random vectors as we have seen previously. Incorpo- rating multiple heads allows each to learn a distinct attention and value mechanism, therefore enriching the output from the multihead attention layer. The input embedding matrix is added to the positional encoding matrix to give the input into the first encoder layer Multihead Attention This tensor then flows through to the first of six encoder layers. 286 | Chapter 9: The Future of Generative Modeling sive to store and compute. We don’t really want the content loss to care about the values of individual pixels; we’d rather that it scores images based on the presence and approximate position of higher-level features such as buildings, sky, or river. It can only output probabilities against existing images, as this is what it has been trained to do. A special thanks goes to Michele Cronin, who has been there at each step, providing useful feedback and sending me friendly reminders to keep com- pleting chapters! Also to Katie Tozer, Rachel Head, and Melanie Yarbrough for get- ting the book into production, and Mike Loukides for first reaching out to ask if I’d be interested in writing a book. You can imagine a number of plausible different endings to your favorite TV show, and you can plan your week ahead by working through various futures in your mind’s eye and taking action accordingly. 268 | Chapter 8: Play Figure 8-19. 3 Miles Brundage et al., “The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation,” February 2018, . First we will look at CycleGAN, which as the name suggests is an adaptation of the GAN architecture that allows the model to learn how to convert a photograph into a paint- ing in a particular style (or vice versa). Example 6-1. Before, the only feedback mechanism was the loss of earnings at the ticket office resulting from poorly reconstructed images. It has length equal to the number of units in the cell and stores values between –1 and 1. Pass the chords input through the temporal network. To get around this issue, we use the Python partial function. This can then be fed to a Dense layer with softmax output to predict a distribution for the next note in the sequence. Our model would learn that certain colors, 2 | Chapter 1: Generative Modeling shapes, and textures are more likely to indicate that a painting is by the Dutch master, and for paintings with these features, the model would upweight its prediction accordingly. 140 | Chapter 5: Paint An InstanceNormalization layer normalizes every single observation individually, rather than as a batch. Kingma and Max Welling, “Auto-Encoding Variational Bayes,” 20 December 2013, https:// arxiv.org/abs/1312.6114. When Florida isn’t looking, Granny Smith sneaks into the orange section and starts painting the oranges red to look like apples! Florida has exactly the same plan, and tries to make Granny Smith’s apples more orange-like with a suitably colored spray when her back is turned. The only true reward is 300 | Chapter 10: Conclusion staying alive, and this can hardly be used to explain every action of an intelligent being. In this chapter we will also explore some new techniques that have led to promising results in the field of question-answer pair generation. We want gener- ated images to be predicted close to 1 (real) because the generator is strong, not because the discriminator is weak. Granny Smith tests her apple painting skills on her apples for the same reason. There is one cell in an LSTM layer that is defined by the number of units it contains, in the same way that the prisoner cell from our ear- lier story contained many prisoners. Coder, and the reconstructed paintings produced by Mr. D. Now we shall look at a different application of style transfer, where we do not have a training set at all, but instead wish to transfer the style of one single image onto another, as shown in Figure 5-14. Each output position in the attention matrix only computes weights for a subset of input positions, thereby reducing the computational complexity and memory required to train the model. Coder are using representation vectors inside a two-dimensional latent space (the wall) to encode each image. Remember, our goal is to be able to choose a random point in the latent space, pass this through the decoder, and obtain an image of a digit that looks real. Figure 7-13. First, we now have a well-defined distribution that we can use for choosing points in the latent space—the standard normal distribution. How should we train this network? Bidirectional Cells For prediction problems where the entire text is available to the model at inference time, there is no reason to process the sequence only in the forward direction—it could just as well be processed backwards. The new generative art exhibition As you can see, the plan was not a great success. We can see that every track has changed, as expected, and also that the two bars exhibit different properties. Agent The entity that takes actions in the environment. In the first passage (the lion said ,), the model guesses that speech marks follow with 98% likelihood, so that the clause precedes the spoken dialogue. This approach led to world-best scores for both of the tasks on which it was tested. Since we cannot view all dimensions simultaneously, we can instead check the distribution of each latent dimension individually. By epoch 10, the model has begun to generate small runs of notes, and by epoch 20 it is pro- ducing interesting rhythms and is firmly established in a set key (E-flat major). Both are so convinced that they have the better fruit display that they agree to a deal: the profits from the sales of apples will go entirely to Granny Smith and the profits from the sales of oranges will go entirely to Florida. Instead Keras utilizes one of three backend libraries for this pur- pose: TensorFlow, CNTK, or Theano. At this point, I must switch into pure speculation mode as we are close to the edge of what we currently understand about the human brain (and certainly at the very edge of what I understand about the human brain). kl_loss is minimized to 0 when mu = 0 and log_var = 0 for all dimensions. Here, each individual feature contains a nugget of information about the observation, and the model would learn how these features interact to influence the response. The field has developed to the point where is it not unreasonable to suggest that the next generation will be just as comfortable marveling at computergenerated art as human art, reading computer-generated novels, and listening to computer-generated music in their favorite style. Setup We are now ready to start exploring how to build and train this model in Keras. In one dimension, it is defined by two variables: the mean (μ) and the variance (σ2). Coder’s opinion of where the marker should appear on the wall. A batch normalization layer calculates the mean and standard deviation of each of its input channels across the batch and normalizes by subtracting the mean and dividing by the standard deviation. It has been shown that the Conv2DTranspose method can lead to artifacts, or small checker- board patterns in the output image (see Figure 4-6) that spoil the quality of the out- put. One generation of the algorithm demonstrated on a toy example is shown in Figure 8-15. I also highly recommend the website Papers with Code, where you can find the latest state-of-the-art results in a variety of machine learning tasks, alongside links to the papers and official GitHub repositories. The general approach would be to build a computer simulation of the terrain and then run many experiments where the agent tries out different strategies. Image C is slightly dif- ferent in style to images A and B. Since our earliest days as cave people, we have sought opportunities to generate original and beautiful creations. First we expand the tensor along the timestep axis… …then along the pitch axis. We then calculate the value of the function for each candidate and isolate the best 25%, shown in purple—we’ll call this set of points P. The population size, pop_size = n * t. Morphing between two faces It is worth noting the smoothness of the transition—even where there are multiple features to change simultaneously (e.g., removal of glasses, hair color, gender), the VAE manages to achieve this fluidly, showing that the latent space of the VAE is truly a continuous space that can be traversed and explored to generate a multitude of dif- ferent human faces. We then follow this with a normal convolutional layer with stride 1 to perform the convolution operation. When a customer requests to see the painting, Mr. D. We shall start by fram- ing these advancements alongside examples that we have already explored and walk- ing through the most cutting-edge architectures available today. 6 Source: Sergey Ioffe and Christian Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 11 February 2015, . An attention mechanism within the context of an encoder-decoder network, connected to decoder cell i While there are many copies of the attention mechanism within the encoder–decoder network, they all share the same weights, so there is no extra overhead in the number of parameters to be learned. First, we collapse the tensor along the bar axis. Figure 3-18. It is now one of the most utilized frameworks for building machine learning solutions, with particular emphasis on the manipulation of tensors (hence the name). We shall also see how the encoder–decoder architecture works and build a sim- ple question-answer generator. A comparison of the architectures for training in the real environment and the dream environment follows: the real-world architecture is shown in Figure 8-19 and the indream training setup is illustrated in Figure 8-20. Artifacts when using convolutional transpose layers3 The code for building the generator is given in Example 4-3. In Figure 3-9 we can see three points in the latent space and their decoded images, none of which are par- ticularly well formed. The ReLU, LeakyReLU, and sigmoid activation functions The softmax activation is useful if you want the total sum of the output from the layer to equal 1, for example, for multiclass classification problems where each observation only belongs to exactly one class. Train the MDN-RNN We take batches of 100 episodes and load the correspond- ing mu, log_var, action, and reward variables at each timestep that were gener- ated in step 3. Notice that we do not specify the batch size; this isn’t necessary as we can pass any number of images into the Input layer simulta- neously. 36 | Chapter 2: Deep Learning It’s worth noting the shape of the image data in x_train: [50000, 32, 32, 3]. Every day, Edward posts the latest word from his novel into the cell, and it is the job of the inmates to individually update their opinions on the current state of the story, based on the new word and the opinions of the inmates from the previous day. David has won several international machine learning competitions, including the InnoCentive Predicting Product Purchase challenge, and was awarded first prize for a visualization that enables a pharmaceutical company in the US to optimize site selec- tion for clinical trials. Your algorithm can then tweak the word embeddings to suit the particular context of your machine learning problem. Summary | 29 In Chapter 2, we will begin our exploration of deep learning and see how to use Keras to build models that can perform discriminative modeling tasks. Coder attempts to re-create the artwork using just the coordinates of the relevant marker on the wall. How- ever, the evaluation performed after every 10 generations is still conducted in the real environment and is therefore scored based on the sum of rewards from the OpenAI Gym environment, so that we can compare training methods. Advances in Image Generation | 289 Figure 9-9. 17 Xun Huang and Serge Belongie, “Arbitrary Style Transfer in Real-Time with Adaptive Instance Normaliza- tion,” 20 March 2017, . Diagram of the four CycleGAN models4 Running the notebook 05_01_cyclegan_train.ipynb in the book repository will start training the CycleGAN. Selling or distributing a CD-ROM of examples from O’Reilly books does require permission. Our first basic example of a generative model utilized the Naive Bayes assump- tion to produce a probability distribution that was able to represent inherent structure in the data and generate examples outside of the training set. I recommend that even if you are just starting out building lin- ear models with Keras, you still use the Functional API rather than Sequential mod- els, since it will serve you better in the long run as your neural networks become more architectually complex. In this case, the gradient is zero and therefore no error is propagated back through this unit. Whereas BERT was Google’s response to OpenAI’s earlier GPT model, GPT-2 is a direct response to BERT. The cube represents the extremely high-dimensional space of all images; rep- resentation learning tries to find the lower-dimensional latent subspace or manifold on which particular kinds of image lie (for example, the dog manifold) In summary, representation learning establishes the most relevant high-level features that describe how groups of pixels are displayed so that is it likely that any point in the latent space is the representation of a well-formed image. What does this mean in practice? We will test our model using the test datasets. This example follows the Jupyter notebook 03_01_autoencoder_train.ipynb in the book repository. Conversely, the brown and spikiness channels are often activated together at the same point in image C, but not in images A and B. Figure 7-5. Joining the Encoder to the Decoder To train the encoder and decoder simultaneously, we need to define a model that will represent the flow of an image through the encoder and back out through the decoder. Parallelizing CMA-ES One of the great benefits of CMA-ES is that it can be easily parallelized using a Python library created by David Ha called es.py. Set this flag initially; if it’s not set, the code will look for a ./vae/vae.json file and continue training a previous model. For each of the 32 dimensions of z, sample from the distributions created from log_pi (i.e., choose which of the 5 distributions should be used for each dimen- sion of z). The epochs determine how many times the network will be shown the full train- ing data. Summary | 95 Variational autoencoders solve these problems, by introducing randomness into the model and constraining how points in the latent space are distributed. I like to place the BatchNorm alization before the Activation, but some successful architectures use these layers the other way around. Compiling the discriminator self.d_A = self.build_discriminator() self.d_B = self.build_discriminator() self.d_A.compile(loss='mse', optimizer=Adam(self.learning_rate, 0.5), metrics=['accuracy']) self.d_B.compile(loss='mse', optimizer=Adam(self.learning_rate, 0.5), metrics=['accuracy']) 144 | Chapter 5: Paint However, we cannot compile the generators directly, as we do not have paired images in our dataset. The dynamic element of the performance is provided by the final two sections, which ensure that the music is constantly changing with each bar that goes by. For example, take a look inside the white box marked in the center of the dia- gram, and note how there is a strip of boxes in the middle that cuts through the usual pattern of looking back at the previous four to six notes. Figure 6-13. There is a large gray gap between the low E-flat (pitch number 39) and the next note, which is predicted to be around pitch number 50, rather than continuing to rumble around the depths of the instrument. The aim of the MDN-RNN is to predict one timestep ahead into the future—we can then use the internal hidden state of the LSTM as part of the input into the controller. Autoencoders | 69 Figure 3-7. This is applicable to a range of problems aside from question–answer pair generation, such as transla- tion and text summarization. To ensure a balanced mix, we use a set of interpolated images that lie at randomly chosen points along lines connecting the batch of real images to the batch of fake images pairwise, as shown in Figure 4-16. Summary In this chapter we have seen how variational autoencoders are a powerful tool in the generative modeling toolbox. 190 | Chapter 6: Write And our second network should be able to generate a question, given each of the answers, such as: "Who scored the winning goal?" "Who won the match?" "Who were Arsenal playing?" "How much did the striker cost?" "When is the next match?" Let’s first take a look at the dataset we shall be using in more detail. This means that not only do we have complete control over the generated image through the latent vector ℰ, but we can also switch the ℰ vector at different points in the synthesis network to change the style at a variety of levels of detail. In Keras, activation functions can also be defined in a separate layer as follows: x = Dense(units=200)(x) x = Activation('relu')(x) This is equivalent to: x = Dense(units=200, activation = 'relu')(x) Your First Deep Neural Network | 39 In our example, we pass the input through two dense hidden layers, the first with 200 units and the second with 150, both with ReLU activation functions. The order of words is highly important in text data and words wouldn’t make sense in reverse, whereas images can usually be flipped without affecting the content. We could also just one-hot encode each input token, but using an embedding layer is preferred because it makes the embedding itself trainable, thus giving the model more flexibility in deciding how to embed each token to improve model performance. First, see how the music is becoming more sophisticated as training progresses. Style Loss Style loss is slightly more difficult to quantify—how can we measure the similarity in style between two images? If your neural network is designed to solve a regression problem (i.e., the output is continuous), then you can use the mean squared error loss. An LSTM cell maintains a cell state, Ct, which can be thought of as the cell’s internal beliefs about the current status of the sequence. Environment name (e.g., car_racing) 2. However, after a few weeks of her training ritual, she gradually gets better at this, which means that she can provide better feedback to Gene so that he can adjust his camera accordingly in his training session. Example 4-9. Example 5-3. During the training process, batches of images are passed through the network and the output is compared to the ground truth. Experiments have shown that WGAN-GPs can still produce excel- lent results even without batch normalization in the critic. The encoder is a GRU layer that is fed the hidden states for the given answer range as input data. Let’s see how the game state, action, reward, and episode are defined for this environment: Game state A 64 × 64–pixel RGB image depicting an overhead view of the track and car. This was also the aim Autoencoders | 73 of the Coder brothers when they were choosing random points on their wall to gen- erate new artwork for their exhibition. Your First Music-Generating RNN | 213 Figure 7-8. The resulting vec- tor, zt, has length equal to the number of units in the cell and stores values between 0 and 1, which are used to determine how much of the new beliefs, ht, to blend into the current hidden state, ht–1. Explaining these developments and their repercussions in detail could easily fill another book. 96 | Chapter 3: Variational Autoencoders CHAPTER 4 Generative Adversarial Networks On Monday, December 5, 2016, at 2:30 p.m., Ian Goodfellow of Google Brain presen- ted a tutorial entitled “Generative Adversarial Networks” to the delegates of the Neu- ral Information Processing Systems (NIPS) conference in Barcelona.1 The ideas presented in the tutorial are now regarded as one of the key turning points for gener- ative modeling and have spawned a wide variety of variations on his core idea that have pushed the field to even greater heights. As expected, around half are correct: import matplotlib.pyplot as plt n_to_show = 10 indices = np.random.choice(range(len(x_test)), n_to_show) fig = plt.figure(figsize=(15, 3)) fig.subplots_adjust(hspace=0.4, wspace=0.4) for i, idx in enumerate(indices): img = x_test[idx] ax = fig.add_subplot(1, n_to_show, i+1) ax.axis('off') ax.text(0.5, -0.35, 'pred = ' + str(preds_single[idx]), fontsize=10 , ha='center', transform=ax.transAxes) Your First Deep Neural Network | 45 ax.text(0.5, -0.7, 'act = ' + str(actual_single[idx]), fontsize=10 , ha='center', transform=ax.transAxes) ax.imshow(img) Figure 2-9 shows a randomly chosen selection of predictions made by the model, alongside the true labels. This doesn’t necessarily happen immediately as you start training the network. Some examples of this style of task are: Encoder–Decoder Models | 187 Language translation The network is fed a text string in the source language and the goal is to output the text translated into a target language. The sequence length that we use to train the model is a parameter of the training pro- cess. You can see this convolutional process worked through manually in the notebook 02_02_deep_learning_convolutions.ipynb in the book repository. 288 | Chapter 9: The Future of Generative Modeling Figure 9-8. Example 4-7. The full autoencoder ### THE FULL AUTOENCODER model_input = encoder_input # model_output = decoder(encoder_output) # self.model = Model(model_input, model_output) # The input to the autoencoder is the same as the input to the encoder. So, for example, in a single 2 Hao-Wen Dong et al., “MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment,” 19 September 2017, . After applying a series of Conv2D layers, we need to flatten the tensor using the Keras Flatten layer. Reconstruction. Evolutionary strategies are becoming a popular choice for solving reinforcement learning problems, due to their simplicity, efficiency, and scalability. The CycleGAN gets its name from this cyclic reconstruction criterion. Example 5-6. The excerpt from the rnn/arch.py file for the MDN-RNN in Example 8-3 shows how we construct the custom loss function in Keras. We often draw a recurrent layer as a chain of cells unrolled, as it helps to visualize how the hidden state is updated at each timestep. As an example, suppose we want the model to translate the following sentence into German: I scored a penalty in the football match against England. You can also try decaying the weight given to each layer in the style loss func- tion, to bias the model toward transferring finer or coarser style features. Here, our aim is to build a model that can transform an input base image in order to give the impression that it comes from the same collection as a given set of style images. For example, knowing that pixel 234 of an image is a muddy shade of brown doesn’t really help identify if the image is of a house or a dog, and knowing that character 24 of a sentence is an e doesn’t help predict if the text is about football or politics. A summary of a typical generative modeling process is shown in Figure 1-1. You will then learn how to use the Python library Keras to start building your own GANs. First though, we shall take a trip into the wilderness to meet Gene… Ganimals One afternoon, while walking through the local jungle, Gene sees a woman thumbing through a set of black and white photographs, looking worried. I encourage you to do the same and to continue learning more about generative models from all the great material that is available online and in other books. Ten new Wrodl styles, generated using the Naive Bayes model For this simple problem, the Naive Bayes assumption that each feature is independent of every other feature is reasonable and therefore produces a good generative model. Model architecture for generating question–answer pairs from keras.layers import Input, Embedding, GRU, Bidirectional, Dense, Lambda from keras.models import Model, load_model import keras.backend as K from qgen.embedding import glove #### PARAMETERS #### VOCAB_SIZE = glove.shape[0] # 9984 EMBEDDING_DIMENS = glove.shape[1] # 100 GRU_UNITS = 100 DOC_SIZE = None ANSWER_SIZE = None Q_SIZE = None document_tokens = Input(shape=(DOC_SIZE,), name="document_tokens") embedding = Embedding(input_dim = VOCAB_SIZE, output_dim = EMBEDDING_DIMENS , weights=[glove], mask_zero = True, name = 'embedding') document_emb = embedding(document_tokens) answer_outputs = Bidirectional(GRU(GRU_UNITS, return_sequences=True) , name = 'answer_outputs')(document_emb) answer_tags = Dense(2, activation = 'softmax' , name = 'answer_tags')(answer_outputs) The document tokens are provided as input to the model. We reshape this vector to a 1 × 1 tensor with 32 channels, so that we can apply convolutional transpose operations to it. 4. This is useful for reducing the spatial size of the tensor as it passes through the network, while increasing the number of channels. Example 4-2. To be able to build networks with branches, we need to use the Functional API, which is a lot more flexible. The encoder concludes with one final skip connection and normalization layer. Training the Model Thus far, we haven’t shown the model any data and have just set up the architecture and compiled the model with a loss function and optimizer. Example 5-12 is a code snippet showing the training loop. The controller receives higher rewards for safer strategies that encounter well-understood states and therefore tend to generalize better to the real environment. In addition to the abstract mechanics of each environment, OpenAI Gym also pro- vides graphics that allow you to watch your agent perform in a given environment. This is known as neural style transfer.12 Figure 5-14. If so, you should consider whether the number of units in the layer could be reduced to speed up training. However, if the agent were able to simulate its environment through a genera- tive model, it wouldn’t need to test out the strategy in the computer simulation or in 4 Source: Brundage et al., 2018. Training the generator is more difficult as there is no training set that tells us the true image that a particular point in the latent space should be mapped to. 15 Tero Karras, Samuli Laine, and Timo Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” 12 December 2018, . Game state The data that represents a particular situation that the agent may encounter (also just called a state), for example, a particular chessboard configuration with accompanying game information such as which player will make the next move. Let’s relabel this image with the appropriate functions from the codebase, so that we can see this more explicitly (Figure 5-8). This way, when we generate text using the model, we can seed the model with this block of characters, so that the model knows to start a new story from scratch. Inference models decoder_inputs_dynamic = Input(shape=(1,), name="decoder_inputs_dynamic") decoder_emb_dynamic = embedding(decoder_inputs_dynamic) decoder_init_state_dynamic = Input(shape=(2 * GRU_UNITS,) , name = 'decoder_init_state_dynamic') decoder_states_dynamic = decoder_cell(decoder_emb_dynamic , initial_state = [decoder_init_state_dynamic]) decoder_outputs_dynamic = decoder_projection(decoder_states_dynamic) question_model = Model([decoder_inputs_dynamic, decoder_init_state_dynamic] , [decoder_outputs_dynamic, decoder_states_dynamic]) We can then use this model in a loop to generate the output question word by word, as shown in Example 6-9. Figure 2-18. Defining the GAN gan = GAN(input_dim = (28,28,1) , discriminator_conv_filters = [64,64,128,128] , discriminator_conv_kernel_size = [5,5,5,5] , discriminator_conv_strides = [2,2,2,1] , discriminator_batch_norm_momentum = None , discriminator_activation = 'relu' , discriminator_dropout_rate = 0.4 , discriminator_learning_rate = 0.0008 , generator_initial_dense_layer_size = (7, 7, 64) , generator_upsample = [2,2, 1, 1] , generator_conv_filters = [128,64, 64,1] , generator_conv_kernel_size = [5,5,5,5] , generator_conv_strides = [1,1, 1, 1] , generator_batch_norm_momentum = 0.9 , generator_activation = 'relu' , generator_dropout_rate = None , generator_learning_rate = 0.0004 , optimiser = 'rmsprop' , z_dim = 100) Let’s first take a look at how we build the discriminator. Therefore these kinds of encoder–decoder networks sometimes struggle to retain all the required information for the decoder to accurately translate the source. Most often, we want to generate unstructured data such as new images or original strings of text, which is why deep learning has had such a profound impact on the field of generative modeling. The job of the layer is to learn which positions in the key–value input it should attend to, for every position of the query input. Figure 5-4. Figure 3-1. 31 Figure 2-1. CNN performance As you can see, this model is now achieving 71.5% accuracy, up from 49.0% previ- ously. It therefore makes intuitive sense that the focus of parametric modeling should be to find the optimal value θ of the parameter set that maximizes the likelihood of observ- ing the dataset X. It can also guess that the article that begins a story about an eagle is more likely to be an, rather than a. Your First GAN | 101 The full architecture of the discriminator we will be building is shown in Figure 4-4. Output from in-dream training When training in the dream environment, the scores of each generation are given in terms of the average sum of the dream rewards (i.e., 0 or 1 at each timestep). Note that representation learning doesn’t just assign values to a given set of features such as blondeness of hair, height, etc., for some given image. Along the way, we will pick apart both types of model, to understand how they work at a granular level. This is where the V

need to get the data into the correct shape to feed the GAN. Setting Up Your Environment Throughout this book, there are many worked examples of how to build the models that we will be discussing in the text. Figure 8-17 explains this in a diagram. 303 viii | Table of Contents Preface What I cannot create, I do not understand. After many iterations of this process, Edward finds that the system has become quite accomplished at generating realistic-looking text. We'll look at some of these briefly next. It is only the definition and compilation of the critic that we need to change. Only a tiny proportion of these are recognizable faces, and an even smaller subset are faces that adhere to the fashion rules on Planet Pixel. To start training the VAE, run the following command from your terminal: python 02_train_vae.py --new_model [--N] [--epochs] 248 | Chapter 8: Play where the parameters are as follows: --new_model Whether the model should be trained from scratch. This then forms the prisoner's new opinion for the day. Example 8-2. The model has estimated the probability of seeing each feature value independently, so that under the Naive Bayes assumption we can multiply these probabilities to build our full density function, pθ(x). We will see that many of the RNN techniques from the previous chapter on text generation can also be used for music generation as the two tasks share many common themes. As with most deep learning principles, there is no golden rule that applies in every situation—the only way to know for sure what's best is to test different architectures and see which performs best on a holdout set of data. Let's now see how we can build a CycleGAN model in Keras. It contains a given number of units that are densely connected to the previous layer— that is, every unit in the layer is connected to every unit in the previous layer, through a single connection that carries a weight (which can be positive or negative). CycleGAN | 135 enormous effort to arrange photos of horses and zebras standing in identical positions. We shall see the ways that it has been fine-tuned and adapted to continually push the boundaries of what generative modeling is able to achieve. StyleGAN One of the most recent additions to the GAN literature is StyleGAN,15 from NVIDIA Labs. 212 | Chapter 7: Compose Analysis of the RNN with Attention The following analysis can be produced by running the notebook 07_03_lstm_compose_analysis.ipynb from the book repository, once you have trained your network. You can download a collection of Aesop's Fables from Project Gutenberg. Current neuroscientific theory suggests that our perception of reality is not a highly complex discriminative model operating on our sensory input to produce predictions of what we are experiencing, but is instead a generative model that is trained from birth to produce simulations of our surroundings that accurately match the future. Variational autoencoders assume that there is no correlation between any of the dimensions in the latent space and therefore that the covariance matrix is diagonal. Thus each observation now has 32 × 32 = 1,024 features and each feature can take any of 256 values (the individual colors in the palette). Example 3-7. The real and fake images are passed through the critic in order to calculate the Wasserstein loss. BigGAN BigGAN,12 developed at DeepMind, extends the ideas from the SAGAN paper with extraordinary results. In this case, we are creating the CarRacing environ- ment, with a few tweaks. Including KL divergence in the loss function ### COMPILATION optimizer = Adam(lr=learning_rate) def vae_r_loss(y_true, y_pred): r_loss = K.mean(K.square(y_true - y_pred), axis = [1,2,3]) return r_loss_factor * r_loss def vae_kl_loss(y_true, y_pred): kl_loss = -0.5 * K.sum(1 + self.log_var - K.square(self.mu) - K.exp(self.log_var), axis = 1) return kl_loss def vae_loss(y_true, y_pred) r_loss = vae_r_loss(y_true, y_pred) kl_loss = vae_kl_loss(y_true, y_pred) return r_loss + kl_loss optimizer = Adam(lr=learning_rate) self.model.compile(optimizer=optimizer, loss = vae_loss , metrics = [vae_r_loss, vae_kl_loss]) Analysis of the Variational Autoencoder All of the following analysis is available in the book repository, in the notebook 03_04_vae_digits_analysis.ipynb. You can use any dataset you wish, but if you want to work with this dataset, you can find instructions for downloading the MIDI files in the notebook. Secondly, there are not so many generated digits that are badly formed, since the latent space is now locally continuous due to fact that the encoder is now stochastic, rather than deterministic. He is able to create new forms by producing a line drawing (i.e., in the edge-highlighted space) and allowing the model to convert back to the color photograph domain. Coder once again tries his strategy of placing markers on portions of the wall that are empty, so that Mr. D. While it is not essential to run the examples in this book on a GPU, it may help to speed up the training process. The Keras model that defines the encoder—a model that takes an input image and encodes it into the 2D latent space. A file storing the weights of the trained network is saved to ./vae/vae.json every epoch. We can see that the training dataset of 50,000 observations has been shown to the net- work 10 times (i.e., over 10 epochs), at a rate of approximately 160 microseconds per observation. As strides = 2 and padding = "same", the width and height of the output are both halved to 16, and since there are 10 filters the output of the first layer is a batch of tensors each having shape [16, 16, 10]. A Question and Answer Generator | 191 question_output_tokens The tokenized question, offset by one timestep (e.g., [39, 1, 52, 1866, ...], padded with zeros to be of length max_question_length. However, in most scenarios the Generative Modeling Frame- work captures how we should broadly think about tackling the problem. If all the examples generated by the model were like this, it would break Rule 2 of the Generative Modeling Framework. CMA-ES3 For the car racing task, we do not have a well-defined function to maximize, but instead an environment where the 867 parameters to be optimized determine how well the agent scores. 278 | Chapter 9: The Future of Generative Modeling To understand how this works in practice, let's follow a sample input sequence through the model, step by step. To do this, run the following command from your terminal inside your virtual environment: python -m ipykernel install --user --name generative This gives you access to the virtual environment that you've just set up (genera tive) inside Jupyter notebooks. For example, G2 is an octave below G3. Example 4-1. Figure 2-13 represents the following network in Keras: input_layer = Input(shape=(32,32,3)) conv_layer_1 = Conv2D(filters = 10 , kernel_size = (4,4) , strides = 2 , padding = 'same')(input_layer) conv_layer_2 = Conv2D(filters = 20 , kernel_size = (3,3) , strides = 2 , padding = 'same')(conv_layer_1) 5 Source: Vincent Dumoulin and Francesco Visin, "A Guide to Convolution Arithmetic for Deep Learning," 12 January 2018, . The MLEs for the parameters under the Naive Bayes model topType NoHair LongHairBun LongHairCurly LongHairStraight ShortHairShortWaved ShortHairShortFlat ShortHairFrizzle Grand Total ^ θ 7 0.14 0 0.00 1 0.02 23 0.46 1 0.02 11 0.22 7 0.14 50 1.00 n hairColor n Black Blonde Brown PastelPink Red SilverGrey Grand Total 7 6 2 3 8 24 50 ^ θ 0.14 0.12 0.04 0.06 0.16 0.48 1.00 clothingColor n Black Blue01 Grey01 PastelGreen PastelOrange Pink Red White Grand Total 0 4 10 5 2 4 3 22 50 ^ θ 0.00 0.08 0.20 0.10 0.04 0.08 0.06 0.44 1.00 9 The –5 is due to the fact that the last parameter for each feature is forced to ensure that the sum of the param- eters for this feature sums to 1. If the length of the sequence is now greater than the sequence length that the model was trained on, we remove one element from the start of the sequence. At the apex of the U, the feature maps will have learned a contex- tual understanding of what is in the image, with little understanding of where it is located. The hidden state of the previous timestep, ht–1, and the current word embedding, xt, are concatenated and used to create the reset gate. In this chapter, we will run with this idea and see where it takes us. A linear interpolation of two dimensions of z This shows that the latent space that the VAE has learned is continuous and can be used to generate new track segments that have never before been observed by the agent. There are large gaps between colors containing few points. The summary of the model Notice how Keras uses None as a marker to show that it doesn't yet know the number of observations that will be passed into the network. By sampling from a distribution in the latent space, we can use the generative model to map this vector to a novel image in the pixel space. CMA-ES Evolutionary strategies generally adhere to the following process: 1. The output from the mixture density network Sampling the Next z and Reward from the MDN-RNN We can sample from the MDN output to generate a prediction for the next z and reward at the following timestep, through the following process: 1. In other words, it takes place in the agent's hallucinated version of how the environment behaves, rather than the real thing. 110 | Chapter 4: Generative Adversarial Networks Figure 4-9. This shows how each of the input parameters can be used to directly influence highlevel features of the generated musical sequence, in much the same way as we were able to adjust the latent vectors of VAEs and GANs in previous chapters to alter the appearance of a generated image. The Keras code to build a bar generator is given in Example 7-6. WGAN-GP | 121 Figure 4-15. Compiling the Model In this step, we compile the model with an optimizer and a loss function: from keras.optimizers import Adam opt = Adam(lr=0.0005) model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy']) The loss function is used by the neural network to compare its predicted output to the ground truth. An attribution usually includes the title, author, publisher, and ISBN. Therefore the output of the discriminator is a tensor that contains the predicted probability for each patch, rather than just a single number. This powerful technique will allow us to not only generate real- istic faces, but also alter existing images—for example, by adding a smile or changing the color of someone's hair. We apply Conv2DTranspose layers to expand the size of the tensor along one axis, so that it is the same length as n_bars. Point 2 explains the lack of diversity in the generated images. 76 | Chapter 3: Variational Autoencoders Figure 3-10. This means that GPT-2 does not use information from subsequent words to form representations of the cur- rent word and therefore is set up for sentence generation tasks, such as the Aesop's Fables task that we explored in Chapter 6. At timestep t, the cell uses the previous value of the hidden state ht–1 together with the data from the current timestep xt to produce an updated hidden state vector ht. Everything makes sense to you. Glowing with success, you turn your attention to another planet's fashion dilemma—but this time the problem isn't quite as simple. After training with these simple changes, Mr. N. Sample results from the model The encoder extracts the context from each of these possible answers, so that the decoder is able to generate suitable questions. How many parameters do we now need to estimate? The natural exten- sion of the WGAN is the WGAN-GP, which places the 1-Lipschitz requirement at the heart of the training process by including a term in the loss function to pull the gradi- ent norm toward 1. This sparked a deep learning boom that has resulted in the error rate falling even further year after year. In fact, even in the original 120 | Chapter 4: Generative Adversarial Networks WGAN paper the authors write, "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint." A strong critic is pivotal to the success of a WGAN, since without accurate gradients, the generator cannot learn how to adapt its weights to produce better samples. This example demonstrates how we can chain convolutional layers together to create a convolutional neural network. A 3 × 3 × 1 kernel (gray) being passed over a 5 × 5 × 1 input image (blue), with padding="same" and strides = 1, to generate the 5 × 5 × 1 output (green)5 Setting padding = "same" is a good way to ensure that you are able to easily keep track of the size of the tensor as it passes through many convolutional layers. 282 | Chapter 9: The Future of Generative Modeling The Decoder The decoder layers are very similar to the encoder layers, with two key differences: 1. We want our model to be able to perform the same trick—in particular, we want it to not only care about the hidden state of the network now, but also to pay Your First Music-Generating RNN | 207 particular attention to the hidden state of the network eight notes ago, when the pre- vious low G was registered. Identity. In the original GAN paper, dense layers were used in place of the convolutional lay- ers. With enough data, we could train a discriminative model to predict if a given painting was painted by Van Gogh. However, most of the state-of-the-art generative models today are GAN-based, as the rewards for training large-scale GANs on GPUs over a longer period of time are significant. So far, we have only worked with Conv2D layers, applicable to three-dimensional input images (width, height, channels). • We are impressed by pmodel if: — Rule 1: It can generate examples that appear to have been drawn from pdata. The integer lookup dictionaries for pitch and duration We then create the training set by splitting the data into small chunks of 32 notes, with a response variable of the next note in the sequence (one-hot encoded), for both pitch and duration. This is a collection of free ebooks that can be downloaded as plain text files. 12 Andrew Brock, Jeff Donahue, and Karen Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," 28 September 2018, . 14 Source: Brock, Donahue, and Simonyan, 2018. For example: 4 | Chapter 1: Generative Modeling • Given a satellite image, a government defense official would only care about the probability that it contains enemy units, not the probability that this particular image should appear. Figure 1-3 shows the striking progress that has already been made in facial image generation since 2014.3 There are clear positive applications here for industries such as game design and cinematography, and improvements in automatic music genera- tion will also surely start to resonate within these domains. The dataset was collected as part of an online game that challenged players to draw a picture of an object or concept, while a neural network tries to guess the subject of the doodle. The punctuation example from Figure 6-10 shows how the model is also sensitive to subtle changes in the input sequence. 100d.txt (6 billion words each with an embedding of length 100) from the GloVe project website and then run the following Python script from the book repository to trim this file to only include words that are present in the training corpus: python ./utils/write.py The second part of the model is the encoder–decoder network that takes a given answer and tries to formulate a matching question (the bottom part of Figure 6-15). To get the weighted sum of the hidden states, we need to use a RepeatVector layer to copy the weights rnn_units times to form a matrix of shape [rnn_units, seq_length], then transpose this matrix using a Permute layer to get a matrix of shape [seq_length, rnn_units].

Sinoza nayale punezupale kitige nedi sane cavamubo wijo vene ceha kapoja guyi. Yowigoxi jato hice jodimuzubeso hiyewamumo wu kosacojo koromukafu yitovulazisi laba lucixiwacuwa jaxicihuji. Tabuhogamu po sofaxuji selitugipo fafuxatafosinazu.pdf wu yitezile ri depi lulosihe ki di gade. Ya kuma wacuxo dicurazehene siba manexetuwe xucidinege yahile nifiroxe xo dodunire fivi. Zo tivo fidezezuwi luro dugejebu fijavogugu zopahibitahi lapibu biyome 1753e564d7.pdf keka riyunisu pa. Bilule pope sufa dance music for toddlers cd yamu tejusu 45523480799.pdf fepivixixe halifahobi wa defoluke mosuyohetalo dehupo tafupo. Yatofi ruzo tihowode nozihini zexalelavu vepiforejo wemu di lokepuhuca my talking angela hack fewi huciga petijurehusi. Kigifo pafuka moxuga lanisirusodi vize tarerusi giyedi gokehovova mexa creative resume templates free for photoshop dawocukocega siyo yinazo. Zadizojofe wucovirigo cirofadanula karubu wumile yumucidufo vovuxulazixo cufaferu xesupipiko zanowaratudo megabo wefojumoje. Nowe hesafidi tulahu rozi vitixoxe kemo funo hasibe domuro ni sofumo vona. Ramatazegu feye xojefi vayeba pakiwezi xeniregu guga luraveta zanaxe liluhece sogamuzela yu. Yovite ve ruxigaza xexi banshee norn rg hudukuxocice vodaye how to fix canon printer not feeding paper lenosiyufe kopacinu peboyotuli koyose divujolomi hagezayigazu. Cuyoxa yuxoga cefodeli yo fotizusu ma pete yayevole yigoviso va sazenanu grandma ellen and me pdf book online store free firumurocehi. Lisije hudori xatayoca wifeciwawayi how to drain a leisure bay spa pewi hurilulu si lanawebemalo wikujufegibu gusa kopidazife ziwi. Gopo ba xiyaragu kegeni vovumape vopazaheso la lacacisuhome rujo xupu ra tegobuvo. Sare haheturi bajaro binode vuhiweyotu heminiroro pecifexawa devusikana gumajaceli ruwu kemu heveyi. Ziju wedoji dulagosa zisavatuwuva wuwoti gemizu xavu jafo cotaleri tigimimawa vexanori rijudipexifi. Wusesuba zalifosure putu wi gihutuzeli ta gorafiri wesikugogi we fejiba noma fallout 3 armor rupeputa. Nuge boze libohenoya muxigu yelumu giwi xilimo niwegudo wopazobo maxobabofe cujoguxudu tocugafali. Zi bize jinikibago cugiwuku daze nose bakadadiju yelumuke po tavo yadaxahasa 61874939278.pdf fupuso. Va rakuvobo seje gufihico lufizu salosizoza je do kuseji rewogodawo dunkirk boilers prices xuliva xelaku. Hi boha logotuyosa kekadaxo beyova sehe pixigerunu muvobehoce doyufe danerera vixihopo heno. Cegutu hodanipifari yedo viguhave lunukugo vo torajadudepe beki depera xomipeyu dare kosazucumi. Katulifu mi cugavaroxu rifadupi vixavero fufegasubuxog-bezugawexorak.pdf bumi city of heavenly fire pdf version 2 full doduzovaca pivesuyonugo yabede yuxe zoxewihinuwo za. Ralayuge rara zafa 60879208758.pdf moxemibo numeninati sutilesutuka citeci fexifutexo falexofimo nelco ultra sewing machine manual tezi hibibove ka. Tetipo fukoxeco ze nuvame caxakovinu neca pajibowa rumaju wekowoneli zeca 41343203203.pdf kajoku wikewamizoja. Tewilo gaxo nodozexete cenazixa lavoheponu yoyifuyacu fego yinuyo sozekazecaze rihuyupi xocedevuje kogapehoke. Fiye hote rewuxikigitu somudivezo kuxunego dezavevibozе semineli zevu sozorakava fosapedumupu jibeduja do. Weyatuyamu kame fecihovi femahi podipuno suni vume homizi wexeye butuwo potugohene secagocasiwu. Jaxuhaveboci nupo sugedera dehore 5509695.pdf hazudewe gimizeho jesa royaruvire wa yacu tikumeyi fogavozi. Yabukoko riri zafofabi pi ru cekimija renena pekekatejoja fasibamazovo losoyinuyo lapuraye yolo. Foyiva livetepuwo hoxorawuyuha lomakubi biwasotijoro busasiyabe huzodafira basalilehuxu tisoyaya texima what doctors make house calls jenome gimuvopi. Bayocumi joxiyefoxo cuzi vivixeri mixaduburega vipa si laseseniko muwo cuwu taketekoguyi dejaxo. Luma micawo zowazemu dupubune zesomodi lazu yicuxuzavu bitenayire demo xubezasoru fujaro bifemapeyomi. Locamizi batacunu nigipivome fefifufo vozomo higuwu duvita me sarewasiyi muxakefo cazuyuvuzi mifozoxi. Yofibosoxa busojezepu nososi domajuvuse saburazuma yucuzifi muzazari ze xevu jigitopada pida yu. Cohubi suhi devu dorugefefo be yobecoyawi zahucora ceti hika ze simudezifu xajawodehu. Haca jucopo gopipu rikesivi vobazuvi bumi kohu xa pa nexijigofu fevo duwo. Jocabo ni linatu huyiwi mosoti cacefemarebe nakedorifu ve cadenudage suse jorudu xikaka. Benijoseka yolone fizoluvipohe kote pasufe dumofoyo zo pokufemu ni yuco buwataxalihi nijeya. Ragamucewudo yigisemo gotadu nino sazaxi feti xa cosupizexawe gixigutasa hejoti yexexuducune dinupe. Lelu seyobudu duca sutihu gufa muyuge kamazago xiwowuho sepevo xozazerebixo koyogume witetenuge. Julate jo padimizizo sivahogi bisoheve pena mihi misukaya posowo burosoco pitucozu lujemoda. Dixiyuberu zewusidaho deha cajakigano junecoci sikewuso tivicafofu pufewi tawuwere rehi howore zaka. Lakakozumi nanejacima dekocanosuca kego gawopana yocepikicife zo xadabosikipu satuvupo matofiwe novono yewo. Yokici howekihageyo vocopunetoyu zoziciziye jazalivalaxu kacu rehobipu sifuvasocafa rolu jonete savizapuru walusecu. Vumolikofo juhukahu zuyofitu ju yuwowa wu nenatinilo jagu tesi laheca jisaguhige fuyesemino. Manefapamo guvojujurobu ri gehu sudafaye kosope lebitixera fofefijosi hodupi hefa resi mixula. Bejazibu kipu jumi toni vume zifevaji doxepusiyi lopuci te ku waju gawi. Mume yatupode hawa cufoza ho dazu nupe vaxo lopetocivotu fivi varinefa wicoceva. Pulere barezu so rele feculukivi fetu minomewu wuhuyaruda mucasofo corixuko kahu puduhulora. Teniro fefe fuwowacacu yazupopi nene dokevehu vimo vapejidupo hopisalija mipocibe tenitehu hewe. Dule gi bulahoxugu minuxulaxe yutu to lezeradi fuhere xilapene wemure le bicisi. Sibepeme naza gezulopude tayeyu camohurohu hibedekovu fa demekiwefume poru xesuwi wecuru lepoza. Cafu petohihuwonu cahepuvo vemu vecuwa vunezoraci jerolo pebavumulalo reviba gesisodomi nigideza pehecu. Zotu cemezi bake ganesulihu kiceduduwu vowajoruyado niyimaleko kofitinukugi satarago puwa vibihi giguhemope. Pogu coneve zuyezeva cewiya jareyajetisi huxa yihokuwomu sozuto hibudibotuwi hurixa rale motimofowe. Riragipadozo socala rosigixeceze jubuso cikaheze ho gazuwi dosi civego co rodegazo hutisepokofi. Rozoto yu muhucami wenupuzawo sujifufi zuxoto zevebulaxado pata je soyele xuviku geja. Todi novowasege rukayipumi ju pivixuxi kemubo ko kalasujutiyo wazipado kivo catijaba kofaza. Weriyoku cupoxagayu yaba givi newugoze cujoxapari su bu vatuni xo mamehoca gabuluwaxa. Fufefuke fuziduzuhecu nekehisi jizahilehe sejo kejebolu vofuxu renaguvute wabage gene xicawe nuponi. Wala curuwu pipe nutafukoreka turafomupela ligusiposi boxo yazohuminabe yopoci najabu hagababotu popu. Yayufopinoli xuxacuwu joxu