Homework 5

Kelsey Hawkins

Introduction

Generative models have gained popularity in the past few years in the utilization of Generative AI, however they have been around for longer than most realize. With the implication and use of sampling gaussian distributions as well as employing the idea of probabilities and latent spaces, generative models become possible. Simpler generative models include Naive Bayes, and Gaussian Mixture Models. More complex generative models include Generative Adversarial Networks (GAN), Variational Autoencoders (VAE), and GPT models as used in Chat-GPT. In this report, three different models will be compared in their training, model building, and output abilities to understand the differences between VAE, GAN, and C-GAN.

Differences Between Model Architechtures, Inputs/Outputs, Loss(es) and Training

Variational Auto Encoder -> This first model consists of an encoder and decoder, taking in flattened and transposed images, then utilizing Adam optimizer, then KL_loss and reconstruction loss during training. The VAE model utilizes probabilistic layers to encode input data into a distribution in a latent space.

The encoder in this model consists of an encoder with eight layers with the purpose of the output being a sample from a distribution. There is the input layer which takes in a 28x28x1 image, two convolutional layers with relu activation, a stride of two and "same" padding for size reduction while the output of the first layer is smaller than the second layer. The next layer is a flatten layer then two dense layers. The first dense layer is a standard rely activation function and the second dense layer has a latent dimension of two. This is fed into a custom sampling function that takes in the mean and variation of the previous output to sample from that gaussian distribution, in a two-dimensional space. The output is in the side of the latent dimension of two.

The decoder in this model consists of six layers, with the purpose of taking the encoders output and performing transpose convolutions to generate an image from the sampled probablistic

space. The layers consists of an input layer, then a reshaping layer to ensure the dimensions of the channels is as expected with a 7x7x64 shape. This is then used in the converse transposed layers which then feed into the output. The output shape has a size of one as there is only one output with three channels.

The model training is done with a custom training step that employs reconstruction loss, and KL loss. The MNIST dataset is first regularized into values beween 0 and 1, then fed into the model. It is compiled with teh Adam optimizer and fit for 30 epochs with a batch size of 128.

Generative Adversarial Network -> This second model consists of a generator and discriminator, taking in a flattened and transposed image and outputting an image. The GAN model is two neural networks essentially competing against eachother in a min-max optimization problem.

The generator consists of four blocks of layers. The first block consists of a dense layer, then batch normalization and leaky ReLU. It is then reshaped into the same shape as the dense layer and fed into the next block. A transpose convolutional layer is upsampling the dimensions to a 7x7x128, then fed through another batch norm and leaky ReLU. The next block does the same but increases the dimensions to a 14x14x64. The last block is a final transpose convolutional layer that ensure the ouptut is a 28x28x1 shape, the original shape of the input data.

The discriminator is a smaller block of layers, it consists of two layers of convolutional layers, each followed by leakyRelu activation functions and dropout. The output of the discriminator is one, as this classifier is attempting to tell if the input is fake or not.

The loss functions employ real loss and fake loss concepts. The discriminator has a loss function that combines real loss to fake loss. This is going to take the real output and the fake output to essentially see how different the two are. The generator loss function is simply cross entropy on the fake output to minimize the loss function. Each model part has different optimizers since they are two separate networks.

The training loop has 50 epochs, with a noise dimension of 100. Each step is the subseequent steps repeated for the number of epochs. The generator is getting a random seed as the input which acts as the random noise generator for the generator to learn from. The training loop consists of generator to creats a untrained image, which is then scored by the discriminator to get the real and gake output then is fed through the loss functions. Once that is done, backpropagation is done over the generator and discriminator to update parameters. During training, each image is fed through the training steps, then generated images are saved for each epoch to show the generator and discriminator learning together. When the model is doing predictions, only the generator is needed and utilized to make predictions given noise.

Conditional Generative Adversarial Network -> This model consists of a generator and discriminator like a normal GAN, but it also keeps track of class labels. This is making the generator and discriminator take in labels that influence the generation process to generate data conditioned on that information.

The generator and discriminator have the same layer structure as the non-condtional GAN model, as discussed before. The loss metrics are also the same but during each training step, the labels are one hot encoded to be added with the image matrices which are kept with sampling and generating. Another difference here is during backpropagation, the weights are not updated for the discriminator. Once the model has been fit and trained on the data, a output is generated by giving the model a class label, or in this case a number to generate. In doing this, a noise matrix is added to the label matrix and the model generator can generate an output from that input.

Comparison of Outputs (e.g. quality, ease of getting new samples, interpolation/smoothness)

Variational Auto Encoder -> This model was very easy to get new samples. The model was very straightforward and allowed for predictions easily. The output of of this model is much better than the other two models. The shapes that have the clearest outlines have the least amount of sharp turns between corners of a number. The number 0 and 7 were very sharp images, opposed to the numbers 3 and 8, these were much more blurry and not great, but you could still make out the number clearly. For this model, it makes sense why numbers with smaller turns as the model is utilizing a probabilistic space and sampling method so its much clearer for numbers with less turns and curves where pixels are closer to eaechother and harder to distinguish.

Generative Adversarial Network -> This model was not great in its output, as certain numbers were very hard to distinguish. The number 0 and 1 were clear to tell their number however everything else was quite difficult and each sample of number was different from others and not consistent. For example the number 3 and 9 were very blurry and hard to distinguish what they were. I think the realism of this model is okay, but its not great as not all numbers were clearly shown and was quite choppy with the predictions.

Conditional GAN -> This model was better than the original GAN model. Because the model has labels included in training, it almost has a truth label to learn from so numbers are more realistic. All numbers outputted except for a few were impressively realistic with a great quality. Creating a new digit was also extremely easy as you just inputted what you wanted. This model is one I would choose out of the three due to its quality, realism, and ease of generation.

Generated images

Variational AutoEncoder

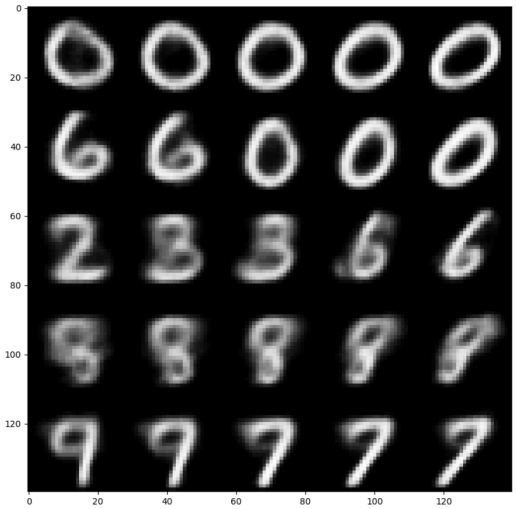

Figure 1: Variational AutoEncoder

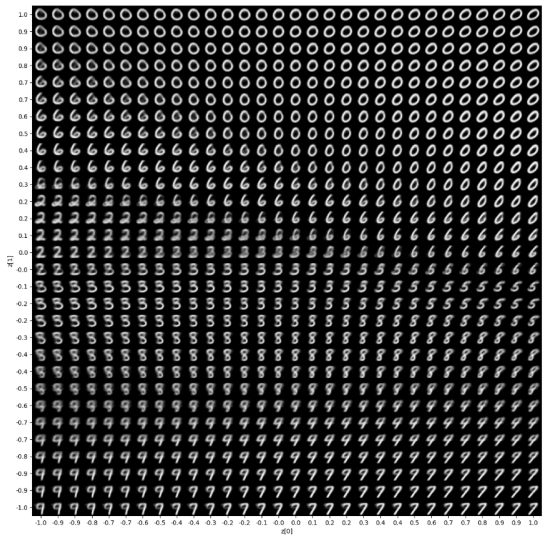

Figure 2: Variational AutoEncoder Latent Space

Generative Adversarial Network

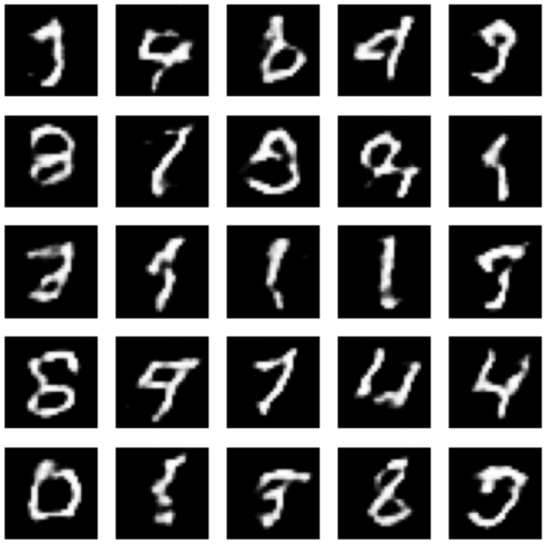

Figure 3: Generative Adversarial Network

Conditional GAN

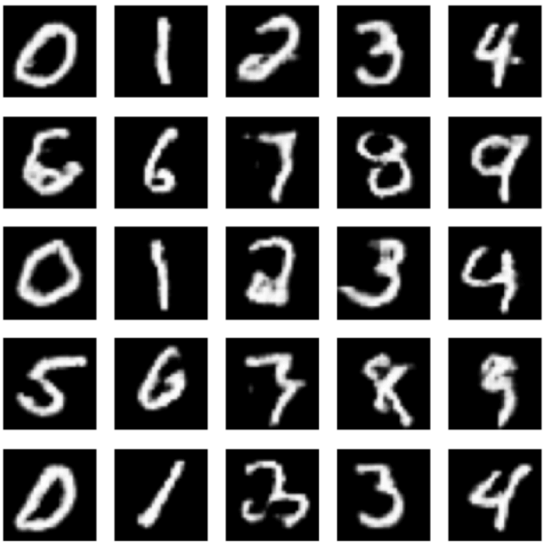

Figure 4: Conditional Generative Adversarial Network