HW5 Technical Report

Bryce Grover

Introduction

Generative models are a type of deep learning model that learn to make new things like pictures or music that look or appear real. Generative models accomplish this by examining many different real examples first. For instance, in this report, a data set consisting of images of handwritten numbers is used to train generative models that makes new images of handwritten numbers would need to learn from the many pre-existing pictures of real handwritten digits. Once a model understands how these real images are compaosed, it can create new images that that did not exist before and no one has seen. These model are useful for making new artworks, improving virtual effects in media, augmenting video game graphics, and generating new written text (such as ChatGPT does).

In this report, the performance of three different types of generative models are compared. The models are trained on images of handwritten numbers. After the models are trained, they are used to generate 25 new images of the numbers. These images are evaluated on their realism and how consistently well generated they are.

Differences Between Models

Variational Autoencoders (VAEs)

VAEs are based on another type of model, autoencoders. Autoencoders consist of an encoder and a decoder and as such, so do VAEs. The encoder compresses input data into a latent representation which it typically done to reduce dimensionality. In VAEs, the encoder outputs a Gaussian probability distribution for each variable. The idea behind outputting a probability distribution in instead of a static value is that if the model receives two inputs that differ only slightly, the model's two ouputs should not be very different from each other.

VAEs are evaluated based on a reconstruction loss function which measurs how well the output matches the input and a KL divergence loss function which measures how much the learned distributions diverge from the original distribution.

VAEs are good for problems where modeling the distribution of the data is important. They are used in situations where smooth interpolation between points is needed or preferred, such as in generating new images and text.

Generative Adversarial Networks (GANs)

GANs consist of two neural network models: a generator and a discriminator. These two neural networks can be thought of as 'competing' with one another. The generator model creates a distribution of data samples from an input of random noise, aiming to generate new data points that are similar to and generally undifferentiable from real data. The discriminator evaluates the generators samples to determine whether they are real, meaning the discrimninator thinks the data is from the training dataset or fake, meaning the discriminator thinks its been made by the generator.

The generator and discriminator go back in forth in a cycle where the generator tries to trick the discriminator by creating better and better outputs that are increasingly realistic. The discriminator tries to become better and better at distinguishing from the real data from fake data.

Training the models involves making updates to both networks until a balance is reached where the generator produces data that is realistic enough that the discriminator reaches a 50% accuracy level, meaning the discriminator can no longer reliably discern between real and generated data.

Conditional Generative Adversarial Networks (CGANs)

CGANs are a modified version of GANs that include conditional information (hence the name) into the neural networks. This conditional information could be class labels, additional data, or any other supplementary information that assists the competing models. Since both the generator and discriminator in a CGAN receive the conditional information, the CGAN follows specific conditions or attributes. In the case of the CGAN used in this report, the conditional information is the class label (what number) the input image is. The objective of a CGAN is similar to that of a vanilla GAN, but the addition of a conditional element guides the process to produce outputs specific to ta given situation.

CGANs are used where the generation needs to be controlled or specific, such as generating images of objects with particular attributes, creating text under specific conditions, or adapting styles in a specific way. They have also been effective in augmentation tasks where labels are crucial, like in supervised machine learning tasks.

Comparison of Outputs

VAE

The output of a VAE shows a surprising level of clarity and smoothness in the representation of the digits/characters. Typically, VAE outputs are expected to be blurrier and less sharp than those of GANs or CGANs due to the VAE's reconstruction-based approach. However, in this case, the VAE outputs are mosty continuous and well-defined. It is evident that the model leaned a good representation of the images and can reproduce them well.

GAN

From the output of a GAN, there is slightly more variation and diversity in the shapes and patterns compared to the CGAN outputs. This aligns with the expectation that GANs can generate more diverse samples due to their adversarial training process. However, the level of clarity and realism are poor. There are some generations that look like numbers, however, suggesting that the model learned meaningful representations

CGAN

The images generated by a CGAN are pixelated and low-resolutionx. The depict only a very loose shape of a number. Typically, CGANs are known for producing high-quality, realistic outputs by leveraging conditional information during training. However, in this case, the outputs lack a level of detail and sharpness. Nonetheless, the shapes and structures of the digits/characters remain reasonably well-defined, indicating that the CGAN has learned meaningful representations.

Generated Images

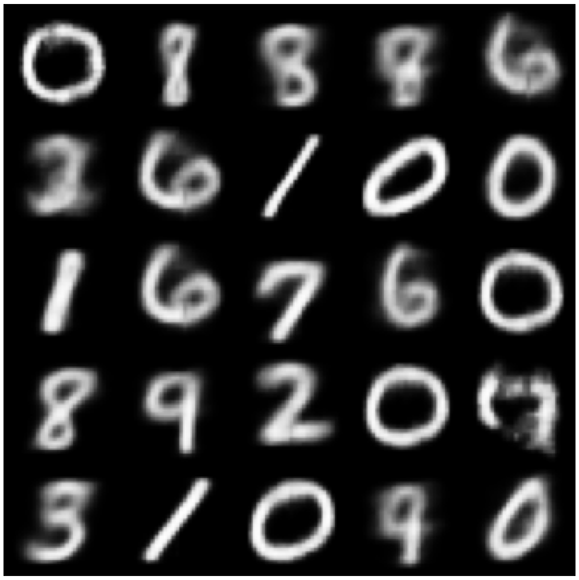

Figure 1: Output of the VAE model.

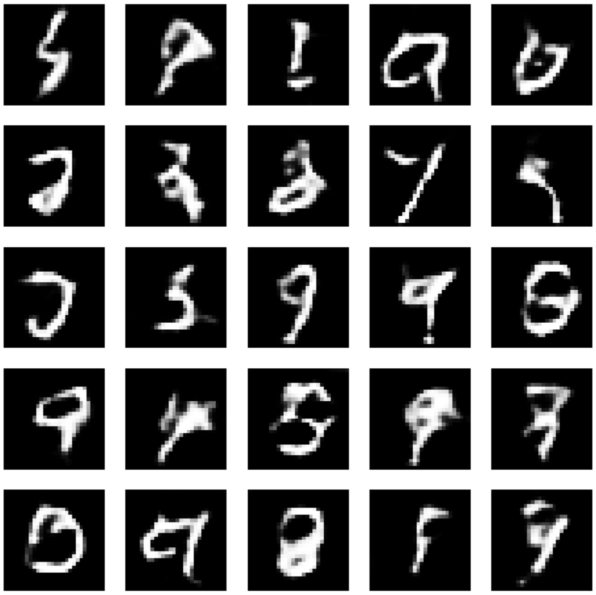

Figure 2: Output of the GAN model.

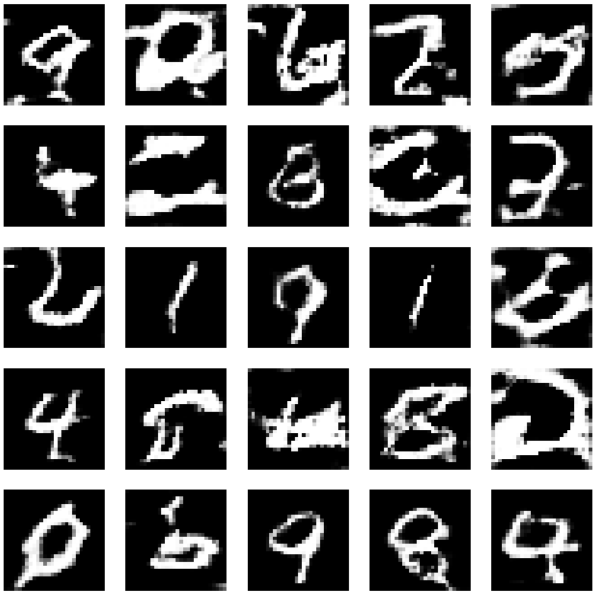

Figure 3: Output of the CGAN model.