CS 615 - Deep Learning - Assignment 6

John George

August 23, 2021

1 Part 1: Theory

1.1 A

Let $\hat{Y}$ be the output of the Discriminator's Sigmoid Activation Layer
Let J be the Generator's negative log objective function

$$J = -ln(\hat{Y}) \qquad \frac{\partial J}{\partial \hat{Y}} = -\frac{1}{\hat{Y}}$$

1.2 B

Let F_{out} be the output of the Discriminator's fully connected layer
Let X be the fake input
Let W_D be the weights of the Discriminator's fully-connected layer
Let B_D be the bias array of the Discriminator's fully-connected layer
Let $\hat{Y}$ be the output of the Discriminator's Sigmoid Activation Layer
Let J be the Generator's negative log objective function

$$F_{out} = X \bullet W_D + B_D \qquad \hat{Y} = \frac{1}{1+e^{-F_{out}}} \qquad J = -ln(\hat{Y})$$

$$\frac{\partial J}{\partial X} = \frac{\partial J}{\partial \hat{Y}} \bullet \frac{\partial \hat{Y}}{\partial F_{out}} \bullet \frac{\partial F_{out}}{\partial X}$$

$$\frac{\partial J}{\partial X} = -\frac{1}{\hat{Y}} \bullet (F_{out} \circ (1 - F_{out})) \bullet W_D^T$$

1.3 C

Let X be the fake input
Let K be the input for the Generator's activation layer

$$X = \begin{cases} K_i & K_i > 0 \\ 0 & K_i \leq 0 \end{cases} \qquad \frac{\partial X}{\partial K} = \begin{cases} 1 & K_i > 0 \\ 0 & K_i \leq 0 \end{cases}$$

1.4 D

Let W_G be the weights of the Generator's fully-connected layer
Let B_G be the bias array of the Generator's fully-connected layer
Let Z be the noise input for the Generator
Let K be the input for the Generator's activation layer

$$K = Z \bullet W_G + B_G \qquad \frac{\partial K}{\partial W_G} = Z^T$$

1.5 E

$$\frac{\partial J}{\partial X} = \frac{\partial J}{\partial \hat{Y}} \bullet \frac{\partial \hat{Y}}{\partial F_{out}} \bullet \frac{\partial F_{out}}{\partial X} \bullet \frac{\partial X}{\partial K} \bullet \frac{\partial K}{\partial W}$$

$$\frac{\partial J}{\partial X} = -\frac{1}{\hat{Y}} \bullet (F_{out} \circ (1 - F_{out})) \bullet W_D^T \bullet \left(\begin{cases} 1 & K_i > 0 \\ 0 & K_i \leq 0 \end{cases} \right) \bullet Z^T$$

2 Part 2: Generative Adversarial Network

I chose to use the EMNIST data set for the training, but submitted the assignment so the provided NIST data set will be run.

The EMNIST data set contains 240,000 labeled digit observations. https://www.nist.gov/itl/products-and-services/emnist-dataset

2.1 Architecture

The architecture for the GAN is much like we discussed in class:

The Generator has a Noise-Generating layer, a Fully-Connected layer, a ReLu Activation layer, and a Negative Log Objective layer.

The Discriminator has an Input layer, a Fully-Connected layer, a Sigmoid Activation layer, and a Log Loss Objective layer.

A Stochastic Gradient Descent strategy was used, with each epoch randomly selecting 200 observations from the training set to train with.

An attempt was made with a Multi-Layer Perceptron with both the Generator and the Discriminator, but the MLP GAN seemed to struggle to train the Generator to create anything that resembled a digit. When both ANNs were reduced to a single Fully-Connected layer, the generator was successful in learning to create a digit-like image.

2.2 Hyperparameters

The Fully-Connected layers in both the Generator and the Discriminator were given learning rates of 0.001.

I found that a regular Fully-Connected layer with a constant learning rate trained the GAN better than a Fully-Connected layer with ADAM learning rate. The Generator didn't seem to converge to creating a good fake image before possibly falling into a local minimum during its gradient descent.

Each digit animation was trained with 2000 epochs. Each animation consists of 21 images, the first image represents the generated image at epoch 0, and the last image represents the generated image at epoch 2000. The interim images were generated every 100 epochs.

2.3 Observations

For each digit, the first 500 epochs appear to generate seemingly random noise.

Epochs 500-1200 generate images that acceptably represent the digits they were trained for. They're not perfect, but they're recognizable.

The generated images deteriorate in quality after epoch 1200. More investigation is needed to fully understand why this deterioration is occurring.

Speculating on the deterioration after epoch 1200, there could be several causes: The Generator's model is overfitting; The generator is finding tricks, such as certain pixels being bright, that defeat the Discriminator without creating a quality image; The Discriminator may not be keeping up with the Generator's deception.

2.4 Future Improvements

Explore more with adaptive learning to see if certain hyperparameters will allow the models to train better.

Implement a Convolutional Neural Network (CNN) in the Discriminator, and possibly the Generator too. CNN's are useful for feature extraction in images. Creating some sort of reverse CNN for the Generator will require some additional research.