Catherine Horng, ch756

Final Project Proposal

I would like to train a GAN to draw doodles trained on doodles from Google's Quick, Draw! Dataset [1].

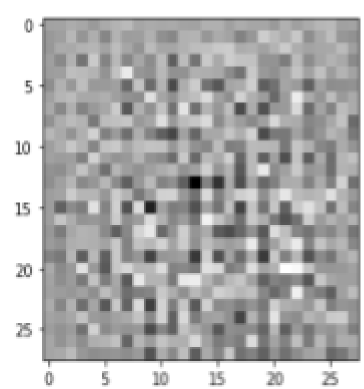
The system I would like to explore is generative adversarial networks (GANs) [2]. GANs consist of a generator and discriminator, two neural networks that compete. The generator (typically a deconvolutional neural network) produces fake data and the discriminator (typically a convolutional neural network) distinguishes between the real and fake data. Other GANs I am considering besides the vanilla GAN are conditional GANs (cGANs) [3] and deep convolutional GANs (dcGANs) [4].

The task will be to draw doodles from the Quick, Draw! Dataset. Google's Quick, Draw! Dataset can be used as a substitute for the MNIST dataset as visually they are similar, but the Quick, Draw! Dataset has more categories of images such as smiley_face, airplane, and apple. The dataset was contributed by players of the game Quick, Draw! [5] in which players, given a category to draw, draw an image in the category in 20 seconds as the game attempts to guess what is being drawn. The dataset is made available by Google and consists of 50 million open-sourced doodles for 345 categories.

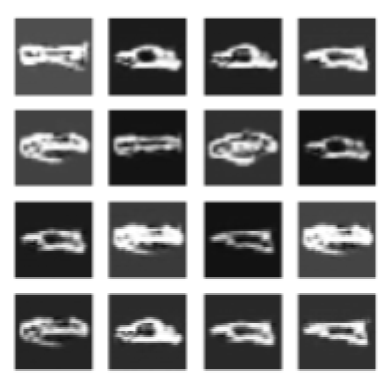
Techniques I will explore are Adam, momentum, and random hyperparameter optimization. With the GAN model, I will vary the optimizers used to train both the generator and the discriminator, using either SGD, SGD with momentum, or Adam. Additionally, I will vary the hyperparameters of these optimizers using random hyperparameter search. I could also vary other hyperparameters such as the network size and batch size, though I would like to focus on optimizers and their hyperparameters. I will use SGD without momentum as a baseline.

The evaluation of drawn GAN images can be subjective, but has been discussed in depth in Pros and Cons of GAN Evaluation Measures [6]. Popular evaluation techniques are Inception Score (IS) and Fréchet Inception Distance (FID). Additionally, I could train a classifier on half of the dataset while the other half is used for training the GAN model and use the classifier as a metric to gauge how well the GAN model performs by producing images and classifying them. If the classifier is known to have good test accuracy, then correctly classifying fake images could be a possible statistical measure of performance. I will also evaluate the wall-clock time of training the GANs as exploring momentum and Adam will likely affect wall-clock time.

As for preliminary work, here are a few images a minimally trained vanilla GAN can produce so far. The model appears to be starting to train on the dataset as it appears there is an image beginning to take shape, far from the original image the untrained model produced which is essentially just noise.



Doodle generated on untrained model.



Doodle generated from minimally trained vanilla GAN model.

[1] Quick, Draw! Dataset https://github.com/googlecreativelab/quickdraw-dataset

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets https://arxiv.org/pdf/1406.2661

[3] Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets https://arxiv.org/pdf/1411.1784

[4] Radford, A., Metz, L. and Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks https://arxiv.org/pdf/1511.06434

[5] Quick, Draw! https://quickdraw.withgoogle.com/

[6] Ali Borji, A. (2018). Pros and Cons of GAN Evaluation Measures https://arxiv.org/pdf/1802.03446

Experiment Plan

Hypothesis: I hypothesize that SGD with momentum and Adam will perform better than SGD. In training a GAN model, essentially we need to train two neural networks at once, therefore training with these optimizers which have been known to improve performance with other neural networks will improve performance of the GAN model.

Proxy: The metric to use to measure this will be most notably wall-clock time. A “better” performance would be lower clock time in training. I would also look at IS, FID, and classifier accuracy as with the same number of epochs, I would expect these values to increase as well though I’m mainly concerned with wall-clock time.

Protocol: If I get the results I expect, I can then explore the hyperparameters of momentum and Adam using random hyperparameter search.

Hypothesis: I hypothesize that random hyperparameter search will perform better than SGD. Similarly to the first hypothesis, random hyperparameter search has been known to be successful on neural networks. I will likely only use random hyperparameter search on the optimizer hyperparameters.

Proxy: The metric to use to measure this will be IS, FID, and classifier accuracy. I will also look at wall-clock time as I expect some hyperparameters for SGD with momentum and Adam to speed up training though I would like to focus on statistical performance with random hyperparameter search.

Protocol: If I get the results I expect, I can use the best hyperparameters to build a final model that can accurately doodle new images.

[1] Quick, Draw! Dataset https://github.com/googlecreativelab/quickdraw-dataset
[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets https://arxiv.org/pdf/1406.2661
[3] Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets https://arxiv.org/pdf/1411.1784
[4] Radford, A., Metz, L. and Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks https://arxiv.org/pdf/1511.06434
[5] Quick, Draw! https://quickdraw.withgoogle.com/
[6] Ali Borji, A. (2018). Pros and Cons of GAN Evaluation Measures https://arxiv.org/pdf/1802.03446