DA24B009_SG3_Submission1

Kirthan

1 Introduction

Conditional GANs pick up where the original paper GANs left off and can be considered an evolution of the same. Under the original GAN framework, images are generated from a sample of random noise z from the latent space. As a result, there is no way to direct the image generation process.
Conditional GANs hope to fix this by essentially incorporating/embedding the labels into the input the Generator receives, thus ensuring that the class of the image is factored into the learning process.

2 Architecture

The specific architecture of the Generator and Discriminator is irrelevant to the scope of this paper as the innovation is made in the input. The authors encode class information as either one-hot encoders or label encoders. This vector is then 'appended' to the noise sample from the latent space and an image from the training dataset. This is then fed into both the Generator and Discriminator.

3 Training

The objective function used is the same as that used for regular GANs, i.e.,

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z}|\mathbf{y})))]$$

where $\mathbf{x}$ belongs to our training data, $\mathbf{y}$ is our class label, and $\mathbf{z}$ is noise.

The same training algorithm is then used where the Discriminator is trained by Gradient Descent for 'k' iterations before one training step for the Generator.

4 Class Embedding

The paper uses a simple one-hot encoding for the MNIST set and a conceptual word embedding system for the Flicks Datset.

5 Conclusion

The paper makes it very clear that their work is merely a proof-of-concept and that their implementation is by no means optimal. They suggest experimenting with different Generator and Discriminator architectures and float the idea of setting up a way to train the word embeddings as well.