HW4: Variational Autoencoders

Alex Lin
alexanderlin01@college.harvard.edu

Melissa Yu
melissayu@college.harvard.edu

March 31, 2018

1   Introduction

2   Problem Description

3   Model and Algorithms

We explored the performance of a Variational Autoencoder and Generative Adversarial Network (and modifications of these models) on the MNIST digits dataset.

3.0.1   Generative Adversarial Networks

The generative adversarial network (GAN) (Goodfellow et al. (2014)) framework establishes a min-max game between two neural networks, the generator G and the discriminator D, in order to match the generative process to the data distribution. Suppose the generator network $G(z)$ maps some latent representation $z \sim p(z)$ to the data space, and the discriminator network $D(x)$ predicts the probability that a sample x comes from the true data distribution p_d (positive samples), and not G's generative process (negative samples). The objective is

$$\min_G \max_D \ \mathbb{E}_{p_d(x)}[\log D(x)] + \mathbb{E}_{p(z)}[\log(1 - D(G(z)))] \tag{1}$$

The network consists of the discriminator, which achieves the best classification accuracy over samples of x, and the generator, which maximally confuses this best discriminator, so that $D(x) = 1/2$. The network is trained in two stages, by first updating the discriminator using both real and generated samples, and then updating the generator using noise samples.

3.1   Variational Autoencoder

Kingma and Welling (2013)

4   Experiments

4.1   Generative Adversarial Network

We normalize all images to have values between -1 and 1, and de-normalize the generated images for visualization purposes. Our final GAN model consists of a generator and discriminator. The

discriminator comprises 2 linear layers with LeakyReLU activations, followed by a linear output layer with Sigmoid activation. The generator comprises 2 linear layers with LeakyReLU activations, followed by a linear output layer with Tanh activation. The hidden dimension size for both D and G is 256. Because the stability of the GAN game suffers when we have sparse gradients, we achieve better results using the LeakyReLU activation (with negative slope 0.2) instead of the ReLU activation.

We train G to maximize $\log(D(G(z))$ instead of minimizing $\log(1 - D(G(z)))$ (Goodfellow et al. (2014)). Note that the new form of the loss is functionally equivalent, but enables stronger gradients early in training when the discriminator can be very confident about samples from the generator. During training for GAN's, the discriminator can be trained for k steps before updating the generator once; if the generator is trained "too much" without updating the discriminator, it may simply learn to map many values of x to the same z and fail to be expressive enough to capture the latent data distribution. We find that setting $k = 1$ is sufficient for our purposes. For both the generator and discriminator, we use the Adam optimizer with learning rate 0.0005 and train the model for 300 epochs.

The input to the generator is a noise vector, which is sampled from a spherical normal prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Figure 1: Visualization of 100 generated samples from random noise vectors sampled from a spherical normal prior with dimension 64.

Figure 2: Visualization of samples generated from interpolations of 2 random noise vectors sampled from a spherical normal prior with dimension 64.

Because the MNIST dataset also provided labels for the data, we can additionally consider improving the model's generator by training the discriminator to also classify the samples. To do

this, we would implement an auxillary GAN, which shares the first 2 layers of the generator's neural network, but whose output layer consists of a softmax activation.

4.2 Variational Autoencoder

5 Conclusion

In this practical, we implemented two models to generate written digit images: A Generative Adverasarial Network and a Variational Autoencoder.

References

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.