Generating Images via Generative Models

MATH6380O Final Project **Team: CAI Haoye, CHEN yuan, BAI Chunyan, GUO Wenshuo**

Project Overview

Generating images from a given dataset remains a challenge and many generative models like Generative Adversarial Networks (GANs) and Variational Autoencoder (VAE) have been developed to tackle this problem.

In this project, we will use several GANs, which includes GAN, GAN with log D trick, WGAN, WGAN_GP, CGAN, and VAE to generate images based on the fashion-mnist dataset and give detailed comparison (both qualitatively and quantitatively) among these generative models.

Dataset

In this project, Fashion-MNIST dataset is used. The Fashion-MNIST is a dataset of Zalando's article images, which consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.

Methodology & Models

In this project, six models for GANs are used, including the original GAN architecture, the improved GAN with log D trick, CGAN, WGAN, WGAN-GP and VAE. In the diagrams below, we show the details of each model.

GAN (Generative Adversarial Network)

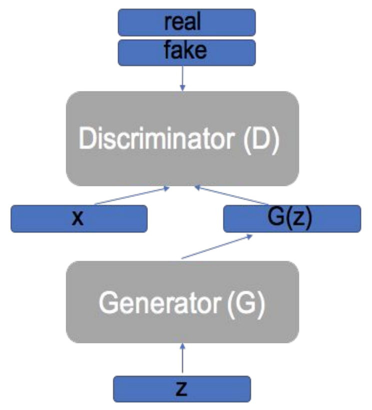
- Deep neural net architectures with two nets (the generator and the discriminator)
- Has shown great success in the realistic image generation
- Suffer from training problems: instability and mode collapse

Improved GAN (with log D trick)

- Original GAN: training the generator easily results in vanishing gradient problem
- Improved GAN: a non-saturating heuristic objective (log D trick) replaces the minimax objective function to penalize the generator

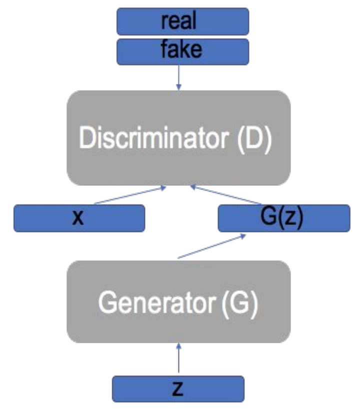
CGAN (Conditional Generative Adversarial Nets)

- Conditional variation of GAN
- Constructed by simply feeding the data, y, we wish to condition on to both the generator and discriminator
- Introduced by Mehdi Mirza, Simon Osindero



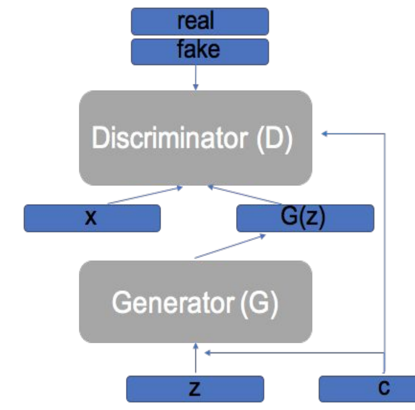
$$Loss(D) = E(\log(D(x))) + E(\log(1 - D(G(z))))$$
$$Loss(G) = E(\log(1 - D(G(z)))$$

Figure 1. Network Structure of GAN and Value Functions



$$Loss(D) = E(\log(D(x))) + E(\log(1 - D(G(z))))$$
$$Loss(G) = E(\log(D(G(z)))$$

Figure 2. Network Structure of improved GAN and Value Functions



$$Loss(D) = E(\log(D(x))) + E(\log(1 - D(G(z),c)))$$
$$Loss(G) = E(\log(D(G(z),c))$$

Figure 3. Network Structure of CGAN and Value Functions

WGAN (Wasserstein GAN)

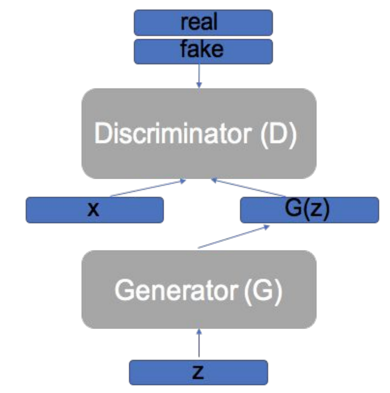
- After every gradient update on the critic function, clamp the weights to a small fixed range, [−c,c].
- New loss function derived from the Wasserstein distance
- Recommended RMSProp optimizer on the critic, rather than a momentum based optimizer

WGAN-GP (Gradient Penalty)

- WGAN performs poorly sometimes due to the use of weight clipping to enforce a Lipschitz constraint on the critic
- WGAN-GP: An alternative to clipping weights: penalize the norm of gradient of the critic with respect to its input
- Better performance than standard WGAN

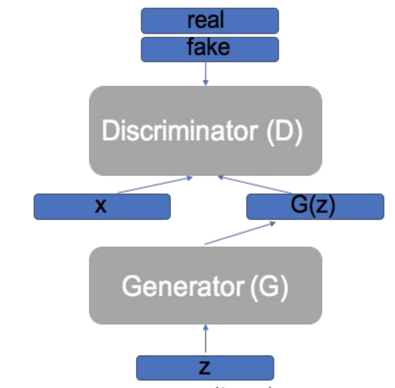
VAE (Variational Autoencoders)

- Autoencoder with added constraints on the encoded representations being learned
- Rooted in bayesian inference: model the underlying probability distribution of data so that it could sample new data from that distribution



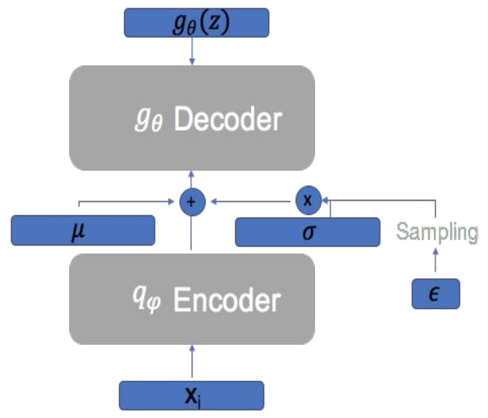
$$Loss(D) = E(D(x)) - E(D(G(z)))$$
$$Loss(G) = E(D(G(z)))$$
$$W_D = clip\_by\_value(W_D, -0.01, 0.01)$$

Figure 4. Network Structure of WGAN and Value Functions



$$Loss(D) = Loss(D)_{WGAN} + \lambda E[(\left|\nabla D\left(\alpha x - (1 - G(z))\right)\right| - 1)^2]$$
$$Loss(G) = E(D(G(z)))$$
$$W_D = clip\_by\_value(W_D, -0.01, 0.01)$$

Figure 5. Network Structure of WGAN-GP and Value Functions



$$Loss(VAE) = -E_{q_\varphi(z|x_i)}[\log(p(x_i|g_\theta(z)))] + KL(q_\varphi(z|x_i)||p(z))$$

Figure 6. Network Structure of VAE and Value Functions

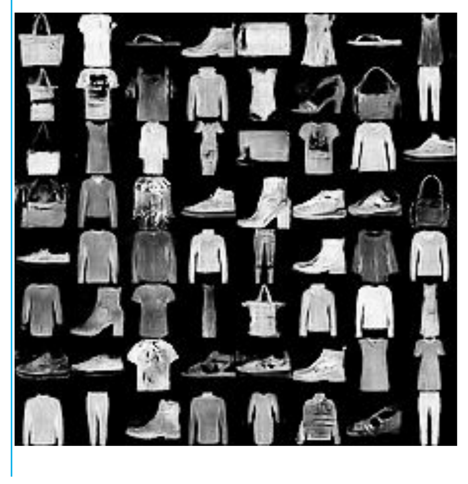
Results

For each model, we ran for 40 epoches and then we generate testing samples from our trained latent space. Here we show 64 samples for each method.

GAN

Improved GAN

CGAN



WGAN

WGAN-GP

VAE

Evaluations

We evaluate our models using the standard Inception Score (IS). Here we also include real images from our dataset (Real) for comparisons.

Table 1. Inception Scores of different generative models.

	Real	GAN	imp-GAN	CGAN	WGAN	WGAN-GP	VAE
IS	4.04±0.84	2.42±0.41	2.45±0.29	2.72±0.43	2.37±0.29	2.47±0.35	2.13±0.26

Observations & Analysis

Qualitative Results

From the image samples shown above, we can see that all our generative models are able to generate plausible fashion item images (of different classes) that are learned from the training set distribution. This indicates that our training is successful overall.

We make our first comparison between GANs and VAE. We notice that the image samples generated by VAE are obviously more blurry than GANs. This is one key shortcoming of VAE inherent in its mechanism to minimize evidence lower bound and the usage of L2 error. Hence, VAE tends to generate blurry results, which coincide with our observation.

Next we compare the results of different GANs. Though they all generate sharper results, WGAN and WGAN-GP seem to produce some bad samples that is not even complete. This may indicate the gap between their theoretical superiority and practical applications. Both original GAN and improved GAN generate fairly good results (improved GAN slightly better), while conditional GAN generate the most realistic and the sharpest images. The good results of CGAN may be attributed to its conditional setting, which utilizes additional class information.

Quantitative Results

From the quantitative comparison shown in the table above, we can see a clearer comparison between different models. We see that our real images yield the highest score, which coincide with our intuition. Then VAE produces the lowest score, consistent with our previous blurry qualitative results. WGAN-GP gains higher score than WGAN, indicating the improvement of gradient penalty over weight clipping. They are all worse than other GANs, showing the same issue as above. GAN and improved GAN are slightly better while improved GAN outperforms the original. This shows the effectiveness of the log-d trick. CGAN has the highest score, which coincides with its best visual quality.

In conclusion, GANs generate results of better visual quality than VAE, and among different GANs the CGAN seems to perform the best.

Acknowledgment

- Work distribution (equal distribution for all members):

CAI Haoye: training models, poster writing
BAI Chunyan: inception score implementation, evaluation
GUO Wenshuo: testing, poster writing
CHEN Yuan: training models, poster writing