An exploration on Wasserstein GAN and Pearson χ^2 f-GAN

Enzo Jia [1]

[1]Department of Computer Science, Stanford University

Abstract: a problem statement

Generative Adversarial Nets (GAN) provides a straightforward yet effective method for image generation after learning from training data, and researchers have proposed variants under GAN framework. In this project I compare performances between Vanilla GAN, Wasserstein GAN (WGAN) and f-GAN, as the later two extend and generalize GAN implementations. Experimental results show that compared to vanilla GAN, WGAN and Pearson χ^2 f-GAN have similar capacities on learning data distributions and generating, given features extracted by convolutional networks. This project also explores hyper parameter tuning for WGAN and the results emphasize importance of gradient clipping threshold.

Introduction and optimization goals

Vanilla GAN

Goal of training a GAN is to have D (discriminator) to maximize the probability of correctly labeling an input image whether it is from actual data or generated by G (generator), and meanwhile to have G to minimize the probability that D labels generated images correctly. Thus we have loss:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \quad (1)$$

Wasserstein-GAN (WGAN)

For vanilla GAN the final output of its discriminator is modeled to be the probability of the input image being an actual image sampled from data and not generated. However there are other approaches we can consider, and one of them is Wasserstein-GAN (WGAN), which does not train its critic (discriminator) as an classifier outputting probabilities, but only trains it to give scores without the constraint of being between 0 and 1 [1]. WGAN optimizes the model by minimizing the Earth-Mover distance between model distribution and real data distribution.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma\in\Pi(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \quad (2)$$

f-GAN

One perspective to see vanilla GAN's training procedure of the generator is minimizing a variant of the Jensen-Shannon divergence between the model distribution and real data distribution [2]. Meanwhile we can generalize the Jensen-Shannon divergence to f-divergence family, also known as Ali-Silvey distances [3]. Under this frame we have the loss function:

$$\min_\theta \max_w \mathbb{E}_{x\sim P}[g_f(V_w(x))] + \mathbb{E}_{x\sim Q_\theta}[-f^*(g_f(V_w(x)))] \quad (3)$$

Experiment settings

Design. This project explores into 3 dimensions, and the goal is to find an optimized combination of approaches on high level framework and low level modeling.

- **Framework**. Performance difference between 3 GAN variants as discussed above.
- **Model**. Performance difference of GANs with vs. without Conv layers in their model networks.
- **Model**. Hyper parameter tuning for WGAN.

Data. This project uses the training set of MNIST data, from which I randomly select 50000 images for model training and 5000 for model validation.

Metrics. Tables of generated images are plotted for comparison.

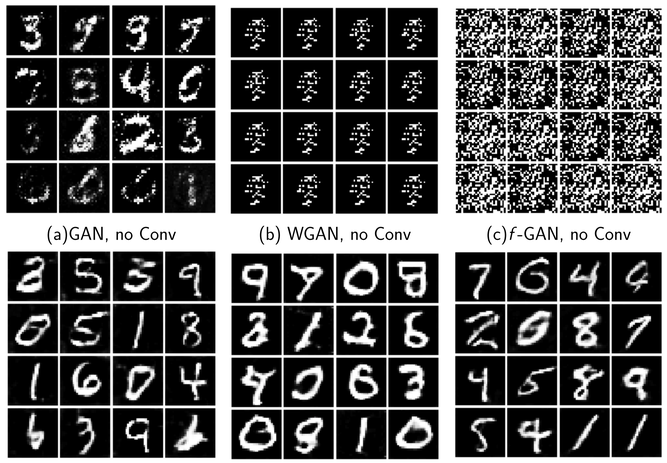
Algorithms for GAN, WGAN, and f-GAN

According to different optimization goals discussed in previous section, I derive Algorithm 1, with lines in black font are shared by all 3 GAN variants, and green lines only for Vanilla GAN, blues lines only for WGAN, and red lines only for Pearson χ^2 f-GAN.

Algorithm 1 GAN, WGAN, and f-GAN. k is number of steps to apply to the discriminator.
In the original papers [Vanilla GAN] k = 1. [WGAN] k = 5. [f-GAN] k = 1.

for number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $z^{(1)}, ..., z^{(m)}$ from noise prior $p_g(z)$
    Sample minibatch of m image samples $x^{(1)}, ..., x^{(m)}$ from data generating distribution $\mathbb{P}_r(x)$
    [f-GAN] Apply g_f as final layer activation function. For Pearson χ^2 it is identical function.
    Update the discriminator with chosen optimizer using gradient:
    [Vanilla GAN] $\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^m[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$
    [WGAN] $\nabla_w[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) + \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$
    [WGAN] w $\leftarrow$ clip(w, -c, c)
    [f-GAN] $\nabla_w[\frac{1}{m}\sum_{i=1}^m g_f(V_w(x^{(i)})) - \frac{1}{m}\sum_{i=1}^m f^*(g_f(V_w(z^{(i)})))]$, which for Pearson ^{2} is:
    [f-GAN] $\nabla_w[\frac{1}{m}\sum_{i=1}^m V_w(x^{(i)}) + \frac{1}{m}\sum_{i=1}^m \left(\frac{1}{4}V_w(z^{(i)})^2 + V_w(z^{(i)})\right)]$
  end for
  Sample minibatch of m noise samples $z^{(1)}, ..., z^{(m)}$ from noise prior $p_g(z)$
  Update the discriminator with chosen optimizer using gradient:
  [Vanilla GAN]$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^m \log(1 - D(G(z^{(i)})))$
  [WGAN] $\nabla_\theta\frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  [f-GAN] $\nabla_w[\frac{1}{m}\sum_{i=1}^m f^*(g_f(V_w(z^{(i)})))]$, which for Pearson ^{2} is:
  [f-GAN] $\nabla_w[-\frac{1}{m}\sum_{i=1}^m \left(\frac{1}{4}V_w(z^{(i)})^2 + V_w(z^{(i)})\right)]$
end for

Eexpriment results: GAN variants and Conv layers

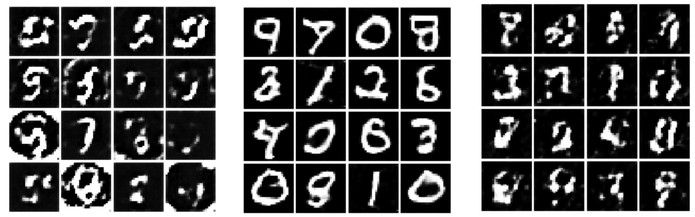


(a)GAN, no Conv (b) WGAN, no Conv (c)f-GAN, no Conv

(a)GAN, with Conv (b) WGAN, with Conv (c)f-GAN, with Conv

Figure 1. Digit images generated by Vanilla GAN, WGAN, and f-GAN, with and without Convolution layers for feature extractions, respectively.

Experiment results: gradient clipping for WGAN



(a)gradient clipped at $1e-5$ (b) gradient clipped at $7.5e-8$ (c)gradient clipped at $1e-10$

Figure 2. Digit images generated by WGAN, with different gradient clipping thresholds. All 3 has weight clipped at 0.001.

Discussions

- **Vanilla GAN vs. WGAN vs. f-GAN** using features extracted by convolutional networks generate digit images of similar qualities, and meanwhile Pearson χ^2 f-GAN generates with arguably better quality because its handwriting style aligns better with human handwriting habits. Most generated digits are readable, with small numbers of them being distorted. These three GAN variants' generations have recognizable different styles, for example WGAN's digit images have thicker strokes.
- **Fully connected layers vs. convolutional networks**. Convolutional networks extract higher-quality image features, as for each GAN variant it performs better using convolutional networks.
- **WGAN tuning**. When gradient clipping threshold is too small ($1e-10$) it has a high probability to cause vanishing gradient thus more random noisy images, and too large clipping threshold ($1e-5$) makes the network unstable and thus generates more distorted digits.
- **WGAN tuning**. On weight clipping threshold dimension, we have similar observation as gradient clipping threshold. Not included in figure 2.

Conclusions

In this project I implement Vanilla GAN, WGAN and Pearson χ^2 f-GAN, with and without convolutional layers. The results prove that under help of features extracted by convolutional layers, both WGAN and Pearson χ^2 f-GAN generate images with qualities comparable to vanilla GAN, and Pearson χ^2 f-GAN's generation has slight better performance in some dimensions including mimicking human handwriting styles. Meanwhile I also observe: 1) convolutional neural networks provide significantly better features for image learning and generating, compared to fully-connected networks, and 2) WGAN has more hyper parameters to tune, and it is sensitive to weight and gradient clipping thresholds, especially to gradient clipping threshold.

The final recommendation of GAN variant is a Pearson χ^2 GAN with its discriminator and generator implemented with convolutional networks.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou.
Wasserstein gan, 2017.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.
Generative adversarial networks, 2014.

[3] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka.
f-gan: Training generative neural samplers using variational divergence minimization, 2016.

https://github.com/enzochia/

enzojia@stanford.edu [frequently-checked email: enzo.jia@gmail.com]