MATH 6038o Mini-Project 3: Generating Images via Generative Model

XU SIAO, TAN JUNHONG, HE YUNJIE, LU JIHUA;
Department of Mathematics

Introduction

Nowadays, two of the most popular generative model for the task of generating graphs are variational auto-encoder(VAE) and generative adversarial network(GAN).

VAE is built on top of standard approximations and can be trained with stochastic gradient descent subject to the following optimization problem: $E_{X\sim D}\left[E_{z\sim Q}[\log P(X|z)] - \mathcal{D}[Q(z|X)\|P(z)]\right]$

On the other hand, GAN is to train two model which are discriminative model D and generative model G subject to the optimization problem: $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$

Data: MNIST

The MNIST database is a large database of handwritten digits hat is commonly used for training various image processing systems. The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from MIST's training dataset, while the other half of the training set and the other half of the test set were taken from MIST's testing dataset.

Graph1: a sample of MNIST.

Methodology

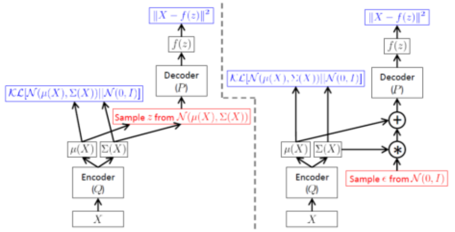
Models

- GAN: very original generative adversarial net which trains two models, one captures data distribution and another estimates the probability a sample comes from training data.
- VAE: variational auto-encoder, just like the frame such that encoding something and decoding what obtains thereafter.
- WGAN: Wasserstein generative adversarial networks, in which D is replaced with 1-Lipschitz function.
- ACGAN: auxiliary classifier generative adversarial networks, a variant to GAN based on label conditioning.
- InfoGAN: a GAN which also maximizes the mutual information between a small subset of the latent variables and the observations.
- Conditional GAN: be constructed by simply feeding data y, which is assumed to condition to both the generator and discriminator.
- Conditional VAE: model data and latent variables, both conditioned on some random variables.

Evaluation

- Inception Score: Combination of two effects (1) conditional label distribution (2) Distribution of label.
- Diversity Score: based on the belief such that generated class distribution mimic real class distribution.

Variational Auto-Encoder VS Generative Adversarial Network



Graph 2: A training-time variational autoencoder implemented as a feedforward neural network, where P(Xjz) is Gaussian. Left is without the "reparameterization trick", and right is with it.

We can see GAN outperforms VAE, indicating two-model frame performs better than encoder-decoder frame for the task of generating images.

VAE

Bad baseline

Modest

Good baseline

GAN

Bad baseline

Modest

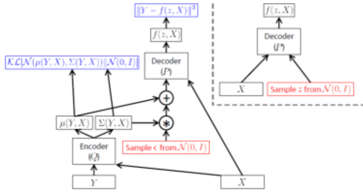
Good baseline

Algorithm GAN

Input: a set of training examples S_*, prior p_z, discriminator d such that $d(\cdot)\in[0,1]$, generator G.
Meta-parameters: mini-batch size b, discriminator update frequency U.

1: **repeat**
2: **for** U steps **do**
3: Sample $x_1^*,\ldots,x_b^*$ from S_*, and sample $z_1,\ldots,z_b$ according to p_z.
4: Update discriminator d by ascending the stochastic gradient:
$$\nabla_{\theta_d}\frac{1}{b}\sum_{i=1}^{b}[\ln d(x_i^*) + \ln(1-d(G(z_i)))]$$
5: **end for**
6: Sample $z_1,\ldots,z_b$ according to p_z.
7: Update the generator by descending the stochastic gradient: $\nabla_{\theta_G}\frac{1}{b}\sum_{i=1}^{b}\ln(1-d(G(z_i)))$
8: **until** some exit criteria are met
9: **return** generator G

Conditional VAE VS Conditional GAN



Graph 3: Left: a training-time conditional variational autoencoder implemented as a feedforward neural network. Right: the same model at test time, when we want to sample from P(YjX).

CVAE

Bad baseline

Modest

Good baseline

CGAN

Bad baseline

Modest

Good baseline

Both the performances of CVAE and CGAN are better then their original counterpart. And the performance of CGAN is still a bit better than that of CVAE.

For Conditional GAN, the object function is now: $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}|\boldsymbol{y})))]$

Wasserstein GAN With/without Gradient Penalty

Algorithm WGAN: Wasserstein GAN with gradient penalty

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.

while θ has not converged **do**
  for $t = 1, \ldots, n_{\text{critic}}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0,1]$.
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}$
      $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
    end for
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m}\sum_{i=1}^m L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^m \sim p(\boldsymbol{z})$.
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m}\sum_{i=1}^m -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while

We can see the performance of WGAN with penalty is very similar to that without penalty, indicating adding a penalty doesn't improve on WGAN much at least for this model.

WGAN


Bad baseline


Modest


Good baseline

WGAN with gradient penalty


Bad baseline


Modest


Good baseline

Information Maximizing GAN

Mutual information: the difference between two entropy terms:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Then, the original optimization problem is replaced with:

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c))$$

With information set considered, the model performs very well, meanwhile, the gap between bad image generated and good image generated is very large, indicating the volatility and instability of the model. We think the final performance of model depends on how much information leaked outside the model. The more information leaked outside and the more we can improve on our model.


Bad baseline


Modest


Good baseline

Auxiliary Classifier GAN

The objective function has two parts: the log likelihood of the correct source, Ls, and the log-likelihood of the correct class, Lc as follows:

$$L_S = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$

$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$$

Now D is trained to maximize Ls + Lc while G is trained to maximize Lc - Ls

compared with traditional GAN.

The GAN conditioned on label performs not so well,


Bad baseline


Modest


Good baseline

Evaluation via Distribution

Inception Score: image that contain meaningful objects should have a conditional label distribution $p(y|x)$ with low entropy and we expect the model generates various graphs, hence, p(y) should be with high entropy. After combination two elements, we have :

$$\exp\{\mathbb{E}_x KL(P(y|x) \| P(y))\}$$

model	Real data	GAN	CGAN	WGAE	InfoGAN	AXGAN
Inception +- std.	11.3 +- .12	7.3 +- .3	8.2 +- .3	6.8 +- .5	3.5 +- .2	5.8 +- .12

Diversity Score: take similarity between real class distribution and generated class distribution as metric. The more similar the two distributions , the higher the score:

$$\exp\{\mathbb{E}_x KL(P(y) \| P_*(y))\}$$

model	Real data	GAN	CGAN	WGAE	InfoGAN	AXGAN
Diversityon +- std.	13.3 +- .6	5.12 +- .7	7.8 +- .15	5.3 +- .2	5.6 +- .7	6.6+- .21

Contribution :
Every team member evenly contributes to all the tasks, including math modeling, data preprocessing, code writing and poster producing.

Reference

- Tutorial on Variatioanl autoencoder.
- Generative Adversarial Nets, Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley,
- Sherjil Ozairy, Aaron Courville, Yoshua Bengio
- InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya.
- Conditional Image Synthesis with Auxiliary Classifier GANs, Augustus Odena 1 Christopher Olah Jonathon.
- Improved Training of Wasserstein GANs, Ishaan Gulrajani1, Faruk Ahmed1, Martin Arjovsky2, Vincent Dumoulin1, Aaron Courville1,3.