Structure

Hi, welcome to this presentation on Info-GAN, what is so called Information maximization generative adversarial network.

Here is the structure of my presentation. In this presentation, I will briefly talk about Varitational autoencoder (VAE) and generative adversarial network (GAN) then I will compare their difference, and then focus on the summary of this Info-GAN model purposed in 2016 which is a simple but effective modified version of GAN with the idea of information theory.

Introduction

Before talking about the details, let's review some backgrounds.

Compared to supervised learning, unsupervised learning is a newly rising area of study. It doesn't rely on labeled data comparing to the supervised learning. So, it has the advantage of being able to utilize huge amount of data. In unsupervised learning, we are trying to learn some underlying hidden structure of the data.

Under such an idea, generative models are examples of unsupervised learning which focus on generating new data under the same distribution of the given training data, that is, it addresses estimation of the underlying distribution of the training data as its main purpose.

As for applications, generative models are really cool, see these generated images which looks super real. It can be used to generate fancy new data and enable inference of latent representation which can be useful general features for other tasks as we will talk more in the introduction of VAE.

Generative models can be roughly divided into two categories, one using explicit density estimation, the other using implicit density estimation.

In this presentation, I will take VAE and GAN as representation of two categories of generative models and mainly focus on summarizing Info-GAN which is a simply modified GAN but have effective performance on discovering semantic and meaningful hidden representation. And such an improvement is surprisingly by simple application of mutual information.

VAE vs GAN

VAE

To illustrate what is a variational autoencoder. First, what is an autoencoder?

Autoencoders:

x -> Encoder -> z -> Decoder -> x hat

Basically, autoencoder is composed by two networks. In the first network which is an encoder, Input data x are mapped to some features z, where z smaller than x and represents the meaningful data in x. Then, a second network is applied to restore x as x hat. The evaluation of the autoencoder, then, is the l2 norm of the difference between input x and output x hat of the second network. That is, we expect the x hat to be as close as possible to x.

After training process, the first network, encoder, can work alone as a general features extractor for unlabeled data.

The problem is we cannot randomly generate new data just using the decoder since we need a input to encoder first to generate latent feature z.

So, how can we generate new data from the autoencoder?

Variational Autoencoder:

Let's just save this question a little longer.

First, lets only look at the decoder part. Similar from decoder part of autoencoder, we assume x is generated from underlying unobserved latent distribution p theta(z) where z is a latent vector which catches the factor variations in the training data. This prior distribution can be chosen to be some simple distribution and usually is chosen to be gaussian. Then, the x can be sampled directly from the conditional distribution p theta (x|z).

Now we have the generative network but how to estimate these parameters?

We know that prior p theta(z) can be chosen to be simple distribution such as gaussian. The conditional p(x|z) can be an arbitrary observation model whose parameters are computed from a parametric function of z and it can be done through a highly flexible function approximator such as a neural network.

However, the data likelihood p theta(x) is intractable since we cannot integral through all z as p(z) is the latent distribution we assumed. Thus, p theta(z|x) is intractable as well.

As hard as to compute p theta(z|x), we can approximate p theta(z|x) by involving an additional encoder network q phi(z|x) to approximate p theta(z|x).

Similar to what have been talked about in class during the discussion of variational inference. Putting everything we have together, we eventually get a lower bound for the data likelihood. Then, by maximizing the data likelihood lower bound, we can compute all the parameters.

After the training process, we can pick out the decoder as a generative network that can generate new data.

Generally, the solution to previous problem by VAE is straightforward: we add limitations to the encoder part forcing it to generate latent features that follows some patterns. In this way, after training, we only need to feed random features that follows the same rule to the decoder part and we can generate new data.

GAN

GAN stands for generative adversarial network.

Usually, it is hard to sample from complex high-dimensional training distribution as the case in VAE encoding part. So, instead of sampling from complex high-dimension, we can simply sample from a simple random noise distribution and learn transform from it to the training distribution. GAN consists of a generator network and discriminator network. The generator generate new data based on the input random noise while the discriminator compares the generated data and the real data to judge if the generated data is real or not.

The two networks are jointly trained in the competition by the minimax objective function, which shows a game theoretic competition between the discriminator and generator. (看图说话) We train D to maximize the probability of assigning the
correct label to both training examples and samples from G while We simultaneously train G to minimize log(1 - D(G(z))) which is the difference between ground truth and the prediction on generated data.

As shown in the graph, the two neuron networks are trained through alternating between gradient ascending and gradient descending methods such that there are k steps of optimizing D and one step of optimizing G. Eventually, our idea is to maximize the probability that discriminator is correct in order to train the discriminator, and maximize the probability that discriminator is wrong in order to train the generator. Note that the alternating between updating two networks should cope with each other in case of one performs too much better than the other.

This figure shows a theoretical good enough process of the training. The blue line represent the classifier output, green line is the generate, and the black dots are training data. (a) initial step (b) shows after a inner loop of training discriminator, it becomes more accurate (c) shows after one update, discriminator guide the generator distribution flow to real data distribution. (d) shows after certain steps the discriminator cannot discriminate anymore, the network reaches its optimal state.

Comparison

Now, lets see the pros and cons of these two models.

VAE

Pros:

Allows inference of input distribution to latent distribution which can be useful for feature representation in other tasks.

Cons:

Sample quality are not as good as GAN. In the perspective of information theory, during the encoder part, the sample z are lower dimension compared to the input x which suggest there are information lost between x and z. And this information cannot be regained by decoder. Therefore generated result are blurred compared to real one.

And application of lower bound of the data likelihood is an approximation which cannot fully represent the true distribution of the input data.

GAN

Pros:

It doesn't require knowledge on the inference structure.

Can generate beautiful, state-of-art samples

In the perspective of information theory, since there is no process of converting input to lower dimension as it is in the VAE case. The generated data doesn't necessarily contains less information than real data.

Cons:

More tricky and unstable to train like we have talked about the alternating between training two networks.

Don't work with an explicit density function. Since we're using neuron network to learn transformation by 2-player competition.

Info-GAN

What is wrong with GAN?

Remember we mention that the GAN takes a random noise distribution as its input and the learning process is basically learning the transform from the noise distribution to the real data distribution. In such a case, GAN doesn't limit how the generator may use this input noise.

However, many domains naturally decompose into a set of semantically meaningful factors of variation. GAN neglects such factors, therefore, the generate uses the input noise in a potentially entangled way. This is not what we want to see, given semantic features always contains many meaningful information.

How to solve it?

In Info-GAN, the input noise vector are divided into two parts: 1. z, which is treated as source of incompressible noise 2. c, which is called latent code and will target the salient structured semantic features of the data distribution.

To avoid the division we made be ignored by GAN, the idea of mutual information is introduced to the measure the relation between latent variable c and generator distribution G(z, c). And it is taken into consideration of the objective function. In information theory, the mutual information between two random variables measures the amount of information learned from one variable of the other. Thus I(c, G(z, c)) should be high which means c should contains clear information on generator distribution x = G(z, c). Remember that

I(c, x)) = H(c) - H(c|x)

As we said, we want the mutual information between the latent variables and the generator distribution to be large. In this paper, latent variable c's distribution is assumed to be fixed for simplicity. Therefore, H(c) can be viewed as a constant. Then, to maximize mutual information between c and x, we only need to make the entropy of c | x as small as possible. Intuitively, this suggests we want x to have as much information as possible about c. In other world, the information of latent variables is not lost in the process of generation of x.

Hence the revised minmax game can be written as:

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

Similarly to the problem we meet in VAE, the posterior probability p(c|x) is hard to compute. So, we can use a distribution q(c|x) to approximate the posterior probability.

$$\begin{aligned} I(c; G(z,c)) &= H(c) - H(c|G(z,c)) \\ &= \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log P(c'|x)]] + H(c) \\ &= \mathbb{E}_{x\sim G(z,c)}[\underbrace{D_{\mathrm{KL}}(P(\cdot|x) \parallel Q(\cdot|x))}_{\geq 0} + \mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\ &\geq \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \end{aligned} \tag{4}$$

Following the similar procedure discussed in VAE, we can estimate a lower bound for the mutual information. Denote this lower bound as Li(G, Q)

$$\begin{aligned} L_I(G,Q) &= E_{c\sim P(c),x\sim G(z,c)}[\log Q(c|x)] + H(c) \\ &= E_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\ &\leq I(c; G(z,c)) \end{aligned}$$

Finally, we get the revised minmax game with a variational regularization of mutual information and a hyperparameter lambda.

$$\min_{G,Q} \max_D V_{\mathrm{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q)$$

Analysis

In practice the auxiliary distribution Q can be parametrized as a neural network, and in many experiments, Q and D share all convolutional layers and there is only one final fully connected layer to output parameters for conditional distribution Q(c|x). As shown in the image, this neural network can be viewed as classifier which judges which c generates the x. And, it is amazing to find if we only look at the generator and the classifier we have a auto-encoder like structure. Note the difference is the order of the generator and classifier.

In analysis of the lower bound Li, it is tight when the auxiliary distribution Q (c|x) approaches the true posterior distribution P(c|x). And we know when maximized the lower bound is achieved, Li(G, Q) = H(c) = I(c, G(z, c)). The maximum entropy is then achieved. Our purpose of finding the latent variables' influence to the final result is achieved at the max mutual information state.

Additionally, remember we said GAN is a unsupervised learning with no need of data. Info-GAN remains its unsupervised feature. Note c is only our pre-defined latent variable which is eventually learned by the model itself. It was confusing to me at first, c can be seen as a reserved vector for potential meaningful factor, and it can be different from case to case. But it turns out to have good pattern of c's influence to x can be learned and it is shown in the experiment results.

Experiments with MNIST

The paper performs Info-GAN on MNIST dataset. To cope with the digit dataset, c is setup as vector with length 3, where c1 is assumed to model some discontinuous variation in data, and c2, c3 are assumed to capture some continuous variations in data.

After the training on MNIST, the result shows c1 has captured some digit type variation, compared to GAN's result, it is more clear. In fact, it is pointed out in the paper that c1 can even be used as a classifier which can match c1 to digit type with only 5% of error rate.

c2 and c3 respectively captures the angel factor of the digit and the width factor of the generated digit. We can see that the generator not just distort the original digit according to different c but adding details to make the digit generated looking natural rotated or getting wider.

And we can see that even with -2 to 2 scale of the c which is out of the scale of the training set. The generated digit still looks natural, suggesting c has greatly capture the semantic factor as we expected.

Conclusion

This presentation has covered a recap of VAE and GAN and a summary of the Information Maximizing Generative Adversarial Network what we called info-GAN. It is amazing to see how a simple modification on GAN with information theory has such potential to capture semantic information. In future work, I think the idea of information theory can be expanded and combined with many other state-of-art models. And it is a really useful theory for interpreting the wanted information in the data to be processed.