Generative Adversarial Nets

Liangjie Cao

July 20, 2018

Abstract

A new framework for estimating generative models, in which simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

1. Introduction

The promise of deep learning is to discover rich, hierarchical models [1]that represent probability distributions over the kinds of data encountered in artificial intelligence applications, such as natural images, audio waveforms containing speech, and symbols in natural language corpora. So far, the most striking successes in deep learning have involved discriminative models, usually those that map a high-dimensional, rich sensory input to a class label [4, 7]. These striking successes have primarily been based on the backpropagation and dropout algorithms, using piecewise linear units [6, 2, 3] which have a particularly well-behaved gradient . Deep generative models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. So the new model estimation procedure can sidesteps these difficulties. It's GAN.

In the proposed adversarial nets framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

2. Adversarial Nets

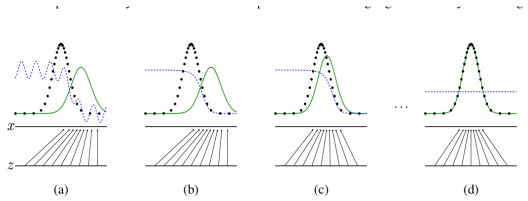

Figure 1. Generative adversarial nets

The function V(G,D) is very familiar to us, D is to maximize the probability of assigning the correct label to both training examples and samples from G. G is to minimize $log(1D(G(z)))$:

$$\min_G \max_D = V(G,D) = \mathbb{E}_{x\ p_{data}(x)}[logD(x)] + \mathbb{E}_{z\ p_z(z)}[log(1 - D(G(z)))] \quad (1)$$

See Figure 1 for a less formal, more pedagogical explanation of the approach. In practice, the authors implement the game using an iterative, numerical approach. Optimizing D to completion in the inner loop of training is computationally prohibitive, and on finite datasets would result in overfitting. Instead, they alternate between k steps of optimizing D and one step of optimizing G. This results in D being maintained near its optimal solution, so long as G changes slowly enough.

In practice, equation 1 may not provide sufficient gradient for G to learn well. Early in learning, when G is poor, D can reject samples with high confidence because

they are clearly different from the training data. In this case, $log(1D(G(z)))$ saturates. Rather than training G to minimize $log(1D(G(z)))$ or train G to maximize log $D(G(z))$. This objective function results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning.

3. Their Experiments

Figure 2. Digits obtained by linearly interpolating between coordinates in z space of the full model.

The authors trained adversarial nets an a range of datasets including MNIST, the Toronto Face Database (TFD), and CIFAR-10. The generator nets used a mixture of rectifier linear activations and sigmoid activations, while the discriminator net used maxout [3] activations. Dropout [5] was applied in training the discriminator net. While our theoretical framework permits the use of dropout and other noise at intermediate layers of the generator,Noise is as the input to only the bottommost layer of the generator network.

Figure 3 and 2 show samples drawn from the generator net after training. These samples are better than samples generated by existing methods, they believe that these samples are at least competitive with the better generative models in the literature and highlight the potential of the adversarial framework.

Figure 3. Visualization of samples from the model

4. Advantages and Disadvantages

This new framework comes with advantages and disadvantages relative to previous modeling frameworks. The disadvantages are primarily that there is no explicit representation of $p_g(x)$, and that D must be synchronized well with G during training (in particular, G must not be trained too much without updating D, in order to avoid "the Helvetica scenario" in which G collapses too many values of z to the same value of x to have enough diversity to model p data), much as the negative chains of a Boltzmann machine must be kept up to date between learning steps. The advantages are that Markov chains are never needed, only backprop is used to obtain gradients, no inference is needed during learning, and a wide variety of functions can be incorporated into the model.

5. Conclusion

This paper is the base of GAN. It has demonstrated the viability of the adversarial modeling framework, suggesting that these research directions could prove useful.

References

[1] Y. Bengio. Learning deep architectures for ai. Foundations & trends in machine learning, *2009. 1*

[2] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In AISTATS, *2011. 1*

[3] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In ICML. *1, 2*

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE signal processing magazine, *2012. 1*

[5] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. Computer science, *2012. 2*

[6] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. Lecun. What is the best multi-stage architecture for object recognition? In ICCV, *2010. 1*

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, *2012. 1*