

Generative Adversarial Nets

Yufeng Jiang

Abstract

Authors propose a novel framework for estimating generative models via an adversarial process. They 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.

1. Introduction

The promise of deep learning is to discover rich, hierarchical models that represent probability distributions over the kinds of data encountered in artificial intelligence applications. 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 [3]. 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. They propose a novel generative model estimation procedure that sidesteps these difficulties¹.

2. Adversarial nets

The adversarial modeling framework is most straightforward to apply when the models are both multilayer perceptrons. To learn the generator's distribution p_g over data \mathbf{x} , authors define a prior on input noise variables $p_z(z)$, then represent a mapping to data space as $G(z; \theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g . They also define a second

multilayer perceptron $D(\mathbf{x}; \theta_d)$ that outputs a single scalar. $D(\mathbf{x})$ represents the probability that \mathbf{x} came from the data rather than p_g . D and G play the following two-layer minimax game with value function $V(G, D)$ [1]:

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

In the next section, they present a theoretical analysis of adversarial nets, essentially showing that the training criterion allows one to recover the data generating distribution as G and D are given enough capacity. See Fig. 1 for a less formal, more pedagogical explanation of the approach.

3. Theoretical Results

The generator G implicitly defines a probability distribution p_g as the distribution of the samples $G(z)$ obtained when $z \sim p_z$. The results of this section are done in a non-parametric setting.

3.1. Global Optimality of $p_g = p_{data}$

They first consider the optimal discriminator D for any given generator G . For G fixed, the optimal discriminator D is [1]

$$D_G^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})} \quad (2)$$

The training criterion for the discriminator D , given any generator G , is to maximize the quantity $V(G, D)$ [1]

$$\begin{aligned} V(G, D) &= \int_{\mathbf{x}} p_{data}(\mathbf{x}) \log(D(\mathbf{x})) + \int_z p_z(z) \log(1 - D(G(z))) dz \\ &= \int_{\mathbf{x}} p_{data}(\mathbf{x}) \log(D(\mathbf{x})) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})) d\mathbf{x} \end{aligned} \quad (3)$$

If G and D have enough capacity, the discriminator is allowed to reach its optimum given G , and p_g is updated so as to improve the criterion [1]

$$\mathbb{E}_{\mathbf{x} \sim p_{data}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log(1 - D_G^*(\mathbf{x}))] \quad (4)$$

¹All code and hyperparameters available at <http://www.github.com/goodfeli/adversarial>

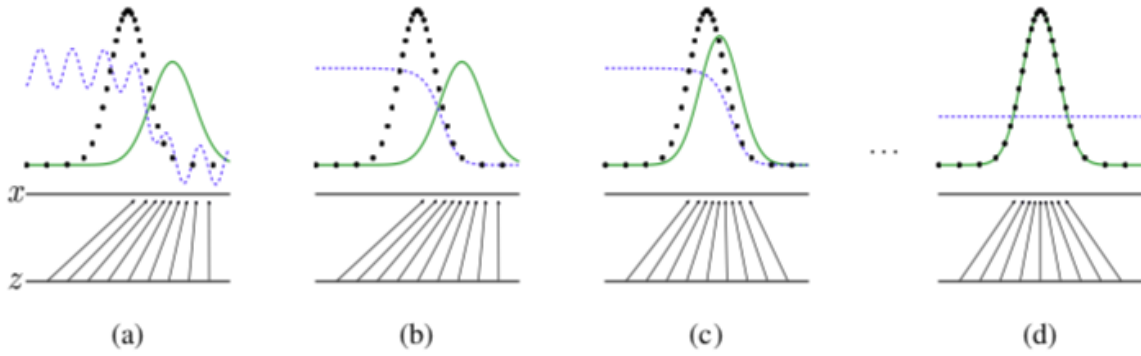


Figure 1. (a) Consider an adversarial pair near convergence: p_g is similar to p_{data} and D is a partially accurate classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to $D^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})}$. (c) After an update to G , gradient of D has guided $G(z)$ to flow to regions that are more likely to be classified as data. (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{data}$. The discriminator is unable to differentiate between the two distributions, i.e. $D(\mathbf{x}) = \frac{1}{2}$.

Model	MNIST	TFD
DBN	138 ± 2	1909 ± 66
Stacked CAE	121 ± 1.6	2110 ± 50
Deep GSN	214 ± 1.1	1890 ± 29
Adversarial nets	225 ± 2	2057 ± 26

Table 1. Parzen window-based log-likelihood estimates. The reported numbers on MNIST are the mean loglikelihood of samples on test set, with the standard error of the mean computed across examples. On TFD, we computed the standard error across folds of the dataset, with a different σ chosen using the validation set of each fold. On TFD, σ was cross validated on each fold and mean log-likelihood on each fold were computed. For MNIST we compare against other models of the real-valued (rather than binary) version of dataset.

then p_g converges to p_{data} . Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion.

4. Experiments

Authors trained adversarial nets on a range of datasets including MNIST [4], the Toronto Face Database (TFD) [5], and CIFAR-10 [2]. Results are reported in Tab. 1. This

method of estimating the likelihood has somewhat high variance and does not perform well in high dimensional spaces but it is the best method available to their knowledge. Advances in generative models that can sample but not estimate likelihood directly motivate further research into how to evaluate such models.

References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 1
- [2] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2
- [3] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1
- [4] S. Rifai, Y. Bengio, Y. Dauphin, and P. Vincent. A generative process for sampling contractive auto-encoders. In *ICML*, 2012. 2
- [5] L. Younes. On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics and Stochastic Reports*, 65(3):177–228, 1999. 2