GAN

Qingyun Li

August 24, 2018

1. Tntroduction

The framework can yield specific training algorithms for many kinds of model and optimization algorithm. In the article, the authors explore the special case when the generative model generates samples by passing random noise through a multilayer perceptron, and this discriminative model is also a multilayer perceptron. We refer to this special case as adversarial nets. In this case, we can train both models and using only the highly successful backpropagation and dropout algorithms [5] and sample from the generative model using only forward propagation. No approximate inference or Markov chains are necessary.

2. Related work

An alternative to directed graphicalmodels with latent variables are undirected graphical models with latent variables, such as restricted Boltzmann machines (RBMs) [9], deep Boltzman machines (DBMs) [8] and their numerous variants. The interactions within such models are represented as the product of unnormalized potential functions, normalized by a global summation over all states of the random variables. This quantity and its gradient are intracble for all but the most trivial instances, although they can be estimated by Markov chain Monte Carlo (MCMC) methods[1]. Mixing poses a significant problem for learning algorithms that rely on MCMC [1].

Deep belief networks (DBNs) are hybrid models containing a single undirected layer and several directed layers. While a fast approximate layer-wise training criterion exists, DBNs incur the computational difficulties associated with both undirected and directed models.

[1] A Markov chanin is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. Roughly speaking, a process satisfies the Markov property if one can make predictions for the future based only on its present state just as well as one could knowing the process's full history. In statistics, MCMC methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by observing the chain after a number of steps. The more steps there are, the more closely the distribution of the sample matches the actual desired distribution.

Alternative criteria that do not approximate or bound the log-likelihood have also been proposed, such as score matching [6] and noise-contrasive estimation (NCE) [4]. Both of these require the learned probability density to be analytically specified up to a normalization to be analytically specified up to a normalization constant. Note that in many interesting generative models with several layers of latent variable (such as DBNs and DBMs), it is not even possible to derive a tractable unnormalized probability density. Some models such as denoising auto-encodes [10] and contractive autoencoders have learning rules very similar to score matching applied to RBMs. In NCE, as in this work, a discriminative training criterion is employed to fit a generative model. However, rather than fitting a separate discriminative model, the generative model itself is used to discriminate generated data from samples a fixed noise distribution. Because NCE uses a fixed noise distribution, learning slows dramatically after the model has learned even an approximately correct distribution over a small subset of the observed variables.

Finally, some techniques do not involve defining a probability distribution explicitly, but rather train a generative machine to draw samples from the desired distribution. This approach has the advantage that such machines can be designed to be trained by back-propagation. Prominent recent work in this area includes the generative stochastic network (GSN) framework [3], which extends generalized denoising auto-encoders [2]: both can be seen as defining a parameterized Markov chain. Compared to GSNs, the adversarial nets framework does not require a Markov chain for sampling. Because adversarial nets do not require feedback loops during generation, they are better able to leverage piecewise linear units [7], which improve the performance of backpropagation but have problems with unbounded activation when used ina feedback loop. More recent examples of training a generative machine by back-propagation into it include recent work on auto-encoding variational Bayes and stochastic backpropagation.

3. Adversarial nets

The adversarial modeling framework is most straightforward to apply when the models are both multilayer perceptions. To learn the generator's distribution p_g over data x, we define a prior on input noise variable $p_z(z)$, 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. We also define a second multilayer perceptron $D(x;\theta_d)$ that outputs a single scalar. D(x) represents the probability that x came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G.

The authors make 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, i.e. in the non-parameter limit. See Figure. 1 for a less formal, more pedagogical explanation of the approach. In practice, we must 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. I stead, we 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. This strategy is analogous to the way that SML/PCD [11] training maintains samples from a Markov chain from one learning step to the next in order to avoid burning in a Markov chain as part of the inner loop of learning.

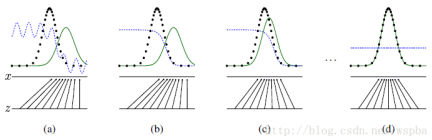

Figure 1. Procedure of training.

4. Theoretical results

The generator G implicitly defines a probability distribution p_g as the distribution of the samples G(z) obtained when z∼p(z). Therefore, we should like Algorithm 1 to converge a good estimator of p_{data}, if given enough capacity and training time. The result of this section are done in a nonparametric setting, e.g. we represent a model with infinite capacity by studying convergence in the space of probability density functions.

4.1. Global optimality of $p_g = p_{data}$

At first the authors consider the optimal discriminator D for any given generator G.

Proposition1. For G fixed, the optimal discriminator D is

$$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{1}$$

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

$$\begin{aligned} V(G,D) &= \int_x p_{data}(x) log(D(x)) dx \\ &+ \int_z p_z(z) log(1 - D(g(z))) dz \\ &= \int_x p_{data}(x) log(D(x)) \\ &+ p_g(x) log(1 - D(x)) dx \end{aligned} \tag{2}$$

References

[1] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai. Better Mixing via Deep Representations. In *ICML*, 2013. 1

[2] Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized Denoising Auto-Encoders as Generative Models. *NIPS*, pages 899–907, 2013. 1

[3] Y. Bengio, ric ThibodeauLaufer, G. Alain, and J. Yosinski. Deep Generative Stochastic Networks Trainable by Backprop. *Computer Science*, pages 226–234, 2014. 1

[4] M. Gutmann and A. Hyvrinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *JMLR*, pages 297–304, 2010. 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*, pages 212–223, 2012. 1

[6] A. Hyvrinen. Estimation of Non-Normalized Statistical Models by Score Matching. *JMLR*, pages 695–709, 2005. 1

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

[8] R. Salakhutdinov and H. Larochelle. Efficient learning of deep boltzmann machines. In *AIStats*, 2010. 1

[9] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Computer Science, 1986. 1

[10] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *ICML*, 2008. 1

[11] L. Younes. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, pages 177–228, 1999. 2