Generative Adversarial Networks

Ian J. Goodfellow, Jean Pouget-Abadiey, Mehdi Mirza,
Bing Xu, David Warde-Farley, Sherjil Ozairz,
Aaron Courville, Yoshua Bengio

Neural Information Processing Systems (NIPS), 2014

Problems Addressed

Deep generative models have problems in discovering models that represent probability distributions over the kinds of data encountered in AI, with reasons being :

(a) Difficulty in approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies

(b) Difficulty in leveraging the benefits of piecewise linear units in the generative context.

Proposed Solution

A 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.

(a) The discriminator and the generator are both multilayer perceptrons. The generator nets used a mixture of rectifier linear activations and sigmoid activations, while the discriminator net used maxout activations.

(b) The networks play a mini-max game. They can also be modeled differently e.g. as a Non-Saturating game. Competition drives both networks to improve themselves until the generated models are no longer differentiable from the training data.

(c) No Markov chains are required. Only back-propagation is used to obtain the gradients and no inference is required during learning.

Claims

(a) The network does not explicitly model the probability density that describes the data. Instead, it uses latent code to quickly recover the density, which is advantageous over Fully Visible Belief Networks, where the calculations soon become intractable.

(b) The method is asymptotically consistent, as if the equilibrium point of the game defining GANs is found, then it is guaranteed that the true probability distribution has been recovered, modulo sample complexity issues.

Results

(a) This methodology is often regarded as producing the best results, but there is no good way to quantify this. PixelCNN (A Fully Visible Belief Net) also produces very good results, but the computation takes a lot more time.

(b) GANs perform better than VAEs in tasks like upscaling images or predicting the next frame of a video, producing much more detailed and crisper images. VAEs obtain better log-likelihood estimates, however.

Bibliography

1. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano : new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.
2. Bengio, Y. (2009). Learning deep architectures for AI. Now Publishers.
3. Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. (2013). Better mixing via deep representations. In ICML'13.
4. Bengio, Y., Thibodeau-Laufer, E., and Yosinski, J. (2014a). Deep generative stochastic networks trainable by backprop. In ICML'14.
5. Bengio, Y., Thibodeau-Laufer, E., Alain, G., and Yosinski, J. (2014b). Deep generative stochastic networks trainable by backprop. In Proceedings of the 30th International Conference on Machine Learning (ICML'14).
6. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano : a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy). Oral Presentation.
7. Breuleux, O., Bengio, Y., and Vincent, P. (2011). Quickly generating representative samples from an RBM-derived process. Neural Computation, 23(8), 2053–2073.
8. Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In AISTATS'2011. 8
9. Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013a). Maxout networks. In ICML'2013.
10. Goodfellow, I. J., Mirza, M., Courville, A., and Bengio, Y. (2013b). Multi-prediction deep Boltzmann machines. In NIPS'2013.
11. Goodfellow, I. J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., and Bengio, Y. (2013c). Pylearn2 : a machine learning research library. arXiv preprint arXiv :1308.4214.
12. Gregor, K., Danihelka, I., Mnih, A., Blundell, C., andWierstra, D. (2014). Deep autoregressive networks. In ICML'2014.
13. Gutmann, M. and Hyvarinen, A. (2010). Noise-contrastive estimation : A new estimation principle for unnormalized statistical models. In Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS'10).
14. Hinton, G., Deng, L., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012a). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29(6), 82–97.

15. Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. Science, 268, 1558–1161.
16. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012b). Improving neural networks by preventing co-adaptation of feature detectors. Technical report, arXiv :1207.0580.
17. Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition ? In Proc. International Conference on Computer Vision (ICCV'09), pages 2146–2153. IEEE.
18. Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Proceedings of the International Conference on Learning Representations (ICLR).
19. Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.
20. Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In NIPS'2012.
21. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.
22. Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. Technical report, arXiv preprint arXiv :1402.0030.
23. Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. Technical report, arXiv :1401.4082.
24. Rifai, S., Bengio, Y., Dauphin, Y., and Vincent, P. (2012). A generative process for sampling contractive auto-encoders. In ICML'12.
25. Salakhutdinov, R. and Hinton, G. E. (2009). Deep Boltzmann machines. In AISTATS'2009, pages 448– 455.
26. Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. Neural Computation, 4(6), 863–879.
27. Susskind, J., Anderson, A., and Hinton, G. E. (2010). The Toronto face dataset. Technical Report UTML TR 2010-001, U. Toronto.
28. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. ICLR, abs/1312.6199.
29. Tu, Z. (2007). Learning generative models via discriminative approaches. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE.

1 Consider the game in Figure 1. Find out its minimax equilibrium. (You may want to reduce the game.)

Player	B1	B2	B3
A1	-1,1	0,0	2,-2
A2	2,-2	3,-3	4.-4
A3	5,-5	-3,3	1,-1

1. We see that B2 strongly dominates B3

$$U_2(A1, B2) > U_2(A1, B3)$$

$$U_2(A2, B2) > U_2(A2, B3)$$

$$U_2(A3, B2) > U_2(A3, B3)$$

Therefore, the game reduces to

Player	B1	B2
A1	-1,1	0,0
A2	2,-2	3,-3
A3	5,-5	-3,3

2. Now, we notice that A1 is strongly dominated by A2

$$U_1(A2, B1) > U_1(A1, B1)$$

$$U_1(A2, B2) > U_1(A1, B2)$$

The game thus reduces to

Player	B1	B2
A2	2,-2	3,-3
A3	5,-5	-3,3

Here,

$$u_{\mathrm{R}} = 2 \; (where \; u_{\mathrm{R}} = max_{\mathrm{i}} \; min_{\mathrm{j}} \; a_{\mathrm{ij}})$$
$$u_{\mathrm{C}} = 3 \; (where \; u_{\mathrm{C}} = min_{\mathrm{j}} \; max_{\mathrm{i}} \; a_{\mathrm{ij}})$$

Therefore, there is no Minimax Equilibrium.
However, there exists a Nash Equilibrium.
Let us assume that Player 2 plays B1 with probability p and B2 with probability 1-p

$$2p + 3(1 - p) = 5p - 3(1 - p)$$

$$9p = 6$$

$$p = \frac{2}{3}$$

Similarly, we can assume that Player 1 plays A2 with probability q and A3 with probability 1-q

$$-2q - 5(1-q) = -3q + 3(1-q)$$

$$9q = 8$$

$$q = \frac{8}{9}$$