International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-8, Issue-3, May 2020
https://doi.org/10.21276/ijircst.2020.8.3.24
www.ijircst.org

Deep Learning Approach To Face Conditioning Using Invertible Conditional Generative Adversarial Networks (ICGAN)

Utpal Srivastav, Vikas Thada, Amit Kumar, Maulik Garach, Adit Paliwal

ABSTRACT- We propose another system for evaluating generative models by means of an ill-disposed process, in which we at the same time train two models: a generative model G that catches the information conveyance, and a discriminative model D that gauges the likelihood that an example originated from the preparation information as opposed to G. The preparation strategy for G is to expand the likelihood of D committing an error. This system compares to a minimax two-player game. In the space of discretionary capacities G and D, an interesting arrangement exists, with G recuperating the preparation information conveyance and D equivalent to 1/2 all over the place. For the situation where G and D are characterized by multilayer perceptions, the whole framework can be prepared with back propagation. There is no requirement for any Markov chains or unrolled estimated deduction systems during either preparing or age of tests. Investigations illustrate the capability of the system through subjective and quantitative assessment of the produced tests.

KEYWORDS- ICGAN, Face Conditioning, Deep Learning.

Manuscript received May 20, 2020

Utpal Shrivastava, Department of Computer Science, Amity University, Gurgaon, Haryana, India (email: utpalshrivastava@gmail.com)

Vikas Thada, Department of Computer Science, Amity University, Gurgaon, Haryana, India

Amit Kumar, Department of Computer Science, Amity University, Gurgaon, Haryana, India

Maulik Garach, Department of Computer Science, Amity University, Gurgaon, Haryana, India

Adit Paliwal, Department of Computer Science, Amity University, Gurgaon, Haryana, India

I. INTRODUCTION

The guarantee of deep learning is to find rich, various leveled models [2] that speak to probability disseminations over the sorts of data experienced in artificial intelligence applications, for example, natural images, sound waveforms containing discourse, and images in natural language.

Up until now, the most striking accomplishments in deep learning have included discriminative models, ordinarily those that map a high-dimensional, rich tangible contribution to a class mark [14, 22]. These striking victories have principally been founded on the backpropagation and dropout algorithms, utilizing piecewise straight units[19, 9, 10] which have an especially respectful inclination. Deep generative models have had less of an effect, because of the trouble of approximating numerous recalcitrant probabilistic calculations that emerge in most extreme probability estimation and related techniques, and because of trouble of utilizing the advantages of piecewise straight units in the generative setting. We propose another generative model estimation technique that avoids these troubles. In the proposed adversarial nets structure, the generative model is set in opposition to a foe: a discriminative model that figures out how to decide if an example is from the model conveyance or the data dispersion. The generative model can be thought of as practically equivalent to a group of forgers, attempting to deliver counterfeit cash and use it without location, while the discriminative model is closely resembling the police, attempting to identify the fake money. Rivalry in this game drives the two groups to improve their strategies until the fakes are unclear from the veritable articles. This structure can yield explicit preparing algorithms for some sorts of model and optimization algorithm. In this article, we investigate the exceptional situation when the generative model produces tests by going arbitrary clamor through a multilayer perceptron, and the discriminative model is additionally a multilayer perceptron. We allude to this unique case as adversarial nets. For this situation, we can prepare the two models utilizing just the profoundly fruitful backpropagation and dropout algorithms [17] and test from the generative model utilizing just forward engendering. No surmised derivation or Markov chains are fundamental.

II. RELATED WORK

An option in contrast to coordinated graphical models with inert factors are undirected graphical models with inactive factors, for example, limited Boltzmann machines (RBMs) [27, 16], deep Boltzmann machines (DBMs) [26], and their various variations. The connections inside such models are spoken to as the result of non-normalized potential capacities, standardized by a worldwide summation/coordination over all conditions of the irregular factors. This amount (the parcel capacity) and its slope are unmanageable for everything except the most inconsequential examples, despite the fact that they can be assessed by Markov chain Monte Carlo (MCMC) techniques. Blending represents a noteworthy issue for learning algorithms that depend on MCMC [3, 5]. Deep conviction systems (DBNs) [16] are hybrid models containing a solitary undirected layer and a few coordinated layers. While a quick estimated layer-wise preparing standard exists, DBNs cause the computational troubles related with both undirected and coordinated models. Elective rules that don't rough or bound the log-probability have additionally been proposed, for example, score coordinating [18] and clamor contrastive estimation (NCE) [13]. Both of these require the educated probability thickness to be systematically indicated up to a normalization consistent. Note that in many fascinating generative models with a few layers of idle factors, (for example, DBNs and DBMs), it isn't even conceivable to infer a tractable non-normalized probability thickness. A few models, for example, de-noising auto-encoders [30] and contractive auto encoders have learning rules fundamentally the same as score coordinating applied to RBMs. In NCE, as in this work, a discriminative preparing measure is utilized to fit a generative model. Be that as it may, instead of fitting a different discriminative model, the generative model itself is utilized to separate produced data from tests a fixed clamor dissemination. Since NCE utilizes a fixed clamor dissemination, learning eases back drastically after the model has learned even an around right conveyance over a little subset of the watched factors. At long last, a few procedures don't include characterizing a probability dispersion expressly, but instead, train a generative machine to draw tests from the ideal circulation. This methodology has the favorable position that such machines can be intended to be prepared by back-spread. Noticeable late work around there incorporates the generative stochastic system (GSN) structure [5], which expands summed up denoising auto-encoders [4]: both can be viewed as characterizing a parameterized Markov chain, i.e., one learns the parameters of a machine that performs one stage of a generative Markov chain. Contrasted with GSNs, the adversarial nets structure doesn't require a Markov chain for examining. Since adversarial nets don't require criticism circles during age, they are better ready to use piecewise direct units [19, 9, 10], which improve the exhibition of backpropagation yet have issues with unbounded actuation when utilized in a input circle. Later instances of preparing a generative machine by back-engendering into it remember late work for auto-encoding variational Bayes [20] and stochastic backpropagation [24]

III. ADVERSARIAL NETS

The adversarial modeling system is generally clear to apply when the models are both multilayer perceptrons. To become familiar with the generator's dispersion p_g over data x, we characterize an earlier on input clamor factors $p_g(z)$, at that point speak to a mapping to data space as G(z; Θg), where G is a differentiable capacity spoken to by a multilayer perceptron with parameters g. We additionally characterize a second multilayer perceptron D(x; Θd) that yields a solitary scalar. D(x) speaks to the probability that x originated from the data instead of p_g. We train D to boost the probability of allotting the right name to both training models and tests from G. We at the same time train G to limit log(1 Θ D(G(z))):

In other words, D and G play the following two-player minimax game with value function V (G;D):

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

In the following segment, we present a theoretical investigation of adversarial nets, basically indicating that the training measure permits one to recoup the data creating circulation as G and D are given enough limit, i.e., in the non-parametric breaking point. See Figure 1 for a less formal, increasingly educational clarification of the methodology. By and by, we should execute the game utilizing an iterative, numerical methodology. Optimizing D to finishing in the internal circle of training is computationally restrictive, and on limited datasets would bring about over fitting. Rather, we shift back and forth between k steps of optimizing D and one stage of optimizing G. This outcomes in D being kept up close to its ideal arrangement, inasmuch as G changes gradually enough. This procedure is practically equivalent to the way that SML/PCD [31, 29] training keeps up tests from a Markov chain starting with one learning step then onto the next so as to abstain from copying in a Markov chain as a component of the internal circle of learning. The system is officially introduced in Algorithm 1. By and by, condition 1 may not give adequate gradient to G to learn well. Right off the bat in learning, when G is poor, D can dismiss tests with high certainty since they are obviously not quite the same as the training data. For this situation, log (1 Θ D(G(z))) soaks. Instead of training G to limit log (1 Θ D(G(z))) we can prepare G to expand log D(G(z)). This target work brings about the equivalent fixed purpose of the elements of G and D however gives a lot more grounded gradients right off the bat in learning.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-8, Issue-3, May 2020
https://doi.org/10.21276/ijircst.2020.8.3.24
www.ijircst.org

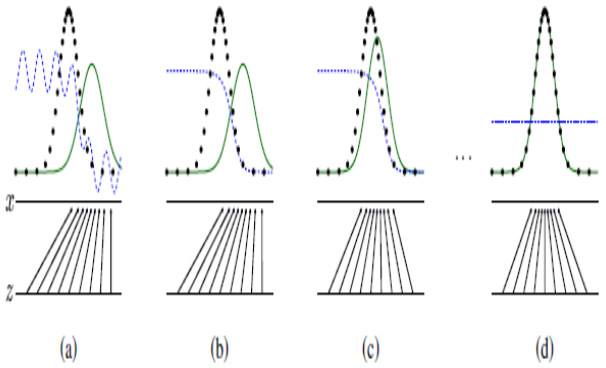

Fig 1: Generative adversarial nets are trained by simultaneously updating the discriminative distribution.

(d, blue, dashed line) so that it differs between instances from the data generating dispersion(distribution) (black, dotted line) p_x from those of the generative dispersion(distribution) p_g (G) (green, solid line). The minor horizontal line is the turf from which z is instanced, in this case systematically. The major horizontal line is part of the turf of x. The upward arrows show how the mapping x = G(z) enforce the non-systematic distribution p_g on reconstructed instances. G converges in area of high density and diverges in area of low density of p_g. (a) Consider an adversarial pair near modification: p_g is similar to p_{data} and D is a partly proper classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to D(x) = $\frac{pdata(x))}{pdata(x}$ + p_g(x). (c) After an improvement to G, gradient of D has instructed G(z) to flow to areas that are more reasonable to be categorized as data. (d) After a few steps of training, if G and D have enough quantity, they will come to a point at which both cannot advance because $p_g = p_{data}$. The discriminator is not able to tell the difference between the two distributions, i.e. D(x) = 1/2.

IV. PROCESS

The generator G certainly characterizes a probability dissemination p_g as the conveyance of the examples G(z) got when z ~ pz. In this manner, we might want Algorithm 1 to merge to a decent estimator of p_{data}, whenever given enough limit and training time. The consequences of this area are done in a nonparametric setting, for example we speak to a model with unending limit by examining combination in the space of probability density capacities. We will appear in segment 4.1 that this minimax game has a worldwide ideal for $p_g = p_{data}$. We will at that point appear in segment 4.2 that Algorithm 1 improves Eq 1, accordingly getting the ideal outcome

Algorithm 1 Minibatch stochastic gradient plunge training of generative adversarial nets. The quantity of steps to apply to the discriminator, k, is a hyper parameter. We utilized k = 1, the most affordable alternative, in our experiments.

for number of training emphases do
for **k** steps do

- Sample mini batch of m clamor tests fz(1); : ; z(m)g from commotion earlier $p_g(z)$.
- Sample mini batch of m models $\{x^{(1)}; : ; x^{(m)}\}$ from data producing dispersion $p_{data}(x)$.
- Update the discriminator by climbing its stochastic gradient:

$$\nabla_{\theta d} \frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right) + \log(1 - D\left(G\left(z^{(i)}\right)\right))\right]$$

end **for**

- Sample mini batch of m noise samples $\{z^{(1)},\ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$g\frac{1}{m}\sum_{i=1}^{m}\log(1 - D(G(Z^{(i)})))$$

end **for**

The weights and biases-based improvements can use any standard gradient-based learning rule. We used momentum in our experiments.

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

Firstly, we examine the optimized discriminator D for any given generator G.

Proposition 1. For G fixed, the optimal discriminator D is

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

Proof: The training basis for the discriminator D, given any generator G, is to expand the amount V (G;D)

$$V(G,D) = \int_{\infty} p_{data}(x)\log\left(D(x)\right)dx + \int_{z} pz^{(z)}\log\left(1 - D\left(g(z)\right)\right)dz$$

$$= \int_{x} p_{data}(x)\log\left(D(x)\right)dx + \int_{z} p_g(x)\log\left(1 - D\left(g(z)\right)\right)dz$$

For any (a; b) $2 \epsilon R^2 \setminus \{0; 0\}$, the function $y \rightarrow a\log(y) + b\log(1 - y)$ achieves its maximum in [0; 1] at a $\frac{a}{a+b}$. The discriminator does not need to be defined outside of Supp (p_{data}) U Supp(p_g), concluding the proof.

The training purpose for D can be explained as maximizing the log-tendency for estimating the conditional probability P(Y = y | x), where Y demonstrates whether x appears from p_{data} (with y = 1) or from p_g (with y = 0). The minimax game in Eq. 1 can now be reformulated as:

$$C(G) = maxD\,V(G,D)$$

$$= \mathbb{E}_{x \backsim p_{data}}[\log D^*_G(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D^*_G((G(z)))]$$

$$= \mathbb{E}_{x \backsim p_{data}}\left[\log D^*_G(x)\right] + \mathbb{E}_{x \sim p_g}[\log(1 - D^*_G(x))]$$

$$= \mathbb{E}_{x \backsim p_{data}}\left[\log\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + \mathbb{E}_{x \backsim p_g}\left[\log\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right]$$

Theorem 1. The global minimum of the virtual training precedent C(G) is achieved if and only if $p_g = p_{data}$. At that point of time, C(G) becomes -ve log 4.

Proof. For $p_g = p_{data}$, $D^*_G(x) = 1/2$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D^*_G(x) = 1/2$,

we find C(G) = log 12 + log 1 2 = −log 4. To identify that this is the best feasible value of C(G), arrived only for $p_g = p_{data}$, examine that:

$$\mathbb{E}_x \sim P_{data}[-log2] + \mathbb{E}_x \sim P_g[-log2] = -log4$$

And that by subtracting this expression from C(G) =V (D^*_G,G), we obtain:

$$C(G) = -\log(4) + K\left(p_{data} \,\|\, \frac{p_{data} + p_g}{2}\right) + KL\left(p_g \,\|\, \frac{p_{data} + p_g}{2}\right)$$

Where KL is the Kullback–Leibler divergence. We perceive in the past articulation the Jensen–Shannon divergence between the model's conveyance and the data producing process:

$$C(G) = -\log(4) + 2JSD\left(p_{data} \,\|\, p_g\right)$$

Considering the Jensen–Shannon divergence between two dispersion (distributions) is always positive and zero only when they are equal, we have shown that C(G) = -ve log(4) is the global minimum of C(G) and that the only solution is $p_g = p_{data}$, i.e., the generative model effectively imitate the data generating process.

B. Convergence of Algorithm 1

Proposition 2. If G and D have enough quantity, and at every step of Algorithm 1, the discriminator is allowed to arrive optimized given G, and p_g is changed so as to improve the criterion.

Proof. Consider V (G;D) = U(p_g;D) as a method of p_g as done in the above precedent. Note that U(p_g;D) is convex in p_g. The sub derivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. In other words, if $f(x) = \sup_{\alpha \in A} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\delta f_\beta(x) \in \delta f$ if $\beta = \arg\sup_{\alpha \in A} f_\alpha(x)$. This is equal to calculating a gradient descent change for p_g at the optimized D given the comparable G. $\sup_D$ U(p_g;D) is convex in p_g with a unique global optima as proven in Thm 1,therefore with sufficiently small updates of p_g, p_g converges to p_x, concluding the proof. In practice, adversarial nets speak to a constrained group of p_g conveyances by means of the capacity G(z; Θ g), and we advance Θ $_g$ instead of p_g itself. Using a multilayer perceptron to define G introduces various basic points in parameter space. Be that as it may, the fantastic presentation of multilayer perceptrons in practice proposes that they are a sensible model to use in spite of their absence of theoretical certifications.

V. EXPERIMENTS

We trained adversarial nets a range of datasets including MNIST[23], the Toronto Face Database (TFD) [28], and CIFAR-10 [21]. The generator nets used a mixture of rectifier linear activations [19, 9] and sigmoid activations, while the discriminator net used max out [10] activations. Dropout [17] was enforced in training the discriminator network. While our philosophical structure allows the use of dropout and other random distribution (noise) at intermediary layers of the generator, we used random distribution (noise) as the input to only the lowest layer of the generator network.

Table 1: Parzen window-based log-likelihood gauges.

Model	MNIST	TFD
DBN[3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	2110 ± 50
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	225 ± 2	2057 ± 26

The announced numbers on MNIST are the mean log likelihood of tests on the test set, with the standard blunder of the mean processed across models. On TFD, we processed the standard blunder across folds of the dataset, with an alternate σ picked utilizing the approval set of each overlay. On TFD, σ it was cross-approved on each overlay, and mean log-likelihood on each overlap was registered. For MNIST we look at against different models of the genuine esteemed (as opposed to double) variant of the dataset. We gauge the probability of the test set data under p_g by fitting a Gaussian window to the examples produced with G and revealing the log-likelihood under this dispersion. The σ parameter of the Gaussians was gotten by cross-approval on the approval set. This method was presented in Breuleux et al. [8] and utilized for different generative models for which the specific likelihood isn't tractable [25, 3, 5]. The outcomes are accounted for in Table 1. This technique for assessing the likelihood has to some degree high fluctuation and doesn't perform well in high dimensional spaces yet it is the best strategy accessible as far as anyone is concerned. Advances in generative models that can test however not gauge likelihood straightforwardly inspire further investigation into how to assess such models. In Figures 2 and 3 we show tests drawn from the generator net in the wake of training. While we make no case that these examples are better than tests produced by existing techniques, we accept that these examples are in any event serious with the better generative models in the writing and feature the capability of the adversarial system

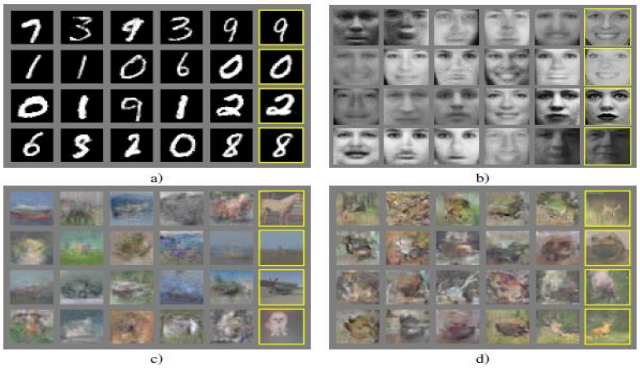

Fig 2: Visualization of tests from the model.

The furthest right segment shows the closest training case of the neighboring example, so as to exhibit that the model has not retained the training set. Tests are reasonable arbitrary draws, not singled out. In contrast to most different visualizations of deep generative models, these images show real examples from the model dispersions, not restrictive methods given examples of concealed units. In addition, these examples are uncorrelated on the grounds that the testing procedure doesn't rely upon Markov chain blending. a) MNIST b) TFD c) CIFAR-10 (completely associated model) d) CIFAR-10 (Figure 2: Visualization of tests from the model. The furthest right segment shows the closest training case of the neighboring example, so as to exhibit that the model has not retained the training set. Tests are reasonable irregular draws, not filtered out. In contrast to most different visualizations of deep generative models, these images show genuine examples from the model dispersions, not contingent methods given examples of shrouded units. Also, these examples are uncorrelated on the grounds that the examining procedure doesn't rely upon Markov chain blending. a) MNIST b) TFD c) CIFAR-10 (completely associated model) d) CIFAR-10 (convolution discriminator and "deconvolutional" generator) discriminator and "deconvolutional" generator)

Fig 3: Digits obtained by linearly interpolating between coordinates in z space of the full model

Table 2: Challenges in generative modeling: a summary of the difficulties encountered by different approaches to deep generative modeling for each of the major operations involving a model

	Deep directed graphical models	Deep undirected graphical models
Training	Inference needed during training	Inference needed during training. MCMC needed to approximate partition function gradient.
Inference	Learned approximate inference	Variational inference
Sampling	No difficulties	Requires Markov chain
Evaluating p(x)	Intractable, may be approximated with AIS	Intractable, may be approximated with AIS
Model design	Nearly all models incur extreme difficulty	Careful design needed to ensure multiple properties

VI. ADVNTAGES AND DISADVANTAGES

This new system accompanies advantages and disadvantages comparative with past modeling structures. The disadvantages are fundamental that there is no express portrayal of $p_g(x)$, and that D must be synchronized well with G during training (specifically, G must not be prepared a lot without refreshing, so as to keep away from "the Helvetica scenario" where G crumples an excessive number of estimations of z to a similar estimation of x to have enough assorted variety to model p_{data}), much as the negative chains of a Boltzmann machine must be stayed up with the latest between learning steps. The advantages are that Markov chains are rarely required, the main scenery is utilized to acquire gradients, no induction is required during learning, and a wide assortment of capacities can be fused into the model. Table 2 sums up the correlation of generative adversarial nets with other generative modeling draw near.

The previously mentioned advantages are fundamentally computational. Adversarial models may likewise increase some factual preferred position from the generator arrange not being refreshed straightforwardly with data models, however just with gradients coursing through the discriminator. This implies parts of the information are not replicated straightforwardly into the generator's parameters. Another favorable position of adversarial systems is that they can speak to exceptionally sharp, even ruffian appropriations, while methods dependent on Makov chains necessitate that the conveyance be to some degree foggy all together for the chains to have the option to blend between modes.

VII. CONCLUSION AND FUTURE WORK

This framework concedes numerous direct augmentations:

1. A contingent generative model p(x j c) can be obtained by adding c as input to both G and D.
2. Learned inexact inference can be performed via training an assistant system to anticipate z given x. This is like the inference net trained by the wake-rest algorithm [15] yet with the bit of leeway that the inference net might be trained for a fixed generator net after the generator net has finished training.7
3. One can around model all conditionals p(xS j x6S) where S is a subset of the indices of x via training a group of

contingent models that share parameters. Basically, one can utilize adversarial nets to actualize a stochastic augmentation of the deterministic MP-DBM [11].

4. Semi-supervised learning: highlights from the discriminator or inference net could improve the presentation of classifiers when restricted marked data is accessible.

5. Effectiveness enhancements: training could be quickened incredibly by divising better methods for coordinating G and D or determining better circulations to test z from during training.

This paper has shown the suitability of the adversarial modeling system, suggesting that these exploration headings could demonstrate valuable.

REFERENCES

[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. (2013a). Better mixing via deep representations. In ICML'13.

[4] Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013b). Generalized denoising auto-encoders as generative models. In NIPS26. Nips Foundation.

[5] Bengio, Y., Thibodeau-Laufer, E., and Yosinski, J. (2014a). Deep generative stochastic networks trainable by backprop. In ICML'14.

[6] 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).

[7] 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.

[8] Breuleux, O., Bengio, Y., and Vincent, P. (2011). Quickly generating representative samples from anRBM-derived process. Neural Computation, 23(8), 2053–2073.

[9] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In AISTATS'2011.

[10] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013a). Maxoutnetworks. In ICML'2013.

[11] Goodfellow, I. J., Mirza, M., Courville, A., and Bengio, Y. (2013b). Multi-prediction deep Boltzmann machines. In NIPS'2013.

[12] 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 preprintarXiv:1308.4214.

[13] Gutmann, M. and Hyvarinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In AISTATS'2010.

[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., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.

[17] 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.

[18] Hyv¨arinen, A. (2005). Estimation of non-normalized statistical models using score matching. J. Machine Learning Res., 6.

[19] 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.

[20] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Proceedings of the International Conference on Learning Representations (ICLR).

[21] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[22] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In NIPS'2012.

[23] 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.

[24] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. Technical report, arXiv:1401.4082.

[25] Rifai, S., Bengio, Y., Dauphin, Y., and Vincent, P. (2012). A generative process for sampling contractive auto-encoders. In ICML'12.

[26] Salakhutdinov, R. and Hinton, G. E. (2009). Deep Boltzmann machines. In AISTATS'2009, pages 448–455.

[27] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing, volume 1, chapter 6, pages 194–281. MIT Press, Cambridge.

[28] Susskind, J., Anderson, A., and Hinton, G. E. (2010). The Toronto face dataset. Technical Report UTML TR 2010-001, U. Toronto.

[29] Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In W. W. Cohen, A. McCallum, and S. T. Roweis, editors, ICML 2008, pages 1064–1071. ACM.

[30] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In ICML 2008.

[31] Younes, L. (1999). On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. Stochastics and Stochastic Reports, 65(3), 177–228.

