WHY ARE DEEP NETS REVERSIBLE: A SIMPLE THEORY, WITH IMPLICATIONS FOR TRAINING

Sanjeev Arora, Yingyu Liang & Tengyu Ma
Department of Computer Science
Princeton University
Princeton, NJ 08540, USA
{arora,yingyul,tengyu}@cs.princeton.edu

ABSTRACT

Generative models for deep learning are promising both to improve understanding of the model, and yield training methods requiring fewer labeled samples. Recent works use generative model approaches to produce the deep net's input given the value of a hidden layer several levels above. However, there is no accompanying "proof of correctness" for the generative model, showing that the feedforward deep net is the correct inference method for recovering the hidden layer given the input. Furthermore, these models are complicated.

The current paper takes a more *theoretical* tack. It presents a very simple generative model for ReLU deep nets, with the following characteristics: (i) The generative model is just the *reverse* of the feedforward net: if the forward transformation at a layer is A then the reverse transformation is A^T. (This can be seen as an explanation of the old *weight tying* idea for denoising autoencoders.) (ii) Its correctness can be *proven* under a clean theoretical assumption: the edge weights in real-life deep nets behave like random numbers. Under this assumption —which is experimentally tested on real-life nets like AlexNet— it is formally proved that feed forward net is a correct inference method for recovering the hidden layer.

The generative model suggests a simple modification for training: use the generative model to produce synthetic data with labels and include it in the training set. Experiments are shown to support this theory of random-like deep nets; and that it helps the training.

This extended abstract provides a succinct description of our results while the full paper is available on arXiv.

1 INTRODUCTION

Discriminative/generative pairs of models for classification tasks are an old theme in machine learning (Ng & Jordan, 2001). Generative model analogs for deep learning may not only cast new light on the discriminative backpropagation algorithm, but also allow learning with fewer labeled samples. A seeming obstacle in this quest is that deep nets are successful in a variety of domains, and it is unlikely that problem inputs in these domains share common families of generative models.

Some generic (i.e., not tied to specific domain) approaches to defining such models include *Restricted Boltzmann Machines* (Freund & Haussler, 1994; Hinton & Salakhutdinov, 2006) and *Denoising Autoencoders* (Bengio et al., 2006; Vincent et al., 2008). Surprisingly, these suggest that deep nets are *reversible*: the generative model is a essentially the feedforward net run in reverse. Further refinements include Stacked Denoising Autoencoders (Vincent et al., 2010), Generalized Denoising Auto-Encoders (Bengio et al., 2013b) and Deep Generative Stochastic Networks (Bengio et al., 2013a).

In case of image recognition it is possible to work harder —using a custom deep net to invert the feedforward net —-and *reproduce* the input very well from the values of hidden layers much higher up, and in fact to generate images very different from any that were used to train the net (e.g., (Mahendran & Vedaldi, 2015)).

To explain the contribution of this paper and contrast with past work, we need to formally define the problem. Let x denote the data/input to the deep net and z denote the hidden representation (or the output labels). The generative model has to satisfy the following: **Property (a)**: Specify a joint distribution of x, z, or at least $p(x|z)$. **Property (b)** A proof that the deep net itself is a method of computing the (most likely) z given x. Past work usually fails to satisfy one of (a) and (b).

The current paper introduces a simple mathematical explanation for *why* such a model should exist for deep nets with fully connected layers. We propose the *random-like nets hypothesis*, which says that real-life deep nets –even those obtained from standard supervised learning—are "random-like," meaning their edge weights behave like random numbers. Notice, this is distinct from saying that the edge weights actually *are* randomly generated or uncorrelated. Instead we mean that the weighted graph has bulk properties similar to those of random weighted graphs. To give an example, matrices in a host of settings are known to display properties —specifically, eigenvalue distribution— similar to matrices with Gaussian entries; this so-called *Universality* phenomenon is a matrix analog of the Law of Large Numbers. The random-like properties of deep nets needed in this paper involve a generalized eigenvalue-like property, which we empirically verified on the real world neural nets.

If a deep net is random-like, we can show mathematically that it has an associated simple generative model $p(x|z)$ (Property (a)) that we call the *shadow distribution*, and for which Property (b) also *automatically* holds in an approximate sense. Our generative model makes essential use of dropout noise and ReLUs and can be seen as providing (yet another) theoretical explanation for the efficacy of these two in modern deep nets.

Note that Properties (a) and (b) hold even for random (and hence untrained/useless) deep nets. Empirically, supervised training seems to improve the shadow distribution, and at the end the synthetic images are somewhat reasonable, albeit cruder compared to say (Mahendran & Vedaldi, 2015).

2 GENERATIVE MODEL AND PROVABLE GUARANTEES

Let x denote the input and h denote the hidden variable computed by the neural network. When h has fewer nonzero coordinates than x, this has to be a many-to one function, and prior work on generative models has tried to define a probabilistic inverse of this function. Sometimes —e.g., in context of denoising autoencoders— such inverses are called *reconstruction* if one thinks of all inverses as being similar. Here we abandon the idea of reconstruction and focus on defining *many* inverses $\tilde{x}$ of h. We define a *shadow distribution* $p(x|h)$, such that a random sample $\tilde{x}$ from this distribution satisfies Property (b), i.e., the feedforward network computes a hidden variable that is close to h. For example, one layer network with weight W and bias b outputs $\mathsf{ReLU}(W^T\tilde{x} + b)$, which we show is close to h. To understand the considerations in defining such an inverse, one must keep in mind that the ultimate goal is to extend the notion to multi-level nets. Thus a generated "inverse" $\tilde{x}$ has to look like the *output* of a 1-level net in the layer below. As mentioned, this is where previous attempts such as DBN or denoising autoencoders run into theoretical difficulties.

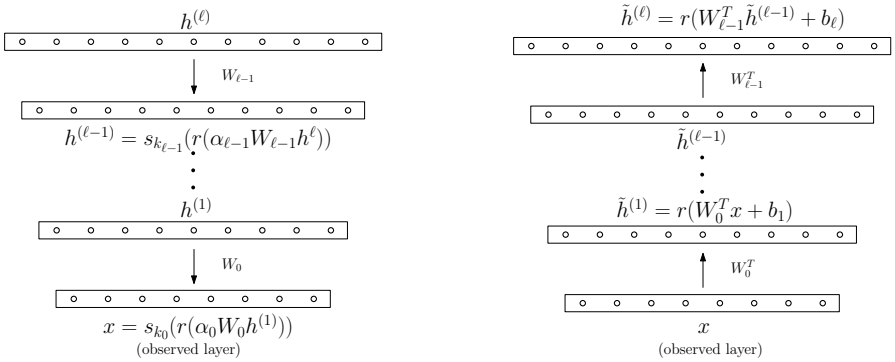
One layer model. For simplicity we start with a single layer neural net. Given $h \in \mathbb{R}^m$, the model $p(x|h)$ generates $x \in \mathbb{R}^n$ as follows: first compute $r(\alpha Wh)$ where a scaling scalar α and and r is the ReLU; then randomly zero-out each coordinate with probability $1 - \rho$. Formally, let $\odot$ denote entry-wise product of two vectors. Then $p(x|h)$ is defined as

$$x = r(\alpha Wh) \odot n_{\text{drop}}\,, \tag{1}$$

where $\alpha = 2/(\rho n)$ is a scaling factor, and $n_{\text{drop}} \in \{0,1\}^n$ is a binary random vector satifying

$$\Pr[n_{\text{drop}}] = \rho^{\|n_{\text{drop}}\|_0}(1-\rho)^{n-\|n_{\text{drop}}\|_0}, \tag{2}$$

where $\|n_{\text{drop}}\|_0$ denotes the number of non-zeros of n_{drop}. We refer to this noise model as "dropout noise", and note that ρ can be reduced to make x as sparse as needed; typically ρ will be small. Let $s_t(\cdot)$ be the



(a) Generative model (b) Feedforward NN

Figure 1: Generative-discriminative pair: a) defines the conditional distribution $\Pr[x|h^{(\ell)}]$; b) defines the feed-forward function $\tilde{h}^{(\ell)} = \text{NN}(x)$.

random function that drops coordinates with probability $1-\rho$, that is, $s_t(z) = z \odot n_{\text{drop}}$. Then (1) is simplified to

$$x = s_{\rho n}(r(\alpha W h)). \tag{3}$$

Multiple layer model. We describe the multilayer generative model for the shadow distribution associated with an ℓ-layer deep net with ReLUs. The feedforward net is shown in Figure 2 (b). The j-th layer has n_j nodes, while the observable layer has n_0 nodes. The corresponding generative model is in Figure 2 (a). The number of variables at each layer, and the edge weights match exactly in the two models, but the generative model runs from top to down.

The generative model starts with the top layer $h^{(\ell)}$, which is from some arbitrary distribution D_ℓ over the set of k_ℓ-sparse vectors in $\mathbb{R}^{n_\ell}$. Then it generates the hidden variable $h^{(\ell-1)}$ below using the same stochastic process as described for one layer: compute $r(\alpha_{\ell-1}W_{\ell-1}h^{(\ell)})$, and then apply a random sampling function $s_{k_{\ell-1}}(\cdot)$ on the vector, where $k_{\ell-1}$ is the target sparsity of $h^{(\ell-1)}$, and $\alpha_{\ell-1} = 2/k_{\ell-1}$ is a scaling constant. Formally, the generative model is

$$x = s_{k_0}(r(\alpha_0 W_0 s_{k_1}(r(\alpha_1 W_1 \cdots)))). \tag{4}$$

Probable Guarantees We now present our formal guarantees for 2 and 3 layers that the feedforward net inverts the generative model, i.e., Property (b). We assume the random-like matrices W_j's to have standard gaussian prior:

$$W_j \text{ has i.i.d entries from } \mathcal{N}(0,1) \tag{5}$$

We also assume that the distribution D_ℓ produces k_ℓ-sparse vectors with not too large entries almost surely:

$$h^{(\ell)} \in \mathbb{R}^{n_\ell}_{\geq 0}\,,\ |h^{(\ell)}|_0 \leq k_\ell, \quad \text{and} \quad |h^{(\ell)}|_\infty \leq O\left(\sqrt{\log N/(k_\ell)}\right)\|h\| \quad \text{a.s.} \tag{6}$$

where $N \triangleq \sum_j n_j$ is the total number of nodes in the architecture. Under this mathematical setup, we prove the following reversibility and dropout robustness results for 2-layer networks.

Theorem 2.1 (2-Layer Reversibility and Dropout Robustness)**.** *For $\ell = 2$, and $k_2 < k_1 < k_0 < k_2^2$, for 0.9 measure of the weights (W_0, W_1), the following holds: There exists constant offset vector b_0, b_1 such that when $h^{(2)} \sim D_2$ and $\Pr[x \mid h^{(2)}]$ is specified as model* (4)*, then network has reversibility and dropout robustness in the sense that the feedforward calculation (defined in Figure 2(b)) gives $\tilde{h}^{(2)}$ satisfying*

$$\forall i \in [n_2], \quad \mathbb{E}\left[|\tilde{h}_i^{(2)} - h_i^{(2)}|^2\right] \leq \epsilon\tau^2 \tag{7}$$

where $\tau = \frac{1}{k_2}\sum_i h_i^{(2)}$ *is the average of the non-zero entries of* $h^{(2)}$ *and* $\epsilon = \widetilde{O}(k_2/k_1)$.

To parse the theorem, we note that when $k_2 \ll k_1$ and $k_1 \ll k_0$, in expectation, the entry-wise difference between $\tilde{h}^{(2)}$ and $h^{(2)}$ is dominated by the average single strength of $h^{(2)}$. We note that the magnitude of the error in the theorem is on the order of the ratio of the sparsities between two layers.

3 SUMMARY OF OTHER RESULTS

Theorem 2.1 is extended to three layers with stronger assumptions on the sparsity of the top layer – We assume additionally that $\sqrt{k_3}k_2 < k_0$, which says that the top two layer is significantly sparser than the bottom layer k_0. We note that this assumption is still reasonable since in most of practical situations the top layer consists of the labels and therefore is indeed much sparser.

Theorem 3.1 (3-layers Reversibility and Dropout Robustness, informally stated)**.** *For* $\ell = 3$*, when* $k_3 < k_2 < k_1 < k_0 < k_2^2$ *and* $\sqrt{k_3}k_2 < k_0$*, the 3-layer generative model has the same type of reversibility and dropout robustness properties as in Theorem 2.1.*

We also present experimental results that support our theory. First, the random-like nets hypothesis was verified on the fully connected layers in a few trained networks, such as those in AlexNet and also those in multilayer nets that we trained on different data sets. Edge weights fit a Gaussian distribution, and bias in the ReLU gates are essentially constant (in accord with the theorems) and the distribution of the singular values of the weight matrix is close to the quarter circular law of random Gaussian matrices.

Second, the knowledge that the deep net being sought is random-like can be used to improve training. Namely, take a labeled data point x, and use the current feedforward net to compute its label z. Now use the shadow distribution $p(x|z)$ to compute a *synthetic* data point $\tilde{x}$, label it with z, and add it to the training set for the next iteration. Experiments show that adding this to training yields measurable improvements over backpropagation + dropout for training fully connected layers. Furthermore, throughout training, the prediction error on synthetic data closely tracks that on the real data, as predicted by the theory.

REFERENCES

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, pp. 153–160, 2006.

Yoshua Bengio, Eric Thibodeau-Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. *arXiv preprint arXiv:1306.1091*, 2013a.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pp. 899–907, 2013b.

Yoav Freund and David Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. Technical report, 1994.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 841–848, 2001.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pp. 1096–1103, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.