Studies in Deep Belief Networks

Jiquan Ngiam
jngiam@cs.stanford.edu

Chris Baldassano
chrisb33@cs.stanford.edu

Abstract

Deep networks are able to learn good representations of unlabelled data via a greedy layer-wise approach to training. One challenge arises in choosing the layer types to use, whether it is an autoencoder, restricted boltzmann machine, with and without sparsity regularization. The layer choice directly affects the type of representations learned. In this paper, we examine sparse autoencoders and characterize their behavior under different parameterizations. We also present preliminary results on quadratic layers with slowness.

1 Introduction

Recent work [2, 6, 10] in Deep Networks have shown that good representations can be learned from unlabelled data with a greedy layer-wise approach to training. In particular, when one adds more layers to a deep network, the deeper layers often learn invariant higher order representations [5, 10]. While autoencoders and restricted boltzmann machines have been popular choices to use as layers, [12, 9] have proposed using sparsity regularization to find better representations. [3, 4] also found that quadratic features with slowness is able to discover good, invariant representations.

In this paper, we examine sparse autoencoders and attempt to characterize the behavior of the model under various parameterizations. One challenge in these models (and unsupervised learning in general) is model evaluation. While prior work have used neuroscience [9] or classification [10] as a means of evaluation, we seek to find a good objective measure to evaluate the model without resorting to fitting neuroscience data or building huge object classification models. This is similar in spirit to related work on measuring invariances [5], and we compare our results against their metrics. We also experiment with the quadratic layer model with slowness.

2 Model

The sparse autoencoder model is essentially a neural network with one hidden layer that is optimized to reconstruct its input. Formally, we can define the model with the following objective function.

$$\begin{aligned} min_{W,W',b,b'} \sum_{x_i} \|x_i - tanh(W' z_i + b')\|_2^2 + \lambda \|E[z] - \rho\|_2^2 + \xi \|W\|_2^2 + \xi \|W'\|_2^2 \\ \text{s.t. } z_i = tanh(W x_i + b) \end{aligned} \tag{1}$$

We perform stochastic gradient descent (backprop) to train the network, except for the sparsity objective. Sparsity of the network is controlled by $\lambda \|E[z] - \rho\|_2^2$ and we use the method suggested in [9] to optimize for the sparsity objective. The method only adjusts the bias term, b without adjusting W. In the rest of the paper, we refer to ρ as target-activation and ξ as weight-decay. These two parameters are the most important for the model. In our natural image experiments, we use $\lambda = 10$ and a learning rate of 0.003, and in our MNIST experiments, we use $\lambda = 1$ and a learning rate of 0.001.

3 Measures

To evaluate the model, we consider the following measures: Reconstruction Error, Hidden Unit Kurtosis, Input Representation Kurtosis, Invariance Measures [5] and Classification Error.

We validate the measures by first visually inspecting the learned bases of the models. For natural images, we expect the model to learn localized gabor filters [9]. For MNIST dataset of handwritten digits [8], good bases are characterized by penstrokes [12]. We also evaluate the MNIST models on classification error.

4 Active Cells

In the process of evaluating the sparse autoencoder model, we often found that the model would sometimes learn to use only a subset of the available hidden units. We hypothesize that the sparsity constraint and weight decay acts to regularize the model so that it automatically selects the number of hidden units to use.

However, the inactive cells (which are always turned off, or exactly at the expected activation) often cause evaluation measures to be inaccurate. [1] In particular, if there are many inactive cells, then the input representation $(z_i|x_i)$ would have an extremely high kurtosis, since it is very sharply peaked at zero. Hence, in order to get reasonable values for the measures, we lesion these inactive cells from the network before computing the metrics. This is performed for the hidden unit kurtosis and input kurtosis.

To lesion inactive cells, we compute and threshold the variance of the pre-sigmoidal value for the cell (i.e. $Var(W_i x)$). The variance of inactive cells are almost always strikingly separate from the active cells. We also note that one can select the inactive cells by looking at other measures such as $\|W_i\|_2^2$ or the coefficient of variation[2] on the post-tanh values.

5 Experiments

5.1 Natural Images

We used a set of natural images from [7] to train the model. We varied the target activation parameter roughly linearly from 0.005 to 0.250. We found that networks with $\rho < 0.005$ were always degenerate (with extremely bad reconstruction error), while networks with $\rho > 0.250$ always failed to learn the localized gabors we sought to find. Weight-decay was varied from 0.0001 to 1.000 in a log-scale. We chose a hidden layer of 400 units after finding it to be reasonable. Networks were trained to $10million$ iterations to ensure that they have converged.

5.2 MNIST Dataset

We also trained networks on the MNIST handwritten digit dataset. The target activation parameter was varied from 0.20 to 0.50. Networks with $\rho < 0.20$ were usually very degenerate and had bad reconstruction error. Interestingly, we tend to observe pen-strokes when $\rho \approx 0.25$, much higher than that for natural images. Weight-decay was varied from 0.001 to 0.0100 in a log-scale. We chose a hidden size of 100 units by comparing preliminary results for 100, 200 and 400 units.

Supervised training was peformed by adding a new layer connected to the hidden layer. This layer had 10 units and was trained to learn the digits in a supervised fashion using the pre-trained hidden layer. The network was first trained unsupervised for $7.5million$ iterations, followed by $5million$ iterations of supervised training.

[1] We hypothesize that [5] had to use the top-scoring proportion p of hidden units for their metrics because of the effect of the inactive cells

[2] The coefficient of variation is defined as the ratio of the standard-deviation to the mean. It is a normalized measure of dispersion.

6 Findings

6.1 Natural Images — Active Input Kurtosis

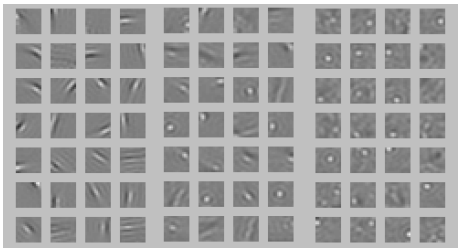

Figure 1: Bases on natural images as target-activation is increased. Left shows the best bases learned by the model. Right shows center-surround point detectors as target-sparsity is set to a very high value. Middle shows that the transition between models is smooth.

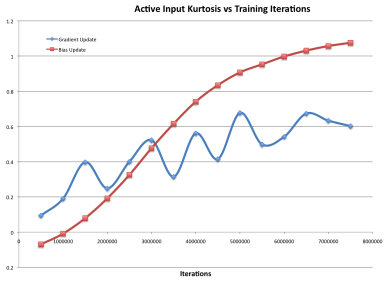

Figure 2: Active Input Kurtosis as a function of training iterations. Note that reconstruction error converges after $500k$ iterations. Blue (curvy) line shows the model trained with the full-gradient for the sparsity objective. Red line shows using just the bias update.

In our experiments with natural images, we mainly evaluated the model by hand through visualizations of the bases. By considering only the kurtosis on the active units, we found that this measure very nicely represented how good the bases were. In particular, the active kurtosis score was very well correlated with how localized and gabor-like the bases were. For space considerations, we do not show the graphs for the other measures (recontruction error, unit kurtosis, etc.) The other measures generally do not correlate well with the how good the bases were (even after considering only active units).

A particularly striking graph is Figure 2, which shows how the kurtosis measure changes as the network progresses in training. Note that reconstruction error converges at $0.5million$ iterations, at the left most point of the graph. The graph shows that although the network might already be at a local minima for the objective function (1), it is still evolving to learn better, more kurtotic representations. In preliminary experiments using the full gradient update (instead of just the bias update), we find that the model still generally converges to good representations, but in a different manner (blue line in Figure 2).

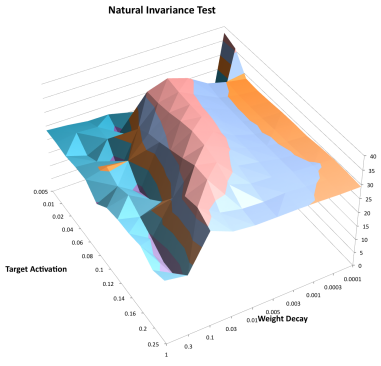

Figure 3: Invariance test on natural videos as a function of target-activation and weight-decay.

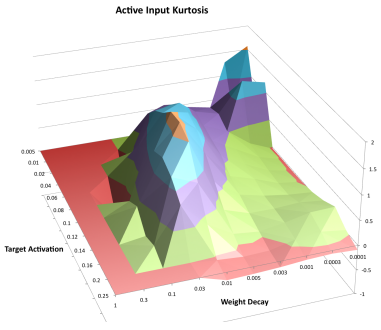

Figure 4: Active Input Kurtosis as a function of target-activation and weight-decay.

The left-most bases in Figure 1 were chosen by selecting the top performing network according to Figure 4. By visualizing the networks, we find that Figure 4 describes the space of networks very well. The best network is found with a target-activation of 0.10, which differs from the usual intuition that good values for target-activation are between 0.02 and 0.05. However, it remains to see if these networks ultimately are better at some more objective task such as classification.

While evaluating on the measures proposed by [5], we found that their measure correlates well with ours. Surprisingly, while the two measure were designed independently, the model parameters which maximizes both measures are exactly the same at $\rho = 0.10, \xi = 0.01$. However, the surfaces of both measures are different and this is a subject for further study. We also note that both measures have problems with outlier models ($\rho < 0.005, \xi < 0.0001$). We hypothesize that the problems are due to active v.s. inactive cells.

6.2 MNIST — Supervised Error

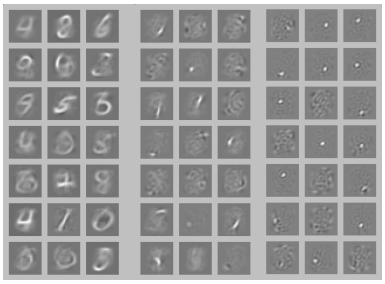

Figure 5: Bases on handwritten digit images as target-activation is increased. At low activation values, digit memorizers are obtained. At mid-level activations, pen-strokes are obtained. At high activations, we see point like features

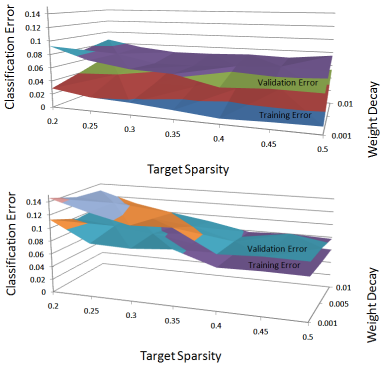

Figure 6: Error on Training and Validation Sets on MNIST. Top: Unlocked Hidden Weights, Bottom: Locked Hidden Weights

After pre-training, we consider adding supervised training in three ways:

1. Update the hidden layer with the supervised layer, discarding the unsupervised criterion.
2. Lock the hidden layer while training the supervised layer.
3. Lock the hidden layer during initial training of the supervised layer, then after $2million$ iterations allow both layers to be updated.

In all cases, supervised training was conducted for $5million$ more iterations after unsupervised training. In case (1) where both hidden and output layers were updated, we found that the weights learned during unsupervised training are mostly discarded. Many of the learned bases are always turned off, and the rest are not easily interpreted. Despite their random appearance, these bases actually give supervised error rates comparable to the best two-layer neural networks on the MNIST website.

In case (2), where the unsupervised layer was locked before the supervised training, we found that the network was still able to obtain reasonable error rates. Training error was much higher in this case and validation error increased, but the gap between training and validation error was much smaller, as in Figure 6. This finding correlates well with the idea that unsupervised pre-training can be viewed as a regularization [1].

One might hypothesize that the learned bases get discarded in case (1) due to the random initialization of the supervised layer. Hence, we experiment with a hybrid model in case (3) where the hiddens weights are allowed to be modified only after $2million$ iterations of supervised training. This case turned out to converge to ones similar to case (1).

7 Quadratic Expansion

We performed preliminary experiments with quadratic expansion layers. Following [4], we used PCA to perform dimension reduction of the input to 50 dimensions, then did a full quadratic basis expansion and trained a sparse autoencoder on top of this. Preliminary findings (left out due to space considerations) show that sparsity gives rise to more structure in the bases, particularly for inputs which minimize the response of the bases.

8 Discussion

While Active Input Kurtosis worked very well for natural images, preliminary results show that it does not work very well for the MNIST dataset. In particular, consider the representation such that each hidden unit memorizes one digit - such a representation will have very high input kurtosis. However, such a representation will not have good reconstruction error. Hence, (active) kurtosis alone is clearly not enough as a measure.

An interesting point to note is that while input kurtosis was very informative, the kurtosis of each individual unit was not informative. Many prior studies have placed emphasis in the unit kurtosis, as they would usually pick out a particular feature (one filter) and observe its distribution. In this work, we show that the opposing point of view is equally important - that the kurtosis of the **input representation** is (more) important. These ideas relate closely to coefficient entropy [11], and a comparison is left as future work. However, the analysis here is limited by the fact that quality of each learned model is evaluated subjectively by a human.

One point that we would like to highlight is the importance of selecting only active cells. In the analysis of any model which transforms the original input into a different encoding, one should consider whether each dimension in the new encoding is informative at all. Perhaps this is particularly important here since the model is able to automatically select the number of units to use.

Finally, our MNIST experiments show that there are unexplored complexities in applying sparse autoencoders to supervised learning problems. Since the bases changed drastically during supervised training (when unlocked), it seems that the naive method of simply adding on supervised training is insufficient, as it discards the benefits gained by pre-training.

9 Future Work

We pose the following open questions for future work:

9.1 Unsupervised Training

- Is there an objective way to evaluate a model learned in an unsupervised fashion ?
- What is the difference between maximizing kurtosis for the input representation and the hidden unit ?
- Is maximizing kurtosis or doing ICA the ideal model ?
- Is there a better alternative to kurtosis ?
- How do the measures compare to coefficient entropy [11] ?
- What does it mean for the current model to automatically select a number of active units, why doesn't it try to use all the available units ?
- How does the bias-update only rule differ from the full-gradient (see Figure 2) ?
- Can we design scores (for natural images) based on how (a) localized and (b) gabor like the bases are ?
- How would the results change with a Sparse RBM ?

9.2 Supervised Training

- Why does supervised training (as opposed to overall fine-tuning) discard the hidden layer representations ?
- How should supervised training be conducted? Should the hidden layer weights be locked or their learning rates lowered ?
- Should sparsity continue to be enforced while supervised fine tuning happens ?
- Can we show theoertical connections between good input representations and performance of supervised classifiers ?

Acknowledgments

We thank Professor Andrew Ng for his insightful advice and Quoc Le, Andrew Saxe, Andrew Maas, and Dan O'Shea for many helpful discussions.

References

[1] Yoshua Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, Université De Montréal, and Montréal Québec. Greedy layer-wise training of deep networks. In *In NIPS*, 2007.

[3] Pietro Berkes and Laurenz Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):579–602, July 2005.

[4] Pietro Berkes and Laurenz Wiskott. On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural Computation*, 18(8):1868–1895, 2006.

[5] Ian Goodfellow, Quoc Le, Andrew Saxe, and Andrew Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems 21*, Cambridge, MA, 2009. MIT Press.

[6] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[7] A. Hyvärinen and P. O. Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423, 2001.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, August 2002.

[9] Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area v2. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 873–880, Cambridge, MA, 2008. MIT Press.

[10] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the Twenth-Sixth International Conference on Machine Learning*, 2009.

[11] Michael S. Lewicki, Terrence J. Sejnowski, and Howard Hughes. Learning overcomplete representations. *Neural Computation*, 12:337–365, 1998.

[12] Marc'Aurelio Ranzato, Y-Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. In *NIPS*, 2007.