Properties of Autoencoders

Justin Johnson
jcjohns@stanford.edu

Bharath Ramsundar
rbharath@stanford.edu

1 Introduction

In recent years a variety of deep learning algorithms, including deep belief networks [2, 6] and deep neural networks, both convolutional [3] and non-convolutional [4]. The types of networks that are typically used in these applications are very complicated, and studying their theoretical properties is very difficult.

Some approaches have utilized unsupervised pretraining of deep neural networks in order to improve performance on classification tasks [4]. One of the fundamental building blocks of this unsupervised pretraining process is the sparse autoencoder. A single-layer autoencoder is a much simpler object than an entire deep network, but even this relatively simple object has not been well-studied theoretically.

In this project we aim to explore different varieties of autoencoders and to try and understand why they work.

2 Sparse Autoencoder

A sparse autoencoder is a neural network with a single hidden layer that attempts to learn the identity function. The transfer function in these networks is nonlinear; in our experiments we use the sigmoid transfer function $f(z) = 1/(1 + e^{-z})$.

Neural net weights are learned by minimizing an objective function consisting of three terms. The first term is the ℓ^2 reconstruction error of the training data. The second term is ℓ^2 regularization of the weight vectors. The third term constrains the mean activation of each hidden layer neuron over the training set. The sparsity constraint can take many forms, but our initial implementation uses a penalty of the form

$$\sum_{j=1}^{p} \left(\rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \right)$$

where the hidden layer contains p neurons, $\hat{\rho}_j$ is the mean activation of the jth hidden layer neuron over the training set, and ρ is a constant controlling the desired sparsity. The weights of the trained network can be viewed as a feature transform for the training data. We implemented this algorithm and ran it on the MNIST dataset [5]; the learned feature transform is shown in Figure 1. Qualitatively, the learned features roughly correspond to local object parts.

3 Equivalence with PCA

An autoencoder that does not include the regularization or sparsity terms learns a feature representation that is closely related to the principal components of the training data [1]. We implemented this type of autoencoder on the MNIST dataset and also ran principal component analysis on the same dataset; the learned features can be seen in Figure 1. The learned features appear to be qualitatively similar.

This equivalence between "vanilla" autoencoders and principal component analysis suggests that the recent success of autoencoders is due to the additional constraints placed upon their parameters. This motivates a more in-depth analysis of the sparsity constraint.

4 Sparse Linear Autoencoders

We next considered a sparse linearized autoencoder where the transfer function is simply the identity function. The objective function for this autoencoder is the sum of the ℓ^2 reconstruction error, an ℓ^2 regularization term on the network weights, and an ℓ^2 sparsity constraint of the form $\|\rho - \hat{\rho}\|_2^2$ where as above $\hat{\rho}_j$ is the mean activation of the jth hidden unit on the training set. The learned features for this autoencoder is shown in Figure 1.

5 Discussion

Our experimentation so far has led us to focus on the sparsity constraint of the autoencoder. For the remainder of the term we will consider other variants on this constraint both on nonlinear and linearized networks. In particular, we will experiment with ℓ^1 and ℓ^2 sparsity constraints on the nonlinear network and an ℓ^1 sparsity constraint on the linearized network. Depending on the results of these experiments, we will attempt to understand the theoretical implications of these sorts of constraints. In addition to the linearized autoencoder, we would also like to experiment with networks whose transfer function is a Taylor approximation to the sigmoid function.

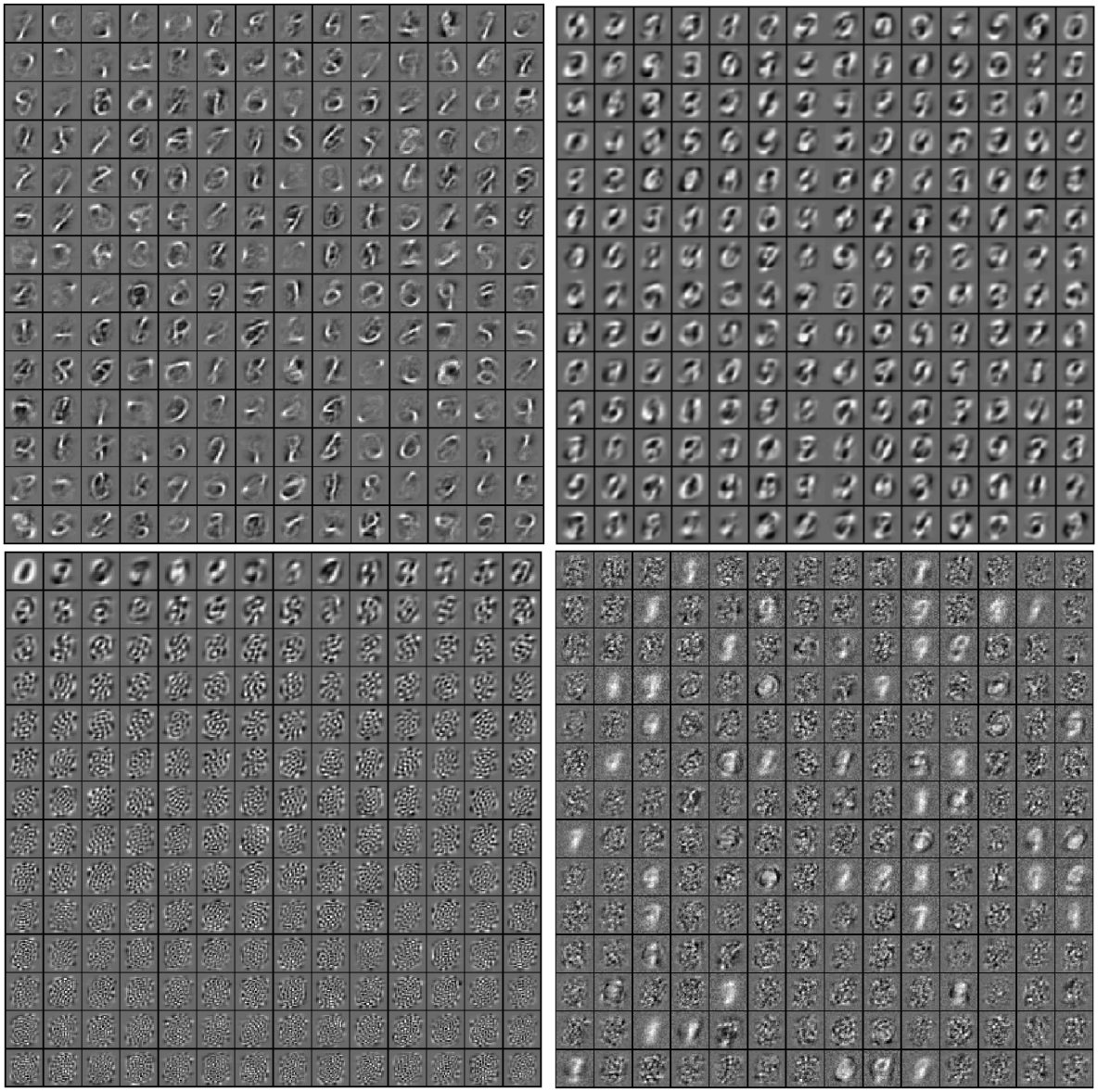

Figure 1: Learned features for MNIST using different methods. Upper left: Sparse (nonlinear) autoencoder. Upper right: Sparse linear autoencoder. Lower left: Principal component analysis. Lower right: nonlinear autoencoder.

References

[1] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

[2] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[4] Quoc V Le, Rajat Monga, Matthieu Devin, Greg Corrado, Kai Chen, Marc'Aurelio Ranzato, Jeff Dean, and Andrew Y Ng. Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112.6209*, 2011.

[5] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits, 1998.

[6] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.