Reducing Dimensionality of Data with Neural Networks

Ayushman Singh Sisodiya - 12188

March 16, 2015

1 Problem and Motivation

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. I will do a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network to transform the high-dimensional data into a low-dimensional code and a similar "decoder" network to recover the data from the code

2 Data Set

This is one of the most popular datasets in image processing and hand-written digit classification tasks. It consists of 70,000 binary images of size 28 28 having 10 different classes each corresponding to a digit from 0-9. The complete data is split into 3 sets - training set having 50,000 images, validation set having 10,000 images and test set having 10,000 images. The training set is used for training the networks while the validation set is used to evaluate the performance of the trained network and to decide the stopping criterion. The performance is reported by applying the best validation model on the test set. Henceforth, I mention this split in the form of a tuple like (50,10,10). Sample image from the MNIST database are given below

3 Challenges and Methodology

As I will be using variant of gradient descent to train one of the main challenge is to overcome the popularly known vanishing gradient descent problem which is prominent in deep neural nets. Once the errors get backpropagated to the first few layers, they are minuscule, and quite ineffectual. This causes the network to almost always learn to reconstruct the average of all the training data. A pretraining technique developed by Geoffrey Hinton[1] for training many-layered "deep" auto-encoders involves treating each neighboring set of two layers like a restricted Boltzmann machine for pre-training to approximate a good solution and then using a backpropagation technique to fine-tune.

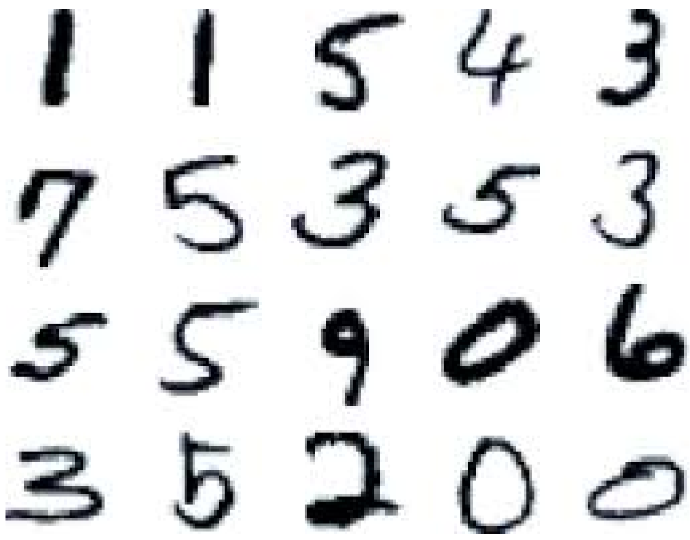

Figure 1: Caption

My main implementation will revolve around the paper "Reconstruction and recognition of face and digit images using autoencoders" by Tan [2]. I will also try to implement the stacked autoencoder[3] and denoising[4][5] it and will try to evaluate the performance.

References

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

[2] Chun Chet Tan and C Eswaran. Reconstruction and recognition of face and digit images using autoencoders. *Neural Computing and Applications*, 19(7):1069–1079, 2010.

[3] Telmo Amaral, J Sá, L Silva, Luís A Alexandre, and J Santos. Improving performance on problems with few labelled data by reusing stacked auto-encoders, 2014.

[4] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[5] 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.