DIMENSIONALITY REDUCTION USING NEURAL NETWORKS

MOHAMMAD NAYEEM TELI
Department of Computer Science
Colorado State University
Fort Collins, CO

ABSTRACT
A multi-layer neural network with multiple hidden layers was trained as an autoencoder using steepest descent, scaled conjugate gradient and alopex algorithms. These algorithms were used in different combinations with steepest descent and alopex used as pretraining algorithms followed by training using scaled conjugate gradient. All the algorithms were also used to train the autoencoders without any pretraining. Three datasets: USPS digits, MNIST digits, and Olivetti faces were used for training. The results were compared with those of Hinton et al. (Hinton and Salakhutdinov, 2006) for MNIST and Olivetti face dataset. Results indicate that while we were able to prove that pretraining is important for obtaining good results, the pretraining approach used by Hinton et al. obtains lower RMSE than other methods. However, scaled conjugate gradient turned out to be the fastest, computationally.

1. INTRODUCTION

Dimensionality reduction is a method of obtaining the information from a high dimensional feature space using fewer intrinsic dimensions. Reducing dimensionality of high dimensional data is good for better classification, regression, presentation and visualization of data.

Recently Hinton et al. (Hinton and Salakhutdinov, 2006) used a deep autoencoder for dimensionality reduction of multiple datasets. The autoencoders are multi-layer identity mapping neural networks represented by a function $f(x) = x$, where x is a multidimensional input vector to the network. They argue that deep autoencoders could be easily trained using a gradient descent method provided the initial weights are near good solutions. They claimed that by pretraining, they were able to obtain a good set of initial weights and the fine tuning which followed the pretraining approach was able to reduce the data dimensionality very efficiently. Their results support their arguments, however, there still remain some areas which require additional studies.

Firstly, they reported deep autoencoders showed significant improvement when pretrained over the ones without pretraining (see supporting material of (Hinton and Salakhutdinov, 2006) for details). However, they studied conjugate gradient algorithm with line search for fine tuning and their results cannot be extended to other gradient based methods which do not use line search. Secondly, their pretraining approach is very complicated. It assigns a probability to every possible image via an energy function. We thought it would be pertinent to pretrain a deep autoencoder with a less complicated approach. Thirdly, one of the problems that has been identified with training multi-layer neural networks using gradient based algorithms, is the problem of local minima. Keeping this in view, we thought it would be important to train our network with a non-gradient based algorithm.

We want to further investigate some of the aspects of the dimensionality reduction using neural networks that were not explored fully by Hinton et al. In particular we want

to focus on the pretraining and weight initialization of multi-layer neural networks. We hypothesize that, gradient based algorithms can train a deep autoencoder and obtain competitive results without pretraining; secondly, simpler pretraining approaches followed by fine tuning using a gradient descent method can obtain competitive results; and finally, non-gradient based algorithm can train a deep autoencoder with competitive results. In these hypotheses, competitive results would mean that the RMSE values between the input and the reconstructed datasets are lower or equal to those obtained by Hinton et al. (Hinton and Salakhutdinov, 2006).

In order to investigate our first hypothesis, we will train our deep autoencoders using scaled conjugate gradient (Moller, 1993) and steepest descent methods without pretraining. The second hypothesis will be verified by, firstly, pretraining layer by layer, using steepest descent followed by fine tuning using scaled conjugate gradient. Secondly, repeat fine tuning phase by replacing steepest descent in the pretraining phase with the correlation based non gradient algorithm, alopex (Unnikrishnan and Venugopal, 1994). To test our third hypothesis, the multi-layer neural network will be trained using alopex without pretraining.

2. BACKGROUND

Over the years many algorithms have been suggested to train autoencoders for efficient dimensionality reduction. DeMers et al. (DeMers and Cottrell, 1993), Nielsen (Nielsen, 1995), and Kambhatla et al. (kambhatla and Leen, 1997) independently observed that optimizing weights in non-linear autoencoders that have multiple layers is a very difficult task because of the choice of initial weights.

Recently Hinton et al. came up with a very efficient approach for dimensionality reduction. They argue that a gradient descent method could be used to train deep autoencoders very efficiently, if our initial set of weights were close to a good solution. They reason that autoencoders would find poor local minima with large initial weights and the gradients in the early layers would be tiny with small initial weights. This would make it infeasible to train autoencoders with many hidden layers (Hinton and Salakhutdinov, 2006). In order to overcome this, they suggested that we should pretrain our deep autoencoders first, so that we initialize the weights close to a good solution and then we could use a gradient descent method to fine tune the network. They pretrained their network using Restricted Boltzmann Machines (RBM). An RBM is a two layer Boltzmann machine without any connections between the hidden units. A Boltzmann machine is a stochastic recurrent neural network with simulated annealing dynamics. In an RBM the first layer is the visible layer, which is observable, and the second layer corresponding to the hidden layer is used for feature detection. The features learned from one RBM act as an input to train the next RBM in the group.

2.1 ALTERNATIVE ALGORITHMS

We will be using three different algorithms to address the hypotheses: steepest descent, scaled conjugate gradient and alopex. The motivation for using these algorithms is three fold: firstly, we wanted to use a very simple approach for training a deep neural network, steepest descent is one of the simplest gradient based algorithms. Secondly, they used a conjugate gradient method for fine tuning. We wanted to formulate a method which would be similar to the one used by them but at the same time we did not want to replicate the results already produced. Therefore, scaled conjugate gradient appeared to be a good alternative because it does not use the line search method unlike other conjugate gradient algorithms. Thirdly, we had to use a non-gradient based algorithm, we chose to use alopex algorithm. It uses error correlations which would avoid poor local

optimum selection. The reason behind using it are two fold: firstly, it would help us know the impact of using gradient information, and secondly, this could not only be an algorithm for fine tuning but we could also see its effect as a pretraining algorithm in combination with a gradient based fine tuning method.

3. RESULTS

The main performance metric of our experiments is the error between the neural network input data and the reconstructed data at the output of the network. We also focus on the CPU time taken by the methods for MNIST dataset for which we were able to replicate Hinton et al.'s results. Each dataset was normalized in all our experiments.

3.1 USPS DATASET

USPS (Hastie et al., 2001) is a hand written zip code dataset. It has 256 features of values 0-255(the brightness value of 16x16 gray-scale image of the character). There were 7291 training samples and 2007 test samples. An eight layer autoencoder represented as, 256-1000-500-250-30-250-500-1000-256, was used for our experiment. This network was chosen based on pilot experiments.

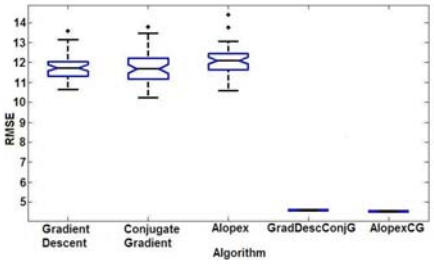

Figure 1: A comparison of the RMSE values for USPS dataset using algorithms with and without pretraining.

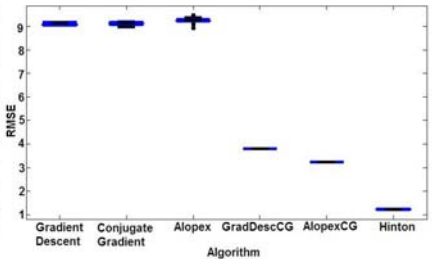

Figure 2: Comparison of test RMSE when different algorithms were used to train an 8 layer autoencoder for MNIST dataset.

A boxplot result of different combinations of the methods is presented in Fig. 1 for comparison. GradDescConjG algorithm in Fig. 1 represents pretraining using steepest descent followed by training using scaled conjugate gradient. AlopexCG is characterized by pretraining using alopex followed by scaled conjugate gradient training. This algorithm did slightly better than the one where we use steepest descent pretraining and much better than the other algorithms where no pretraining is being used. The variations in the results of the algorithms without pretraining are due to the multiple runs and the random initialization of the weights.

3.2 MNIST DATASET

MNIST (Roweis, 2007a) is a subset of the NIST digits dataset with 60000 training samples and 10000 test samples. The dimensionality of the dataset is 784. We used the same dataset and the same network, 784-1000-500-250-30-250-500-1000-784, that was used by Hinton et al. We not only ran our experiments with and without pretraining, but also reproduced results using Hinton et al.'s source code.

In order to visualize how our experiments did in comparison to each other and Hinton et al.'s approach, a boxplot is presented in Fig. 2. GradDescCG algorithm is a combination of steepest descent pretraining and conjugate gradient training. AlopexCG is alopex pretraining and conjugate gradient training. A visual comparison of Fig. 2 indicates Hinton et al.'s approach has produced the best results. Against an RMSE of 1.22 obtained using Hinton et al.'s approach we could get as close as 3.22 using conjugate gradient to train a network whose initial weights were set by alopex algorithm. However, this is not a big difference since it is over 784 intensity values. Again, the variations in the results of the algorithms without pretraining are due to the multiple runs and the random initialization of the weights.

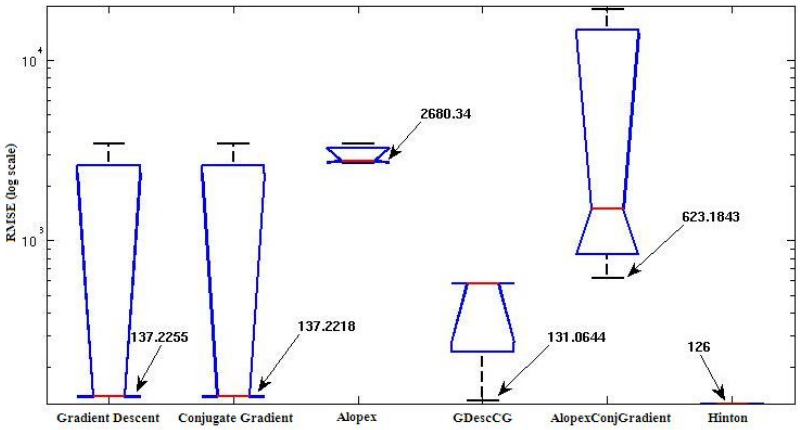

Figure 3: Comparison of test RMSE when different algorithms were used to train an autoencoder for Olivetti face dataset. GDescCG is steepest descent pretraining with scaled SCG training. AlopexConjGradient is an algorithm with alopex pretraining and SCG training.

3.3 OLIVETTI FACE DATASET

Olivetti (Roweis, 2007b) dataset was also the same as the one used by Hinton et al. It contains 10 64x64 images of each of forty different people. The authors' code was not accessible so we had to compare our results to the ones reported by Hinton et al. This dataset required some preprocessing as was done by Hinton et al. We constructed a dataset of 166000, 25x25 images from the original 400 images, by rotating (-90 to +90), scaling (1.4 to 1.8), cropping and subsampling. The rotation and scaling ranges were the same as the ones reported by Hinton et al. (Hinton and Salakhutdinov, 2006). The dataset was further divided into 41400 test images and 124600 training images. A network, 625-2000-1000-500-30-500-1000-2000-625 used by Hinton et al., was used to train this dataset. The results were obtained using algorithms without pretraining and with pretraining by steepest descent and alopex, followed by training using scaled conjugate gradient. These results are presented in Fig. 3 along with a least RMSE value reported by Hinton et al. (Hinton and Salakhutdinov, 2006). We found the least RMSE of 131.0644 against Hinton's 126. This is a very small difference over 625 intensity values. In Fig. 3 the numbers at the tails of the arrows indicate the lowest RMSE value obtained using an algorithm.

4. STATISTICAL ANALYSIS

We carried out pairwise t-tests between the different algorithms for each dataset. We used multiple comparison t-tests with pooled standard deviation and Holms method for p-value adjustment. A summary of these tests is presented in Table 1. The p-value reported is the least we could obtain using the corresponding algorithm whose difference is statistically significant than all other algorithms. For USPS dataset, steepest descent with conjugate gradient has statistically significant difference to other methods without pretraining, than alopex as a pretraining method. The two methods with pretraining for this dataset were statistically similar. For Olivetti, since we had only one value for Hinton et al. results (reported in (Hinton and Salakhutdinov, 2006)) we could not carry out any t-tests using their results. Within other methods there was no statistically significant difference between any of the approaches. For this dataset the lowest p-value is reported.

Table 1: Pairwise t-test results showing the lowest p-value of various algorithms for different datasets.

Dataset	Algorithm	p-value
USPS	Steepest Descent with Conjugate Gradient	2e-16
MNIST	Hinton	2e-16
OLIVETTI	Steepest Descent with Conjugate Gradient	0.39

We found that conjugate gradient was the fastest algorithm. We saved the CPU time using cputime command of Matlab used by each algorithm for each dataset. However, since we could compare our results with Hinton et al.'s code for only MNIST datasets, so we carried out pairwise t-test on that dataset alone. We found that conjugate gradient was significantly faster than the rest of the algorithms including Hinton et al.'s method with a p-value of 2e-16.

5. DISCUSSION

We started with three hypotheses which could be summarized as: gradient based algorithms like steepest descent and scaled conjugate gradient and non gradient based algorithm like alopex could produce competitive results in comparison to the approach used by Hinton et al. We also hypothesized that a simpler pretraining approach than RBM, using steepest descent or alopex followed by training using scaled conjugate gradient algorithm would get us competitive results when compared with Hinton et al.

Our results indicate some interesting findings. Firstly, Hinton et al.'s argument that pretraining followed by training would get better results holds true. A closer look at Fig.'s 1, 2, and 3 indicate that training followed by pretraining does indeed improve the results. Secondly, pretraining helps to find a good set of initial weights and the training that follows it tries to fine tune those weights in order to obtain better results. In Fig.'s 1 and 2, we can notice that the algorithms without pretraining have a variation around the median RMSE values while as the RMSE values obtained using pretraining and training do not show any such variation. This is because the training phase only tries to fine tune the weights such that it restricts itself to a particular search area so that it can find the global optimum.

Alopex by itself did not perform well on Olivetti dataset unlike gradient descent and scaled conjugate gradient. However, alopex performed equally well as the other methods for other two datasets without pretraining. It performed better than steepest descent when

used as a pretraining algorithm for MNIST dataset, shown in Fig. 2. This leads us to an important observation. It seems that weight optimization in deep autoencoders is not entirely related to poor local optima while using gradient based algorithms.

An important finding of this research is that we were able to come up with alternative pretraining approaches, which found a low RMSE value for Olivetti face dataset, very close to Hinton et al.'s approach. We also came very close to Hinton et al.'s results for MNIST dataset. These results were achieved because steepest descent as well as alopex did well as pretraining methods. Of these two pretraining approaches, statistical results indicate steepest descent to be slightly better than alopex.

Another important result is that computationally, scaled conjugate gradient is the fastest approach. This points to a possibility of obtaining a more robust and faster training of deep autoencoders using a scaled conjugate gradient based approach with different combinations of steepest descent and alopex as pretraining algorithms. One of the possibilities would be to explore the effect of varying the number of units in the hidden layers. Another possible approach would be to train the autoencoder using Hinton et al.'s pretraining approach in combination with scaled conjugate gradient training algorithm. We could also try reversing the order of our pretraining and training algorithms.

6. CONCLUSIONS

In this research three different algorithms in different combinations for both pretraining and training were used to train deep autoencoders for three different datasets. One of the main intrinsic goals regarding pretraining was proved. Pretraining does indeed help to find a good set of initial weights and thereby help find a good solution using a gradient descent based method in a deep autoencoder. We were able to prove that other pretraining methods could perform well. However, further tuning of these other methods as pretraining and training algorithms might obtain much better results. The performance of alopex pointed towards the possibility that deep autoencoders might not be prone to local minima as we had initially thought. A more careful analysis using these algorithms could lead to more insights into training deep autoencoders and a better understanding of the weight optimization in such networks.

REFERENCES

Hinton, G, and Salakhutdinov, R., 2006, "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, pp. 504–507.

Møller, M. F., 1993, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, Vol. 6, pp. 525–533.

Unnikrishnan, K. P. , and Venugopal, K. P., 1994, "Alopex: A correlation-based learning algorithm for feedforward and recurrent neural networks," *Neural Computation*, Vol. 6, pp. 469–490.

DeMers, D., and Cottrell, G., 1993, "Non-linear dimensionality reduction," *Advances in Neural Information Processing Systems*, Morgan Kaufmann, San Mateo, CA, Vol. 5, pp. 580–587.

Hecht-Nielsen, R., 1995, "Replicator neural networks for universal optimization source coding," *Science*, Vol. 269, pp. 1860–1863.

Kambhatla, N., and Leen, T. K., 1997, "Dimension reduction by local principal component analysis," *Neural Computation*, Vol. 9, pp. 1493–1516.

Hastie, T., Tibshirani, R., and Friedman, J., 2001, "Datasets for the elements of statistical learning," *http://www-stat.stanford.edu/~tibs/ElemStatLearn/data.html*, Stanford University.

Roweis, S., 2007a, "MNIST handwritten digits,"*http://www.cs.toronto.edu/ roweis/data.html*, Computer Science Department, University of Toronto, Canada.

Roweis, S., 2007b, "Olivetti faces," *http://www.cs.toronto.edu/ roweis/data.html.* Computer Science Department, University of Toronto, Canada.