ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

An Augmented Efficient Backpropagation Training Strategy for Deep Autoassociative Neural Networks

Mark J. Embrechts[1], Blake J. Hargis[1], and Jonathan D. Linton[2] *

1- Rensselaer Polytechnic Institute - Department of Decision Sciences and Engineering Systems, Troy, New York - USA

2- University of Ottawa - Telfer School of Management Ottawa - Canada

Abstract. We introduce Augmented Efficient BackProp, a strategy for applying the backpropagation algorithm to deep autoencoders, i.e. autoassociators with many hidden layers, without relying on a weight initialization using restricted Boltzmann machines (RBMs). This training method, benchmarked on three different types of application datasets, is an extension of Efficient BackProp, first proposed by LeCun et al. [12].

1 Introduction

An autoassociator is a feedforward neural network, which is trained to map approximations of input vectors to its corresponding outputs and is distinguished by a central, low-dimensional bottleneck or coding layer. Such networks have long been of interest for dimensionality reduction and feature extraction, which often help reveal underlying patterns in large, high-dimensional datasets [10, 4].

Error surfaces associated with training deep architectures are non-convex and have many local minima, making gradient-based optimization difficult [3, 11]. This difficulty is particularly pronounced in networks with bottlenecks [1]. Furthermore, backpropagation scales poorly and can be very slow. As a result, it has often been claimed that the traditional backpropagation algorithm alone is not of practical interest in the effort to efficiently train deep networks. Our results suggest otherwise. Erhan at al. [3] note that no good training methods for deep architectures were known prior to the breakthrough development by Hinton et al. of an algorithm for deep belief networks [7]. The purpose of this paper is to introduce a backpropagation-based training strategy that does not require pre-training using RBMs.

2 Augmented Efficient BackProp

Efficient training of deep autoencoders with multiple constriction layers is an outstanding research issue. Larochelle et al. [11] state explicitly that, although not widely reported in the literature, the backpropagation algorithm is empirically known to find poor solutions for networks with three or more hidden

*Support for this research has been provided by the Canadian Natural Sciences and Engineering Research Council.

layers. However, semi-heuristic procedures can be developed, which allow efficient training of such networks. To establish such procedures, we extend the guidelines reported in "Efficient BackProp," introduced by Yann LeCun et al. [12] and reported in Haykin's heuristics for improving the backpropagation algorithm [6]. Through experimentation on numerous large datasets of the University of California Irvine data repository [15], an efficient approach was developed. It was found that for fast, efficient, and robust training of neural networks, the ratios of the learning parameters, from one layer to the next, are of key importance. Efficient BackProp [12] proposes the following guidelines for training a multi-layered perceptron by backpropagation:

(i) Use the tanh activation function.
(ii) Standardize (or normalize) the inputs.
(iii) Initialize weights layerwise by choosing uniform random numbers in the interval $[-\sqrt{k}, \sqrt{k}]$, where k is the number of inputs of the layer.
(iv) Assign smaller learning rates in the latter layers.
(v) Assign smaller learning rates to neurons with many inputs.

We modify this approach for deep autoassociative networks by adding the following procedures sequentially and refer to the expanded guidelines as Augmented Efficient BackProp:

(vi) Initially assign the learning parameter, η, for each layer to be $\eta = \frac{1}{\sqrt{k}}$, where k is the number of inputs to neurons of that particular layer.
(vii) Leave the learning parameter for the last layer as is and reduce the learning parameters of previous layers by $\sqrt{2}$.
(viii) Proportionally scale all learning parameters such that the largest is $\frac{1}{n}$, where n is the number of training samples in the batch (or the epoch, when weights are updated after using smaller sets of random samples).
(ix) Apply a momentum factor α to each layer, starting with $\alpha = 0.5$.
(x) If the network initially converges monotonically, keep the learning parameters constant and continue training. Otherwise, divide all learning parameters by 2 and restart the training. Repeat this until an initial, uniform convergence has been established.
(xi) Once the error starts increasing, rather than decreasing, reduce all the learning parameters by a factor of 2 and increase the momentum, such that applying $\frac{1}{1-\alpha}$ increases by a factor of 2.

While (i)-(xi) represent a heuristic approach for choosing reasonable initial values for the learning and momentum parameters, it can be justified by: (1) experimentation on a large number of datasets to determine a reasonable estimate for (iv) in Efficient BackProp; (2) the observation that a good choice for a constant learning parameter for most neural networks trained with the backpropagation algorithm is $\frac{1}{\eta}$, in order to show good initial convergence; and (3) the fact that when the learning parameter is halved, one should double $\frac{1}{1-\alpha}$, as reported in [6].

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

To establish good initial learning rates for small networks, step (x) typically leads to division of initial weights by a total factor of eight; i.e., training must typically be restarted three times. Extensive experimentation has shown that the procedure usually trains within a few hundred to a few thousand epochs and is often remarkably efficient for large networks. In general, we trained on all available samples in one batch. Only when training progressed exceptionally slowly did we proceed with epoch training, in which 30 samples were selected at random for each epoch.

3 Benchmarking of Augmented Efficient BackProp

Three datasets are considered for benchmarking the Augmented efficient training strategy introduced in this paper:

(i) tobacco data, including 26 tobacco samples, with 16 features containing chemical constituents, for classification as sun-dried or flue-dried tobaccos. These data were obtained from [9], in which results for nonlinear principal components from autoassociators trained using a modification of the backpropagation algorithm also appear.

(ii) Italian olive oil data, consistsing of 572 samples of olive oils containing 8 fatty acids. Oils in the dataset come from 9 different regions of Italy [5]. Processing can be considered an example of a multi-class, unbalanced classification problem. This dataset, too, was previously discussed in the context of autoassociative neural networks [2].

(iii) toxicity data for small molecules and drug design used in the ICANN 2009 challenge for prediction of the toxicities of drug-like molecules [16, 14]. This dataset includes 1093 training data, 110 test data, and a total of 2223 descriptors. We include this dataset, because it is representative of a relatively large regression dataset and should pose a challenge for autoencoders trained by backpropagation.

The literature is vague on how to assess the performance of unsupervised learning, in general, and deep autoassociative networks, in particular. A meaningful pair of metrics is the fraction of unexplained variance (FUV) and, correspondingly, the fraction of explained variance (FEV), as defined by [13]:

$$FEV = 1 - FUV = 1 - \frac{\sum_{j=1}^{m} var(x_j) - \sum_{j=1}^{m} var(\hat{x}_i)}{\sum_{j=1}^{m} var(x_j)}$$

The FEV is a useful metric for establishing the number of neurons that should be present in the bottleneck, and by evaluating this metric or the root mean square error (RMSE) on training and validation data, it is easy to implement an early stopping criterion.

4 Results

Figure 1 summarizes the outputs of the two bottleneck neurons of various autoassociators and qualitatively compares the outputs of the two bottleneck neurons with the first two principal components 1 for the tobacco set.

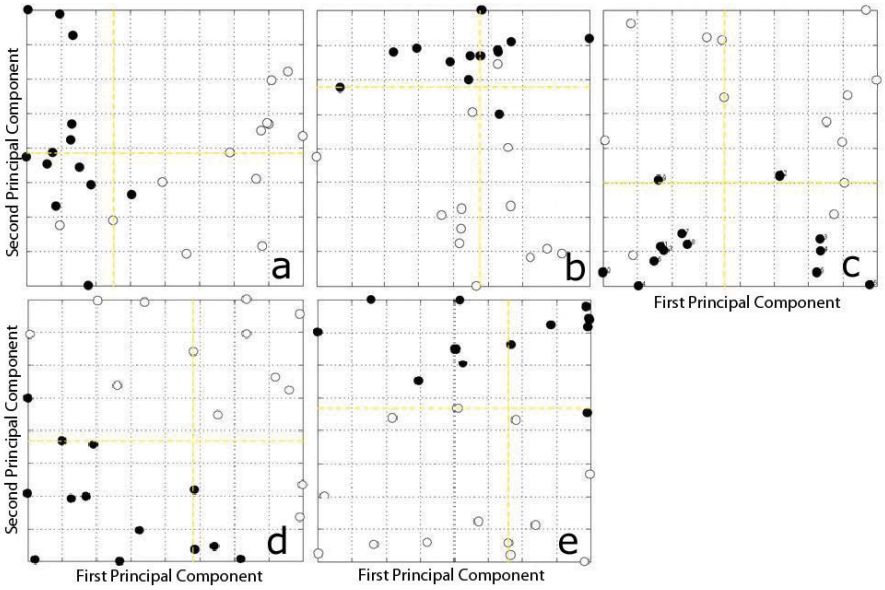

Fig. 1: Outputs of the two bottleneck neurons of various autoassociators: (a) PCA, (b) 16x2x16, (c) 16x8x2x8x16, (d) 16x12x8x2x8x12x16, and (e) 6x16x12x8x2x8x12x16x16 for sun-dried (open circles) and flue-dried tobaccos (filled circles).

Deeper architectures, as seen in Figure 1, tend to separate the tobacco classes in a more linear fashion. Each network was well-trained within a few thousand iterations, taking a fraction of a second to a few minutes on a standard laptop computer with a 1.5 GHz Pentium processor. The trend of the observed results is consistent with observations reported in [9].

Figure 2 compares outputs of the bottleneck layers of various autoencoders with PCA for dataset (ii). Again, classes to become more linearly separated by deeper encoders. These results are consistent with [2]. We include a large 8x1000x100x10x2x10x100x1000x8 architecture. Networks required between 1000 and 4000 iterations and trained from within a fraction of a second to a few minutes on a 1.5 GHz Pentium processor. Table 1 presents performance metrics for the networks identified in Figure 2: the FUV, the RMSE, and the required number of iterations.

A 2223x500x100x30x500x100x23 network was trained on the toxicity data within a few thousand iterations to $RMSE = 0.16$ and $FUV = 0.22$, indicating

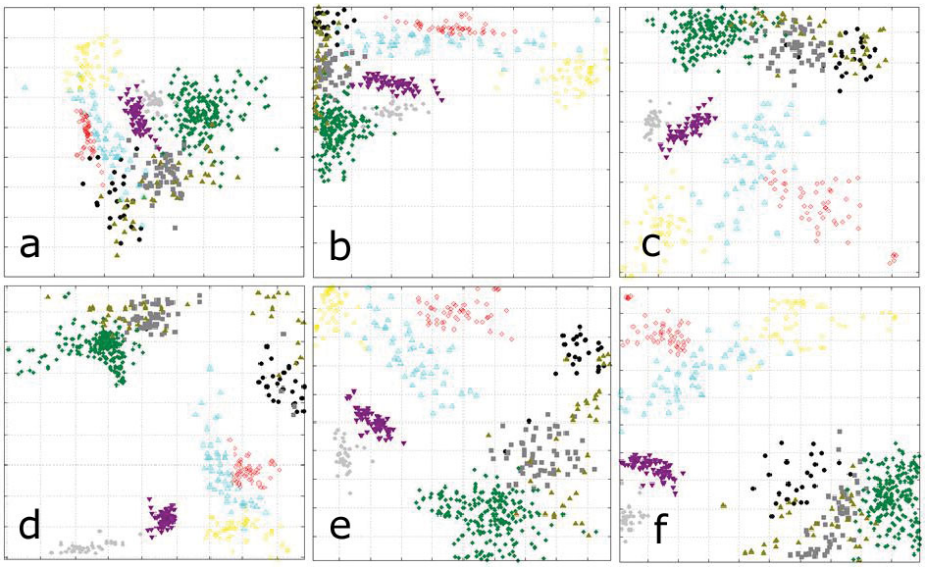

Fig. 2: Outputs of bottleneck layers of various autoencoders for the Italian olive oil data. Regions of origin are represented by different grey-scale colors: (a) PCA, (b) 8x2x8, (c) 8x6x2x6x8, (d) 8x8x6x4x2x4x6x8x8, (e) 8x100x10x2x10x100x8, and (f) 8x1000x100x10x2x10x100x1000x8 architectures

Network Structure	FUV	RMSE	Iterations
8x2x8	0.30097	0.20154	4000
8x6x2x6x8	0.17269	0.15137	4000
8x6x4x2x...	0.13146	0.13771	4200
8x8x6x4x2x...	0.11108	0.12559	4200
8x100x10x2...	0.13540	0.13820	1500
8x1000x100x10x2...	0.11195	0.13079	1240

Table 1: Performance metrics for Italian olive oil data

that the Augmented Efficient BackProp strategy is applicable to larger datasets.

5 Conclusion

We introduced Augmented Efficient BackProp as a training strategy for deep autoencoders and showed for the first time that deep autoencoders can be trained efficiently by the backpropagation algorithm alone. The establishment of good ratios of the learning parameters of the different layers is of prime importance. Only datasets with much larger numbers of features, such as $128x128$ image data, remain a challenge for Augmented Efficient BackProp, driving neuron outputs into saturation. While adjustments will be required for such datasets,

there is hope that fine-tuning can be calibrated using a design-of-experiments approach.

References

[1] Y. Bengio and Y. LeCun, Scaling algorithms toward AI. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large-Scale Kernel Machines*, MIT Press, 2007.

[2] M. Daszykowski, B. Walczak, and D. L. Massart, A journey into low-dimensional spaces with autoassociative neural networks, *Talanta*, Vol. 59, pp. 1095-1105, 2003.

[3] D. Erhan, P. A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, The difficulty of training deep architectures and the effect of unsupervised pre-training, In D. van Dyk and M. Welling, editors, *Proceedings of the* 12^{th} *International Conference on Artificial Intelligence and Statistics* (AISTATS 2009), Vol. 5, pp. 153-160, April 16-18, Clearwater Beach, Florida USA, 2009.

[4] I. K. Fodor, A survey of dimension reduction techniques. Technical Report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, California, USA, June 2002.

[5] M. Forina and C. Armanino, Eigenvector projection and simplified nonlinear mapping of fatty acid content of Italian olive oils, *Annali di Chemica (Rome)*, Vol. 72, pp. 127-141, 1981.

[6] S. Haykin, *Neural Networks and Learning Machines: 3rd Edition*. Prentice Hall, Pearson Education Inc., Ipper Saddle River, New Jersey, pp. 144-150, 2009.

[7] G. E. Hinton, S. Osindero, and Y. Teh, A fast learning algorithm for deep belief nets, *Neural Computation*, Vol. 18, pp. 1527-1554, MIT, 2006.

[8] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, Vol. 313, pp. 504-507, 2006.

[9] J Jiang, J. Wang, X. Chu, and R.Yu, Neural network learning to non-linear principal component analysis, *Analytica Chemica Acta*, Vol. 336, pp. 209-222, 1996.

[10] M. A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *American Institute of Chemical Engineers Journal*, Vol. 37, pp. 233-243, 1991.

[11] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, Exploring strategies for training deep neural networks, *Journal of Machine Learning Research*, Vol. 1, pp. 1-40, 2009.

[12] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, Efficient backprop, In G. B. Orr and K. R. Müller, editors, *Neural Networks: Tricks of the Trade*, Springer, 1998.

[13] A. Monahan, Nonlinear principal component analysis by neural networks: theory and application to the Lorenz system, *Journal of Climate*, Vol. 13, No. 4, pp. 821-835, 2000.

[14] I. V. Tetko, I. Sushko, A. K. Pandey, H. Zhu, A. Tropsha, E. Papa, T. Oberg, R. Todeschini, D. Fourches, and A. Varnek, Critical assessment of QSAR models of environmental toxicity against tetrahymena pyriformis: focusing on applicability domain of overfitting by variable selection, *Journal of Chemical Information and Modeling*, Vol. 48, pp. 1733-1746, 2008.

[15] UCI Machine Learning Repository [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[16] H. Zhu, A. Tropsha, D. Fourches, A. Varnek, E. Papa, P. Gramatica, T. Oberg, P. Dao, A. Cherkasov, and Igor V. Tetko, Combinatorial QSAR modeling of chemical toxicants tested against tetrahymena pyriformis, *Journal of Chemical Information and Modeling*, Vol. 48 pp. 766-784, 2008.