AutoMLP: Simple, Effective, Fully Automated Learning Rate and Size Adjustment

Thomas M. Breuel, Faisal Shafait
University of Kaiserslautern
67663 Kaiserslautern, Germany
www.iupr.com

In ease of use assessment of pattern recognition algorithms as part of the STATLOG project (King, 1995), neural networks received the lowest possible score: users had trouble finding learning rates and numbers of hidden units that worked well. It appears that this continues to be an impediment to the use of neural networks. In contrast, methods like SVMs combine predictable optimizers with a simple grid search and reproducibly achieve generally high recognition rates. MLPs continue to be important classifiers because, when trained correctly, they generally yield compact and fast classifiers with good classification performance, and because they scale up to very large training sets and large numbers of classes; they also allow simple domain adaptation via stochastic gradient descent. These are reasons why we are using MLPs for OCR. A significant obstacle to their use has been the need for manual intervention in the training process. An evaluation of existing learning rate adjustment methods from the literature showed inconsistent and unpredictable performance when applied to a wide range of problems.

Here, we report on the evaluation of a simple algorithm (AutoMLP) for both learning rate and size adjustment of neural networks during training. The algorithm combines ideas from genetic algorithms and stochastic optimization. It maintains a small ensemble of networks that are trained in parallel with different rates and different numbers of hidden units. After a small, fixed number of epochs, the error rate is determined on a validation set and the worst performers are replaced with copies of the best networks, modified to have different numbers of hidden units and learning rates. Hidden unit numbers and learning rates are drawn according to probability distributions derived from successful rates and sizes.

In our experiments, we compared AutoMLP against MLP and libsvm with a full grid search over 90 data sets from the UCI database. Training time with grid search was 120 hours, 3 hours with AutoMLP. Grid search and libsvm performed very similarly (with some outliers in favor of grid search), while AutoMLP generally performed close to both grid search and libsvm (Figure 1) at 1/40th of the computational cost. Differences could be further reduced by continuing AutoMLP training (additional training time will only improve performance, so AutoMLP can be kept running based on how much CPU time is available). Of course, in practice, for problems of the size of the benchmark problems, there is little reason not to perform the full grid search or use libsvm. But these results give us confidence that AutoMLP is a reasonable procedure to use for problem instances that are so large that grid search and libsvm are not feasible choices anymore.

On MNIST (no deskewing, no distortions or other dataset augmentation), AutoMLP achieves an error rate of 2.5% with 90 hidden units, which compares favorably to results reported in the literature for the same input data (LeCun et al, 1998; 4.5% and 4.7% with 1000 and 300 hidden units, respectively). This suggests that MLP results reported in the literature may have been limited by choices of learning rates and hidden units, and it also suggests that a meaningful comparison of classifiers should include an automated procedure for picking learning parameters.

We have used AutoMLP on very large classification problems (60M samples, 900 features, 130 classes) and found it to be effective and robust. The fact that training is fully

automated also means that we can use AutoMLP for boosting and style adaptation using stochastic gradient descent, applications in which manual intervention in the training process would be laborious or impossible.

A multicore AutoMLP implementation is available as part of iulib/OCRopus and can be used from both C++ and Python.

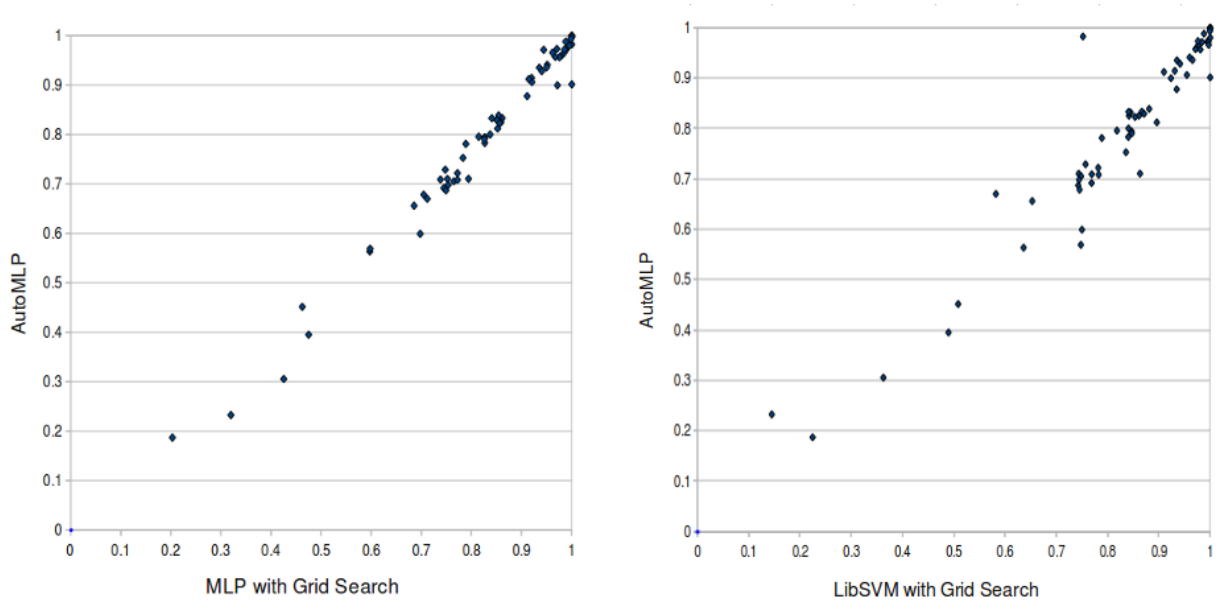

Figure 1: Scatterplot showing relative MLP, AutoMLP, and LibSVM performance. Axes show fraction of correctly classified samples (higher is better).

Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998

R. D. King, C. Feng, and A. Sutherland, "STATLOG: COMPARISON OF CLASSIFICATION ALGORITHMS ON LARGE REAL-WORLD PROBLEMS," Applied Artificial Intelligence 9, no. 3 (1995): 289–333.