Speed Training: Improving the Rate of Backpropagation Learning through Stochastic Sample Presentation

Michael E. Rimer, Timothy L. Andersen and Tony R. Martinez

Brigham Young University
Computer Science Department
Provo, UT 84602, USA
{mrimer, tim}@axon.cs.byu.edu, martinez@cs.byu.edu

Abstract

Artificial neural networks provide an effective empirical predictive model for pattern classification. However, using complex neural networks to learn very large training sets is often problematic, imposing prohibitive time constraints on the training process. We present four practical methods for dramatically decreasing training time through dynamic stochastic sample presentation, a technique we call speed training. These methods are shown to be robust to retaining generalization accuracy over a diverse collection of real world data sets. In particular, the SET technique achieves a training speedup of 4278% on a large OCR database with no detectable loss in generalization.

1 Introduction

Artificial neural networks have received substantial attention as robust learning models for tasks including classification [5]. Much research has gone into improving their ability to generalize beyond the training data. Many factors play a role in their ability to learn, including network topology, learning algorithm, and the nature of the problem at hand. In particular, the measure to which the training set represents the underlying distribution influences ultimate classification accuracy. Overfitting the training data is often detrimental to generalization. In theory, amassing an infinite training set would provide an exact measure of test accuracy (complete representation of the data distribution) and discourage overfitting. Hence, it is desirable to incorporate as large a training set as possible into the learning phase. However, training on very large data sets is problematic, as training time tends to increase more than linearly with the size of the training set [3]. The time required to converge on large data sets can be prohibitive. We provide four novel learning approaches that have shown to decrease training time by over an order of magnitude on very large data sets. Notably, the SET method achieves a training speedup of up to 4278% on the data tested with no detectable loss in generalization.

We give an overview of related work in section 2 and present four novel methods for speed training in section 3. Experiments are described in section 4. Results and analysis are given in section 5, followed by further work in section 6 and conclusion in section 7.

2 Related work

There have been many algorithms used to speed up the training of backpropagation neural networks, most of which are gradient descent "optimizing" algorithms. Two noteworthy approaches are QuickProp [2] and RProp [4]. QuickProp introduces a new error function, weight decay, and an alternative momentum equation. RProp uses an exponentially adaptive step size for each parameter in the network [7]. These techniques allow quicker convergence. However, little research has involved how the nature and size of the training set affects the training speed and resultant generalization. Zhang [9] creates a training set by selecting only critical examples and then expands this set if necessary for proper convergence.

A simpler method of improving generalization through reducing overfitting is to provide a maximum error tolerance threshold, d_{max}, which is the smallest absolute output error to be back propagated [6]. In other words, for a given d_{max}, target value, t_j, and network output, o_j, no weight update occurs if the absolute error $| t_j - o_j | < d_{max}$. This threshold is arbitrarily chosen to represent a point at which a sample has been sufficiently approximated. With an error threshold, the network is permitted to converge with much smaller weights, translating to a reduction in overfitting.

When class data is unbalanced, techniques such as sub-sampling and re-sampling the training data can provide a way to reduce training time and improve generalization on

the less represented classes [3]. Along with these techniques, Owens trains a committee of networks, each network learning from a distinct (balanced) subset of the training data. However, while this can improve training time and generalization, it results in a much more complex solution involving several networks instead of one. This technique's training time is reduced at the expense of testing time. In problem domains where a large amount of high-dimensional data is being classified, such solutions introduce a new problem by slowing down classification.

3 New approach

Our proposed methods differ from Zhang's and Owens' in two main respects. First, we use a stochastic data selection mechanism based solely on the network's ability to learn the given data rather than statistical approaches focusing on feature redundancy. Second, whereas Zhang only adds more examples with time and does not allow them to be removed from the training set and Owens selectively determines the data as a step preliminary to training, we provide a temporally dynamic stochastic data inclusion mechanism that presents samples to the network according to present learning need. These differences are based on inferred feature correlation and data replication (identical or almost-identical samples) existing in artificial and real world data sets. Equivalent generalization is achieved in less time without increasing the complexity of the network.

Rather than initially selecting a small subset of the training data to present to the network during training, the network retains access to all data samples during the training process. Sample presentation is determined exclusively by the ability of the network to learn the data. These methods result in a large reduction in training time through selectively "pruning" correctly classified samples from the training set to exclude their (redundant) presentation to the network each epoch. In other words, only the samples currently affecting the learning process are presented. We refer to this method of reducing training time through selective sample presentation as *speed training*.

3.1 Error based presentation (Error Based)

Each sample from the training data is presented to the network during the first epoch. The output error of the net for each sample is recorded. In subsequent epochs, samples are stochastically presented to the network based on the previous amount of error, where the error translates to the probability of subsequent presentation. That is, the probability of a sample, x_i, being presented on the following epoch is equal to its absolute training error (a value between 0 and 1), or formally,

$$P(x_i) = \frac{|t_i - o|}{\|O\|} \quad (1)$$

where t_i is the sample's target value, o is the net output, and $\| O \|$ is a normalization factor describing the range of the activation function (e.g., 1 for a standard sigmoid function).

Therefore, samples already learned to a high degree of accuracy are rarely presented to the network, while samples with a high error are presented more often. This approach provides a mechanism to progressively speed up training as the network converges by bypassing unneeded examples (those that do little to update the network parameters) and focusing on the more difficult parts of the problem.

3.2 Stochastic presentation with error threshold (SET)

An error tolerance threshold, d_{max}, is incorporated so that network weights are only updated on samples that output an error greater than this threshold (as described in Section 2). The probability of presenting a sample to the network is proportional to how close the sample is to overstepping the threshold. Formally,

$$P(x_i) = \begin{cases} \dfrac{|t_i - o|}{d_{\max}} & \text{if } |t_i - o| < d_{\max} \\ 1 & \text{otherwise} \end{cases}$$

This crudely equates to the probability the sample has of affecting the network parameters. Thus, samples with error far below the threshold will be seen rarely, while samples closer to the threshold will be seen often to maintain their correctness. This effectively bypasses samples that do not affect the performance of the network. Note that this method is more "conservative" than equation (1), skipping fewer samples on average.

3.3 Skip when correct (*n*-SKIP)

When the network classifies a sample correctly for n epochs, do not present it again for n epochs:

$$P(x_i) = \begin{cases} 0 \text{ if } (\text{last } n \text{ epochs correct}) \wedge \\ \quad (\text{skipped less than last } n \text{ epochs}) \\ 1 \text{ otherwise} \end{cases}$$

where n is a parameter and "skipped" is when x_i is not presented during an epoch; we define "correct" as error within d_{max} for the experiments presented below. These parameters are determined by the problem at hand, and can include the network outputting in a range of values (e.g., above 0.6 or according to winner-take-all). The intuition behind this method is that when the network incorrectly classifies a sample, it will probably incorrectly classify it again. Conversely, when the network is consistently correct on a sample, it will probably be correct again, and can therefore be skipped without adversely affecting the training process with high probability.

The tendency is that the more data there are, even when some samples are skipped, there will exist neighboring samples (closer to the decision surface) that are not skipped. This serves to keep the decision surface "in line" in the temporary absence of sample points. Re-including a sample after *n* epochs provides a quick check that the sample is still being classified correctly, and then if it is still correct it is skipped for another *n* epochs. The larger the value of *n*, the greater the speed up will be on large data sets, with the greater risk of samples falling "out of line" during their absence from several training epochs. This might result in greater deviation from standard training, but does not necessarily translate to a loss in generalization accuracy.

3.4 Stochastic presentation based on correctness history (Correct Ratio)

The probability of *not* presenting a sample is the ratio of the number of epochs for which it is correctly classified to the total number of epochs. We implement the probability of presentation through the formula

$$P(x_i) = 1 - \frac{\#\text{epochs correct}}{\#\text{epochs}} \quad (2)$$

where *# epochs* includes the current epoch (so that there is always a chance for presentation). The more often a sample is classified correctly the less often it is presented. For our experiments, we did *not* consider a sample correctly classified when skipped. This conservatively avoids skipping samples more and more often with time without justification. Other variants are possible and are discussed in section 8.

3.5 Resource Requirements

For the above methods, additional resource requirements are modest, limited to $O(n)$ in both space and time over the number of samples.

4 Experiments

To measure the speedup achieved through these approaches as well as validate their integrity we tested them on various problem domains, from small toy problems to very large real world data sets.

4.1 Data

1. *4-AND*. A small "toy" problem (although it certainly can appear in real data) consisting of a 4-input *AND* function with 16 samples that completely cover the problem space.

2. *Breast cancer*. A medium-sized real world problem taken from the UCI machine learning database repository [8], consisting of nine input attributes, one binary output, and 549 patterns, randomly split into 439 training patterns and 110 test patterns.

3. *OCR*. A very large set of machine printed alphanumeric characters used for OCR. It consists of over 495,000 samples, randomly split into roughly 415,000 training samples and 80,000 test samples. For training, each sample was normalized onto an 8x8 grid, resulting in 64 inputs. We trained a network to distinguish each character, but for simplicity only the results for the character "a" (a typical category with about 15,000 samples) are presented here.

4.2 Parameters

We used fully connected feed-forward neural networks trained through standard on-line backpropagation (minimizing SSE) for all experiments. For learning the *4-AND, breast cancer,* and *OCR* problems the network contained a single hidden layer comprised of 4, 5, and 32 hidden nodes, respectively. Weights were initialized to uniform random values in the range [-0.3,0.3]. For a given data set, the same initial weight values were used for all training runs. We used a learning rate of 0.2, momentum of 0.5, and error threshold (d_{max}) of 0.1 in all experiments presented here. Training was stopped when no samples were classified incorrectly on *4-AND*, and when a maximum number of epochs was reached (1000 for *breast cancer* and 500 for *OCR*).

5 Results and analysis

Tables 1-3 display the results of each data set. *Epochs* is the number of epochs until convergence. *Samples* is the total number of samples presented to the network during the training run. *Time* is real training time in seconds. *% SpdUp* is the speedup in training time over the standard method, in percent. *Train* is the final training set accuracy (above 0.5 for positive samples, below 0.5 for negative samples) in percent. *Train MSE* is the mean squared error for the training set at convergence. *Test* is the test set accuracy in percent. *Test MSE* is the mean squared error for the test set. Best values for each column are in italics.

The Error Based presentation technique results in the greatest training speed up in general, from a 78% increase in speed on *breast cancer* to a 4487% speed up on *OCR*. Of all four methods, this one prunes samples most aggressively. This is at the expense of a slight decrease in generalization accuracy compared to standard sample presentation. Speed up on *breast cancer* is not as great as on other sets because the MSE is higher on this data set. Higher average error causes samples to be presented more often during Error Based presentation.

Table 1: Results on *4-AND* data set.

Method	Epochs	Samples	Time	% SpdUp	Train	Train MSE	Test	Test MSE
Standard	1499	23984	0.047	N/A	*100.0*	*0.0313*		
Error Based	*559*	*3126*	*0.016*	*193.75*	*100.0*	0.0945		
SET	1495	12539	0.032	46.88	*100.0*	*0.0313*		
3-SKIP	1165	7122	0.032	46.88	*100.0*	0.0609	N/A	N/A
6-SKIP	1325	8289	0.032	46.88	*100.0*	0.0409		
9-SKIP	1464	9333	0.032	46.88	*100.0*	0.0326		
Correct Ratio	1502	11092	0.032	46.88	*100.0*	*0.0313*		

Table 2: Results on *breast cancer* data set.

Method	Epochs	Samples	Time	% SpdUp	Train	Train MSE	Test	Test MSE
Standard		439000	1.281	N/A	94.76	*0.0947*	*90.91*	0.1293
Error Based		137990	0.719	78.16	*97.04*	0.1076	88.18	0.1478
SET		201248	0.859	49.13	94.76	0.0949	*90.91*	*0.1291*
3-SKIP	1000	*84959*	*0.484*	*164.67*	94.76	0.1289	*90.91*	0.1611
6-SKIP		92423	0.515	148.74	94.99	0.1335	90.00	0.1726
9-SKIP		95770	0.531	141.24	95.22	0.1239	90.00	0.1618
Correct Ratio		120293	0.640	100.15	95.22	0.1129	90.00	0.1544

Table 3: Results on *OCR* data set.

Method	Epochs	Samples	Time	% SpdUp	Train	Train MSE	Test	Test MSE
Standard		207100000	8527.946	N/A	99.99	*0.0002*	99.96	0.0006
Error Based		*939790*	*185.898*	*4487.43*	99.96	0.0011	99.93	0.0014
SET		1188387	194.773	4278.40	*100.00*	*0.0002*	*99.97*	*0.0005*
3-SKIP		52760243	2312.724	268.74	*100.00*	*0.0002*	99.96	0.0006
6-SKIP	500	31710579	1401.750	508.38	*100.00*	*0.0002*	99.95	0.0006
9-SKIP		24262191	1114.810	664.97	*100.00*	*0.0002*	99.96	0.0006
12-SKIP		20566468	942.520	804.80	*100.00*	*0.0002*	99.96	0.0006
18-SKIP		18116504	854.524	897.98	*100.00*	*0.0002*	*99.97*	*0.0005*
24-SKIP		18161186	857.508	894.50	*100.00*	*0.0002*	99.96	0.0006
Correct Ratio		4378508	328.290	2497.69	99.99	0.0005	99.94	0.0010

SET proves superior in terms of accuracy, generalizing equally well or better than standard training on all three data sets. It is more conservative than Error Based in choosing what samples to exclude, hence yields slightly slower training. It still improves training time by 4278% on *OCR*. In other words, training on this large data set is performed in less than 2.3% of the standard time required. This translates to a drop in training time over one and a half orders of magnitude, or from hours to minutes (see Figures 1 and 2).

All variants of *n*-SKIP produced roughly equivalent results in generalization compared to standard training. They achieve a speed up roughly proportional to their *n* factor on large data sets. On fewer data, smaller *n* perform better. 3-SKIP learns *breast cancer* the quickest of all methods tested. 18-SKIP generalizes as well as SET on *OCR*, although it does not display as marked a decrease in training time (since 18 full epochs must occur before any samples are pruned).

Correct Ratio achieves higher accuracy and is faster on *breast cancer* than Error Based, although it is 76.6% slower on *OCR*. It is only slightly worse in generalizing than standard training. Its training time is roughly the median over all four methods on these data sets. As training continues, this technique tends to prune more and more samples. The percent of samples pruned per epoch is equivalent to the training set accuracy in the limit.

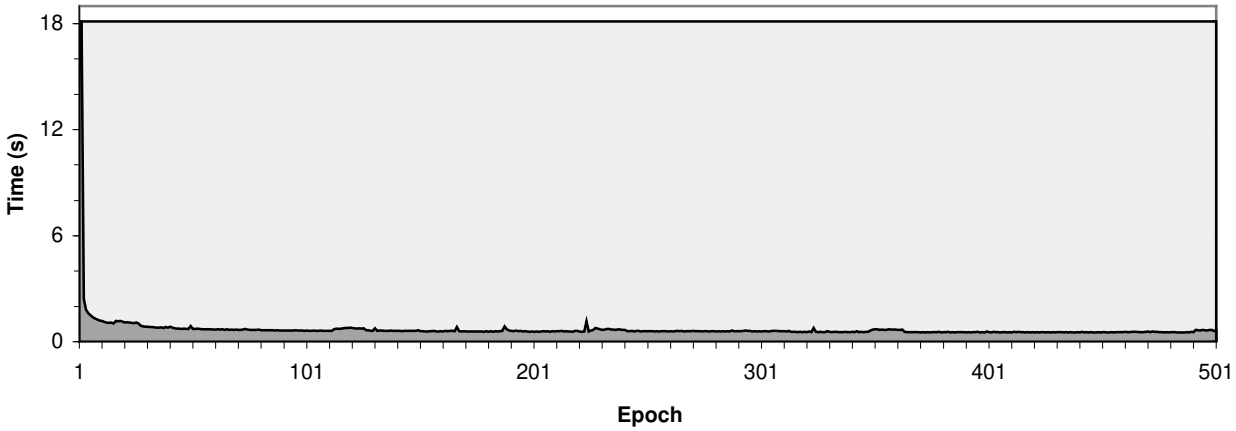

Figure 1: Training time per epoch (log scale) on *OCR* with SET (darker) vs. standard training.

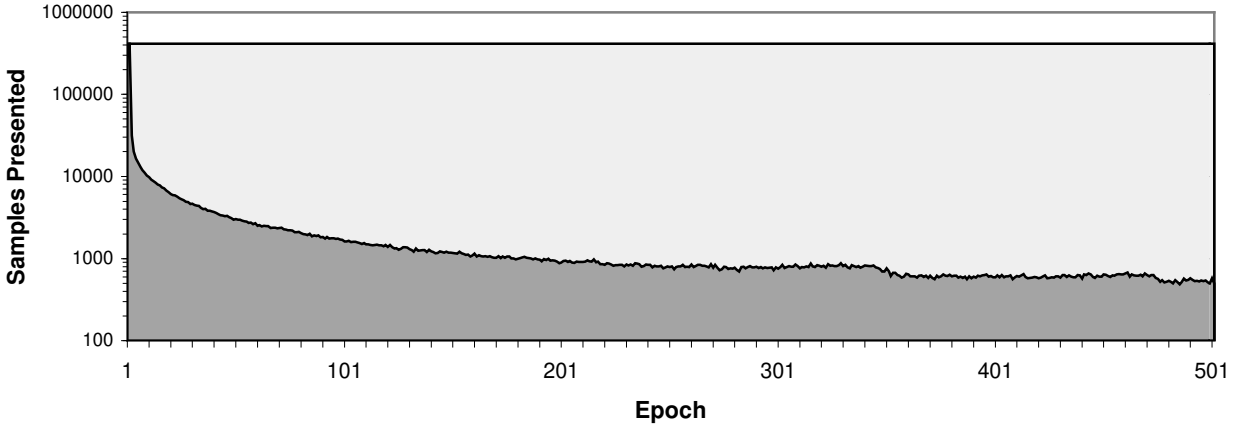

Figure 2: Samples presented per epoch (log scale) on *OCR* with SET (darker) vs. standard training.

6 Further Work

Further efforts will combine speed training with other "optimized" backpropagation algorithms (e.g., Quickprop and RProp). Together, it is conceivable that they will speed up convergence as well as reduce time spent per epoch in sample presentation.

Extending speed training to other iterative learning models, where the effectiveness or need of sample presentation varies over time, will also be studied. In particular, speed training will be tested with *batch* learning, where training time is very slow and epoch speed up is extremely desirable.

In addition to speeding up training, presenting samples with the most error more often may in general discourage overfitting. As proposed in [1], generalization is affected most by the size of the network parameters. When learning continues until weight saturation, generalization can be compromised. Excluding well-learned samples from further training can be a mechanism for keeping weights small, thereby improving generalization over techniques that saturate weight parameters. The usefulness of this principle will be investigated.

Several variations exist on the four methods proposed here. For example, when a sample is excluded from presentation on a given epoch, the probability that it will be presented in subsequent epochs can be gradually increased by a nominal value. This provides a more conservative approach to stochastic data exclusion, not allowing samples to be removed from training for too long.

Similarly, the way skipped samples affect sample presentation probability in Correct Ratio can be incorporated by extending equation (2) as follows:

$$P(x_i) = 1 - \frac{\#\text{epochs correct} + \alpha(\#\text{epochs skipped})}{\#\text{epochs}}$$

where α, ranging from zero to one, provides a pruning "aggressiveness" factor. For α approaching zero, skipping a sample increases the probability of presentation in subsequent epochs. This conservative approach reflects our experiments conducted here. For α close to one, skipping a sample gradually reduces the probability of subsequent presentation, a more aggressive pruning model.

Another improvement is to automate the choosing of n in n-SKIP in order to reduce training time as much as possible without requiring repeat training runs. An extension to this would be to dynamically alter the value of n during the training process to encourage further speedup.

Furthermore, the value from which $P(x_i)$ is derived in Error Based, SET, and Correct Ratio speed training can be augmented by a scaling factor to provide more conservative or aggressive sample pruning. However, a non-linear function of error to $P(x_i)$ is more general and may prove more effective. Investigation of these modifications will be presented in future work.

In the experiments presented here, no parameter optimizations were performed; commonly used, standard parameter values were incorporated for learning rate, momentum and error threshold. Work will be done to observe the effect of modifying these parameters on the time and accuracy of these speed training techniques.

7 Conclusion

Speed training provides an alternative to standard sample presentation in neural network training. It is a viable solution to overcoming prohibitive training costs in learning very large data sets with complex networks, and is an alternative to techniques such as subsampling [3] to reduce training time. It has proven effective on a variety of data sets with vastly different properties. Training time is reduced by roughly an order of magnitude and generalization is preserved.

A major weakness of standard backpropagation neural network learning is its slow training speed. Any of the proposed stochastic sample presentation schemes are appropriate if rapid training speeds are required while a very minimal drop in accuracy is acceptable. If accuracy is paramount, then conservative sample exclusion techniques, such as SET, provide dramatic speedup with no detectable loss of accuracy.

8 References

[1] Bartlett, Peter L., "The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network", *IEEE Trans. Inf. Theory*, 44(2), 1998, pp. 525-536.

[2] Fahlman, S.E., "Faster-learning Variations on Back-propagation: An Empirical Study", *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann.

[3] Owens, Aaron J., "Empirical Modeling of Very Large Data Sets Using Neural Networks", *IJCNN 2000*, vol. 6, pp. 6302-10.

[4] Riedmiller, Martin and Braun, Heinrich, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm", *Proceedings of the IEEE Conference on Neural Networks,* San Francisco, 1993.

[5] Rumelhart, David E., Hinton, Geoffrey E. and Williams, Ronald J., *Learning Internal Representations by Error Propagation*, Institute for Cognitive Science, University of California, San Diego; La Jolla, CA, 1985.

[6] Schiffmann, W., Joost, M. and Werner, R., "Comparison of Optimized Backpropagation Algorithms", *Artificial Neural Networks*, European Symposium, Brussels, 1993.

[7] Schiffmann, W., Joost, M. and Werner, R., "Optimization of the Backpropagation Algorithm for Training Multilayer Perceptions", University of Koblenz: Institute of Physics, 1994.

[8] University of California, Irvine, Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html

[9] Zhang, Byoung-Tak, "Accelerated Learning By Active Example Selection", *International Journal of Neural Systems*, 5(1), Germany, 1994, pp. 6775-79.