A Fast Learning Strategy Using Pattern Selection for Feedforward Neural Networks

Szilárd Vajda, Yves Rangoni, Hubert Cecotti, Abdel Belaïd

A Fast Learning Strategy Using Pattern Selection for Feedforward Neural Networks

Szilárd Vajda
Hubert Cecotti

Henri Poincaré University
Loria Research Center
READ Group
Campus Scientifique BP.239
Vandœuvre-lès-Nancy,
54506, France
{vajda,cecotti}@loria.fr

Yves Rangoni
Abdel Belaïd

Nancy 2 University
Loria Research Center
READ Group
Campus Scientifique BP.239
Vandœuvre-lès-Nancy,
54506, France
{rangoni,abelaid}@loria.fr

Abstract

Intelligent pattern selection is an active learning strategy where the classifiers select during training the most informative patterns. This paper investigates such a strategy where the informativeness of a pattern is measured as the approximation error produced by the classifier. The algorithm builds the training corpus starting from a small randomly chosen initial dataset and new patterns are added to the learning corpus based on their error sensitivity. The training dataset expansion is based on the selection of the most erroneous patterns. Our experimental results on MNIST[1] *separated digit dataset show that only 3.26% of training data are sufficient for training purpose without decreasing the performance (98.36%) of the resulting neural classifier.*

Keywords: pattern selection, incremental learning, handwritten digit recognition, neural networks, support vector machines

1. Introduction

The most commonly used scheme for character recognition is a feedforward neural network (NN) with a supervised learning strategy [1, 2, 11]. The training procedure is often based on the least mean squares (LMS) error minimization rule by adjusting proportionally the weights linking the units in the different network layers [15].

In principle, to train such a recognition system a huge amount of data is needed to cover the different intra-class and inter-class variations. Hence, neural network scheme based approaches are time costly solutions. The system convergence can be long as the adjustment of the decision surface (hyperplane) in the function of the network's free parameters needs many patterns. To achieve a good generalization the patterns belonging to training corpus should be considered several times. The excessive data amount, the network architecture and the course of dimensionality are always an endless trade-off in the neural networks theory [15].

In order to tackle these problems, different solutions have been proposed in the literature. They are based mainly on William of Ockham's (1285-1349) statement: "*What can be done with fewer is done in vain with more*" [5]. Considering the nature of the improvements based on the Ockham's statement, different research axis could be distinguished.

Some of the methods modify the network topology in order to reduce the number of free parameters. Another solution is the optimal brain damage (OBD), proposed by Le Cun in [12] which removes the unimportant weights in order to achieve a better generalization and a speedup of the training/testing procedure. The optimal cell damage (OCD) and its derivatives are based also on the idea to prune the network structure. Some others have been tried to construct the network by varying the number of the hidden units [8].

A fairly new approach considered in the last decade is the reduction of the size (dimension) of the input pattern called also feature selection/reduction [7]. The objective of such approach is three-fold: improving the prediction performances of the predictor, providing faster and more cost-effective predictors and providing a better understanding of the underlaying process that generated the data.

Among these solutions, the most appropriate seems to be the so called active learning. In such an approach, the classifier is guided during the training process while some information control mechanism is implanted in the system.

Rather than passively accepting all the available training samples, the classifier on his own, guides its learning process by finding the most informative patterns. With this guided training the computation is considerably reduced as the training patterns are rigorously selected. Supposing that the pattern selection method does not exceed

[1] Modified NIST

the reduction in training complexity (due to the reduction of the training samples) the training will be reduced [20]. A better generalization may be obtained as all non interesting data are discarded during the process. As stated in [16], the authors have found that the empirical error decreases more rapidly for active learning than for passive learning. In this way by implementing such a training algorithm we can build faster and more generic recognition systems.

Engelbrecht [5] is grouping these active learning strategies in two classes:

1. Selective learning, where the classifier selects at each selection interval a new training subset from the original candidate set. In each selection process, all the candidate patterns have the same chance to be considered in the training subset.

2. Incremental learning, where the classifier starts with an initial subset selected somehow from the candidate set. During the learning process, at specified selection intervals, some new samples are selected from the candidate set and these patterns are added to the training set. During the training, the candidate pattern set decreases while the size of the actual learning set grows.

We should note that the term "incremental learning" has been used rather loosely in the literature, where the term referred to as diverse concepts as incremental network, on-line learning or relearning of formerly misclassified patterns. We prefer to use the generic definition of Engelbrecht [5] instead of the definition given by Polikar et al. in [13].

While Engelbrecht [5], Foody [6] and Seung et al. [16] have developed active learning techniques for feedforward NNs and for SVMs, similar approach systems have been proposed in [10, 17, 19]. For SVM, the pattern selection algorithms are based on the idea than the hyperplane constructed by the SVM is depending only on a reduced subset of the training patterns called support vectors that lie close to the decision surface. Mostly the selection methods are based on k-NN, clustering, confidence measure, Hausdorf distance, etc. The drawback of such systems is the difficulty to fix different parameters of the systems as stated by Shin et al. [17]. Another limitation is the second training procedure. This drawback can also be observed in case of the different network pruning algorithms proposed by Le Cun [12]. To avoid this inconvenience we are proposing a wrapper method which acts like a filter.

However, there are many benefits of these active learning methods: facilitating data visualization and data understanding, reducing the data measurement and storage, reducing training and utilization time, defying the course of dimensionality to improve prediction performances [7].

Our training algorithm is based on active learning technique and it focuses on time reduction and accuracy. More precisely, it belongs to the branch of incremental learning, where the dataset is constructed dynamically during the training. Using this learning strategy, based mainly on LMS error minimization, we have considerably reduced the training time without loss of accuracy. Using this fast training algorithm allows to the researchers to test quickly different neural network topologies and different system initializations as there is no exact strategy available to set these values.

The rest of the paper is organized as follows. In Section 2, a description of the proposed reduction strategies is provided. Section 3, is dedicated to the results and finally in Section 4, concluding remarks are given.

2 Incremental learning algorithm description

2.1 General considerations

Our pattern selection strategy is based on incremental learning using error selection. The approach is based on a neural classifier. The main idea of the algorithm is to build-up in run-time a data driven learning-corpus based on the LMS by selecting "hard patterns" from the available training corpus. Instead of using the error selection method proposed by Engelbrecht [5], our method propose a selection based on the error of the classifier, selecting as training candidates those samples that have been misclassified, considering as error function the last means square function.

Let $D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ a training dataset where $x_i \in \mathbb{R}^d$ and $y_i \in \{C_1, \ldots, C_M\}$ where n denotes the number of samples in the training corpus, d denotes the dimension of the input space while M denotes the number of classes which should be separated. The main objective of the training is to achieve a good generalization performance of the neural network by giving acceptable responses to new inputs.

This learning mechanism can be reduce to a nonlinear optimization problem whose goal is to minimize the additive error function:

$$E(D_n \mid W) = \sum_{i=1}^{n} E(y_i \mid x_i, W) \tag{1}$$

where E is the error function while W is the neural network with the corresponding weights.

The main feature of the algorithm is that we do not train the network on the entire training corpus composed by n patterns. Rather we start the training with a reduced subset $D_0 \subset D_n$ and increasing the training set incrementally. Using such a technique, rather than attempting to minimize the error function (1), we try to minimize the error function components.

$$E(D_n \mid W) = E(D_{n_0} \mid W) + \cdots + E(D_{n_k} \mid W) \tag{2}$$

where n_i is the i^{th} size of the training set satisfying the following relation:

$$D_{n_0} \subset D_{n_1} \subset \cdots \subset D_{n_k} = D_n \tag{3}$$

where $n_0 < n_1 < \cdots < n_k = n$.

When a backpropagation learning strategy is considered, the algorithm tries to satisfy from the beginning all the constraints described by the given samples. As some samples are contradictory or some of them carry redundant information, it takes usually a long time to evolve a reasonable approximation curve. By such a decomposition we can decrease the number of constraints. Hence, the computation time is also reduced.

2.2 Algorithm description

Let us denote by *GlobalLearningCorpus* (GLC), the whole set of patterns which can be used during the training procedure, by *GlobalValidationCorpus* (GVC), the pattern set which helps to quide the training, by *GlobalTestingCorpus* (GTC), the whole set of patterns which can be used for the test and by *DynamicLearningCorpus* (DLC), the minimal set of patterns which can serve to train the network. Let us also denote by *NN* the neural network and by N the iterator, which provides the number of new pattern candidates to be considered at each learning level. M denotes the number of classes to be separated by the classifier.

$DLC = \{x_i \in GLC \mid i = \overline{1, M}\}$
$GLC = GLC - DLC$
Repeat
  Repeat
    TrainNet(NN, DLC)
  until (NetErr(NN, DLC, ALL)$\geq \Delta\epsilon$)
  TestNet(NN, GVC)
  If (NeErr(NN, GVC, ALL)$< \beta_1$) **then**
    STOP
  else
    TestNet(NN, GLC)
    If (NetErr(NN, GLC, ALL)$< \beta_2$) **then**
      STOP
    else
      $DLC = DLC \cup \{y_i \in GLC \mid i = \overline{1, N}\}$
      $GLC = GLC - DLC$
    end If
  end If
until ($GLC \neq \emptyset$)

Algorithm 1. The fast pattern selection algorithm

where: *TrainNet(NN,DATASET)* will train the NN with the given DATASET using classical LMS error minimization and error backpropagation, *TestNet(NN,DATASET)* will test the NN with the given DATASET, *NetErr(NN,DATASET,SAMPLES_NUMBER)* calculates the error given by NN using SAMPLES_NUMBER of patterns from the DATASET using the LMS criterion, y_i denotes the pattern from the GLC giving the i-th highest error during the test, while $|DATASET|$ denotes the cardinality of the DATASET.

The algorithm is starting with an initialized DLC set where we have selected for each class one random representative pattern (x_i) in order to avoid biasing one or another class initially. The algorithm performs the network training with these samples. Once the training error variation is less than an threshold value ($\Delta\epsilon$), the training process stops and we test the network with the samples belonging to the GVC. If the error criterion (β_1) is satisfied the algorithm stops as the training was successful and we test the network considering the GTC as test data. Otherwise, we should continue by adding new samples to the DLC set. To do this, we are looking from the GLC for the N samples (y_i) giving the highest error in the classification. If this error is less than a threshold value (β_2) we are stopping the algorithm, as we cannot add extra helpful information to the network. Otherwise we are picking these N elements from GLC and moving them to DLC and restart the training with this new and extended dataset. The algorithm stops when the error criterion is satisfied or there are no more available patterns in GLC set. In the second case we are in the classical training as finally we are using the whole dataset. So there is no restriction in the algorithm. In the worst case we should get almost the same results as in case of using the whole dataset. We should only consider the fact that the training is not equilibrated. While in the classical on-line learning each samples is presented once during one epoch, here there are samples which have been presented many times while some others just a few times.

The parameters β_1 and β_2 have been fixed based on tests described in a previous work [1] considering the same neural network and the whole learning corpus.

A modified version of the algorithm consists to feed the network with class candidates having the same distribution. This precaution could be necessary as stated in [9] not to influence the system in a way or another. For that reason we have modified the conditions of the DLC set creation. Now, at each iteration we add N samples for each pattern class based on their highest error contribution in their class instead of using the first N samples of the dataset. Using such a selection process we can guarantee the distribution uniformity of the patterns in the classes.

3 Experiments and results

The experiments performed by the pattern selection algorithm used as input data the MNIST benchmark database. This dataset contains 60,000 samples for learning and 10,000 samples for test. The 28×28 normalized grey-scale images contain separated handwritten digits from 0 to 9.

The main tests were performed with a fully connected MLP with one hidden layer. As input raw images were used. The input layer contains 784 nodes, each one corresponding to the grey level of each pixels in the image. The hidden layer contains 500 nodes while the output layer has 10 nodes, one for each digit. Using this network topology with the classical error back-propagation algorithm, considering the whole learning corpus the recognition performance achieved is 98.59% good recognition [1, 4]. To achieve such a performance 30 epochs were necessary to train the model. One epoch is considered when all the samples belonging to the training corpus have been given once to the model. That means we have presented

Table 1. Results obtained with different datasets constructed by the algorithm

N	Generated learning set (DLC)	Presented patterns	Recognition rate
50	1,260	64,390	98.30%
100	1,310	63,400	98.29%
150	1,960	93,180	**98.36**%
200	2,810	121,410	98.32%
250	3,010	130,900	98.21%
300	4,810	176,670	98.15%
350	3,510	147,720	98.20%
400	3,210	128,120	98.22%
450	3,160	128,270	98.23%

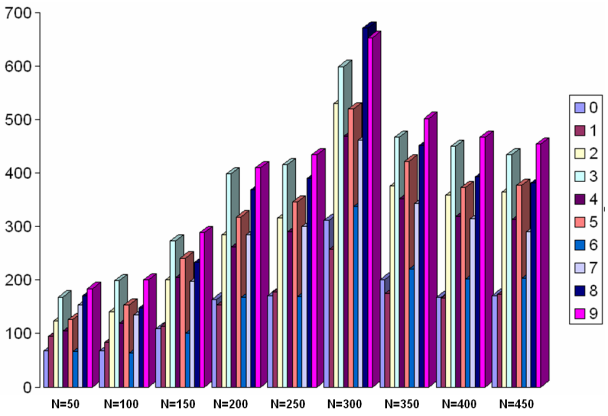

Figure 1. The samples distribution in the classes for the different constructed datasets reported in Table 1

$30 \times 60,000 = 1,800,000$ patterns to the network.

To test our designed pattern selection algorithm, the original training corpus is split in 50,000 patterns for training while the remaining 10,000 have been considered as a validation dataset. The selection was done using a random procedure but still considering an uniform distribution of the patterns in the classes.

In Table 1 we show different constructed datasets (DLC), the number of patterns given to the system and the obtained results by the algorithm on the test set. The reported results are based on different values of the N parameter which measures the amount of the new information added during each epoch.

We achieve comparable result using just 1,960 samples from the possible 60,000 used initially. We have considered just 3.26% of the training corpus. That means the remaining 96.74% patterns can be considered as being non relevant information for the decision function and there is no need to use them. The selected patterns lie close to the decision boundaries allowing an optimal decision surface design. The computation of the training process can be reduced substantially as it is possible to reach almost similar recognition scores presenting only 93,180 patterns to the network instead of 1,800,000 in the classical case. We can speedup the learning process by 13, which is a considerable gain even for a high-tech computer.

The modified selection algorithm result, 98.01%, is close to the original one's but it needs much more iterations and patterns (9,010 different samples were selected while 864,600 patterns were presented to the network). The uniform distribution of the data in the DLC set has a lower contribution as the original algorithms selects always the hard patterns enlarging the different class boundaries handling the lack of uniformity in the classes.

Considering the parameter N of the algorithm controlling the amount of the new candidates we can show than the algorithm is stable as varying this parameter the recognition score is almost similar.

In the Figure 1, we can observe the class distribution for the different datasets built by selection algorithm according to the N parameter. The x-axis indicates the different classes, and the y-axis denotes the distribution percentage of the different classes. We can see that the pattern distribution variance is not significant for the different datasets, so, the N parameter can influence just the learning convergence speed and the size of the built dataset.

The empirical value $N = 50$ was established after some bootstraps performed with different N values. We have found this starting value as being optimal to achieve a considerable speed gain.

Similarly, the results presented in Table 1 prove that N has no major influence on the results. It can influence the size of the built dataset and the speed of the building process.

Using the same pattern distribution as in Figure 1 but random choice for the patterns selection, the recognition accuracy cannot achieve higher average recognition scores than 91.01%. For that purpose 1,000 random databases have been generated.

Analyzing the dynamic learning corpus, we can pronounce also in the matter of the intra-class and inter-class variance. In the MNIST database the class '0' contains the least variance and the class '9' contains the most variation, so much more samples are needed belonging to class '9' in order to achieve a good recognition score.

In pattern complexity terms speaking, the classes '0', '1', '6' are the simplest ones while the classes '3', '8', '9' are the more complex ones, which is natural as their shape can be confused.

To test the generic aspect of the proposed selection algorithm we have considered a convolusional network too. This neural network is designed with a specific topology. The goal of the topology based on convolutional neural network is to classify the image given as input by analyzing it through different "receptive fields". Each layer is composed of several maps, each one corresponding to an image transformation. These transformations extract features like edges, strokes, etc [11].

Table 2. State of art results concerning the MNIST dataset using MLP like approaches considering the full dataset for training purpose

Ref	Method	Recognition rate
[11]	28x28-1000-10	95.50%
[11]	28x28-300-100-10	96.95%
[11]	28x28-500-150-10	97.05%
[18]	28x28-800-10	98.4%
[4]	28x28-500-10	98.59%
[18]	28x28-800-10 (distorsion)	99.3%

The neural network is composed of 5 layers. The first one corresponds to the input image, normalized by its center and reduced to a size of 29×29 pixels. The next two layers correspond to the information extraction, performed by convolutions. The second and third layers are composed of 10 and 50 maps respectively. Each map describes a convolution and a sub-sampling. In these maps, neurons share the input weights represented by a pivot neuron in each map. The last two layers are fully connected and finally the last layer is the output: 10 neurons, one output for each digit [4].

The system can achieve 98.74% good recognition score when the clasifier is trained with the whole learning corpus, using at least 10 consequent training epochs. Training the network just with 4,960 samples selected by the proposed algorithm, we can achive 97.49% accuracy. Although the recognition rate is not improved, the training speed is improved as 5 times less training samples were considered for this second training procedure.

In order to compare the results of the presented method, we show in Table 2 considering similar approaches on the MNIST dataset.

The result achieved by Simard et al. in [18] can be explained with the fact that in this case the original training corpus was extended considering geometrical distortions which extends considerably the training corpus as well the training process. Meanwhile, in this paper the goal is to achieve comparable results with much less data with special consideration on the time complexity of the training.

In [17], the authors provide results of their pattern selection method on MNIST benchmark dataset using SVMs, so a comparison study could be performed.

Nine SVM type binary classifier were considered: class 8 is paired with each of the rest. The reported recognition error average over nine classifiers is 0.28% using all the available patterns and 0.38% for the pattern selection based technique. The loss of accuracy is similar to the obtained in our case. Unfortunately, there is no result reported concerning the real recognition accuracy for each separated digit class, so a direct comparison in term of accuracy cannot be performed with our method. The time factor is reduced with a factor of 11.8 which is much less as in our case. Similarly, the number of used patterns (16.76%) serving as support vector are considerably more than our 3.26% selected patterns to train the system.

Table 3. MNIST digit classification accuracy while decreasing the number of features. Here the all samples have been considered

#features	Whole MNIST		The 1,960 samples	
	Raw Rate	Normal. Rate	Raw Rate	Normal. rate
784(max)	93.8%	100%	82.4%	100%
500	93.1%	99.2%	81.6%	99.0%
300	92.3%	98.4%	79.0%	95.8%
150	90.5%	96.5%	70.8%	85.9%
100	88.4%	94.2%	69.1%	83.9%
50	82.4%	87.8%	62.5%	75.9%
25	63.4%	67.6%	47.5%	57.7%

For the following tests, the method is shown interesting for another application that uses also the MNIST database. In [14], the author suggests a feature selection algorithm based on a filter approach (the classifier is not considered in the selection process). The sample choice is a critical task because the selection relies only on these database samples for finding the best variable subset, which will create a good classifier. To build this subset, the correlation matrix $C \in \mathbb{R}^n$ of the training database is computed. Then, the KL-Transform is applied $C = V \times \lambda \times V^T$. As the PCA, the new basis V is reduced to create a lower-space representation. The new dimension q is determined according to Cattell method [3]. The n eigenvectors $V_i \in \mathbb{R}^q$ are clusterized by a k-means in order to keep together the V_i corresponding to high correlated variables. By choosing in each cluster the nearest eigenvector from the center, a subset containing at the same time high predictive power and independent variables is created.

The selection method is evaluated using a MLP. The best recognition rate with the whole features is kept as a standard, the new subsets of the selection are compared to this standard by computing the recognition rate using only these few features. The same test protocol is also done but just with 1,960 samples selected by the algorithm and not on the whole learning corpus.

Comparing the results, we can conclude that in the second case, when the 1,960 patterns are taking into account, the results are almost similar but the knowledge is considerably reduced (Table 3). The differences can be explained with the fact that in the second case is not possible to achieve similar scores as the training corpus is not equilibrated and for the low represented classes (class 0, 1 and 6) the algorithm cannot extract the most informative features. The presented selection method has not been considered to select the best variables, rather to rank the features for another artificial neural network [14].

4 Conclusion

The proposed algorithm is based on a dual Least Mean Squares error estimation, which can guarantee the convergence of the algorithm. The first LMS minimization is used in the training process in the error back-propagation. The second one is used when the LMS error is calculated for the samples during the recognition. The misclassified patterns should be added to the Dynamic Learning Corpus set in order to minimize the recognition error by learning these new items which have contributed to the error accumulation. The method reduces substantially the learning running and discards the redundant information in order to avoid the overfitting.

The performed tests on MNIST showed that is possible to achieve 98.36% good recognition accuracy using only 1,960 different samples. The learning process computation can be reduced by a factor of 13, and such a gain is also considerable considering the algorithm complexity.

The algorithm tries to enlarge the different class boundaries using the extreme patterns for learning which lie close to the decision borders. The algorithm increases the number of forward steps (propagation) but substantially decreases the number of backward steps (error back-propagation) which are much more costly in computation terms.

This speedup achievement let us consider new ways to tune fast the different neural networks. Moreover, the creation of fast classifiers is very useful for creating multi-classifiers schemes.

We would like also to ameliorate the selection method by using a homogeneous distribution of the patterns in the different classes in order to be able to select the most representative features for all the classes without penalizing a given class. Considering the success of the wrapper selection method described in this article we would like to propose a corresponding wrapper feature selection technique.

References

[1] U. Bhattacharya, S. Vajda, A. Mallick, B. B. Chaudhuri, and A. Belaïd. On the choice of training set, architecture and combination rule of multiple mlp classifiers for multiresolution recognition of handwritten characters. In IWFHR, pages 419–424, 2004.

[2] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

[3] R. Cattell. The scree test for the number of factors. Multivariate Behavioral Research, 1966.

[4] H. Cecotti, S. Vajda, and A. Belaïd. High performance classifiers combination for handwritten digit recognition. In ICAPR, volume 3686 of LNCS, pages 619–626, 2005.

[5] A. P. Engelbrecht. Selective learning for multilayer feedforward neural networks. J. Mira and A. Prieto (eds.) In LNCS, 2084:386–393, 2001.

[6] G. M. Foody. The significance of border training patterns in classification by a feedforward neural network using back propagation learning. International Journal of Remote Sensing, 20(18):3549–3562, 1999.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. Machine Learning Research, (3):1157–1182, 2003.

[8] Y. Hirose, K. Yamashita, and S. Hiyija. Back-propagation algorithm which varies the number of hidden units. Neural Networks, (4):61–66, 1991.

[9] N. Japkowicz. The class imbalance problem: significance and strategies. In International Conference on Artificial Intelligence, 2000.

[10] R. Koggalage and S. Halgamuge. Reducing the number of training samples for support vector machine classification. Neural Information Processing - Letters and Reviews, 2(3):57–65, 2004.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Intelligent Signal Processing, pages 306–351, 2001.

[12] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. Advances in Neural Information Processing Systems 2 (NIPS*89), 1990.

[13] R. Polikar, L. Udpa, S. Udpa, and V. Honavar. Learn++: An incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man and Cybernetics - Part C: Application and Reviews, 31(4):497–508, 2001.

[14] Y. Rangoni and A. Belaïd. Data categorization for a context return applied to logical document structure recognition. ICDAR, 2005.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. pages 318–362, 1986.

[16] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. 5th Annual ACM Workshop on Computational Learning Theory, pages 287–299, 1992.

[17] H. Shin and S. Cho. Fast pattern selection for support vector classifier. 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNCS, (2637), 2003.

[18] P. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. ICDAR, pages 958–962, 2003.

[19] J. Wang, P. Neskovic, and L. N. Cooper. Training data selection for support vector machines. International Conference on Neural Computation, 2005.

[20] B. Zhang. Accelerated learning by active example selection. International Journal of Neural Systems, 5(1):67–75, 1994.