

Faculty of Informatics

Faculty of Informatics - Papers

University of Wollongong

Year 2008

Efficient Supervised Learning with Reduced Training Exemplars

G. H. Nguyen* A. Bouzerdoum†

S. L. Phung‡

*University of Wollongong, giang_nguyen@uow.edu.au

†University of Wollongong, a_bouzerdoum@uow.edu.au

‡University of Wollongong, lam_phung@uow.edu.au

This conference paper was originally published as Nguyen, GH, Bouzerdoum, A, Phung, SL, Efficient Supervised Learning with Reduced Training Exemplars, 2008 International Joint Conference on Neural Networks (IJCNN 2008), Hong Kong, 1-6 June 2008, 2981-2987. Copyright Institute of Electrical and Electronics Engineers 2008. Original conference paper available <[a href="http://dx.doi.org/10.1109/IJCNN.2008.4634217"](http://dx.doi.org/10.1109/IJCNN.2008.4634217)>here

This paper is posted at Research Online.

<http://ro.uow.edu.au/infopapers/691>

Efficient Supervised Learning with Reduced Training Exemplars

G. H. Nguyen, A. Bouzerdoum *Senior Member, IEEE* and S. L. Phung *Member, IEEE*

Abstract— In this article, we propose a new supervised learning approach for pattern classification applications involving large or imbalanced data sets. In this approach, a clustering technique is employed to reduce the original training set into a smaller set of representative training exemplars, represented by weighted cluster centers and their target outputs. Based on the proposed learning approach, two training algorithms are derived for feed-forward neural networks. These algorithms are implemented and tested on two pattern classification applications - skin detection and image classification. Experimental results show that with the proposed learning approach, it is possible to design networks in a fraction of time taken by the standard learning approach, without compromising the generalization ability and overall classification performance.

I. INTRODUCTION

Over the past two decades, machines that learn from examples, such as neural networks, support vector machines and decision trees, have proven to be important pattern classification tools, with growing applications in financial forecasting [1], text document classification [2], image and video retrieval [3], handwritten digit recognition [4], speech recognition [5], gender classification [6]–[8] (and references therein), face detection [7], [9] (and references therein), and face recognition [10], among others. To tackle the various applications, many network models have been proposed which differ in architecture and connection topology, but share similar learning strategies. Most learning algorithms are based on optimization theory, statistical learning theory, or evolutionary computation [11].

Although significant progress has been achieved in using neural networks for pattern classification, several issues still remain. A problem that we focus on in this paper is how to learn a classification task from large-scale or imbalanced data sets. For many real-world applications, as the size of data increases the computational resources required to learn the task become prohibitive. For example, it is a non-trivial task to design a neural network having thousands of parameters and using millions of samples because training could take days or even weeks. The problem is even more severe for systems that must learn in real-time.

In general, learning algorithms for large-scale problems can be classified into two categories: on-line learning and batch learning. Online algorithms, such as stochastic gradient-based learning [12] and non-target incremental learning [13], update the network parameters after the presentation of each training sample. These algorithms are used because of their ability to cope with a large data set. However, because only one training sample is considered

each time, online algorithms are not able to fully optimize the cost function, and it is possible that the network will “forget” previous training samples [14].

In batch training, the optimization process is performed with respect to the entire training set. While batch training works well for medium-sized networks and training sets, it is not efficient for large problems [15]. There exist two major approaches to addressing these shortcomings. The first approach, called passive learning, selects randomly a smaller number of training samples from the original set. However, it is difficult to determine the suitable number of samples to ensure that training will converge. The second approach, known as active learning [16]–[18], attempts to find the most informative training samples according to a predefined cost function; however, evaluation of the cost function can result in significant computational load.

In this paper, we introduce a new, efficient approach for training feed-forward neural networks with large-scale or imbalanced data sets. The proposed approach consists of two main stages: unsupervised clustering and supervised learning. First, a clustering technique is applied to partition the training patterns into a smaller number of clusters. Next, a supervised learning algorithm is applied that utilizes weighted cluster centers to achieve efficient learning. Compared with random sampling or using only cluster centers, not only does the proposed approach accelerate network training, but it also improves network generalization because training is based on a small yet more informative set of training exemplars.

The paper is organized as follows. Section II describes the proposed learning approach and derives two training algorithms for feed-forward neural networks. Section III presents experimental results where the proposed supervised learning method is applied to two different pattern classification tasks: skin detection and image classification. Finally, Section IV presents concluding remarks.

II. THE NEW SUPERVISED LEARNING APPROACH

Suppose that a multi-layer feed-forward neural network is to be trained using a set of M samples $\{\mathbf{x}^m, \mathbf{d}^m; m = 1, 2, \dots, M\}$, where \mathbf{x}^m is the m -th input pattern and \mathbf{d}^m is the corresponding desired output vector. Let L be the number of network layers and $f^l(\cdot)$ be the transfer function of the l -th network layer. Let \mathbf{w} be a vector consisting of all free network parameters, including weights and biases. The objective of supervised learning is to find a vector \mathbf{w}^o that minimizes a cost function. A common cost function is the *mean square error* (MSE), defined as

$$E(\mathbf{w}) = \frac{1}{M \times N^L} \sum_{m=1}^M \sum_{i=1}^{N^L} (y_i^{L,m} - d_i^m)^2, \quad (1)$$

Authors are with the School of Electrical, Computer and Telecommunication Engineering, University of Wollongong, Australia (email: giang@uow.edu.au, a.bouzerdoum@uow.edu.au, phung@uow.edu.au).

where the subscript i denotes the i -th element of a vector, and N^L is the number of neurons in the output layer L .

When the number of samples M is very large, calculating the error gradient is costly in terms of both time and memory storage required. Hence, we propose a more efficient algorithm for training feed-forward neural networks. In this approach, a pre-processing step is introduced to reduce the number of training samples. To this end, unsupervised learning or *clustering* is applied to the original data set $\{\mathbf{x}^m\}$ to extract cluster centers $\{\mathbf{c}^k\}$ that yield a compact representation of the original data. Here, clustering is applied independently to all the training samples representing a particular class. Therefore, each cluster represents samples from a single class, and each class is represented by several clusters. One way of dealing with imbalanced data sets is to simply assign the same number of clusters to each class. There exist many clustering techniques including the K-means [19], fuzzy C-means [20], hierarchical clustering [21], and self-organizing maps [22]; for a detailed review, the reader is referred to [21]. Although any of the aforementioned clustering techniques can be used, a suitable clustering is usually application-dependent and could be guided by the probability distribution of the input data.

After clustering, the data set is reduced to K exemplars ($K \ll M$), each is represented by a cluster *centroid* \mathbf{c}^k and *size*. Here, the cluster size z^k is simply the number of samples in the cluster—other measure of cluster size could be used. In the following, we present two training algorithms that integrate the cluster sizes and centroids into the learning rule.

A. Modified error gradient

During the supervised learning stage, the original data set $\{\mathbf{x}^m; m = 1, 2, \dots, M\}$ is replaced by the set of cluster centroids $\{\mathbf{c}^k; k = 1, 2, \dots, K\}$, which is then presented to the network along with the target outputs. To take into account the cluster sizes z^k , we modify the error function as follows:

$$E_p(\mathbf{w}) = \frac{1}{N^L} \sum_{k=1}^K \sum_{i=1}^{N^L} p_k (y_i^{L,k} - d_i^k)^2, \quad (2)$$

where d^k is the i -th element of the target or desired output vector \mathbf{d}^k and p_k is the cluster weight. It is defined as follows,

$$p_k = \frac{z^k}{\sum_{m=1}^M \omega_k \gamma_{mk}}, \quad (3)$$

where ω_k is the size of the class to which centroid \mathbf{c}^k belongs, and γ_{mk} is the degree of membership of x^m in the cluster k ,

$$\gamma_{mk} = \begin{cases} 1 & \text{if } x_m \in \text{cluster } k \\ 0 & \text{othersize} \end{cases} \quad \text{with } \sum_{m=1}^M \gamma_{mk} = 1 \quad \forall k.$$

To calculate the error gradient ∇E , we first compute the error sensitivities. The error sensitivity of neuron i in layer

l is defined as

$$\delta_i^{l,k} = \partial E_p / \partial s_i^{l,k}, \quad (4)$$

where $s_i^{l,k}$ is the weighted sum input to the neuron. With the error function in (2), the error sensitivities can now be expressed as follows.

- ◇ For the i -th output unit, $i = 1, 2, \dots, N^L$,

$$\delta_i^{L,k} = \frac{2}{N^L} p_k (y_i^k - d_i^k) f'_L(s_i^{L,k}). \quad (5)$$

- ◇ For the hidden layers, the sensitivity of the i -th neuron ($i = 1, 2, \dots, N^l$) in layer l ($l = L - 1, L - 2, \dots, 1$) is

$$\delta_i^{l,k} = f'_l(s_i^{l,k}) \sum_{j=1}^{N^{l+1}} \delta_j^{l+1,k} w_{i,j}^{l+1}. \quad (6)$$

Once the error sensitivities are found, the error gradient can be computed as follows:

- ◇ For weight $w_{i,j}^l$, $i = 1, 2, \dots, N^{l-1}$ and $j = 1, 2, \dots, N^l$,

$$\frac{\partial E_p}{\partial w_{i,j}^l} = \sum_{k=1}^K \delta_j^{l,k} y_i^{l-1,k}. \quad (7)$$

- ◇ For bias b_j^l , $j = 1, 2, \dots, N^l$,

$$\frac{\partial E_p}{\partial b_j^l} = \sum_{k=1}^K \delta_j^{l,k}. \quad (8)$$

B. Modified training algorithms

Once the error gradient is computed, numerous algorithms can be derived to train the feed-forward neural network. The list includes gradient descent (GD), gradient descent with momentum and variable learning rate (GDMV), resilient back-propagation (RPROP), conjugate gradient (CG) and Levenberg-Marquardt (LM). All these algorithms have been implemented with our proposed learning approach. However, in this paper, we focus our analysis on two modified algorithms: the *modified RPROP*, denoted as *Mod-RPROP* and the *modified Levenberg-Marquardt*, or *Mod-LM* for short. Because details of the standard algorithms can be found in [23], [24], we only summarize their main characteristics herein.

1) *Resilient back-propagation*: The resilient back-propagation algorithm updates the network weights and biases based on the sign of the error gradient,

$$\Delta w_i(t) = -\text{sign}\left\{\frac{\partial E_p}{\partial w_i}(t)\right\} \times \Delta_i(t), \quad (9)$$

where $\Delta_i(t)$ is an adaptive step specific to weight w_i .

The step size is adjusted using the following rule:

$$\Delta_i(t) = \begin{cases} \eta_{\text{inc}} \Delta_i(t-1), & \text{if } \frac{\partial E_p}{\partial w_i}(t) \frac{\partial E_p}{\partial w_i}(t-1) > 0 \\ \eta_{\text{dec}} \Delta_i(t-1), & \text{if } \frac{\partial E_p}{\partial w_i}(t) \frac{\partial E_p}{\partial w_i}(t-1) < 0 \\ \Delta_i(t-1), & \text{otherwise,} \end{cases} \quad (10)$$

where η_{inc} and η_{dec} are two scalar terms, $\eta_{\text{inc}} > 1$ and $1 > \eta_{\text{dec}} > 0$.

2) *Levenberg-Marquardt*: The Levenberg-Marquardt is a very fast training algorithm for neural networks [24]; it is based on the Gauss-Newton approximation of the Hessian matrix. The MSE cost function can be expressed in matrix forms as follows:

$$E_p(\mathbf{w}) = \frac{1}{N^L} \text{tr}(\Gamma^T P \Gamma), \quad (11)$$

where Γ is the error matrix, and P is the cluster weight matrix and defined as $P = \text{diag}(p_k), k = 1, 2, \dots, K$. Let \mathbf{e} be an $N^L K$ column vector obtained by stacking the columns of the error matrix Γ ; and let \mathbf{p} be the vector obtained by replicated the trace of matrix P into an $N^L K$ row vector. Then the modified Levenberg-Marquardt weight update rule is given by

$$\Delta \mathbf{w}(t) = [J^T \mathcal{P} J + \mu I]^{-1} \nabla E_p, \quad (12)$$

where μ is an adaptive learning rate, I is the identity matrix, J is the Jacobian matrix, and $\mathcal{P} = \text{diag}(\mathbf{p})$ is the expanded cluster weight matrix. Given N_w is the size of the weight vector, the Jacobian is a matrix of $N^L K$ rows and N_w columns, whose entries are defined as

$$J_{(q-1)K+k,i} = \frac{\partial e_q^k}{\partial w_i} \quad (13)$$

where $q = 1, 2, \dots, N^L$ and e_q^k is the error term of output neuron q for training sample k ,

$$e_q^k = y_q^k - d_q^k. \quad (14)$$

Calculation of the Jacobian matrix is similar to computation of the gradient ∇E_p shown in Equations (4) to (8). We only need to modify the definition of error sensitivities:

$$\delta_{q,i}^{l,k} = \partial e_q^k / \partial s_i^{l,k}. \quad (15)$$

We should also note that error gradient can be expressed in terms of the Jacobian matrix as

$$\nabla E_p = J^T \mathcal{P} \mathbf{e}, \quad (16)$$

where \mathbf{e} is a column vector of the error terms $\{e_q^k\}$.

III. EXPERIMENTS AND ANALYSIS

In this section, we apply the proposed learning approach to two pattern recognition tasks: (i) skin detection; and (ii) image classification for automatic image annotation. Our aim is to study the convergence speed and generalization capability of the propose approach, compared to the standard approach for neural network training.

A. Skin detection task

Skin detection aims to identify human skin regions in a colour image. It is used for web image filtering and face detection. Most existing skin detection techniques rely on classification of each image pixel (Red, Green, Blue) into skin or non-skin [25]. The difference in our approach is that skin classification is based on not only one center pixel but also pixels in its neighbourhood region (in this paper, we use the 3-by-3 region). Therefore, the input to the neural

TABLE I
NETWORK CONFIGURATION FOR SKIN DETECTION PROBLEM

Network configuration	Layer sizes and network weights				
	Input	Layer 2	Layer 3	Output	Weights
Net A	27	10	none	1	291
Net B	27	6	3	1	193

TABLE II
NETWORK CONFIGURATION FOR IMAGE CLASSIFICATION PROBLEM

Network configuration	Layer sizes and network weights				
	Input	Layer 2	Layer 3	Output	Weights
Net C	80	20	none	4	1704
Net D	80	16	8	4	1468

network is a 27-element vector containing the Red, Green, Blue values of nine pixels. The network output is a scalar that indicates the class of the center pixel. The network has 27 input neurons and one output neuron. To evaluate the modified training algorithms, we use two network structures that are summarized in Table I.

The skin data used for this study is taken from a large face and skin detection database of about 4,000 images [25]. Images in the database are taken from various sources and contain people of different skin tones: blackish, yellowish, brownish and whitish. The results presented here are based on data set consists of 250 images, of which 200 images are used for training and 50 images are used for testing. From the training images, 120,000 samples are randomly selected to form the training set and 30,000 samples are extracted for the test set. We should note that the training and test samples are extracted from separate images. Furthermore, the number of skin and non-skin samples are equal in both the training and test sets.

B. Image classification task

The second task is the classification of images into conceptual classes; this is a key step in automatic annotation of images for content-based retrieval. The experiments are conducted using a data set of 14,400 images with four classes: landscape, cityscape, vehicle and portrait [3], with each class comprising 3,600 images. Shao et al. extract MPEG-7 visual descriptors and classify these descriptors into the four categories [3]. Since our main objective is to compare the proposed and the traditional supervised learning approach, we only use one descriptor, the edge histogram, which has been found to have more discriminative power compared to other MPEG-7 visual descriptors [3].

In the experiment, we use 8,400 images for training and 6,000 images for testing; the four classes have equal numbers of images. For this four-class classification problem, the network input is a 80-element vector containing the edge histogram of the input image. The network output is a vector of 4 elements representing the image class. We analyze two network structures that are shown in Table II.

C. Reduction of training samples

Two approaches for the reduction of the original data set were implemented. The first approach selects the training samples randomly from the original set. The second approach finds representative training samples using clustering. In this study, we adopt the K-means clustering algorithm. This algorithm requires little parameter tuning and is quite effective in handling large data sets [19].

The first set of experiments investigates the effects of replacing the original data by cluster centroids and their weights, we compare three techniques for data reduction.

- *Ran-RPROP*: The training samples are randomly selected from the original training set.
- *Clus-RPROP*: The training samples are the cluster centroids; no information on cluster size is used.
- *Mod-RPROP*: The proposed training algorithm which takes into account cluster centroids and cluster sizes.

The experiment steps can be summarized as follows.

- Each training technique is applied to train 20 networks with different initial weights.
- The training sets are partitioned into 80% for training and 20% for validation.
- In the skin detection task, the number of training samples varies from 0.025% to 0.5% of the original data set of 96,000 samples.
- In the image classification task, the number of training sample varies from 1% to 7% of the original data set of 8400 images.
- Classification rate of each training technique is evaluated on the test set and averaged across all 20 networks.

The classification rates (CRs) of the different training techniques on the skin detection task and the image classification task are presented in Table III and IV, respectively. The same results are presented in Fig. 1, which illustrates the classification rates of the three training techniques versus the number of training samples. Clearly, using unsupervised clustering to select training samples (*Clus-RPROP* and *Mod-RPROP*) achieves higher classification rates compared to selecting training samples randomly (*Ran-RPROP*). Furthermore, the proposed approach, *Mod-RPROP*, achieves the highest CR. The improvement in the classification rate of *Mod-RPROP* is more significant when the number of training samples is small. For example, for the skin detection task and net B with 24 training samples, the classification rates of *Ran-RPROP*, *Clus-RPROP* and *Mod-RPROP* techniques are 77.97%, 80.72% and 82.37%, respectively. For the image classification task and net C with 84 training samples, *Mod-RPROP* technique has a CR of 71.53, 95% confidence interval of [70.39,72.67], whereas the random sampling method *Ran-RPROP* achieves a 63.49% CR only.

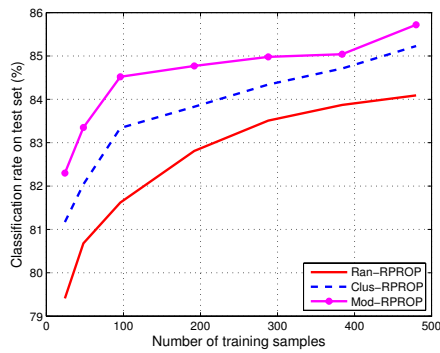
These results also show that the modified training approach can handle the case when the number of free parameters (network weights and biases) is larger than the number of training samples. For instance, in Table III the *Mod-RPROP* has a CR of 84.77% while training with only 192 samples on a network that has 291 parameters. In Table IV, the *Mod-RPROP* has a CR of 73.52% while training with 588 samples on a network that has 1704 parameters. Here, the original samples are still used for training but in a compressed form. We can conclude that the combination of clustering and the new cost function provides extra information in the extracted

TABLE III
COMPARISON OF TRAINING TECHNIQUES FOR THE SKIN DETECTION TASK. THE 95% CONFIDENCE INTERVAL OF THE CR FOR MOD-RPROP IS ALSO SHOWN.

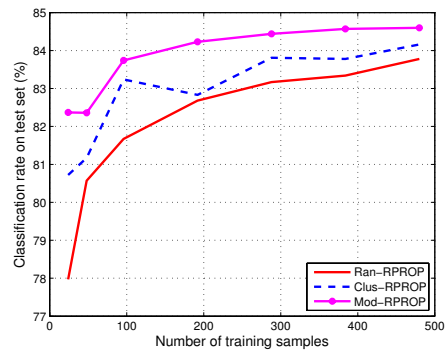
Size of train data		Net A: 291 weights and biases				Net B: 193 weights and biases			
		Classification rate on test set (%)				Classification rate on test set (%)			
%	size	Ran-RPROP	Clus-RPROP	Mod-RPROP	95% C.I.	Ran-RPROP	Clus-RPROP	Mod-RPROP	95% C.I.
0.025	24	79.41	81.17	82.30	[81.9, 82.7]	77.97	80.72	82.37	[81.9, 82.8]
0.05	48	80.68	82.04	83.35	[82.9, 83.8]	80.57	81.17	82.36	[81.9, 82.8]
0.1	96	81.62	83.34	84.52	[84.1, 84.9]	81.67	83.24	83.74	[83.3, 84.2]
0.2	192	82.81	83.83	84.77	[84.4, 85.2]	82.68	82.83	84.23	[83.8, 84.6]
0.3	288	83.51	84.34	84.98	[84.6, 85.4]	83.17	83.81	84.44	[84.0, 84.8]
0.4	384	83.87	84.71	85.04	[84.6, 85.4]	83.34	83.78	84.57	[84.2, 84.9]
0.5	480	84.09	85.23	85.72	[85.3, 86.1]	83.78	84.16	84.60	[84.2, 85.0]

TABLE IV
COMPARISON OF TRAINING TECHNIQUES FOR THE IMAGE CLASSIFICATION TASK. THE 95% CONFIDENCE INTERVAL OF THE CR FOR MOD-RPROP IS ALSO SHOWN.

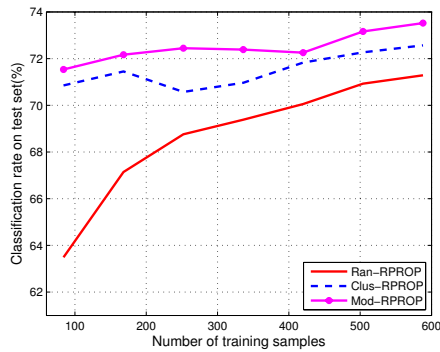
Number of train data		Net C: 1704 weights and biases				Net D: 1468 weights and biases			
		Classification rate on test set (%)				Classification rate on test set (%)			
%	size	Ran-RPROP	Clus-RPROP	Mod-RPROP	95% C.I.	Ran-RPROP	Clus-RPROP	Mod-RPROP	95% C.I.
1	84	63.49	70.85	71.53	[70.39, 72.67]	61.71	69.67	70.67	[69.52, 71.82]
2	168	67.14	71.45	72.17	[71.03, 73.30]	66.01	70.08	71.06	[69.91, 72.21]
3	252	68.76	70.57	72.44	[71.31, 73.57]	67.81	69.85	71.08	[69.93, 72.23]
4	336	69.38	70.96	72.39	[71.25, 73.52]	68.67	69.56	71.77	[70.63, 72.91]
5	420	70.05	71.83	72.26	[71.12, 73.39]	68.88	70.62	71.84	[70.71, 72.98]
6	504	70.93	72.27	73.16	[72.04, 74.28]	70.46	71.12	72.25	[71.12, 73.38]
7	588	71.29	72.57	73.52	[72.40, 74.66]	70.47	71.70	72.33	[71.19, 73.46]



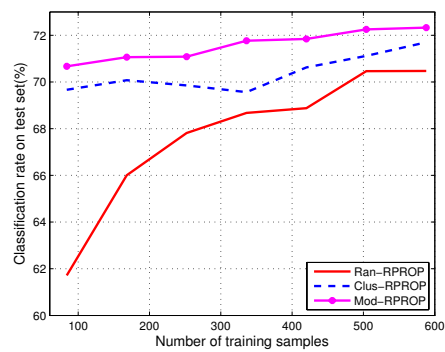
(a) Skin detection, network size 27-10-1



(b) Skin detection, network size 27-6-3



(c) Image classification, network size 80-20-4



(d) Image classification, network size 80-16-8-4

Fig. 1. The classification rates of the training algorithms versus the number of training samples that are actually used.

training samples.

D. Generalization performance

Here, we compare the generalization performances of the proposed training approach and the standard training approach. The comparison is based on the five-fold cross validation on the training set. The entire training set is divided into five subsets. In each fold, one four subsets are used for training and the remaining subset for validation. Several networks are trained and the best performing network on the validation set is selected for testing; its performance is evaluated on the test set. The average classification rate on the test set, over the five folds, is used as an estimate of generalization performance. The standard training approach (RPROP and LM) employs the entire original training set whereas the proposed training approach (Mod-RPROP and Mod-LM) uses reduced number of training samples: 480 samples for skin detection task and 588 for image classification task.

TABLE V
COMPARISON OF STANDARD AND MODIFIED ALGORITHMS ON THE SKIN
DETECTION TASK.

Training methods	Classification rate on test set (%)	95% confident interval
RPROP	87.14	[86.8, 87.5]
Mod-RPROP	87.51	[87.1, 87.9]
LM	87.87	[87.5, 88.2]
Mod-LM	87.24	[86.8, 87.6]

The classification rates of different training algorithms are shown in Table V for the skin detection task and Table VI for the image classification task. The modified training algorithms and the standard training algorithms achieve almost similar classification rates. For skin detection task, the CRs of different algorithms are: RPROP = 87.12%, Mod-RPROP = 87.51%, LM = 87.87%, and Mod-LM = 87.24%. For image classification task, the CRs of different algorithms are: RPROP = 78.43% and Mod-RPROP = 77.70%. This is remarkable because the modified training algorithms use only

TABLE VI
COMPARISON OF STANDARD AND MODIFIED ALGORITHMS ON THE
IMAGE CLASSIFICATION TASK.

Training methods	Classification rate on test set (%)	95% confident interval
RPROP	78.43	[77.4, 79.5]
Mod-RPROP	77.70	[76.6, 78.8]

a fraction number of training examples.

E. Convergence speed

In this section, we investigate the speed of the proposed training approach (Mod-RPROP and Mod-LM) and compare it to that of the standard supervised learning approach (RPROP and LM). Each of the four algorithms is applied to train 50 networks of the same structure but with different initial weights. The number of training epoches is 500. The RPROP and LM algorithms are applied on the original training sets whereas the Mod-RPROP and Mod-LM algorithms are run on a reduced training set of 480 samples for the skin detection task, and 588 samples for the image classification task. The training speed of an algorithm is defined as the time taken to learn the original training set. For comparison purposes, the maximum, minimum and average training times in seconds are recorded. All the experiments are conducted on a PC with a P4 3GHz CPU and 1GB RAM.

The comparative speed of the training algorithms are shown in Table VII and Table VIII. The same results are also illustrated in Fig. 2. The results show that the modified training algorithms converge faster compared to their standard counterparts. In the skin detection task, the Mod-LM algorithm takes on average only 84.49 seconds (including the clustering time) to learn the entire training set. In comparison, the standard LM algorithm takes on average 692.1 seconds.

TABLE VII
CONVERGENCE SPEEDS ON THE SKIN CLASSIFICATION TASK.

Training Algorithms	Net A: 291 parameters			Clustering time (s)
	Max. (s)	Min. (s)	Aver. (s)	
RPROP	241.30	42.24	134.30	none
Mod-RPROP	5.00	0.58	1.66	82.83
LM	1395.00	131.20	692.10	none
Mod-LM	1.57	0.76	1.16	82.83

TABLE VIII
CONVERGENCE SPEEDS ON THE IMAGE CLASSIFICATION TASK.

Training Algorithms	Net C: 1704 parameters			Clustering time (s)
	Max. (s)	Min. (s)	Aver. (s)	
RPROP	17.23	7.72	14.64	none
Mod-RPROP	4.57	1.40	2.54	4.08

Note that the one-time cost of finding clusters depends on the clustering algorithm and the number of clusters. In the skin detection task, for a data set of 96,000 samples in a 27-dimensional space, the time taken to form 480 clusters is 82.83s. In the image classification task, for a data set of

8400 samples in a 80-dimensional space, the time taken to form 588 clusters is 4.08s.

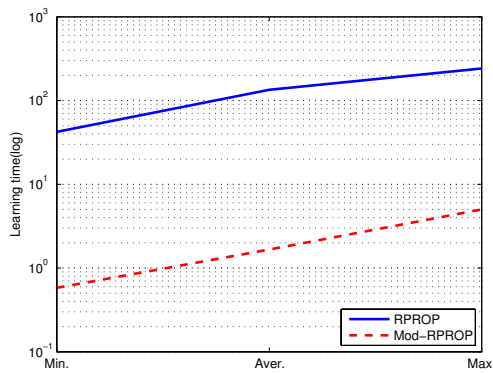
The results presented in this section show that it is possible to train a neural network using only a fraction of the original training set and achieve a similar classification rate. In this paper, the main issue of interest is, therefore, the computation efficiency. For comparison purposes, we have used here the data sets that the standard training approach can handle. However, in many practical applications the standard training approach is infeasible because of the amount of training data; our approach can be easily applied to train networks in much shorter time and produce networks of smaller size.

IV. CONCLUSIONS

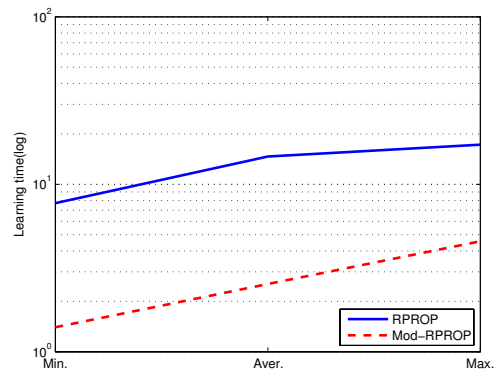
In this article, a new training approach for feed-forward neural networks that combines unsupervised clustering and supervised learning has been presented. The proposed approach can be applied to existing training algorithms. Several experiments have been conducted to compare the performance of the proposed approach and the standard training approach on two different pattern recognition tasks: skin detection and image classification. The results show that the our approach can achieve similar classification rates as the standard training approach. More importantly, the new approach has a much lower computation time and can cope with large data sets. We show that it is possible to learn large data sets efficiently by combining unsupervised clustering with supervised learning. Future work will address the theoretical framework of the proposed approach, and investigate how it can be used in conjunction with meta-learning algorithms.

REFERENCES

- [1] L. Cao and F. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506 – 1518, 2003.
- [2] Z. Chen, L. Huang, and Y. L. Murphey, "Incremental learning for text document classification," in *International Joint Conference on Neural Networks (IJCNN 2007)*, 2007, pp. 2592 – 2597.
- [3] W. Shao, G. Naghdy, and S. L. Phung, "Automatic image annotation for semantic image retrieval," in *Lecture Notes in Computer Science*. Heidelberg: Springer Berlin, 2007, vol. 4781, pp. 369–378.
- [4] F. H. C. Tivive and A. Bouzerdoum, "Application of siconnets to handwritten digit recognition," *International Journal of Computational Intelligence and Applications*, vol. 6, no. 1, pp. 45–59, 2006.
- [5] N. Morgan and H. Bourlard, "Neural networks for statistical recognition of continuous speech," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 742 – 772, 1995.
- [6] F. H. C. Tivive and A. Bouzerdoum, "A shunting inhibitory convolutional neural network for gender classification," in *Proc. 18th Intern. Conf. Pattern Recognition (ICPR-2006)*, vol. 4, 2006, pp. 421–424.
- [7] —, "A brain-inspired visual pattern recognition architecture and its applications," in *Pattern Recognition Technologies and Applications: Recent Advances*, B. Verma and M. Blumenstein, Eds. IGI Global Press, 2008.



(a) Skin classification task



(b) Image classification task

Fig. 2. The training time of the standard and modified algorithms: (a) skin classification (b) image classification.

- [8] S. Phung and A. Bouzerdoum, "A pyramidal neural network for visual pattern recognition," *IEEE Transactions on Neural Networks*, vol. 18, no. 2, pp. 329–343, March 2007.
- [9] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
- [10] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.
- [11] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 30, no. 4, pp. 451–462, Nov. 2000.
- [12] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive algorithms and stochastic approximations*. Berlin: Springer Verlag, 1990.
- [13] C. E. Vivaracho, J. Ortega-Garcia, L. Alonso, and Q. I. Moro, "Extracting the most discriminant subset from a pool of candidates to optimize discriminant classifier training," in *ISMIS*, 2003, pp. 640–645.
- [14] D. Saad, *On-line learning in neural networks*, 1st ed. Cambridge University Press, 1999.
- [15] R. O. Duda, P. E. Hart, and D. G. Stock, *Pattern Classification*, 2nd ed. New York: John Wiley and Son, Inc, 2001.
- [16] K. Fukumizu, "Statistical active learning in multilayer perceptrons," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 17–26, Jan. 2000.
- [17] M. Li and K. Senthil, "Confidence-based active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, Aug. 2006.
- [18] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.
- [19] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [20] J. Yu, "General c-means clustering model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1197–1211, Aug. 2005.
- [21] R. Xu and D. W. II, "Survey of clustering algorithm," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, May 2005.
- [22] F. Badran, M. Yacoub, and S. Thiria, "Self-organizing maps and unsupervised classification," in *Neural Networks: Methodology and Applications*. Berlin: Springer, 2005, pp. 379–442.
- [23] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *IEEE International Conference on Neural Networks*, vol. 1, 1993, pp. 586–591.
- [24] M. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, pp. 989–993, 1994.
- [25] S. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation using color pixel classification: analysis and comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148–154, Jan. 2005.