Brief Papers

Training Multilayer Perceptron Classifiers Based on a Modified Support Vector Method

J. A. K. Suykens and J. Vandewalle

***Abstract*— In this paper we describe a training method for one hidden layer multilayer perceptron classifier which is based on the idea of support vector machines (SVM's). An upper bound on the Vapnik–Chervonenkis (VC) dimension is iteratively minimized over the interconnection matrix of the hidden layer and its bias vector. The output weights are determined according to the support vector method, but without making use of the classifier form which is related to Mercer's condition. The method is illustrated on a two-spiral classification problem.**

***Index Terms*— Classification, multilayer perceptrons, support vector machines.**

I. Introduction

IT IS well known that multilayer perceptrons (MLP's) are universal in the sense that they can approximate any continuous nonlinear function arbitrarily well on a compact interval. As a result MLP's became popular in order to parametrize nonlinear models and classifiers, often leading to improved results compared to classical methods [1], [2], [5], [10], [16]. One of the major drawbacks is that for batch training of MLP's one usually solves a nonlinear optimization problem which has many local minima. Recently, support vector machines (SVM's) have been introduced for which classification and function estimation problems are formulated as quadratic programming (QP) problems [12]–[15]. The idea of SVM originates from finding an optimal hyperplane in order to separate two classes with maximal margin. It has been extended later to one-hidden layer multilayer perceptrons, radial basis function networks, and other architectures. Being based on the structural risk minimization principle and capacity concept with pure combinatorial definitions, the quality and complexity of the SVM solution does not depend directly on the dimensionality of the input space.

Manuscript received September 1, 1998; revised February 18, 1999. This work was carried out at the ESAT Laboratory and the Interdisciplinary Center of Neural Networks ICNN of the Katholieke Universiteit Leuven, Belgium, in the framework of the FWO project G.0262.97 *Learning and Optimization: An Interdisciplinary Approach*, the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology, and Culture (IUAP P4-02 & IUAP P4-24) and the Concerted Action Project MIPS (*Modelbased Information Processing Systems*) of the Flemish Community.

The authors are with the Department of Electrical Engineering, Katholieke Universiteit Leuven, ESAT-SISTA, Kardinaal Mercierlaan 94, B-3001 Leuven (Heverlee), Belgium. J. A. K. Suykens is also with the National Fund for Scientific Research FWO, Flanders.

Publisher Item Identifier S 1045-9227(99)05969-X.

However, taking the case of a MLP-SVM, only the output weights of the MLP are found by solving the QP problem. The interconnection matrix is directly related to the training data points itself, up to two additional constants. Hence the overall problem of finding the output weights together with these additional constants is in fact nonconvex. The number of hidden units follows from solving the QP problem and is equal to the number of support vectors. In this paper we describe a modified support vector method approach for training a MLP with a given number of hidden units. An upper bound on the Vapnik–Chervonenkis (VC) dimension is iteratively minimized over the interconnection matrix and the bias vector of the hidden layer. The output weights are determined according to the support vector method. We illustrate the method on a two-spiral benchmark problem. An advantage of this approach compared to backpropagation is the optimization of the generalization performance in terms of the upper bound on the VC dimension. In backpropagation one usually incorporates a regularization term (norm on interconnection weights vector or weight decay) in order to obtain an improved generalization performance, being related to the bias-variance tradeoff [1].

For MLP-SVM's Mercer's condition is not satisfied for all possible values of the hidden layer parameters and the SVM theory is less developed for this type of kernels than, e.g., for RBF kernels where additional links with regularization theory have been demonstrated [9]. The present method does not require the additional Mercer condition and could be applied to other activation functions than $\tanh$ such as circular units [7]. On the other hand for the QP subproblem the matrix is not guaranteed to be positive definite (which is related to the fact that the QP solution is global and unique [3]), but the overall design problem is nonconvex anyway. While SVM's have been successfully applied to large scale problems, the modified method is applicable to moderate size problems due to the fact that all the weights of the hidden layer have to be estimated instead of the two additional constants. Drawbacks of the proposed method are the high computational cost and the larger number of parameters in the hidden layer, compared to a standard SVM approach.

This paper is organized as follows. In Section II we review some basic facts about support vector machines for classification problems. In Section III we discuss the multilayer perceptron classifier with the modified support vector training method. In Section IV we give an example for a two-spiral classification problem.

1045–9227/99$10.00 © 1999 IEEE

II. Support Vector Method for Classification

In this section we shortly review some basic work on SVM's for classification problems. For more details we refer to [12]–[15].

Given a training set of N data points $\{y_k, x_k\}_{k=1}^N$, where $x_k \in \mathbb{R}^n$ is the kth input pattern and $y_k \in \mathbb{R}$ is the kth output pattern, the support vector method approach aims at constructing a classifier of the form

$$y(x) = \text{sign}\left[\sum_{k=1}^{N} \alpha_k y_k \psi(x, x_k) + b\right] \tag{1}$$

where α_k are positive real constants and b is a real constant. For $\psi(\cdot, \cdot)$ one typically has the following choices: $\psi(x, x_k) = x_k^T x$ (linear SVM); $\psi(x, x_k) = (x_k^T x + 1)^d$ (polynomial SVM of degree d); $\psi(x, x_k) = \exp\{-\gamma \|x - x_k\|_2^2\}$ (radial basis SVM), where γ is a positive real constant. This paper on the other hand is related to the two-layer neural SVM

$$\psi(x, x_k) = \tanh[\kappa x_k^T x + \nu] \tag{2}$$

where κ and ν are constants.

The classifier is constructed as follows. One assumes that

$$\begin{cases} w^T \varphi(x_k) + b \geq 1, & \text{if } y_k = +1 \\ w^T \varphi(x_k) + b \leq -1, & \text{if } y_k = -1 \end{cases} \tag{3}$$

which is equivalent to

$$y_k[w^T \varphi(x_k) + b] \geq 1, \qquad k = 1, \cdots, N \tag{4}$$

where $\varphi(\cdot)$ is a nonlinear function which maps the input space into a higher dimensional space. However, this function is not explicitly constructed. For the nonseparable case, variables ξ_k are introduced such that

$$\begin{cases} y_k[w^T \varphi(x_k) + b] \geq 1 - \xi_k, & k = 1, \cdots, N \\ \xi_k \geq 0, & k = 1, \cdots, N. \end{cases} \tag{5}$$

According to the structural risk minimization principle, the risk bound is minimized by formulating the optimization problem:

$$\min J(w, \xi_k) = \tfrac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k \tag{6}$$

subject to (5). Therefore one constructs the Lagrangian

$$\begin{aligned} &\mathcal{L}_1(w, b, \xi_k; \alpha_k, \gamma_k) \\ &\quad = \tfrac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k - \sum_{k=1}^{N} \alpha_k \{[w^T \varphi(x_k) + b] y_k \\ &\qquad - 1 + \xi_k\} - \sum_{k=1}^{N} \gamma_k \xi_k \end{aligned} \tag{7}$$

by introducing Lagrange multipliers $\alpha_k \geq 0$, $\gamma_k \geq 0$ $(k = 1, \cdots, N)$. The solution is given by the saddle point of the Lagrangian by computing

$$\max_{\alpha_k, \gamma_k} \min_{w, b, \xi_k} \mathcal{L}_1(w, b, \xi_k; \alpha_k, \gamma_k). \tag{8}$$

One obtains

$$\begin{cases} \dfrac{\partial \mathcal{L}_1}{\partial w} = 0 \;\rightarrow\; w = \displaystyle\sum_{k=1}^{N} \alpha_k y_k \varphi(x_k) \\ \dfrac{\partial \mathcal{L}_1}{\partial b} = 0 \;\rightarrow\; \displaystyle\sum_{k=1}^{N} \alpha_k y_k = 0 \\ \dfrac{\partial \mathcal{L}_1}{\partial \xi_k} = 0 \;\rightarrow\; 0 \leq \alpha_k \leq c, \, k = 1, \cdots, N \end{cases} \tag{9}$$

which leads to the solution of the following quadratic programming problem:

$$\begin{aligned} &\max_{\alpha_k} \mathcal{Q}_1(\alpha_k; \varphi(x_k)) \\ &\quad = -\tfrac{1}{2} \sum_{k, l=1}^{N} y_k y_l \, \varphi(x_k)^T \varphi(x_l) \, \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \end{aligned} \tag{10}$$

such that

$$\begin{cases} \displaystyle\sum_{k=1}^{N} \alpha_k y_k = 0 \\ 0 \leq \alpha_k \leq c, \, k = 1, \cdots, N. \end{cases}$$

The function $\varphi(x_k)$ in (10) is related then to $\psi(x, x_k)$ by imposing

$$\varphi(x)^T \varphi(x_k) = \psi(x, x_k) \tag{11}$$

which is motivated by Mercer's Theorem. For the two layer neural SVM, Mercer's condition only holds for certain parameter values of κ and ν.

The classifier (1) is designed by solving

$$\begin{aligned} &\max_{\alpha_k} \mathcal{Q}_1(\alpha_k; \psi(x_k, x_l)) \\ &\quad = -\tfrac{1}{2} \sum_{k, l=1}^{N} y_k y_l \psi(x_k, x_l) \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \end{aligned} \tag{12}$$

subject to the constraints in (10). Note that one does not have to calculate w nor $\varphi(x_k)$ in order to determine the decision surface. Because the matrix associated with this quadratic programming problem is not indefinite, the solution to (12) will be global [3].

Furthermore, for a fixed prespecified basis, one can show that hyperplanes (4) satisfying the constraint $\|w\|_2 \leq a$ have a VC-dimension h which is bounded by

$$h \leq \min([r^2 a^2], n) + 1 \tag{13}$$

where $[.]$ denotes the integer part and r is the radius of the smallest ball containing the points $\varphi(x_1), \cdots, \varphi(x_N)$. Finding this ball is done by defining the Lagrangian

$$\mathcal{L}_2(r, q, \lambda_k) = r^2 - \sum_{k=1}^{N} \lambda_k (r^2 - \|\varphi(x_k) - q\|_2^2) \tag{14}$$

where q is the center of the ball and λ_k are positive Lagrange multipliers. In a similar way as for (6) one finds that the center

is equal to $q = \sum_k \lambda_k \varphi(x_k)$, where the Lagrange multipliers follow from

$$\max_{\lambda_k} \mathcal{Q}_2(\lambda_k; \varphi(x_k)) = -\sum_{k,l=1}^{N} \varphi(x_k)^T \varphi(x_l)\, \lambda_k \lambda_l + \sum_{k=1}^{N} \lambda_k \varphi(x_k)^T \varphi(x_k) \quad (15)$$

such that

$$\begin{cases} \sum_{k=1}^{N} \lambda_k = 1 \\ \lambda_k \geq 0, \qquad k = 1, \cdots, N. \end{cases}$$

Based on (11), $\mathcal{Q}_2$ can also be expressed in terms of $\psi(x_k, x_l)$. Finally, one selects a support vector machine with minimal VC dimension by solving (12) and computing (13) from (15).

III. Modified Support Vector Method for MLP's

Instead of constructing the classifier (1) we are interested here in a classifier with $\varphi(x_k) = \tanh(Vx_k + \beta)$

$$y(x) = \text{sign}[w^T \tanh(Vx + \beta)] \text{ s.t.}$$
$$w = \sum_{k=1}^{N} \alpha_k y_k \tanh(Vx_k + \beta) \quad (16)$$

with $w \in \mathbb{R}^{n_h}$ the output weight vector, $V \in \mathbb{R}^{n_h \times n}$ the interconnection matrix for the hidden layer, and $\beta \in \mathbb{R}^{n_h}$ the bias vector, where n_h denotes the number of hidden units. The coefficients α_k are the solution to (10).

Two important differences between the classifiers (1) and (16) are

1) The number of hidden units in (16) is fixed beforehand, while for (1) this is equal to the number of support vectors, which follows from solving the QP problem (12). Both for (1) and (16) the nonzero α_k coefficients correspond to support vectors.
2) The classifier (1) is based on Mercer's condition, while (16) not. For MLP-SVM's Mercer's condition imposes additional constraints on the choice of κ and ν.

The training of the classifier is done as follows:

$$\min_{V,\beta} \|rw\|_2^2 \text{ such that}$$
$$\begin{cases} \mathbf{C1}. \text{ QP subproblem:} \\ \quad \begin{cases} w = \sum_{k=1}^{N} \alpha_k^* y_k \tanh(Vx_k + \beta) \\ \alpha_k^* = \arg\max_{\alpha_k} \mathcal{Q}_1(\alpha_k; \tanh(Vx_k + \beta)) \\ 0 \leq \alpha_k^* \leq c \end{cases} \quad (17) \\ \mathbf{C2}. \quad \|V(:); \beta\|_2 \leq \gamma \\ \mathbf{C3}. \quad r \text{ is radius of smallest ball containing } \{\tanh(Vx_k + \beta)\}_{k=1}^{N} \end{cases}$$

where $(:)$ denotes a columnwise scan of a matrix and γ is positive real constant. The cost function is related to the upper bound on the VC dimension (13).

Remarks:

- According to the structural risk minimization (SRM) principle a bound on the risk, being the sum of the empirical risk and the confidence interval, is minimized in this way. The generalization ability of learning machines depends on the capacity of a set of functions, characterized by the VC dimension. According to the SRM inductive principle a function with low capacity which describes the data well will have a good generalization, regardless the dimensionality of the space [15].
- SVM's are compared and selected based on the same criterion as (17), but usually by trying a number of possible choice for κ and ν [2]. Here it is formulated explicitly as an additional optimization problem in V, β. This is one of the underlying reasons for a higher computational cost.
- For the QP subproblem (**C1**) the matrix is not guaranteed to be positive definite (which is related to the fact that the QP solution is global and unique [3]), because Mercer's condition hasn't been imposed. On the other hand the overall design problem is nonconvex anyway. One may solve the constrained nonlinear optimization problem by sequential quadratic programming (SQP) [3], [4], where at a certain iteration step one solves the QP subproblem (**C1**) for given V, β values and the QP subproblem related to (**C3**). A QP method for large-scale problems has been discussed in [7].
- The condition (**C2**) is imposed because experiments show that omitting it may lead to very large weights for the hidden layer, due to (**C3**). This condition avoids that the hidden neurons are going to far in saturation. It can be considered as an additional regularization to the problem formulation.

IV. Example

Here we illustrate the modified support vector method for MLP classifiers on a two-spiral benchmark problem. The training data are shown on Fig. 1 with two classes indicated by $*$ and $\times$ (60 points with 30 for each class). Points in between the training data located on the two spirals are often considered as test data for this problem but are not shown on the figure. The generalization is clear by visual inspection from the decision boundaries shown on the figure.

A MLP with 20 hidden units is taken for the classification ($n_h = 20$). Note that this amounts into a parameter vector ($[w; V(:); \beta]$) of dimension 80, which is more than the number of training data points. The hidden layer V, β is randomly initialized by taking V and β normally distributed with zero mean and standard deviation 0.2. The corresponding classifier, obtained by computing w from (**C1**), has many training data which are misclassified. On the other hand, by optimizing this result further according to (17) one obtains a perfect classification on the training data with good generalization. The support vectors are indicated by big circles. During this optimization, one can observe that the number of support vectors for the optimized result is smaller than for the initial classifier, which is a well-known desirable feature in order to have a good generalization. For the simulations the parameters

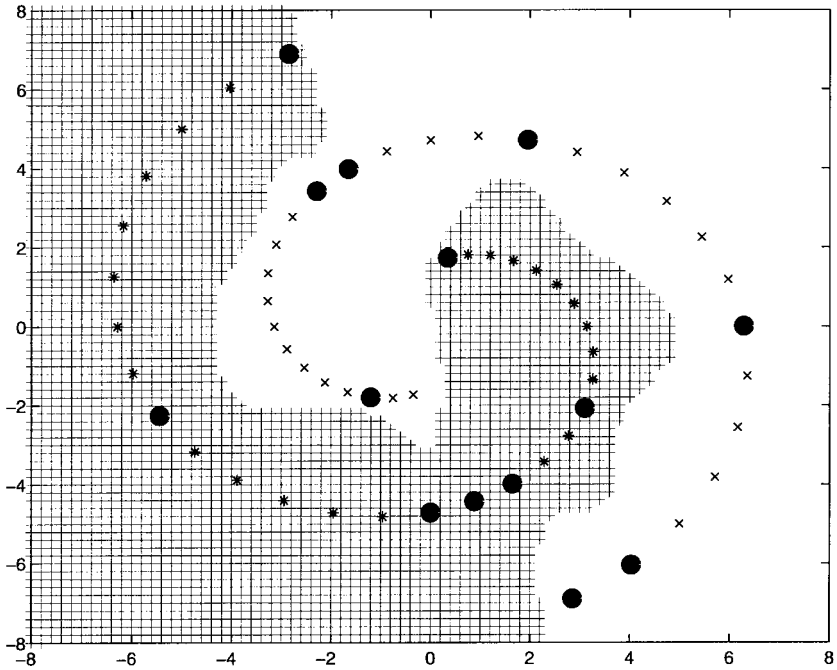

Fig. 1. Two-spiral classification problem: shown is the (x_1, x_2) plane with $x = [x_1; x_2]$ taken as input of a MLP classifier with 20 hidden units. The output weights of the MLP are found by solving the QP problem for given randomly chosen weights of the hidden layer. An optimization of the hidden layer is done by minimizing an upper bound on the VC dimension. All 60 training data (two classes indicated by $*$ and $\times$) are correctly classified. A good generalization is obtained, indicated also by the small amount of support vectors (big circles).

$\gamma = 5$ and $c = 500$ were taken. The choice of γ was *ad hoc* and needed for the algorithm in order to give meaningful solutions, rather than fine-tuning of generalization performance. Sequential quadratic programming has been applied (constr in Matlab) with numerical calculation for the gradient of the cost function and constraints. The constraint on the α_k^* coefficients has been realized by applying a saturation function to it at level 500. In order to speed up simulation results evaluations of the cost function and the QP problem have been programmed in C making use of the cmex facility.

A standard SVM approach for a $\tanh$ kernel (2), which is based upon (12) instead of (10), has been applied to this two-spiral problem for the typical choices $\kappa = 2/n = 1$, $\nu = -1$ and $\kappa = 2$, $\nu = 1$ from [12, pp. 143–145] and [2, p. 369], respectively. Both results lead to a larger number of support vectors and bad generalization. Finally, RBF kernels are a better choice for solving two-spiral problems as shown, e.g., in [11].

V. Conclusions

We described a modified support vector method for training a multilayer perceptron architecture with a given number of hidden units. Opposed to the standard SVM method, Mercer's condition is not applied for the present classifier. The hidden layer weights appear as additional parameters, while for standard MLP-SVM's there are only two additional parameters to be determined because the hidden-layer interconnection matrix is directly expressed in terms of the training data points. From the viewpoint of Mercer's condition, MLP-SVM's are less attractive because it is not sufficiently understood for which values of the hidden layer parameters the condition is satisfied. This is a main motivation why we selected MLP classifiers for the discussion. However, the ideas of this paper can also be applied to RBF kernels and in fact to all types of activation functions as there are no restrictions imposed by Mercer's condition. The main result that we want to stress is that a classical MLP classifier can be trained based upon a support vector method. A good generalization is obtained by optimizing a bound on the VC dimension, even if there are more interconnection weights than the number of training data points. Drawbacks of the proposed method in comparison to a standard SVM approach are the higher computational cost, the larger number of parameters in the hidden layer and a choice for the number of hidden units. The proposed method is intended to be complementary to existing work on SVM's for cases where it is difficult to exploit Mercer's condition or in case the architecture of the classifier is given but one would still be interested to apply SVM methodology.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford, U.K.: Oxford Univ. Press, 1995.
[2] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory and Methods.* New York: Wiley, 1998.
[3] R. Fletcher, *Practical Methods of Optimization.* New York: Wiley, 1987.
[4] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization.* London, U.K.: Academic, 1981.
[5] S. Haykin, *Neural Networks: A Comprehensive Foundation.* New York: Macmillan, 1994.
[6] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Proc. NNSP'97*, Amelia Island, FL.
[7] S. Ridella, S. Rovetta, and R. Zunino, "Circular backpropagation networks for classification," *IEEE Trans. Neural Networks*, vol. 8, pp. 84–97, 1997.

[8] B. Schölkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Processing*, vol. 45, pp. 2758–2765, 1997.

[9] A. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, pp. 637–649, 1998.

[10] J. A. K. Suykens, J. Vandewalle, and B. De Moor, *Artificial Neural Networks for Modeling and Control of Nonlinear Systems.* Boston, MA: : Kluwer, 1996.

[11] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Lett.*, to be published.

[12] V. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.

[13] ———, *Statistical Learning Theory.* New York: Wiley, 1998.

[14] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Advances in Neural Information Processing Systems.* Cambridge, MA: MIT Press, 1997, vol. 9.

[15] V. Vapnik, "The support vector method of function estimation," in *Nonlinear Modeling: Advanced Black-Box Techniques*, J. A. K. Suykens and J. Vandewalle, Eds. Boston, MA: Kluwer, 1998, pp. 55–85.

[16] J. M. Zurada, *Introduction to Artificial Neural Systems.* St. Paul, MN: West, 1992.