Neurocomputing 152 (2015) 369–376

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A neural network method for solving support vector classification problems

Alireza Nazemi, Mehran Dehghan *

Department of Mathematics, School of Mathematical Sciences, University of Shahrood, P.O. Box 3619995161-316, Shahrood, Iran

ARTICLE INFO

Article history:
Received 24 March 2014
Received in revised form
15 July 2014
Accepted 21 October 2014
Communicated by Y. Liu
Available online 15 November 2014

Keywords:
Neural network
Support vector classification
Quadratic programming
Convergent
Stability

ABSTRACT

This paper presents a recurrent neural network to support vector machine (SVM) learning in pattern classification arising widespread applications in a variety of setting. The SVM learning problem in classification is first converted into an equivalent quadratic programming (QP) formulation, and then a recurrent neural network for SVM learning is proposed. The proposed neural network is guaranteed to obtain the optimal solution of support vector classification. It is also shown that the proposed neural network model is stable in the sense of Lyapunov and it is globally convergent to an exact optimal solution of the QP problem. Several illustrative examples are provided to show the feasibility and the efficiency of the proposed method in this paper.

1. Introduction

It is well known that support vector classification (SVC) problems arise in a wide variety of scientific and engineering applications [1–9]. In many engineering and scientific applications, such as classification in complex electromagnetic environments and recognition in medical diagnostics radar object recognition in strong background clutter [10], real-time online solutions of the SVC problems are often desired. Over the years, a variety of numerical optimization algorithms for SVM learning have been proposed [11–17]. However, these traditional algorithms may not be applicable for digital computers since the computing time required for a solution is greatly dependent on the dimension and the structure of the problem and the complexity of the algorithm used. For such real-time applications, recurrent neural networks based on hardware implementation are more competent [18,19]. Compared with traditional numerical optimization algorithms, the neural network approach has several potential advantages in real-time applications. First, the structure of a neural network can be implemented effectively using very large scale integration (VLSI) and optical technologies. Second, neural networks can solve many optimization problems with time-varying parameters. Third, the dynamical techniques and the numerical ODE techniques can be applied directly to the continuous-time neural network for solving constrained optimization problems effectively.

The main idea of the neural network approach for an optimization problem is to construct a nonnegative energy function and establish a dynamic system that represents an artificial neural network. The dynamic system is usually in the form of first-order ordinary differential equations. Furthermore, it is expected that the dynamic system will approach its static state (or equilibrium point), which corresponds to the solution for the underlying optimization problem, starting from an initial point. An important requirement is that the energy function decreases monotonically as the dynamic system approaches an equilibrium point. Because of the dynamic nature and the potential of electronic implementation, neural networks can be implemented physically by designated hardware such as application specific integrated circuits, where the computational procedure is truly distributed and parallel [20]. It is well known that the real-time processing ability of neural networks is one of their most important advantages [18].

In the past two decades, recurrent neural networks for optimization and their engineering applications have been widely investigated [21–38,40–47]. In particular, neural networks for solving SVMs, which can be modeled as a quadratic optimization problem, have been rather extensively studied and some important results have also been obtained (see [21,36,39,41–43,47]). These papers present several recurrent neural networks to train SVMs for classification. These models can also reach the exact saddle point of the QP problem in SVC problems and are globally stable in the Lyapunov sense. But some of their structure are rather complicated and further simplification can

* Corresponding author. Tel./fax: +98 23 32300235.
E-mail addresses: nazemi20042003@yahoo.com (A. Nazemi), mehrandgn91@gmail.com (M. Dehghan).

http://dx.doi.org/10.1016/j.neucom.2014.10.054
0925-2312/

be achieved. Moreover, the QP formulation of SVC problems may not be strictly convex in many applications (see Remark 2.1). There exist some other networks which can potentially train SVMs for classification without strict convexity assumption (e.g., [22–24]). Thus, on the basis of the above notations, proposing an efficient neural network for solving the SVC problems with a simple structure, good stability and convergence results is very necessary and meaningful.

With motivation from the above discussions, in this paper, we proposed a suitable neural network for analog hardware implementation which gives a good solution of SVC learning. According to the Karush–Kuhn–Tucker (KKT) optimality conditions of convex programming, a neural network model for solving the QP formulation of SVC will be proposed. The equilibrium point of the proposed neural network will be proved to be equivalent to the KKT point of the QP formulation. The existence and uniqueness of an equilibrium point of the proposed neural network will also be analyzed. By constructing a suitable Lyapunov function, a sufficient condition to ensure the existence and global asymptotical stability for the unique equilibrium point of the proposed neural network will be obtained. The proposed neural network model has also been successfully used for solving the minimax problem [31], geometric programming problem [32], maximum flow problem [34] and shortest path problem [35].

The remainder of this paper is organized as follows. In Section 2, we briefly describe the learning problem of SVMs for classification. In Section 3, the system model for SVC problem and some necessary preliminaries are given. The stability and convergence of the proposed neural network are studied in Section 4. Numerical simulations are provided in Section 5. Finally, some concluding remarks are given in Section 6.

2. Problem setting

Given a training set of input–target pairs

$$\mathcal{D}=\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\},$$

where the ith sample $x_i\in\mathbb{R}^n$ (n is the dimension of the input space) belongs to two separate classes labeled by $y_i\in\{-1,1\}$. The classification problem is to find a hyperplane in a high dimensional feature space Z, which divides the set of examples in the feature space such that all the points with the same label are on the same side of the hyperplane [1,2].

The SVM classification technique requires the solution of the following convex quadratic programming problem [3]:

$$\text{minimize}\quad \tfrac{1}{2}\alpha^T G\alpha-e^T\alpha \tag{1}$$

$$\text{subject to}\quad \begin{cases} y^T\alpha=0,\\ 0\le\alpha\le ce, \end{cases} \tag{2}$$

where $\alpha\in\mathbb{R}^N$, G is a $N\times N$ positive semidefinite matrix, $e\in\mathbb{R}^N$ is the vector of all ones, $y\in\{-1,1\}^N$ and the parameter $c>0$ is a user-defined constant to control the tradeoff between the maximization of the margin and the minimization of errors. The generic element g_{ij} of the matrix G is given by $y_iy_jK(x_i,x_j)$, where $K(x_i,x_j)=\phi(x_i)\phi(x_j)$ is called kernel function related to the nonlinear function ϕ that maps the data from the input space into the feature space. The most widely used kernels are the following:

- linear: $K(x,y)=x^Ty$;
- polynomial: $K(x,y)=(\gamma x^Ty+r)^d$, with $\gamma>0$;
- Gaussian: $K(x,y)=\exp(-\|x-y\|^2/2\delta^2)$, with $\delta>0$;

where γ,r,d,δ are kernel parameters.

The support vector classification technique is to find an optimal decision function of the classifier given by

$$f(x)=\operatorname{sgn}\left[\sum_{i=1}^{N}\alpha_iy_i.K(x_i,x)+\beta\right],$$

where $\{\alpha_i\}$ is an optimal solution of the QP (1) and (2) and β is given by an average over all of the support vectors in S as

$$\beta=\frac{1}{N_s}\sum_{s\in S}\left(y_s-\sum_{m\in S}\alpha_my_mK(x_m,x_s)\right), \tag{3}$$

where S is determined as the set of support vectors by finding the indices such that $0<\alpha\le ce$ (see [4]). Therefore, the learning problem in SVC is equivalent to the QP problem in (1) and (2) with N bounded variables. The problem (1) and (2) is referred to as the dual formulation of the SVM.

Remark 2.1 (*Yang et al. [47]*). In many applications, G may not be positive definite. For instance, in linear SVM, $K(x_i,x_j)=x_i^Tx_j$ and G can be written as AA^T where $A=(y_1x_1,y_2x_2,\ldots,y_Nx_N)^T\in\mathbb{R}^{N\times M}$. When $N>M$, which is often the case in practice, $rank(G)<N$ and G is a positive semidefinite only. For another instance, it is easy to show that when there are repeated samples in the training set, G is singular and thus positive semidefinite only.

3. A neurodynamic model

In this section, based on an equivalent QP formulation in SVC, we propose a neural network for the SVC problem. Thus, we consider a general form of the quadratic programming problems given by

$$\text{minimize}\quad \tfrac{1}{2}x^TQx+D^Tx \tag{4}$$

subject to

$$Ax-b\le 0, \tag{5}$$

$$Ex-f=0, \tag{6}$$

where $Q\in\mathbb{R}^{n\times n}$ is only a symmetric and positive semidefinite matrix i.e. $\nu^TQ\nu\ge 0,\forall\nu\ (\ne 0)\in\mathbb{R}^n, A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$, $E\in\mathbb{R}^{l\times n}$, $f\in\mathbb{R}^l, x\in\mathbb{R}^n$ and $\operatorname{rank}(A/E)=m+l$ $(0\le m,l<n)$.

Theorem 3.1 (*Bazaraa et al. [48]*). *$x^*\in\mathbb{R}^n$ is an optimal solution of (4)–(6) if and only if there exist $u^*\in\mathbb{R}^m$ and $v^*\in\mathbb{R}^l$ such that $(x^{*T},u^{*T},v^{*T})^T$ satisfies the following Karush–Kuhn–Tucker (KKT) system:*

$$\begin{cases} u^*\ge 0,\quad Ax^*-b\le 0,\quad u^{*T}(Ax^*-b)=0,\\ Qx^*+D+A^Tu^*+E^Tv^*=0,\\ Ex^*-f=0. \end{cases} \tag{7}$$

Theorem 3.2 (*Bazaraa et al. [48]*). *x^* is an optimal solution of (4)–(6), if and only if x^* is a KKT point of (4)–(6).*

Now, let $x(\cdot),u(\cdot)$ and $v(\cdot)$ be some time dependent variables. A neural network model for solving (4)–(6) and its dual is proposed as

$$\frac{dx}{dt}=-(Qx+D+A^T(u+Ax-b)^+ +E^Tv), \tag{8}$$

$$\frac{du}{dt}=(u+Ax-b)^+-u, \tag{9}$$

$$\frac{dv}{dt}=Ex-f, \tag{10}$$