Low-complexity, linear circuit implementation of support vector machines training

S. Decherchi, P. Gastaldo, M. Parodi and R. Zunino

A sparse, positive-definite kernel function is adopted as the basis for the formulation of a simple support vector machine (SVM) model in which the bias term is removed. The resulting functional to be minimised in the SVM training process is shown to be equivalent to the potential function (co-content) of a linear, resistive circuit featuring low complexity.

Introduction: The support vector machine (SVM) model [1] is a very effective methodology for tackling classification problems in complex, nonlinear data distributions. The success of SVMs in real-world domains motivates the ongoing research towards hardware implementations, and programmable logic devices are quite popular for the realisation of off-line trained SVMs [2].

Digital technologies, however, may prove inefficient when supporting systems that require on-line training capabilities. This Letter presents an innovative circuit-design method for the hardware support of SVM training, and shows that by a sparse, positive-definite kernel one can map efficiently the learning process into a linear, low-complexity, resistive circuit.

Issues in circuit support to SVM training: The training process of SVMs, the model of which derives from statistical learning theory [1], can be stated as a convex quadratic programming (QP) problem; it requires the minimisation of a quadratic functional, subject to linear equality and inequality constraints. Let $Z = \{(\boldsymbol{x}_l, y_l);\ l = 1, \ldots, n_p;\ \boldsymbol{x}_l \in \Re^n,\ y_l \in \{-1, +1\}\}$ denote the (training) set of labelled patterns, belonging to two classes y_l; the resulting binary classification setting requires one to solve the QP problem:

$$\min_{\alpha}\left\{\frac{1}{2}\sum_{l,m=1}^{np}\alpha_l\alpha_m y_l y_m K(\boldsymbol{x}_l,\boldsymbol{x}_m) - \sum_{l=1}^{np}\alpha_l\right\}$$
$$\begin{cases} 0 \le \alpha_l \le C, \forall l \\ \sum_{l=1}^{np} y_l\alpha_l = 0 \end{cases} \tag{1}$$

where the scalar quantities α are the model parameters to be adjusted, the quantity C upper bounds the SVM parameters, and $K(\circ,\circ)$ is a kernel function, i.e. the basis for the SVM series expansion. SVM supports a linear class separation in a Hilbert space; the classification rule for a trained SVM is:

$$f(\boldsymbol{x}) = \sum_{l=1}^{np}\alpha_l y_l K(\boldsymbol{x},\boldsymbol{x}_l) + b \tag{2}$$

where b is a bias.

The functional (1) can be viewed as the co-content potential of a resistive circuit. This poses the basis for Chua and Lin's circuit for trained SVMs [3], which is extended to the hardware support of the learning phase in [4]. That approach exhibits two crucial drawbacks, which ultimately limit its overall effectiveness. First, Chua and Lin's circuit model cannot support the linear equality constraint shown in (1), and the trick of replacing the equality constraint with a pair of inequalities [4] may compromise the stability of the overall circuit. Secondly, the complexity of Chua and Lin's circuit model increases as the density of the kernel matrix, $K(\boldsymbol{x}_l, \boldsymbol{x}_m)$, increases.

The following discussion addresses both of these issues, and shows that the properties of positive-definite kernels allow one to set up a simpler model, leading to an efficient circuit support to the SVM training task.

Sparse kernels for efficient circuit support of SVM training: A theoretical result [5] allows one to overcome the equality-constraint issue: for any positive-definite kernel, the representer theorem [6] holds true for SVMs even in the absence of a bias term. As a consequence, choosing a positive-definite kernel takes out the bias term, b, from (2); it can be easily proved that this eliminates the linear constraint in (1), while the generalisation ability of the SVM remains unaffected.

When selecting the required positive-definite kernel function, however, one can take into account the second issue, i.e. circuit complexity. The present approach adopts a kernel formulation [7] that leads to sparse matrixes, and therefore attains the goal of limiting circuit complexity. The kernel function can be formalised as follows: let $d(\boldsymbol{x}_l, \boldsymbol{x}_m) = \|\boldsymbol{x}_l - \boldsymbol{x}_m\|^2$ denote the distance between points $(\boldsymbol{x}_l, \boldsymbol{x}_m)$, and let r be a cutoff distance. By definition, the kernel-based inner product, $K(\boldsymbol{x}_l, \boldsymbol{x}_m)$, nullifies whenever $d(\boldsymbol{x}_l, \boldsymbol{x}_m) \ge 2r$; otherwise, the quantity $K(\boldsymbol{x}_l, \boldsymbol{x}_m)$ is worked out by means of a recursive procedure:

$$\begin{aligned} k_{n,1}(\boldsymbol{x}_l,\boldsymbol{x}_m) &= 1 - \frac{\|\boldsymbol{x}_l - \boldsymbol{x}_m\|}{2r} \\ k_{n,2}(\boldsymbol{x}_l,\boldsymbol{x}_m) &= \arccos\left(\frac{\|\boldsymbol{x}_l - \boldsymbol{x}_m\|}{2r}\right) - \frac{\|\boldsymbol{x}_l - \boldsymbol{x}_m\|}{2r} \\ &\quad \times \sqrt{1 - \left(\frac{\|\boldsymbol{x}_l - \boldsymbol{x}_m\|}{2r}\right)^2} \\ k_{n,j}(\boldsymbol{x}_l,\boldsymbol{x}_m) &= \frac{j-1}{j} k_{n,j-2}(\boldsymbol{x}_l,\boldsymbol{x}_m) \\ &\quad - \frac{1}{j}\frac{\|\boldsymbol{x}_l - \boldsymbol{x}_m\|}{2r}\left(1 - \left(\frac{\|\boldsymbol{x}_l - \boldsymbol{x}_m\|}{2r}\right)^2\right)^{j-1/2} \end{aligned} \tag{3}$$

Recursions stop when the index, j, reaches the dimension, n, of the original data space, hence: $K(\boldsymbol{x}_l, \boldsymbol{x}_m) \equiv k_{n,n}(\boldsymbol{x}_l, \boldsymbol{x}_m)$. As shown in [7], this kernel ensures good prediction performances while yielding sparse kernel matrixes. From a circuit-design perspective, it can be proved that the above kernel-based SVM training process directly maps into a simple co-content minimising circuit. By assuming a voltage reference, V_0, and a reference resistance, R_0, one implements the terms α_l as v_l/V_0 and reformulates the overall SVM function as the equivalent co-content function:

$$\min_{v}\left\{\frac{1}{2R_0}\sum_{l,m=1}^{np} v_l v_m y_l y_m K(\boldsymbol{x}_l,\boldsymbol{x}_m) - \frac{V_0}{R_0}\sum_{l=1}^{np}\alpha_l\right\} \tag{4}$$

subject to the set of 'box' constraints, $0 \le v_l \le CV_0, \forall l$.

The first sum is the co-content function of a multi-terminal linear resistor G, the conductance matrix elements of which are $G_{lm} = y_l y_m K(\boldsymbol{x}_l, \boldsymbol{x}_m)$. Each term in the second sum is the co-content of constant current sources, V_0/R_0. The 'box' constraints on the set of voltages, v_l, is satisfied by as many identical, nonlinear resistors (diodes) the characteristic of which is represented in Fig. 1*a*. By connecting the n_p terminals to (identical) capacitors C as per Fig. 1*b*, one allows the free voltages, $v_1, \ldots, v_{np}$, to reach their stable values imposed by:

$$\sum_{l=1}^{np} y_k y_i K(\boldsymbol{x}_k,\boldsymbol{x}_i) v_l = V_0 \quad 0 \le v_1 \le V_0 C \tag{5}$$

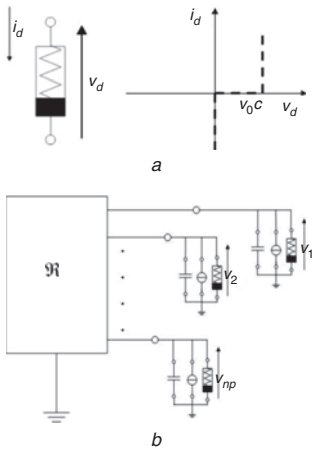

Fig. 1 *Nonlinear resistor characteristic and global circuital structure*

a Nonlinear resistor characteristic
b Global circuital structure

Experimental results: The well-known Sonar dataset from UCI repository provided a complex testbed for evaluating the proposed circuit implementation; the dataset includes 208 patterns evenly distributed

over two classes. To verify the generalisation performance of the circuit support, the dataset was randomly split into a training set holding 150 patterns, and a test set including the remaining 58 patterns. The circuit-design parameters C, V_0, R_0, and I_0 were set as follows: $C = 1$ nF, $V_0 = 1$ V, $R_0 = 1$ kΩ and $I_0 = 1$ mA. In the graph in Fig. 2, the x-axis gives the cutoff distance, r, and three different quantities are reported on the co-ordinate axis: 1. circuit complexity, measured by a scalar in the range [0,1] that quantifies the sparseness of the kernel matrix; 2. 'software' classification accuracy, measured as the number of classification errors scored by an ideal SVM model (including, bias) on the test set; 3. 'hardware' classification accuracy, measured as the number of classification errors (test set) scored by the circuit implementation described in this Letter. All values were measured upon completion of a transient interval of 80 μs.

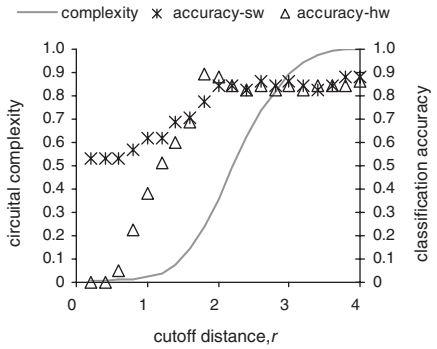

Fig. 2 *Circuital complexity and classification accuracies against cutoff distance*

Stars denote classification accuracies obtained with software implementation of the SVM model; triangles denote classification accuracies obtained with circuital implementation of SVM model

Remarkably, numerical results show that the performance of the circuit implementation of the SVM model decays only when the circuit complexity is lower than 0.2. This occurs when the cutoff distance, r, yields a kernel matrix such that more than 80% of the matrix elements are null. The lack of a bias term in the circuit setup determines the degradation in generalisation performance as compared with the ideal model; at the same time, one might observe that such a degradation begins in a region of the graph in which circuit complexity is sufficiently low.

Conclusion: This Letter presents an effective circuit implementation of the SVM training process. In the proposed approach, theoretical properties of the SVM paradigm have been exploited to efficiently map the training process on a linear, resistive circuit characterised by low complexity.

29 June 2008
Electronics Letters online no: 20082210
doi: 10.1049/el:20082210

S. Decherchi, P. Gastaldo, M. Parodi and R. Zunino (*Department of Biophysics and Electronics (DIBE), University of Genoa, Genova, Italy*)

E-mail: paolo.gastaldo@unige.it

References

1 Vapnik, V.N.: 'Statistical learning theory' (Wiley, 1998)
2 Anguita, D., Boni, A., and Ridella, S.: 'A digital architecture for support vector machines: theory, algorithm and FPGA implementation', *IEEE Trans. Neural Netw.*, 2003, **14**, pp. 993–1009
3 Chua, L.O., and Lin, G.N.: 'Non linear programming without computation', *IEEE Trans. Circuits Syst.*, 1984, **31**, pp. 182–188
4 Anguita, D., Ridella, S., and Rovetta, S.: 'Circuital implementation of support vector machines', *Electron. Lett.*, 1998, **34**, pp. 1596–1597
5 Poggio, T., Mukherjee, S., Rifkin, R., Rakhlin, A., and Verri, A.: 'B', 2001, AI Memo 2001-011 CBCL Memo 198, Massachutes Institute of Technology
6 Scholkopf, B., Herbrich, R., and Smola, A.J.: 'A generalized representer theorem', *Neural Netw. Comput. Learn. Theory*, 2001, **81**, pp. 416–426
7 Boughorbel, S., Tarel, J., Fleuret, F., and Boujemaa, N.: 'GCS kernel for SVM-based image recognition'. Proc. Int. Conf. on Artificial Neural Networks,ICANN 2005, Warsaw, Poland, September 2005, Vol. II, pp. 595–600