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 equalities and inequalities [4] may compromise the stability of the overall circuit. First, Chua and Lin’s circuit for trained SVMs [3], which is extended to the hardware support of the 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, \( K(x_i, x_m) \), 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 representor 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 matrices, and therefore attains the goal of limiting circuit complexity. The kernel function can be formalised as follows: let \( d(x_i, x_m) = ||x_i - x_m||^2 \) denote the distance between points \( (x_i, x_m) \), and let \( r \) be a cutoff distance. By definition, the kernel-based inner product, \( K(x_i, x_m) \), nullifies whenever \( d(x_i, x_m) \geq 2r \); otherwise, the quantity \( K(x_i, x_m) \) is worked out by means of a recursive procedure:

\[
k_{n,1}(x_i, x_m) = 1 - \frac{||x_i - x_m||}{2r}
k_{n,2}(x_i, x_m) = \arccos \left( \frac{||x_i - x_m||}{2r} \right) \times \sqrt{1 - \left( \frac{||x_i - x_m||}{2r} \right)^2}
k_{n,j}(x_i, x_m) = \frac{-1}{j} k_{n,j-2}(x_i, x_m) - \frac{1}{j} ||x_i - x_m|| \left( 1 - \left( \frac{||x_i - x_m||}{2r} \right)^2 \right)^{j/2}
\]

Recurions stop when the index, \( j \), reaches the dimension, \( n \), of the original data space, hence: \( K(x_i, x_m) = k_{n, n}(x_i, x_m) \). As shown in [7], this kernel ensures good prediction performances while yielding sparse kernel matrices. 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 \( a_i \) as \( v_i / V_0 \) and reformulates the overall SVM function as the equivalent co-content function:

\[
\min_v \left\{ \sum_{l=1}^{n_p} v_l y_l K(x_i, x_m) - \frac{V_0}{R_0} \sum_{l=1}^{n_p} a_i \right\}
\]

subject to the set of ‘box’ constraints, \( 0 \leq v_l \leq CV_0 \), \( \forall l \).

The first sum is the co-content function of a multi-terminal linear resistor \( G \), the conductance matrix elements of which are \( G_{mn} = y_m K(x_i, x_m) \). 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. 1a. By connecting the \( n_p \) terminals to (identical) capacitors \( C \) as per Fig. 1b, one allows the free voltages, \( v_l \), to reach their stable values imposed by:

\[
\sum_{l=1}^{n_p} v_l y_l K(x_i, x_m) = V_0 \quad 0 \leq v_l \leq CV_0
\]

Fig. 1 Nonlinear resistor characteristic and global circuitual structure
\( a \) Nonlinear resistor characteristic
\( b \) Global circuitual 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.

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.

© The Institution of Engineering and Technology 2008
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