Neural Networks 23 (2010) 607–613

Contents lists available at ScienceDirect

Neural Networks

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

A neural support vector machine

Magnus Jändel *

Agora for Biosystems, Box 57 SE-193 22, Sigtuna, Sweden
Swedish Defence Research Agency, SE-164 90, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 24 November 2008
Received in revised form 4 October 2009
Accepted 2 January 2010

Keywords:
Support vector machine
Neural systems
Pattern recognition
Perceptual learning
Associative memory
Olfactory system

ABSTRACT

Support vector machines are state-of-the-art pattern recognition algorithms that are well founded in optimization and generalization theory but not obviously applicable to the brain. This paper presents Bio-SVM, a biologically feasible support vector machine. An unstable associative memory oscillates between support vectors and interacts with a feed-forward classification pathway. Kernel neurons blend support vectors and sensory input. Downstream temporal integration generates the classification. Instant learning of surprising events and off-line tuning of support vector weights trains the system. Emotion-based learning, forgetting trivia, sleep and brain oscillations are phenomena that agree with the Bio-SVM model. A mapping to the olfactory system is suggested.

1. Introduction

Human and animal brains excel in complex classifications. Friend or foe? Edible or poisonous? Survival depends on such quick appraisals. How does the brain implement trainable general-purpose classifiers that learn instantly, yet avoid overwriting relevant lessons and match outputs to appropriate behaviours?

Instant learning is vital in an unforgiving environment. Overwriting or diluting old but still valid experiences is dangerous. Yet there is not enough memory or search resources for remembering everything. Memory must be managed so that vital knowledge is conserved while trivial experiences are discarded. Connecting the output of plastic neural classifiers to predetermined behavioural triggers is crucial. Predator scent detection must for example be coupled to flight behaviour. Haberly (2001) found, however, that biologically plausible algorithms for trainable pattern recognition generating predetermined output codes are in short supply.

This paper introduces Bio-SVM, a biologically feasible support vector machine that instantly learns surprising examples, forgets trivial examples and trains an optimal generalizing classifier with predetermined output codes. Bio-SVM is consistent with the observed mix of fast and slow brain oscillations and maps well to the architecture of the olfactory system.

The generic pattern recognition task is to classify a test sample by generalizing from known classifications of training examples. We shall only consider binary classifications. Multi-value classifications can readily be produced by a bank of binary classifiers. The training set S consists of examples $(\mathbf{x}, y)$, where $\mathbf{x}$ is a real-valued input vector and $y \in \{+1, -1\}$ indicates the correct classification. Bold letters signify vector quantities. The training examples are presented in a batch or more realistically one-by-one in online learning.

Support vector machines (SVMs) (see Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002 for reviews) have recently emerged as a strong alternative for any classification application. An SVM works by projecting input vectors $\mathbf{x}$ to a high-dimensional feature space. Features $\boldsymbol{\phi}(\mathbf{x})$ are typically non-linear functions of the input vector. The training algorithm finds a hyperplane in feature space separating positive cases from negative cases with maximum margin. The set of feature-space hyperplanes provides the broad hypothesis domain that is vital for solving substantial classification tasks. Enforcing maximal margins ensures a generalization performance that is optimal in a certain well-defined sense. The key insight of SVM pioneers Boser, Guyon, and Vapnik (1992) is that the SVM optimization problem can be solved without explicitly constructing the feature space.

The solution to a classification problem is the set of support vectors **SV**. Each support vector $\mathbf{x}_i$ is drawn from the training examples and has an associated positive real-valued weight α_i. The support vectors are borderline members of the training data used for defining the partitioning feature-space hyperplane. Positive support vectors are close to the negative domain. Negative support vectors are similarly bordering to the positive realm. An SVM classifies test samples $\mathbf{x}$ using a real-valued classification function $f(\mathbf{x})$. The test sample belongs to the negative class $y = -1$ if $f(\mathbf{x}) < 0$ and to the

* Corresponding address: Vårvägen 10, SE-19460 Upplands Väsby, Sweden. Tel.: +46 709277264; fax: +46 855503700.
E-mail addresses: magnus@jaendel.se, magjan@foi.se.

0893-6080/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2010.01.002

positive class $y = 1$ otherwise. The classification function is

$$f(\mathbf{x}) = \sum_{i \in \mathbf{SV}} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{1}$$

where b is a bias parameter. The positive definite kernel function K defines the implicit projection to feature space. For a given pair of input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, $K(\mathbf{x}_i, \mathbf{x}_j)$ is a measure of alignment in feature space.

2. The Bio-SVM model

SVMs are correctly viewed as founded on rigorous mathematics rather than biological analogies. Solution algorithms suggest implementation in a digital computer. There is, however, one aspect of SVMs that stands out as similar to biological systems. An SVM ignores typical examples but pays attention to borderline cases and outliers. It remembers surprises and forgets run-of-the-mill events. Life learns also from odd emotionally charged events. We remember the support vectors. Given their mathematical soundness, efficiency and a certain high-level similarity to biological learning, could SVMs be implemented in the brain?

This section casts the abstract SVM concept into a form that can be implemented by biological neural systems—the Bio-SVM model. This hypothesis is then compared to the gross features of brain pattern recognition systems.

2.1. Zero-bias ν-SVM

We must first find a formal SVM model that is malleable to neural form. The base-line is ν-SVM (Schölkopf, Smola, Williamson, & Bartlett, 2000), a soft-margin SVM in which a dimensionless parameter $0 < \nu < 1$ controls the trade-off between generalization and accuracy. Soft margin means that outlier support vectors may violate margins. Such mavericks are expected in noisy training sets. Schölkopf et al. (2000) show that ν is an upper bound on the fraction of margin errors. The ν-SVM model is solved, for a set of m training examples, by maximizing the dual objective function,

$$W(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

subject to

$$0 \le \alpha_i \le \frac{1}{m} \tag{3}$$

and

$$\sum_{i=1}^{m} \alpha_i \ge \nu. \tag{4}$$

The classification function is defined by Eq. (1). The solution to this problem is the optimal feature-space hyperplane.

We use a modified version of ν-SVM in which the bias parameter in Eq. (1) is set to zero. This is achieved by embedding the feature space vector $\boldsymbol{\phi}(\mathbf{x})$ of the original problem in a larger space $\{\boldsymbol{\phi}(\mathbf{x}), \tau\}$, thus increasing the dimensionality by one (Cristianini & Shawe-Taylor, 2000). This operation corresponds to replacing the old kernel K with a new kernel $K + \tau^2$. Removing the bias means less freedom for optimization and thus potentially smaller feature space margin, leading to reduced generalization performance (see Cristianini & Shawe-Taylor, 2000, p. 131). As explained in Section 2.5, it is, however, an essential simplification for mapping the model to a biological substrate.

The solution of the ν-SVM problem in Eqs. (2)–(4) is the weight vector $\boldsymbol{\alpha}$. We need a solution algorithm that is suitable for physiological modelling even if it may be suboptimal as a serial computer algorithm. We note that in general there exists an optimal solution in the α-space hyperplane,

$$\sum_{i=1}^{m} \alpha_i = \nu, \tag{5}$$

(Chang & Lin, 2001). The strategy is to start at an arbitrary point in the allowed domain of the α-hyperplane, e.g. by initializing all α_i to ν/m, and then follow the projection of the gradient of $W(\boldsymbol{\alpha})$ in the α-hyperplane until an optimum is found.

The gradient projection is

$$\mathbf{grad_P}(W) = \mathbf{grad}(W) - \mathbf{e}_\perp(\mathbf{grad}(W) \cdot \mathbf{e}_\perp), \tag{6}$$

where $\mathbf{e}_\perp$is the unit normal vector of the α-hyperplane and $\mathbf{grad}(W) = (\frac{\partial W}{\partial \alpha_1}, \frac{\partial W}{\partial \alpha_2}, \ldots, \frac{\partial W}{\partial \alpha_m})$. The ith gradient component is

$$\frac{\partial W}{\partial \alpha_i} = -y_i \sum_{j=1}^{m} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) = -y_i f(\mathbf{x}_i) = -marg_i, \tag{7}$$

where $marg_i$ is the margin in feature space between the example and the classification hyperplane. A positive margin means that the example is classified correctly. The ith component of the gradient projection is, therefore,

$$\mathbf{grad_P}(W)_i = \langle marg \rangle - marg_i, \tag{8}$$

where $\langle marg \rangle = \frac{1}{m} \sum_{j=1}^{m} y_j f(\mathbf{x}_j)$ is the average margin. Each weight shall hence be updated in proportion to the difference between the average margin and the margin of the associated example. This rule strives to make the margin of each example equal to the average margin as the hypothesis $\boldsymbol{\alpha}$ progresses towards the optimum. It is not always possible to reach equality. The converged ν-SVM partitions the training examples into three distinct sets.

Trivial examples are non-support vectors. Their weights are driven to zero since the margin of such examples is larger than the average margin. Note that trivial examples can be removed from the training set once a solution has been found.

Outliers are possibly misclassified support vectors that consistently fall beyond of the average margin. Their weights are pushed to the maximum value $1/m$.

Regular support vectors converge to the average margin. Their weights fall within $0 < \alpha_i < 1/m$.

Eq. (2) is quadratic with respect to $\boldsymbol{\alpha}$ and the maximum is sought in the $\boldsymbol{\alpha}$-space hyperplane defined by Eqs. (3) and (5). This guarantees that there are no false maxima (see Chang & Lin, 2001 for a proof). It is hence easy to evaluate convergence. With plenty of time and computational resources one can simply move in very small steps along $\mathbf{grad_P}(W)$ until the maximum is found. The challenge is to find a biological apparatus that does just this.

2.2. The Bio-SVM concept

The general architecture and key operational processes for mapping zero-bias ν-SVM to brain systems are first outlined here and then detailed in the following sections. The main modules of the Bio-SVM are the Oscillating Memory (OM) for learning and storing support vectors and the Classification Pathway (CP) for performing classifications. The OM is the only plastic part of the system. The overall architecture is shown in Fig. 1.

The Bio-SVM executes three processes:

(1) **Classification**, where sensory inputs are classified.
(2) **Surprise learning**, where new training examples are engraved. A supervising unit, called the Critic, detects failed classifications and triggers the OM to remember the anomalous event.
(3) **Importance learning**, where trivial examples are forgotten and support vectors get optimal weights. The OM inputs training examples to the CP while the brain sleeps and adjusts weights according to resulting feedback. Examples are forgotten if weights consistently fall to zero.

Higher brain systems control which process to employ.

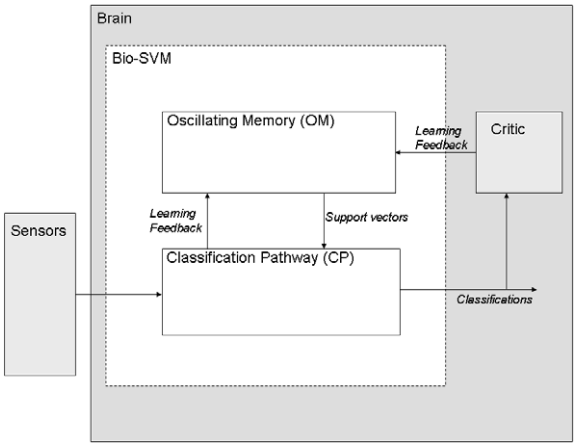

Fig. 1. Bio-SVM high-level architecture and interfaces. A Bio-SVM classifier (dashed box) interacts with surrounding systems shown in gray tone. It receives sensory inputs and outputs classifications. The Critic is an abstraction of higher brain systems providing feedback in case of misclassifications.

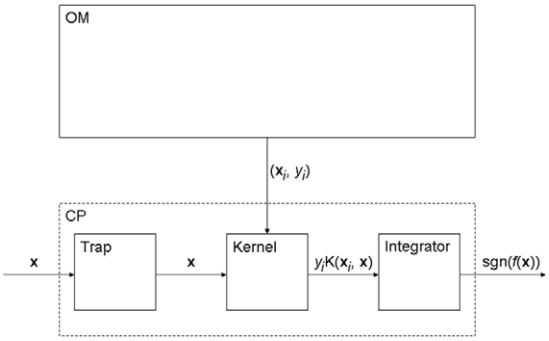

Fig. 2. Details of the Classification Pathway (CP) and signal flow in the classification process. OM is the Oscillating Memory. Sensor signal $\mathbf{x}$ is captured by the Trap. The OM outputs the support vector $(\mathbf{x}_i, y_i)$. K is the kernel function and f is the classification function according to Eq. (1).

2.3. Classification

The classification process operates when the animal is awake and captures sensory data. We assume that the system is fully trained so that the OM contains support vectors with appropriate weights. The CP consists of three sub-units: the Trap, the Kernel and the Integrator, connected as shown in Fig. 2.

Sensor signal $\mathbf{x}(t_0)$ is captured by the Trap at time t_0. This means that the Trap locks on the signal and outputs $\mathbf{x}(t_0)$ for a time T_{trap} however the input fluctuates. The Trap acts like a sensory memory stabilizing the input. After a time T_{trap}, the Trap will reset, capture the presently available incoming signal $\mathbf{x}(t_0 + T_{trap})$ and repeat the cycle.

The OM oscillates between support vector memories. It will at any given moment output support vector $(\mathbf{x}_i, y_i)$ with probability $p_i = \kappa\alpha_i$, where κ is a constant and α_i is the support vector weight. The Kernel unit gets the input vector $\mathbf{x}$ from the Trap and the present support vector $(\mathbf{x}_i, y_i)$ from the OM, computes the SVM kernel and outputs $y_i K(\mathbf{x}_i, \mathbf{x})$ to the Integrator. This is done continuously as the inputs change. The Integrator resets as a new input is captured by the Trap. The system needs means, such as a clock signal, for synchronizing the Trap and Integrator reset. The clock frequency $1/T_{trap}$ must be much smaller than the OM frequency.

The Integrator estimates the zero-bias classification function (Eq. (1)) by temporal integration over the Kernel output,

$$f(\mathbf{x}) \approx c' \int_{t_0}^{t_0+T_{trap}} y_{i(t)} K(\mathbf{x}_{i(t)}, \mathbf{x}) \mathrm{d}t, \tag{9}$$

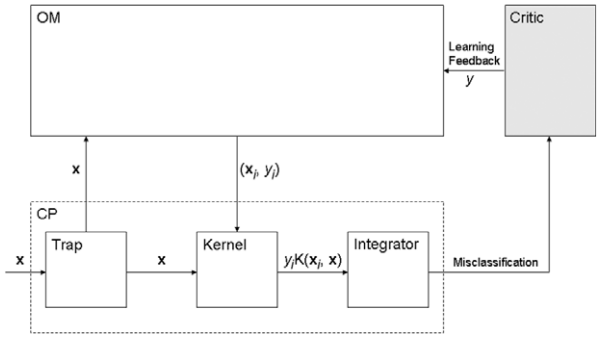

Fig. 3. Signal flow in the surprise learning process. CP is the Classification Pathway. OM is the Oscillating Memory. Sensor signal $\mathbf{x}$ is captured by the Trap. The OM outputs the support vector $(\mathbf{x}_i, y_i)$. K is the kernel function. The feedback from the Critic provides implicitly the correct valence y of the misclassified example.

where c' is some positive constant and t_0 is the starting time of the integration over the holding time of the Trap. The support vector index $i(t)$ is a function of time since the OM oscillates between support vector states. The time integral in Eq. (9) becomes, over many OM oscillations, asymptotically proportional to

$$\sum_{i=1}^{m} y_i p_i K(\mathbf{x}_i, \mathbf{x}) = \kappa \sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) = \kappa f(\mathbf{x}). \tag{10}$$

The Integrator estimates the SVM classification function multiplied by an immaterial constant. The final output is $\mathrm{sgn}(f(\mathbf{x}))$, where sgn is the sign function.

2.4. Surprise learning

There is no learning feedback for correct classifications. Faulty classifications, however, cause admonishment from higher brain systems represented by the Critic (see Fig. 3). Consider a binary snake/non-snake classifier in the brain of an animal. A snake is first thought to be a non-snake. The mistake is eventually discovered, causing fear. The Critic sends a negative surprise signal to the OM. The Trap is still providing a copy of the misclassified input $\mathbf{x}$ to the OM. The surprise signal forces the OM into a plastic state where a new training example $(\mathbf{x}, -1)$ instantly is imprinted. The correct classification of the input is implicit in the feedback from the Critic. Life can also provide positive surprises. The classifier takes a branch for a snake eliciting an erroneous fear response. Relief follows as the mistake is exposed. The Critic ensures that the OM stores the misclassified event $\mathbf{x}$ as a new example $(\mathbf{x}, +1)$ with a positive classification. All misclassifications are here considered to be surprises. The mapping of emotional valence to SVM valence will differ depending on the nature of the classification.

Note that surprise learning requires no repetition of the stimuli. Good support vector candidates are found since the process identifies borderline cases and outliers. The system is, however, off-balance after incorporating a surprise because the new example has an inappropriate weight. The importance learning process finds the new maximum of the SVM dual objective function where fresh examples as well as previous support vectors get proper weights. Before describing the importance learning process we need a more comprehensive model of the OM.

2.5. The oscillating memory

The base-line for the OM is the Hopfield associative memory (Hopfield, 1982; see Hertz, Krogh, & Palmer, 1991 for a review). The Hopfield memory consists of artificial binary-state neurons. The output of each node depends on inputs from all the other nodes. The Hopfield model can learn a new memory instantly

and therefore supports surprise learning, given that the training examples are appropriately coded to a binary format.

Hopfield networks have spurious attractors, including combinations of the intended memories and spin glass states. It has been found that under certain favourable combinations of noise and memory load only regular memories persist. Hopfield models can also be augmented to suppress mirror states where all the bits of the regular memory states have been flipped. We assume here that only regular memories are activated in the OM.

Hopfield networks relax into one of the memory patterns. A node that fires in the selected state will just keep firing. Biological associative memories can have a much richer variety of dynamic behaviour including spontaneous oscillations. Exhaustion of vigorously firing neurons that are characteristic of one memory pattern may for example cause the associative memory to switch to a different pattern. This could cause a perpetual oscillation.

Chaotic itinerancy (Ikeda, Matsumoto, & Otsuka, 1989; Kaneko, 1990; Tsuda, 1992; see Kaneko & Tsuda, 2003 for a review), where a chaotic dynamic system orbits between a set of quasi-attractors, offers a framework for a deeper understanding of spontaneous switching between memory states. Tsuda (1996) investigated chaotic itinerancy in artificial neural networks. Chaotic itinerancy has been applied to perception and episodic memory in the brain (Tsuda, 2001). Chaotic switching in brain systems is also discussed for example by Kozma and Freeman (2001) and Kay, Lancaster, and Freeman (1996). Spontaneous stochastic switching between attractor states is just one facet of the rich phenomenology of chaotic itinerancy. Simulations in several different types of artificial neural networks confirm the ubiquity of such stochastic switching. Pantic', Torres, Kappen, and Gielen (2002) studied models of associative memories with depressible synapses and found a phase with fast oscillations between stored memories. Horn and Usher (1989) showed that fatigue in the artificial neuron's threshold function causes a similar behaviour. Liljenström (2003) describes a dynamical model of self-organized cortical oscillations.

As a simplified model of stochastic switching we assert that each memory state has an endurance time T_i. Once a given memory pattern is triggered it will remain active for the duration of T_i and then spontaneously deactivate. The OM then becomes unstable and relaxes to a new memory pattern. Sparsely populated Hopfield networks have attractors with equally large basins of attraction. The system is influenced by external and internal stochastic processes that can be modelled as thermal fluctuations. In the context of our simple model it is hence reasonable to assume that the new state is randomly selected with an equal probability distribution. The probability of finding the OM in memory state i during a given oscillation cycle is, hence,

$$p_i' = \frac{1}{m}. \tag{11}$$

The probability of finding the OM in memory state i at a given time is

$$p_i = \frac{T_i}{T_{tot}}, \tag{12}$$

where T_{tot} is the sum of the endurance times of all memory patterns. We further postulate that there is a maximum endurance time T_{max} and that the OM operates in a domain where $mT_{max} > T_{tot}$ so that all states cannot have maximum endurance.

The present high-level OM model is expressed in terms of memory patterns rather than neurons. It would be best matched by a sparsely populated Hopfield memory where active cells are involved in only one memory pattern. More realistic representations should involve overlapping populations where the same neuron fires in several memory patterns. It remains to demonstrate that such detailed models produce approximately the same behaviour as our simple model.

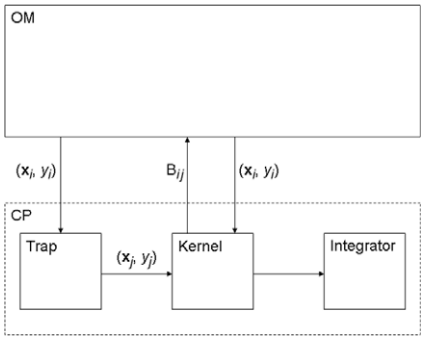

Fig. 4. Signal flow in the importance learning process. CP is the Classification Pathway. OM is the Oscillating Memory. The OM output $(\mathbf{x}_j, y_j)$ is captured and held by the Trap. The Kernel merges the trapped $(\mathbf{x}_j, y_j)$ and the present $(\mathbf{x}_i, y_i)$ OM memory and provides the learning feedback B_{ij} to the OM (see Eq. (16)).

We now show that dimensionless auxiliary variables α_i, related to endurance time according to

$$T_i = m\alpha_i T_{max}, \tag{13}$$

take on the role of the SVM weights. Eq. (3) is fulfilled by definition. Eqs. (12) and (13) imply that

$$p_i = m\alpha_i \frac{T_{max}}{T_{tot}}. \tag{14}$$

Hence $p_i = \kappa\alpha_i$, as assumed in the classification process.

The OM model locks the SVM weights to the $\sum_{i=1}^{m} \alpha_i = \nu$ hyperplane, since $\sum_{i=1}^{m} p_i = 1$ entails

$$\sum_{i=1}^{m} \alpha_i = \frac{T_{tot}}{mT_{max}} = \nu. \tag{15}$$

We note that $0 < \nu < 1$, as required by the ν-SVM model.

At this point we can reconsider the reason for dropping the bias parameter b in Eq. (1). This parameter gives rise to a third constraint $\sum_{i=1}^{m} \alpha_i y_i = 0$ in the optimization problem that is defined by Eqs. (2), (3) and (5) (Schölkopf et al., 2000). While the constraints (3) and (5), as we have seen, are natural consequences of the endurance time model, it is difficult to accommodate the third constraint where strength and valence of memory states are combined.

2.6. Importance learning

The importance learning process implements zero-bias ν-SVM gradient search. This section will explain how the SVM weights slowly move towards the optimum of the dual objective function (Eq. (2)) while the brain sleeps. External senses and the Critic are turned off. The OM oscillates and is in a slightly plastic phase where memory endurance is tuned.

The Trap locks on inputs from the OM since dominant external inputs are absent in a sleeping state (see Fig. 4). The OM sends the example corresponding to its present state to the Trap. The Trap locks and reproduces the example for the duration of the holding time T_{trap}. At the end of the holding time, the Trap becomes unstable again and locks on the pattern that presently is provided by the OM. The effect is that the Trap randomly outputs support vectors with probability distribution $p_j = \kappa\alpha_j$. The Trap hence oscillates at a rate $1/T_{trap}$ which is assumed to be much lower than the OM oscillation frequency.

Consider an OM oscillation in which the example with index i is the active memory and the example with index j is locked in

the Trap. The Kernel receives $(\mathbf{x}_j, y_j)$ from the Trap and the present oscillation state $(\mathbf{x}_i, y_i)$ from the OM. It outputs

$$B_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{16}$$

to the OM. Note that this is almost identical to the signal $y_i K(\mathbf{x}_i, \mathbf{x}_j)$ from the Kernel to the Integrator. The sleeping brain ignores the Integrator output.

The OM has a plastic phase in each oscillation where memories can be reinforced or weakened according to the following:

(1) Depress the present memory pattern in proportion to B_{ij}.
(2) Potentiate all memory patterns in proportion to B_{ij}/m.

Note that B_{ij} is signed. In terms of the endurance time model we get for $0 < T_i < T_{max}$,

$$T_i \leftarrow T_i - \eta T_{tot} B_{ij}, \tag{17}$$

$$\forall k : T_k \leftarrow T_k + \eta T_{tot} \frac{B_{ij}}{m}, \tag{18}$$

where η is a dimensionless learning rate. This should be understood as a physical process proceeding in infinitesimal steps while conforming to the constraints $0 \leq T_i \leq T_{max}$. Note that T_{tot}, and hence ν is conserved under the transformations (17) and (18). The corresponding update rules for SVM weights follow from Eq. (13).

The average net result of m OM cycles on a weight $\alpha_i < 1/m$ is

$$\alpha_i \leftarrow \alpha_i + \frac{m\eta}{\kappa} \left(\frac{1}{m} \sum_{k,j=1}^{m} p'_k p_j B_{kj} - p_i \sum_{j=1}^{m} p_j B_{ij} \right). \tag{19}$$

Using Eqs. (11) and (12), the symmetry of the kernel and $\kappa^{-1} \sum_{j=1}^{m} p_j B_{ij} = y_i f(\mathbf{x}_i) = marg_i$, we find that Eq. (19) evaluates to

$$\alpha_i \leftarrow \alpha_i + \eta(\langle marg \rangle - marg_i), \tag{20}$$

which is identical to the component-wise gradient search rule in Eq. (8). The Bio-SVM model solves the zero-bias ν-SVM problem with a stochastic version of gradient search. Statistical fluctuations away from the true gradient are not a problem since there are no false optima, the SVM weight constraints are automatically enforced, and the optimum can be reached from any point in the α-hyperplane.

The update rules take slightly different forms if some of the SVM weights are locked to the minimum or maximum value. It can be shown that the Bio-SVM update rules again correspond to zero-bias ν-SVM gradient ascent in such special cases. Asymptotic convergence of the gradient search process is ensured for a sufficiently small learning rate. Serviceable implementations must use learning rates that are small enough to avoid overshooting but yet give adequately fast convergence. Biological systems would have to tune the learning rate by genetic or individual trial and error. Importance learning has, in numerical experiments, been compared with optimal solutions. Convergence to useful approximate solutions is always found although the margin of regular support vectors, due to the probabilistic nature of the algorithm, varies asymptotically in an interval centred on the optimal margin. The mean deviations are typically less than 5% of the optimal margin.

The OM implements garbage collection of persistently inactive memory states where the corresponding endurance times consistently are suppressed to zero. Such states will eventually decay and be forgotten. This means that a steady-state OM only contains support vectors. Experiments with SVMs that forget trivial examples show that the resulting classifiers are efficient and robust. This is a special case of the well-known memory-saving "chunking" methods in the SVM toolkit (Schölkopf & Smola, 2002). New examples, introduced by the surprise learning process, can be initiated to a default endurance time by setting a large initial learning rate in Eqs. (17) and (18).

3. Discussion

In the description of the Bio-SVM model we have glossed over many important aspects of biological systems. Much of the huge complexity of perception, pattern recognition and memory in the brain has deliberately been ignored for the purpose of making the introduction of the Bio-SVM model more transparent. It is now time to catch up with some of these issues.

The Bio-SVM model is suggested as a mechanism for one-shot low-level trainable pattern recognition in the brain. It could, for example, model the ability of a rabbit to recognize the scent of a grey fox after one single exposure. Humans and animals, however, have many intertwined memory systems, probably including several different systems for hard-wired and trainable pattern recognition, several types of short-term memory and also long-term episodic and declarative memory. In describing the Bio-SVM model we have emphasised how animals are prone to remember emotionally charged exceptional events (support vectors) and forget successful classifications. Note, however, that the total memory system of humans and advanced animals certainly also provides means to remember typical examples and events of no apparent importance.

The present model appears to be consistent with the gross pattern of observed oscillations in the brain, in particular high-frequency cortical oscillations combined with the low-frequency olfactory "sniff cycle". There are, however, many oscillatory phenomena of the brain and there is presently no evidence that any observed oscillation actually is caused by Bio-SVM pattern recognition.

It should also be understood that it is unlikely that the brain implements any algorithm exactly as envisaged in convenient mathematical formulas or computer simulations. Temporal summation in biological neurons is, for example, not a clean linear temporal integration but is inherently non-linear and limited in range. The present speculation can only be confirmed by the interplay of experimental observations and increasingly realistic modelling.

The convergence time of the temporal integration in Eq. (9) is an Achilles' heel of the model since a good classification must be delivered within a reasonable time to be of any use for the animal. Convergence times depend strongly on problem details, the number of support vectors and the margin of the solution. Numerical experiments are shown in Fig. 5. Quick assessments require, in general, a small number of support vectors.

We have focused on how neural systems could optimize classification accuracy by finding the maximum of the SVM dual objective function Eq. (2). Real-world performance depends, however, on robust joint optimization of time, resources and accuracy. Fig. 5 demonstrates how Bio-SVM temporal integration early delivers coarse results that successively become more accurate. Higher-order systems can therefore be flexible about when to act on the emerging classification.

The types of components and functions that are needed for building a Bio-SVM classifier appear to be available in nature. The Trap is essentially sensory memory (Baddeley, 1999). Neural assemblies can implement non-linear multi-input functions that may serve as kernels. An SVM works with a wide range of kernels. Ensuring a positive definite kernel is the main challenge. Bio-SVM requires temporal integration in Eq. (9). Biological neurons provide a similar function through the mechanism of temporal summation (Johnston & Wu, 1995). The OM is an oscillating associative memory. The biological plausibility of associative memories is explained in Haberly (2001). The oscillating behaviour is a speculation grounded in the concept of chaotic itinerancy (Ikeda et al., 1989; Kaneko, 1990; Kaneko & Tsuda, 2003; Tsuda, 1992) and supported by computational experiments (Horn & Usher, 1989; Liljenström, 2003; Pantic' et al., 2002). The underlying

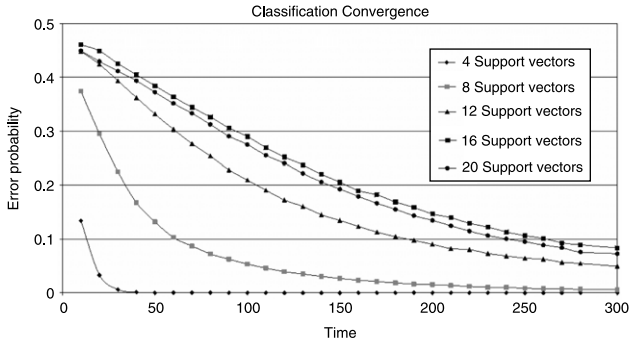

Fig. 5. Examples of Bio-SVM classification convergence. A Bio-SVM classifier is trained to recognize a chessboard pattern using $\nu = 0.5$ and a Gaussian kernel. The Bio-SVM error probability versus integration time is shown. The error probability is the probability that the Bio-SVM classification, computed by temporal integration (Eq. (9)), differs from the standard SVM classification according to Eq. (1). The error probability is estimated by averaging over 100 000 random test samples. Time is measured in units of OM oscillations because convergence depends on how many OM oscillations are included in Eq. (9). The convergence time increases in general with the number of support vectors in the solution but depends also on specific support vector positions and weights. A support vector's contribution to Eq. (9) depends on the weight and the distance to the test sample. The curve for 20 support vectors is, for example, below the curve for 16 support vectors because the former solution includes a smaller set of high-weight support vectors that dominate in Eq. (9).

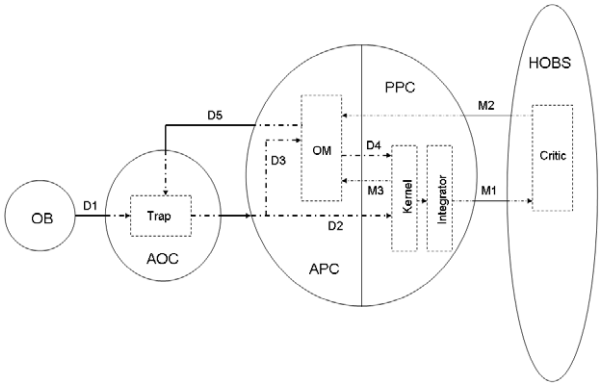

Fig. 6. Bio-SVM mapping to the olfactory system. One of many olfactory kernel machines is outlined. Solid ovals stand for known brain parts. Higher-order brain systems (HOBS) are management functions in the cortex and the limbic system. OB is the olfactory bulb. AOC is the anterior olfactory cortex. APC and PPC are the anterior and posterior piriform cortex, respectively. Dashed boxes indicate hypothetical components of a Bio-SVM system. The Trap is a register for input data in the AOC. OM is an oscillating associative memory in the APC. The Kernel and the Integrator are the core classification logic in the PPC. Solid lines are known neural projections. Dot–dashed lines are hypothetical connections. Broad driving connections carrying current or recalled sensory data are D1, D2, D3, D4 and D5. Narrow modulatory projections are M1, M2 and M3. Afferents (D1) carry pre-processed odor data from the OB to the AOC Trap. Trapped inputs are forwarded to the Kernel (D2) and to the OM (D3). The OM projects support vectors both to the Kernel (D4) and to the Trap (D5). The Integrator sends classification results (M1) to HOBS. HOBS triggers learning of misclassified examples (M2). The Kernel sends learning feedback (M3) to the OM. Anatomical facts are highly simplified but follow reviews in Haberly (1998, 2001). Known anatomical projections that are unused in the mapping are hidden. The feedback from the piriform cortex to the OB is, for example, not shown in the figure.

neural mechanism of the endurance time model could be synaptic depression (Johnston & Wu, 1995), for example, according to the vesicle depletion model (Abbott, Varela, Sen, & Nelson, 1997; Tsodyks & Markham, 1997).

An advantage of Bio-SVM is that learning modifies only the OM and not feed-forward sensory data pathways. Bio-SVM learning will hence not interfere with other processes using the same perceptual pathways. Learning new support vectors will not weaken skills that depend on old support vectors since SVM weights are continuously adapted in the importance learning process.

The Bio-SVM architecture is robust against many of the unavoidable haphazard changes in a living neural system. Associative memories are stable against minor damage to neurons and synapses. Some modifications of the classification pathway can be absorbed as tweaking of the kernel function. The recurrent training of the SVM weights compensates for such changes.

Support vectors are memories of significant percepts. Each support vector was once imprinted in memory because it was surprising and hence had emotional impact. Support vectors remain in memory since they collectively define classification boundaries. Higher-order brain systems could hence reason about the state of SVM-based classifiers. SVMs would be useful components of modular semantic systems.

Bio-SVM associative memory oscillations could be turned on and off as higher brain systems switch attention between different contexts. A possible mechanism for this is to append a context code to input vectors stored in the OM. Presenting the context code as a partial memory makes the OM fluctuate between support vectors carrying the context code.

SVMs are pattern recognizers with a sound mathematical foundation and an impressive track record of successful applications. Bio-SVM is an SVM architecture based on biologically inspired components. To find if such systems actually exist in brains one should look for a low-frequency oscillating sensory memory and a high-frequency associative memory that keeps oscillating rather than stabilizing in a recognized pattern. A non-linear kernel and a temporal integrator should be found in the feed-forward sensory pathway.

The olfactory system discriminates between many different molecular stimuli and hence includes many classifiers. Fig. 6 suggests a speculative mapping of the Bio-SVM to the olfactory architecture. The anterior piriform cortex (APC) resembles an associative memory (Haberly, 2001). The fast oscillations of the APC are not specific to the input (Haberly, 1998). The 4–8 Hz theta oscillation is connected to odor input capture (Macrides, Eichenbaum, & Forbes, 1982) while fast 40–50 Hz oscillations are related to the piriform cortex (Haberly, 1998). The anterior olfactory cortex (AOC) forwards processed sensory inputs to the piriform cortex (Haberly, 1998). Backprojections from the APC allow associatively recalled memories from the APC to be copied to the AOC so that "…a facsimile of the odor-evoked firing pattern …" is recreated (Haberly, 2001). The posterior piriform cortex is mainly a feed-forward network and is hence suited for hosting the Kernel and the Integrator. This mapping is certainly crude and tentative but suggests that the gross anatomical facts of the olfactory system are not inconsistent with the Bio-SVM hypothesis.

An alternative hypothesis would be to map the CP to the olfactory bulb while the OM remains in the piriform cortex. The glomeruli in the bulb would then take the role of the Bio-SVM Trap and mitral and tufted cells together with granule cells would implement the Kernel and Integrator functions. Backward projections from the piriform cortex to the granule layer would, in this model, carry support vector data from the OM to the Kernel. Projections to the glomeruli accounting for inputs from the OM to the Trap, however, appear to be missing.

Can we conclude anything about the functional form of the presumed neural SVM kernel? The Bio-SVM model inherits the prerequisite that the kernel must be positive definite from the base-line ν-SVM model but is otherwise agnostic with respect to the form of the kernel. Many different kernels are used for pattern recognition, including polynomial, $(\mathbf{x} \cdot \mathbf{x}')^n$; inhomogeneous polynomial, $(\mathbf{x} \cdot \mathbf{x}' + c)^n$; Gaussian, $\exp(-\|\mathbf{x} - \mathbf{x}'\|^2/2\sigma^2)$ and sigmoid, $\tanh(\omega(\mathbf{x} \cdot \mathbf{x}') + \theta)$, $\omega > 0$, $\theta \geq 0$. Computational experiments show that pattern recognition performance often is quite insensitive to the precise form of the

kernel (Schölkopf, Burges, & Vapnik, 1995). The Gaussian kernel used in the experiments of Fig. 5 can for example be replaced with several different inhomogeneous polynomial kernels without any significant variation in learning or recognition performance. Even kernels that violate the condition of positive definiteness sometimes work well in practise (Ong, Mary, Canu, & Smola, 2004). A specific functional form of the Bio-SVM kernel is hence not part of the generic model.

There is little basis for speculation on specific kernels for olfactory pattern recognition. The kernel must adapt to the format of odor signals reaching the piriform cortex. Presently there remains much to learn about the neural code. There are, however, good reasons to believe that neural systems would be able to develop effective kernel functions. Feed-forward artificial neural networks with at least one layer of hidden neurons can, under quite unrestrictive conditions, approximate any continuous multivariate function with arbitrary accuracy (Cybenko, 1989). Gaussian radial basis functions can for example be constructed by a hidden layer computing $(\mathbf{x}_i - \mathbf{x'}_i)^2$ components followed by an output neuron performing the exponential of the weighted and summed outputs from the hidden layer.

The present tentative mapping of the Bio-SVM model to the olfactory system shows intriguing correspondences. Note, however, that there is a comprehensive literature on computational models of olfaction with many alternative models (see Cleland and Linster (2005) for a review) including those where much of the information processing is in the olfactory bulb (see e.g. Freeman (1975); Skarda and Freeman (1987)). Odor recognition with an associative memory for oscillating patterns in the piriform cortex and encoding of inputs in the olfactory bulb was modelled by Li and Hertz (2000).

Neural classifiers can gradually develop the Bio-SVM architecture along an evolutionary path where each phase has increased utility. A hard-wired non-linear classification pathway comes first. Sensory memory stabilizes the output. Associative memory, connecting to the classification pathway, adds flexibility for handling exceptions. Memory oscillations and downstream temporal summation provide a crude mechanism for averaging over the relevant exceptional states. Evolution discovers Bio-SVM by gradually tuning this machinery.

Acknowledgment

This work was supported by the Swedish Foundation for Strategic Research.

References

Abbott, L. F., Varela, J. A., Sen, K., & Nelson, S. B. (1997). Synaptic depression and cortical gain control. *Science*, *275*, 220–224.
Baddeley, A. D. (1999). *Essentials of human memory*. New York: Psychology Press.
Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In D. Haussler (Ed.)., *Proceedings of the 5th annual ACM workshop on computational learning theory* (pp. 144–152). ACM Press.
Chang, C-C, & Lin, C-J. (2001). Training ν-Support vector classifiers: Theory and algorithms. *Neural Computation*, *13*, 2119–2147.
Cleland, T. A., & Linster, C. (2005). Computation in the olfactory system. *Chemical Senses*, *30*, 801–813.
Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based methods*. Cambridge: Cambridge University Press.
Cybenko, G. (1989). Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*, 303–314.
Freeman, W. J. (1975). *Mass action in the nervous system*. New York: Academic Press.
Haberly, L. B. (1998). Olfactory cortex. In G. M. Shepherd (Ed.)., *The synaptic organization of the brain* (4th ed.) (pp. 377–416). Oxford: Oxford University Press.
Haberly, L. B. (2001). Parallel-distributed processing in olfactory cortex: New insights from morphological and physiological analysis of neuronal circuitry. *Chemical Senses*, *26*, 551–576.
Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. New York: Addison-Wesley.
Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*, 2554–2558.
Horn, D., & Usher, M. (1989). Neural networks with dynamical thresholds. *Physical Review A*, *40*(2), 1036–1044.
Ikeda, K., Matsumoto, K., & Otsuka, K. (1989). Maxwell–Bloch turbulence. *Progress of Theoretical Physics. Supplement*, *99*, 295–324.
Johnston, D., & Wu, S. M-S. (1995). *Foundations of cellular neurophysiology*. Cambridge MA: MIT Press.
Kaneko, K. (1990). Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements. *Physica D*, *41*, 137–172.
Kaneko, K., & Tsuda, I. (2003). Chaotic itinerancy. *Chaos*, *13*, 926–936.
Kay, L. M., Lancaster, L. R., & Freeman, W. J. (1996). Reafference and attractors in the olfactory system during odor recognition. *International Journal of Neural Systems*, *7*, 489–495.
Kozma, R., & Freeman, W. J. (2001). Chaotic resonance—methods and applications for robust classification of noisy and variable patterns. *International Journal of Bifurcation and Chaos*, *11*(6), 1607–1629.
Li, Z., & Hertz, J. (2000). Odor recognition and segmentation by a model olfactory bulb and cortex. *Network: Computational Neural Systems*, *11*, 83–102.
Liljenström, H. (2003). Neural stability and flexibility: A computational approach. *International Journal of Neuropsychopharmacol*, *28*, 64–73.
Macrides, F., Eichenbaum, H. B., & Forbes, W. B. (1982). Temporal relationship between sniffing and the limbic θ rhythm during odor discrimination reversal learning. *Journal of Neuroscience*, *2*, 1705–1717.
Ong, C. S., Mary, X., Canu, S., & Smola, A. J. (2004). Learning with non-positive kernels. In *Proceedings of the 21st international conference on machine learning* (pp. 81–89). ACM Press.
Pantic', L., Torres, J. J., Kappen, H. J., & Gielen, S. (2002). Associative memory with dynamic synapses. *Neural Computation*, *14*, 2903–2923.
Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. In U. M. Fayyad, & R. Uthurusamy (Eds.), *Proceedings, first annual conference on knowledge discovery & data mining* (pp. 252–257). AAAI Press.
Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge MA: MIT Press.
Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*, 1207–1245.
Skarda, C. A., & Freeman, W. J. (1987). How brains make chaos to make sense of the world. *Behavioral Brain Science*, *10*, 161–195.
Tsodyks, M. V., & Markham, H. (1997). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences of the United States of America*, *94*, 719–723.
Tsuda, I. (1992). Dynamic link of memory: Chaotic memory map in nonequilibrium neural networks. *Neural Networks*, *5*, 313–326.
Tsuda, I. (1996). A new type of self-organization associated with chaotic dynamics in neural networks. *International Journal of Neural Systems*, 7, 451–459.
Tsuda, I. (2001). Towards an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behavioral Brain Science*, *24*, 793–810.

Magnus Jändel received his Ph.D. in theoretical physics in 1985. As Senior Fellow in the theory division of CERN 1986–88, he specialized in nuclear fusion in dense matter. He became an Associate Professor at the Stockholm Royal Institute of Technology in 1988. As Ericsson Senior Scientist (1995–98) he established a laboratory for multimedia coding and initiated the EU Fifth framework project SCALAR. Jändel founded IT start-up Terraplay Systems and worked as Chief Technology Officer (2000–2007). He now serves as Chief Scientist at the Swedish Defence Research Agency. His present research interests include distributed artificial intelligence and computational models of brain systems.