in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

Generating Motion Trajectories by Sparse Activation of Learned Motion Primitives

Christian Vollmer[1], Julian P. Eggert[2], and Horst-Michael Gross[1]

[1]Ilmenau University of Technology,
Neuroinformatics and Cognitive Robotics Lab,
98684 Ilmenau, Germany
christian.vollmer@tu-ilmenau.de
[2]Honda Research Institute Europe GmbH
63073 Offenbach/Main, Germany
julian.eggert@honda-ri.de

Abstract. We interpret biological motion trajectories as composed of sequences of sub-blocks or *motion primitives*. Such primitives, together with the information, *when* they occur during a motion, provide a compact representation of movement. We present a two-layer model for movement generation, where the higher level consists of a number of spiking neurons that trigger motion primitives in the lower level. Given a set of handwritten character trajectories, we learn motion primitives, together with the timing information, with a variant of shift-NMF that is able to cope with large data sets. From the timing information for a class of characters, we then learn a generative model based on a stochastic Integrate-and-Fire neuron model. We show that we can generate good reconstructions of characters with shared primitives for all characters modeled.

Keywords: non-negative matrix factorization, motion primitives, spiking neurons

1 Introduction

Studies in animal motor control suggest that the motor system consists of a control hierarchy, where a number of low-level motor primitives control muscle activations to perform small movements and a higher level controls the sequential activation of those motor primitives to perform complex movements [1].

We present a model for the generation of motion trajectories that is inspired from those results. We demonstrate our model on the generation of handwritten character velocity trajectories. Our model consists of two layers. The lower layer consists of a set of *motion primitives* that, once activated, generate characteristic temporal sequence of values in 2D space for a short time period. Multiple motion primitives can be activated in sequence to generate complex trajectories. The upper layer consists of a mechanism and the knowledge to control activation of the motion primitives over longer time scales, i.e. *when* a particular motion primitive has to be activated to generate an instance of a certain class of trajectories, i.e. a certain character.

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

In the lower layer, we use a sparse coding algorithm, specifically the Non-negative Matrix Factorization (NMF), for learning the motion primitives, together with their activation times from a number of complex training trajectories. Early works for sparse coding of time series have shown that one can interpret the resulting representation as spike-like temporal activations of basis functions, i.e. our motion primitives, that are adapted to the problem domain [2]. In general, with NMF one can decompose a set of N input samples into a small number $K \ll N$ of basis vectors and coefficients, called activations, to superimpose these to reconstruct the inputs. By imposing a non-negativity constraint and specific sparsity constraints on the activations, the resulting basis vectors are interpretable as parts that are shared amongst the inputs and constitute an alphabet or dictionary underlying the data [3].

NMF has been applied to find patterns in data like neural spike trains [4] or walking cycles of human legs with constant frequency [5]. The length of the basis vectors must be specified manually and is typically chosen to be of the length of the expected patterns, e.g. a single spike pattern or a single walking cycle. However, for human movement data like handwriting, where a pattern in this sense is a whole character, NMF in this form can not be applied due to temporal variations of the underlying patters, like different speed profiles. Our approach is to interpret a pattern to be a combination even of smaller sub-parts (see Fig. 1), where the parts themselves have low temporal variability and the variability of the whole pattern is captured by shifting the parts in a small local region.

In the upper layer of our two-layered model, the exact order and timing of the primitives is controlled with a timing model that stores knowledge about the typical activation times of the primitives for a desired class of trajectories, i.e. a character. Inspired by [6], we use the Integrate-and-Fire (I&F) spiking neuron model, which is parametrized by an *intensity matrix* that stores the relative frequency of activation for each primitive for a certain time interval. We learn this model by aligning and averaging over the primitive activations computed for the training trajectories.

Our work is conceptually similar to [6], where the primitives are modeled by a factorial HMM (fHMM), and the primitive control is also modeled by an I&F model. By using NMF in the lower level, we present an alternative approach, which is computationally less demanding.

We describe our approach in detail in Sec. 2. In Sec. 3, we show that we can generate visually appealing characters and illustrate some crucial parameter dependencies. Finally, we discuss our work in Sec. 4.

2 Method

Motion Primitive Learning We use a combination of two variants of NMF called semi-NMF and shift-NMF for learning the motion primitives from the handwritten character velocity profiles. Semi-NMF [7] relaxes the non-negativity constraint, such that only the activations are required to be non-negative. This allows the motion primitives to have positive and negative values. Shift-NMF [8]

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

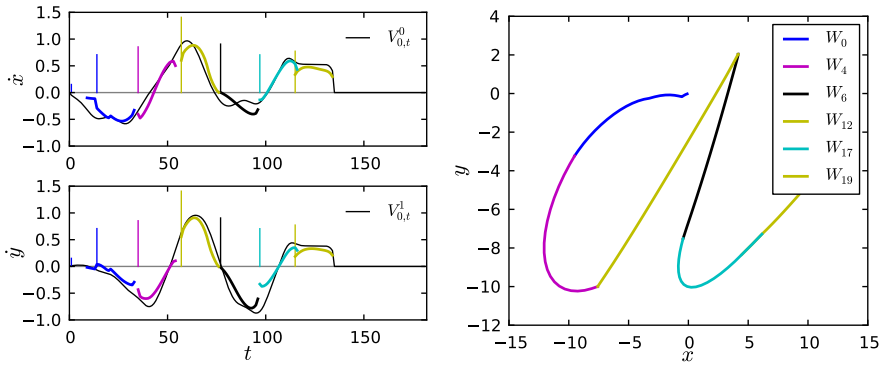

Fig. 1: Reconstruction of one character 'a' from the training data set after decomposition with NMF, according to eqs. 1 to 6. Left: reconstruction of the velocity profile of one input character (black line) by the learned parts (colored thick lines), scaled by their corresponding learned activations (vertical colored lines) Right: velocity reconstruction (left) integrated over time, resulting in the position of the pen. The parts have also been colored. Note that shown here are the temporally integrated versions of the actual parts.

introduces a translational degree of freedom to the basis vectors, i.e. there is not only one coefficient for each basis vector anymore, but one coefficient for each possible translation of a basis vector.

Let $\mathbf{V}^d \in \mathbb{R}^{N\times T}$ denote the matrix of N training trajectories of length T, with elements $V^d_{n,t}$. The the single trajectories are denoted as vectors $\mathbf{V}^d_n$. For ease of notation, we separate the spatial dimensions of the trajectories into distinct matrices, denoted by the upper index d. Let $\mathbf{W}^d \in \mathbb{R}^{K\times L}$ be the matrix of K basis vectors of length L, with elements $W^d_{k,l}$. We denote the single basis vectors by $\mathbf{W}^d_k$. Let $\mathbf{H} \in \mathbb{R}^{N\times K\times T}$ be the tensor of contributions $H_{n,k,t}$ of the k-th basis vector to the n-th input under translation t.

The NMF can be formulated by minimizing the following energy function

$$F = \frac{1}{2}\sum_d \left\|\mathbf{V}^d - \mathbf{R}^d\right\|_2^2 + \lambda_g g(\mathbf{H}) + \lambda_h h(\mathbf{H}) \ . \tag{1}$$

$\mathbf{R}^d \in \mathbb{R}^{N\times T}$ is the reconstruction matrix that is formed by temporal convolution of the activities with basis vectors

$$R^d_{n,t} = \sum_{k,m} H_{n,k,m}\hat{W}^d_{k,t-m} \tag{2}$$

Here, we introduced *normalized basis vectors* $\hat{\mathbf{W}}^d_k$, where the normalization is done jointly over all dimensions. The normalization is necessary to avoid scaling problems as described in [8]. The functions g and h implement the sparseness constraints and will be described later.

This optimization problem can be solved by alternatingly updating one of the factors $\mathbf{H}$ or $\mathbf{W}^d$, while holding the other fixed. For semi-NMF usually a

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

combination of least-squares regression of the basis vectors and multiplicative update of the activations is used [7]. The former, however has very high computational demands in the case of shift-NMF and is not applicable for our problem. We thus have to resort to gradient descent techniques. The following steps are repeated iteratively until convergence after initializing $\mathbf{H}$ and $\mathbf{W}^d$ randomly.

1. Build reconstruction according to Eq. 2
2. Update the activities by gradient descent and make them non-negative

$$H_{n,k,t} \leftarrow \max\left(H_{n,k,t} - \eta_H \nabla_{H_{n,k,t}} F, 0\right) \tag{3}$$

$$\nabla_{H_{n,k,t}} F = -\sum_{d,t'} \left(V^d_{n,t'} - R^d_{n,t'}\right) \hat{W}^d_{k,t'-t} + \nabla_{H_{n,k,t}} g + \lambda_h \nabla_{H_{n,k,t}} h \tag{4}$$

3. Build reconstruction according to Eq. 2
4. Update the basis vectors by gradient descent

$$W^d_{k,l} \leftarrow W^d_{k,l} - \eta_W \nabla_{W^d_{k,l}} F \tag{5}$$

$$\nabla_{W^d_{k,l}} F = -\sum_{n,d'} \sum_{t} \left(V^{d'}_{n,t} - R^{d'}_{n,t}\right) H_{n,k,t-l} \nabla_{W^d_{k,l}} \hat{W}^{d'}_{k,l} \tag{6}$$

The factors η_H and η_W are the learning rates. Note that the temporal correlations (sums over t and t', respectively) can be computed very efficiently in Fourier space. Note further, that expansion of the gradient in eq. 6 leads to a computationally simpler form, which is omitted here due to lack of space.

The function g enforces sparseness of the activities by penalizing the overall sum of activities. It's effect is the emergence of interpretable basis vectors as described in [8].

$$g(\mathbf{H}) = \sum_{n,k,t} H_{n,k,t} \ , \tag{7}$$

Since smooth basis vectors shifted only slightly are similar to themselves, there are multiple non-zero activities at adjacent locations, which contradicts our idea of spike-like activations that are temporally isolated. In most approaches this is handled by a heuristic approach called Matching Pursuit [2], which is suboptimal. Instead, we add a term h to the energy function, that introduces a competition between adjacent activities. The competition is implemented by convolution with a triangular kernel function $z_H(k, k', t - t')$.

$$h(\mathbf{H}) = \sum_{n,k,t} H_{n,k,t} \sum_{k',t'} z_H(k, k', t - t') H_{n,k',t'} \tag{8}$$

$$z_H(k, k', t - t') = \begin{cases} 0 & : k = k', t - t' = 0 \\ 1 - |\left(t - t'\right)/w| & : |\left(t - t'\right)/w| < 1 \end{cases} \ , \tag{9}$$

where w is the kernel width, which we set to twice the length of the basis vectors. In the case of $k = k'$, activities of the same basis vector and adjacent to t are penalized, such that isolated spike-like activities emerge. In the case of $k \neq k'$, the activities of all other basis vectors that try to reconstruct the same part

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

of the input are penalized. Thus, we enforce, that only one basis vector can be active during a time interval of L (the length of a basis vector) steps.

After applying NMF to the data, we interpret the basis vectors as motion primitives and their corresponding activities as temporal activations thereof. See Fig. 1 for an illustration of the resulting representation.

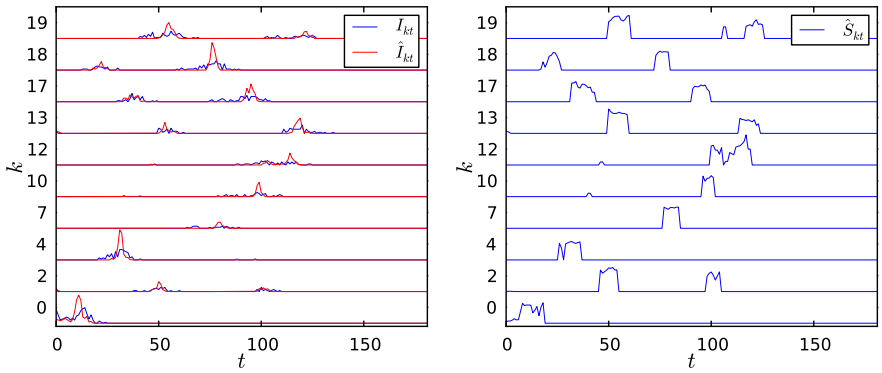

Fig. 2: Left: unaligned ($\mathbf{I}$) and aligned ($\hat{\mathbf{I}}$) intensity matrices for the character 'a' for the first ten basis vectors with highest maximum intensity. The intensities have been normalized to the maximum intensity of 0.235. Right: scaling matrix obtained from the aligned activity patterns for the same set of basis vectors. The scaling values have been normalized to the maximum value of 5.83.

Alignment of Activity Patterns On top of the primitive extraction we build a generative model for the generation of a character trajectory, given a character class. This model will be parameterized by an intensity matrix $\mathbf{I} \in \mathbb{R}^{K \times T}$ which is the relative frequency of an activation greater than zero of primitive $\mathbf{W}_k$ at time t and a scaling matrix $\mathbf{S} \in \mathbb{R}^{K \times T}$, which is the average value or *strength* of an activation of the k-th primitive at time t

$$I_{k,t} = \frac{1}{N} \sum_n \bar{H}_{n,k,t} \ , \qquad S_{k,t} = \frac{\sum_n H_{n,k,t}}{N I_{k,t}}, \tag{10}$$

where $\bar{H}_{n,k,t} = \Theta(H_{n,k,t})$ is the binarized activity and Θ is the Heavyside function. The training trajectories in the data set, however, exhibit some variation in start time and average speed, which is also reflected in the activation patterns after the NMF step. This negatively affects the computation of $\mathbf{I}$ (see blue line in Fig. 2 (left)) and $\mathbf{S}$. Thus, we associate with each training trajectory an offset a_n and stretching factor b_n. We optimize a_n and b_n iteratively by gradient ascent on the correlation Q between the individual activity pattern $\bar{H}_{n,k,t}$ and $I_{k,t}$

$$Q(a,b) = \sum_{n,k} \sum_t \bar{H}_{n,k,t} \tilde{I}_k(\tau(t, a_n, b_n)), \qquad \tilde{I}_k(\tau) = \sum_{t'} I_{k,t'} z_I(\tau - t') \ , \tag{11}$$

where $\tau(t,a,b) = b(t+a)$ is a translation and stretching of time index t, and $\tilde{I}_k(\cdot)$ is an interpolation of I_{kt} with a triangular interpolation kernel z_I that is

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

evaluated at the time index transformed by τ. After this optimization, we sum the aligned activations to get the aligned intensity matrix $\hat{\mathbf{I}}$. Accordingly, the aligned scaling matrix $\hat{\mathbf{S}}$ is computed from the aligned intensity $\hat{\mathbf{I}}$ (eq. 10). See Fig. 2 for an illustration of the resulting alignment and scaling matrix.

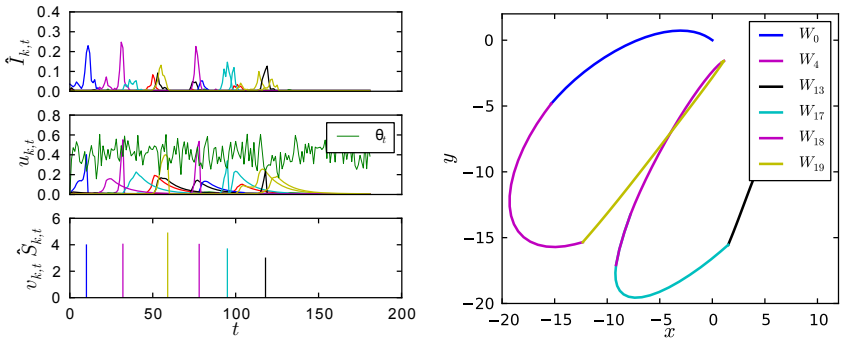

Fig. 3: Generation of trajectories through spiking neurons. Left: Process of spike generation, where the upper plot shows the aligned intensity matrix $\hat{\mathbf{I}}$, the middle plot shows the resulting load $u_{k,t}$ from the integration by the I&F model according to eq. 12, and the lowest plot shows the spikes $v_{k,t}$ that are generated and scaled by $\hat{S}_{k,t}$. Right: the trajectory that results from the convolution of the basis vectors with the scaled spikes in pen space, according to eq. 13.

Activity Generation To model the generation of activities in the upper layer, we use a stochastic Integrate-and-Fire (I&F) model, that is parametrized by the aligned intensity matrix $\hat{I}_{k,t}$ and generates spikes $v_{k,t} \in \{0, 1\}$, which are interpreted as activation times of basis primitives, and thus are the generated counterpart of $\bar{H}_{n,k,t}$. The internal state $u_{k,t}$ of the k-th neuron is modelled by a leaky integrator

$$u_{k,t} = \begin{cases} u_{k,t-1} - \nu u_{k,t-1} + \hat{I}_{k,t} & : t - t' \geq \delta t_{ref} \\ \hat{I}_{k,t} - \nu u_{k,t-1} & : t - t' = 1 \\ 0 & : 1 < t - t' < \delta t_{ref} \ , \end{cases} \tag{12}$$

where t' is the time of the last spike before t, δt_{ref} is the absolute refractory time, during which the load remains zero, and $\nu \in (0, 1)$ controls the amount of leakage. The neuron fires, i.e. $v_{k,t} = 1$, when $u_{k,t}$ exceeds a noisy threshold θ_t, which is sampled from a Gaussian. Our model is conceptually similar to that of [6], but also models a hyperpolarizing spike-afterpotential by an absolute refractory time δt_{ref} to prevent multiple firing in regions with high intensity. For reconstruction of the actual trajectory, we also need the scaling of the activation, which we computed earlier as the scaling matrix $\hat{S}$. The generated trajectory $\mathbf{Y}^d \in \mathbb{R}^T$ is then computed by the convolution of the basis vectors with the scaled spikes

$$Y_t^d = \sum_k \sum_{t'} v_{k,t} \hat{S}_{k,t} \hat{W}_{k,t-t'}^d \ . \tag{13}$$

See Fig. 3 for an exemplary illustration of the generation process.

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

3 Results

We demonstrate our model on the Character Trajectories Data Set [6] available from the UJI Machine Learning Repository. The motion primitive learning was done jointly over 20 character classes (all small one-stroked characters), with 119 samples in each class. The alignment and generation was done per class. Figure 4 (left) shows the reconstruction error dependent on the number of character classes $|C|$ in the training data set and the number of basis components used. It shows that for numbers of basis vectors of greater than 15, the reconstruction error only decreases insignificantly. If we train on smaller data sets, less basis vectors are necessary to get the same error. Figure 4 (right) shows the 20 learned basis vectors. Figure 5 shows a number of generated characters for all classes.

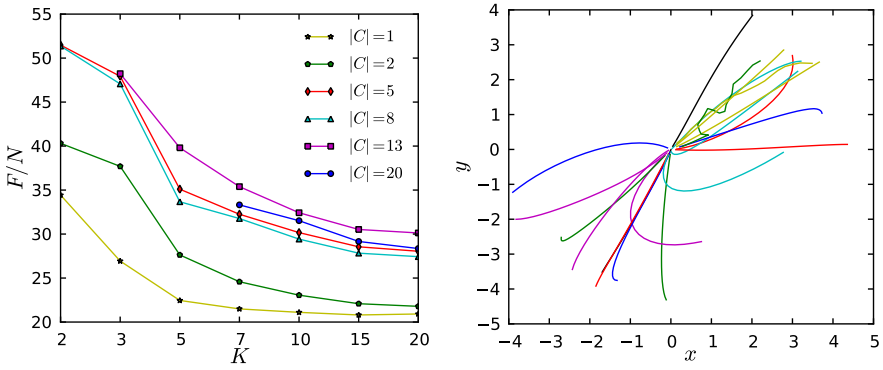

Fig. 4: Left: relation between cost F (normalized on number of inputs N), number of basis vectors K and number of classes $|C|$. Choosing more then $K = 15$ basis vectors does not result in significant decrease of reconstruction error. Right: 20 learned basis primitives in pen space (i.e. temporally integrated). Basis vectors that appear very similar here, differ in the speed of execution.

The quality of the generated characters is sensible on the parameters of the Gaussian firing threshold μ and σ. If μ is chosen too high, some parts are not activated and thus missing in the trajectory, which results in defects in some characters. Further the scaling of the basis vectors sometimes results in overlong strokes like in the characters 'l' and 'm'.

4 Conclusion

We presented a model for the generation of handwritten characters based on a locally sparsified and translationally invariant NMF decomposition followed by an event-based activation through spiking neurons. The decomposition of the input patterns into smaller parts and their corresponding composition by learning their timing regime allows for an efficient handling of the temporal variations inherent in human movement data. We have shown that with the

in: Artificial Neural Networks and Machine Learning (ICANN 2012), Lausanne, Switzerland, Part I, LNCS 7552, pp. 637-644, Springer 2012

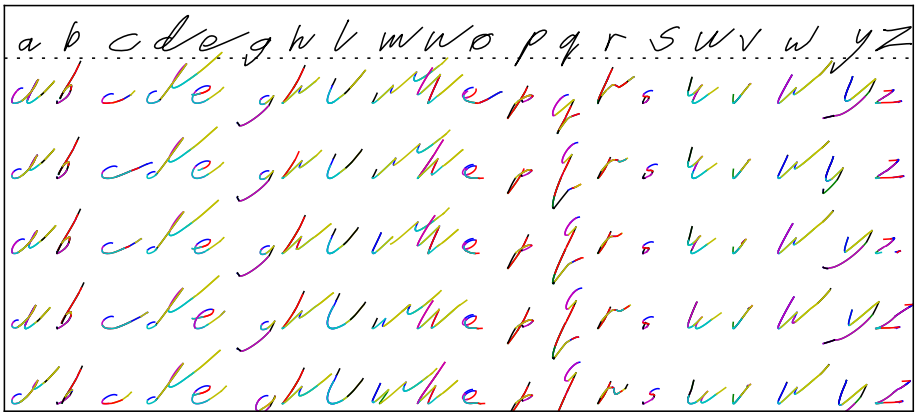

Fig. 5: The top row shows one example training character for each class from the training data set. The other rows show generated samples for all 20 character classes. In some classes of the generated characters, like 'd' and 'z', defects like missing parts are obvious.

proposed model the handwritten characters can be synthesized as a sequence of successive stroke parts.

The Integrate-and-Fire model for activation of primitives, however, sometimes results in defects in the resulting trajectories. Here we see room for improvement, and the fact that our model delivers single, isolated spikes in regions with high intensity, invites for direct statistical models e.g. of Hidden Markov type. This will be investigated in future research.

References

1. Bizzi, E.: Modular organization of motor behavior in the frog's spinal cord. Trends in Neurosciences 18(10), 442–446 (Oct 1995)
2. Smith, E., Lewicki, M.S.: Efficient coding of time-relative structure using spikes. Neural Computation 17(1), 19–45 (2005)
3. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–91 (Oct 1999)
4. Roux, J.L., de Cheveigne, A., Parra, L.C.: Adaptive Template Matching with Shift-Invariant Semi-NMF. Advances in neural information processing systems 21 (2009)
5. Kim, T., Shakhnarovich, G., Urtasun, R.: Sparse Coding for Learning Interpretable Spatio-Temporal Primitives. Advances in neural information processing systems 22 (Dec 2010)
6. Williams, B., Toussaint, M., Storkey, A.J.: A primitive based generative model to infer timing information in unpartitioned handwriting data. In: Int. Joint Conf. on Artificial Intelligence (IJCAI 2007). No. 1 (2007)
7. Ding, C., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorization for clustering and low-dimension representation (2006)
8. Eggert, J., Wersing, H., Körner, E.: Transformation-invariant representation and NMF. In: Proc. IEEE International Joint Conference on Neural Networks, vol. 4, pp. 2535–2539. (2004)