ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

Analytical comparison of the Temporal Kohonen Map and the Recurrent Self Organizing Map.

Markus Varsta Jukka Heikkonen
Jouko Lampinen
Laboratory of Computational Engineering
Helsinki University of Technology

Abstract

The basic SOM is indifferent to the ordering of the input patterns. Real data, however, is often sequential in nature thus context of a pattern may significantly influence its correct interpretation. One simple SOM model that takes the context of a pattern into account is the Temporal Kohonen Map (TKM), which was modified into the Recurrent Self Organizing Map (RSOM). We show analytically and with experiments that the RSOM is a significant improvement over the TKM because the RSOM model allows simple derivation of a consistent update rule.

1 Introduction

The Self Organizing Map (SOM) [4] is probably the most popular unsupervised neural network model. The basic SOM is indifferent to the ordering of the input patterns. Real data, however, is often sequential in nature thus temporal context of a pattern may significantly influence its correct interpretation.

One simple SOM model that takes the context of a pattern into account is the Temporal Kohonen Map (TKM) [1]. In the TKM the outputs of the units are replaced with leaky integrators, which effectively low pass filter the unit activities over the sequence of inputs. The TKM model was modified into the Recurrent Self Organizing Map (RSOM) [7, 6] for better resolution, but it later turned out that the real improvement came in the form of a consistent update rule for the network parameters.

In this paper we analyze the properties of the TKM and the RSOM models. This analysis may also serve as an example of the risks of modifying a model without considering all aspects of the related algorithm and subsequently testing the modification with too few or too simple experiments. In the TKM the problem of the modification lays in the difficulty of updating the learning rule to accommodate for the modified activity rule. We show that the RSOM is a significant improvement over the TKM since it allows simple derivation of a consistent update rule.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

2 TKM and RSOM

In the TKM model leaky integrators, that gradually lose their activity, are added into the outputs of the otherwise normal competitive units. These integrators and consequently the decay of activation is modeled with the difference equation

$$U_i(n,d) = dU_i(n-1,d) - 1/2\|x(n) - w_i(n)\|^2 \ , \quad (1)$$

where $0 \leq d < 1$ is a time constant, $U_i(n,d)$ is the activation of the unit i at step n while $w_i(n)$ is the weight vector of the unit i and $x(n)$ is the input pattern. The unit with the maximum activity is the *bmu* in analogy with the normal SOM.

The update rule for the TKM is not specifically addressed in [1]. In the experiments, however, weights were updated toward the last sample of the input sequence using the normal stochastic SOM update rule

$$w_i(n+1) = w_i(n) + \gamma(n)h_{i,j}(n)(x(n) - w(n)) \ ,$$

where $\gamma(n)$ is the learning rate and $h_{i,j}(n)$ is the value of the neighborhood function for the unit i at step n when the *bmu* is j. This corresponds with the situation where the leaking coefficient d is 0 for the update.

The leaked quantity in the *Recurrent Self-Organizing Map (RSOM)* is the difference vector instead of its squared norm. These leaky integrators are modeled with

$$y_i(n,\alpha) = (1-\alpha)y_i(n-1,\alpha) + \alpha(x(n) - w_i(n)) \ , \quad (2)$$

where $y_i(n,\alpha)$ is the leaked difference vector for unit i at step n. The leaking coefficient α is analogous to the value of $1-d$ in the TKM but in the RSOM formulation the sum of the factors is one to ensure stability when α is positive but less than one.

After moving the leaky integrators into the difference vector computation we can treat the remainder of the map much like the normal SOM when the unit with minimum $\|y(n,\alpha)\|$ is treated as the *bmu*. To derive an update rule for the RSOM we first formulate an error function $E(n)$ for the current sample $x(n)$ $E(n) = \frac{1}{2}\sum_{i \in V} h_{i,j}(n)\|y_i(n,\alpha)\|^2$, where V is the map. The gradient direction of $E(n)$ with respect to $w_i(n)$ is simply $y_i(n,\alpha)$ and thus the stochastic weight update rule for w_i to minimize error $E(n)$ is

$$w_i(n+1) = w_i(n) + \gamma(n)h_{i,j}(n)y_i(n,\alpha) \ .$$

This derivation ignores the discontinuities of the error function $E(n)$ due to discontinuities of the neighborhood function at the boundaries of the Voronoi cells. The key properties of the learning rules of the TKM and the RSOM models are summarized in Table 1.

Model	Bmu selection criterion	Weight update target
TKM	$\max U(\cdot, d)$	$\max U(\cdot, 0)$
RSOM	$\min \|y(\cdot,\alpha)\|^2$	$\min \|y(\cdot,\alpha)\|^2$

Table 1: The properties of the TKM and the RSOM. The second column is *bmu* selection criterion and the third column is the update rule target.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

3 Comparison of TKM and RSOM

In this section we will discuss the learning properties of the TKM and the RSOM models. First in section 3.1 we derive the optimal or the activity maximizing weights for a set of sequences and a single unit for both TKM and RSOM. The analysis directly extends to multiple units in the zero neighborhood case

$$h_{i,j}(n) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

when the boundaries of the Voronoi cells are ignored. In 3.2 we look into the update rule of the TKM to see what the map actually learns and compare the results with the RSOM results.

3.1 Optimal weights

Brief mathematical analysis is sufficient to show how maximizing activity in the TKM should lead to similar weights as minimizing the norm of the leaked difference vector in the RSOM when the maps share the same topology and data. Let us first consider a single TKM unit and a set $S = \{X_1, X_2, ..., X_N\}$ of sequences. The samples of the sequence $X_j \in S$ are $x_j(1), x_j(2), ..., x_j(n_j)$, where n_j is the length of the sequence X_j. In the TKM the goal is to distinguish different sequences by maximizing the activity of the corresponding *bmu*. For the set S of sequences and weights w_T the activity $U(S, w_T)$ over S is the sum

$$U(S, w_T) \quad = \quad -1/2 \sum_{X_j \in S} \sum_{k=1}^{n_j} d^{(n_j-k)} \|x_j(k) - w_T\|^2 \,. \tag{3}$$

Since the activity $U(S, w_T)$ is a parabola, it is everywhere continuous and differentiable with respect to w_T. Consequently its maximum lies either at an extreme or at the single zero of $\partial U(S, w_T)/\partial w_T$. From $\partial U(S, w_T)/\partial w_T = 0$ we obtain

$$w_T \quad = \quad \sum_{X_j \in S} \sum_{k=1}^{n_j} d^{(n_j-k)} x_j(k) / \sum_{X_j \in S} \sum_{k=1}^{n_j} d^{(n_j-k)} \,. \tag{4}$$

The weights w_T maximize the activity $U(S, w_T)$ of the unit for the set S. When all sequences have the same length n, the term $\sum_{k=1}^{n_j} d^{(n_j-k)}$ is constant and thus we can simplify the the equation to $w_T = 1/\Omega_S \sum_{X_j \in S} w_T^j$ where Ω_S is the cardinality of S and w_T^j are the optimal for the sequence $X_j \in S$ defined with $w_T^j = \sum_{k=1}^{n} d^{(n-k)} x_j(k) / \sum_{k=1}^{n} d^{(n-k)}$. These weights are the mean of the per sequence optimal weights, and they also are a good approximation when all sequences are sufficiently long for the chosen d.

For the RSOM unit the leaked difference vector $y(X, w_R)$, where $X = x(1), ..., x(n)$ is the input sequence and w_R are the RSOM weights, is

$$y(X, w_R) = \alpha \sum_{k=1}^{n} (1 - \alpha)^{(n-k)} (x(k) - w_R).$$

Since the goal is to minimize the norm of the leaked difference vector, for the

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

set S we can write

$$E(S, w_R) = 1/2 \sum_{X_j \in S} \|y(X_j, w_R)\|^2$$

for the error function $E(S, w_R)$, which is minimized at the optimum weights. $E(S, w_R)$ defines a parabola just like $U(S, w_T)$ for the TKM and thus the optimal weights are either at an extreme or at the single zero of the derivative of the error function with respect to the weights w_R. From $\partial E(S, w_R)/\partial w_R = 0$ we obtain

$$w_R = \sum_{X_j \in S} \left(\sum_{k=1}^{n_j} (1-\alpha)^{(n_j-k)} \sum_{k=1}^{n_j} (1-\alpha)^{(n_j-k)} x_j(k) \right) / \sum_{X_j \in S} \left(\sum_{k=1}^{n_j} (1-\alpha)^{(n_j-k)} \right)^2 . \tag{5}$$

The weights in Eq. 5 are quite close to the weights specified in Eq. 4 The small difference comes from the location of the leaky integrators.

Much like with the TKM we can simplify Eq. 5 if we assume that all sequences have the same length n. We get $w_R = 1/\Omega_S \sum_{X_j \in S} w_R^j$, where w_R^j are the optimal weights for the sequence $X_j \in S$ defined with $w_R^j = \sum_{k=1}^{n} (1-\alpha)^{(n-k)} x_j(k) / \sum_{k=1}^{n} (1-\alpha)^{(n-k)}$. These weights are identical with the corresponding TKM weights when $d = 1 - \alpha$. From the analysis we observe that the optimal weights for both models are linear combinations of the samples in the sequences.

3.2 Learning algorithms

Since the update rule of the RSOM is gradient descent to minimize the sum of the squared norms of the leaked difference vectors regularized by the neighborhood, the map explicitly seeks to learn the weights defined in the previous section. With the TKM this is not the situation: We show that generally the steady state weights of the TKM do not maximize the activity and use simulations to show how this affects the behavior of the TKM. To simplify the analysis we only considered the zero neighborhood case.

By definition, in a steady state further training causes no changes in weights. In practice this means that the derivative of the objective function is zero with respect to the weights given a static set of input patterns. Though in the stochastic training scheme reaching a steady state is not possible in finite time, criteria for a steady state can be defined and their impact considered when we study the equivalent batch approach. For the batch approach we split the TKM algorithm in two. In the first stage the data is Voronoi partitioned among the units with the network activity function. In the second stage the new weights given the partitioning are computed. While proving convergence for any SOM model is very difficult, possibly impossible [3, 2], if the TKM converges the weights have to satisfy the criteria we define here.

We have a set $S = \{X_1, ..., X_N\}$ of discrete sequences and a map V. Last sample of each sequence $X_j \in S$ is $x_j(n_j)$ where n_j is the length of the sequence $X_j \in S$. In a steady state the TKM weights have to be in the centroids of the last samples of the sequences in the Voronoi cells of the units because the

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

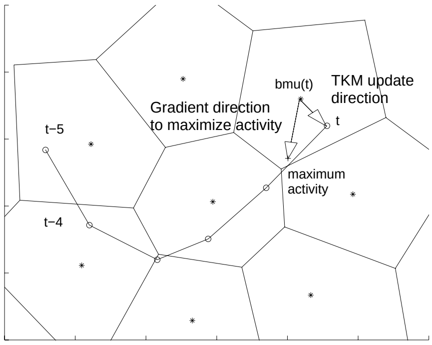

Figure 1: A piece of a TKM during training. The units, and their Voronoi cells, are marked with asterisks ($*$) and the input sequence with little circles ($\circ$). The plus ($+$) is drawn at the activity maximizing weights. The arrows show the optimal and the actual TKM update directions.

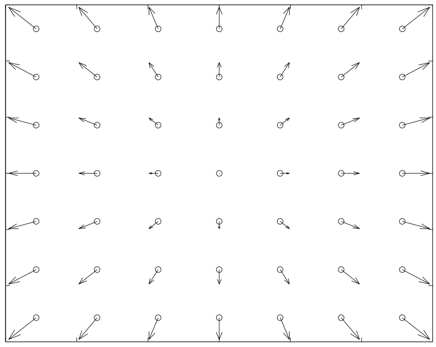

Figure 2: Approximation of mean bias when $d = 0.15$ between the activity maximizing update directions and the TKM update directions toward the last sample of a random sequence in the 7×7 grid in 2D input manifold.

weights are updated toward the last samples of the sequences. This condition follows from the update rule toward the last samples of the sequences which corresponds with the situation $d = 0$. When $d = 0$ TKM activity for unit i in Eq. 3 reduces to $U_i(w_i, S_i) = -1/2 \sum_{X_j \in S_i} (w_i - x_j(n_j))^2$ and the corresponding steady state weights at $\partial U_i(w_i, S_i)/\partial w_i = 0$ are

$$w_i = 1/\Omega_{S_i} \sum_{X_j \in S_i} x_j(n_j) \ , \forall i \in V \ , \tag{6}$$

where $S_i \subset S$ is the set of sequences in the Voronoi cell of i and Ω_{S_i} is the cardinality of S_i. These weights are necessary for a steady state. The optimal TKM weights with respect to the activity rule were defined in the previous section. The weights $w_{T_i} = \sum_{X_j \in S_i} \sum_{k=1}^{n_j} d^{(n_j - k)} x_j(k) / \sum_{X_j \in S_i} \sum_{k=1}^{n_j} d^{(n_j - k)} \ , \forall i \in V$ maximize activity with our simplifying assumptions.

The problem with the TKM is the discrepancy between the optimal weights and the necessary steady state weights. Fig. 1, which has a portion of a TKM during training, shows this graphically. The arrow "Gradient direction to maximize activity" shows the optimal direction to maximize activity while the arrow "TKM update direction" shows the actual update direction toward the last sample of the sequence.

We ran several simulations to show the impact of the discrepancy between the *bmu* selection and the weight update in the TKM. The first simulation involves a 1D map in a discrete 1D input manifold with seven input patterns. We initialized a 25 unit map with optimal weights (see axis 1 in Fig. 3) to maximize the total activity when the 1D inputs were 1...7 and the leaking coefficient d was 0.1429. The selection of d leads to a uniform optimal distribution of weights in the input manifold. The nearly optimally initialized map was further trained

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

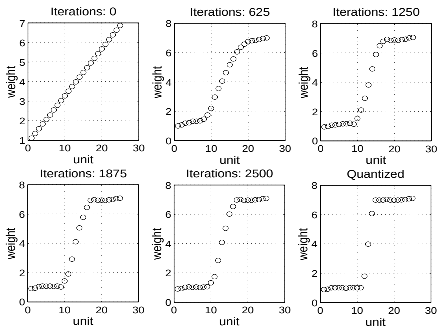

Figure 3: A map initialized with near optimal weights and trained with the TKM approach. Notice how most of the units are drawn into the edges.

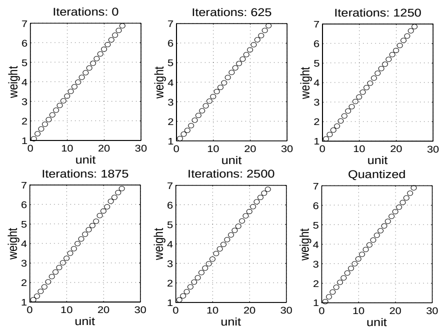

Figure 4: A map initialized with near optimal weights and trained with the RSOM approach.

by randomly picking one of the inputs, thus creating long random sequences, and updating the weights using the stochastic training scheme. The samples of the random sequences were corrupted with additive Gaussian ($\sim N(0, 0.125)$) noise.

Fig. 3 shows the progress of a sample run for the TKM. The TKM quickly "forgets" the initial weights because they do not satisfy the steady state criterion we derived earlier. Notice how the units are drawn toward the extremes of the input space leaving only a couple of units to cover bulk of the space. Similar 1D experiment with the RSOM in Fig. 4 yields a practically unchanged result.

We can intuitively explain the reason for the units being drawn toward the edges in the TKM with Figs. 1 and 2. For sequences that end near the edges of the input manifold the activity maximizing TKM weights and consequently the *bmus* are systematically closer to the center of the manifold than the last samples of the sequences which the units are updated toward. We can see this bias in Fig. 1 in the difference between the activity maximizing update direction and the actual update direction. The bias causes units to be attracted toward the edge and especially corner samples. Once a unit is close enough it will no longer be the *bmu* for any non trivial sequence of moving value.

Fig. 2 shows an approximation of mean bias between the activity maximizing update directions and the TKM update directions for a 7×7 grid in a 2D input manifold. We considered random sequences composed of the 49 input patterns in the manifold and computed the approximation for $d = 0.15$. The approximation was created using all sequences of length seven. The bias is zero only at the center of the manifold and becomes larger the closer the input is to the edge. The lengths and the directions of the arrows show the relative magnitude and direction of the bias for the sequences ending at that particular input. Formally

$$\mathbf{u}_j \approx \sum_{X_k \in S_j} x_j - w_{X_k}$$

where $\mathbf{u}_j$ is the arrow drawn at input x_j, S_j is the set of sequences which end at x_j, X_k is a sequence in S_j and w_{X_k} are the activity maximizing TKM weights

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

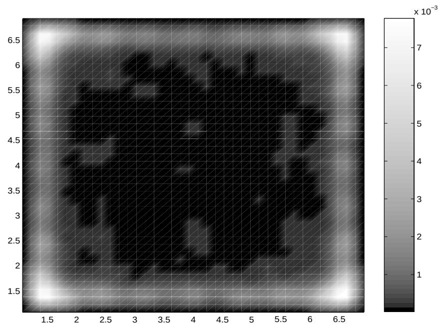

Figure 5: An estimate of the TKM weight distribution for a 10×10 map without neighborhood in the 7×7 grid when $d = 0.15$. Lighter shade means higher density.

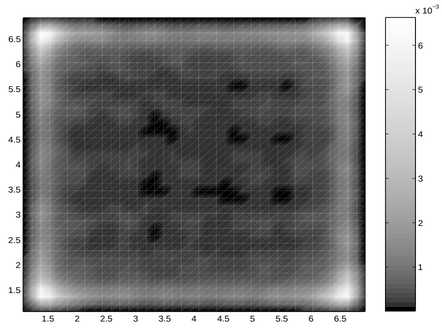

Figure 6: An estimate of the TKM weight distribution for a 10×10 map with box neighborhood of radius one in the 7×7 grid when $d = 0.15$.

for X_k. The arrows form what resembles a gradient field of smooth bump. The behavior of the TKM in the 2D simulations supports the intuitive result in the figure.

In the second set of simulations we trained hundred TKM and RSOM maps to estimate the the weight distributions with Gaussian kernels. The manifold we used to compute Fig. 2 was used in this experiment also and the input sequences were generated by randomly picking one of the 49 input patterns much like in the 1D case. The results are summarized in Figs. 6–8. We ran two simulations for both models. One, where the neighborhood was gradually turned off with quantization in the end and the other where neighborhood was retained to the end. The maps were trained with Luttrell's incremental approach [5]. The 10×10 maps were given sufficient time to organize from random initial configurations and we used $d = 0.15$ and $\alpha = 0.85$. Like in the 1D case the input samples were corrupted with Gaussian distributed noise $\sim N(0, 0.125)$.

The results show that regardless of initial configuration and input data which was independently generated for each map both TKM and RSOM behave consistently. In the case of the TKM without neighborhood the units were always concentrated near the edges and the corners of the input space in accordance with the the intuitive result in Fig. 2. Likewise for the RSOM the units form an approximately uniform lattice into the input space. Using box neighborhood with radius one did not have major impact on the results for either TKM or RSOM.

Now recall the optimal weights we derived for TKM and RSOM in Eqs. 4 and 5. In these 2D simulations the optimal weights for both models were approximately uniformly distributed in the input manifold. The TKM, however, concentrated most of its units in the edges and the corners of the manifold leaving only a few units to cover its bulk. As a consequence in these simulations the TKM model wasted a considerable part of it expressive power. The RSOM on the contrary systematically learned weights that nearly optimally spanned the input manifold.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 273-280

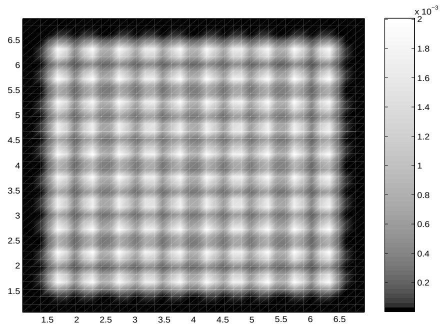

Figure 7: An estimate of the RSOM weight distribution for a 10×10 map without neighborhood in the 7×7 grid when $\alpha = 0.85$.

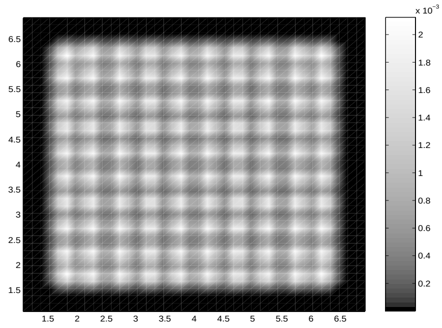

Figure 8: An estimate of the RSOM weight distribution for a 10×10 map with box neighborhood of radius one in the 7×7 grid when $\alpha = 0.85$.

References

[1] Geoffrey J. Chappell and John G. Taylor. The temporal Kohonen map. *Neural Networks*, 6:441–445, 1993.

[2] Marie Cottrell. Theoretical aspects of the som algorithm. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 246—267. Helsinki University of Technology Neural Networks Research Centre, Espoo, Finland, 1997.

[3] E. Erwin, K. Obermeyer, and K. Schulten. Self-organizing maps: ordering, convergence properties and energy functions. *Biological Cybernetics*, (67):47–55, 1992.

[4] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Lecture Notes in Information Sciences*. Springer, second edition, 1997.

[5] S.P Luttrell. Image compression using a multi-layer neural network. *Pattern Recognition Letters*, 10:1–7, 1989.

[6] Markus Varsta, Jukka Heikkonen, and José del Ruiz Millán. Context learning with the self organizing map. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 197–202. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997. ISBN 951-22-3589-7.

[7] Markus Varsta, Jukka Heikkonen, and José del Ruiz Millán. A recurrent self-organizing map for temporal sequence processing. In *Proceedings of the ICANN'97*. Springer-Verlag Berlin Heidelberg New York, October 1997. ISBN 3-540-63631-5.