T-61.181 Special Course in Information Science I 11.11.2002
Time in Self-Organizing Maps
Fall 2002

Seminar paper based on papers:

Dobrzewski, Ruwisch & Bode:
Wave propagation in self-organizing feature maps as means for the representation of temporal sequences

Fancourt and Principe:
A Neighborhood Map of Competing One Step Predictors for Piecewise Segmentation and Identification of Time Series

Fancourt and Principe:
Temporal Self-Organization Through Competitive Prediction

Esa Rinta-Runsala
44449K

1. Introduction

This paper gives an overview of three algorithms for handling sequences and time series. The paper by Dobrzewski et al.[1], presented in chapter 2, is about incorporating temporal ordering of a sequence to a self-organizing map. The other two papers by Fancourt & Principe [2], [3] consider segmentation and identification of time series with competitive predictors. These algorithms of these papers are presented shortly in chapter 3. The paper ends with a discussion about the algorithms presented.

2. Wave propagation in SOM

Self-organizing maps suit well for classification of static feature vectors. Wave propagation algorithm suggested by Dobrzewski et al. [1] is a way to capture also the temporal aspects of an input sequence to a self-organizing map. In the algorithm the previous features affect the choice of the winning nodes by changing the activation sensitivity of the nodes in waves around the previous winners.

The wave propagation algorithm starts with a conventional self-organizing map procedure. A feature vector **U** is presented to the map and a winning reference vector **W** is found. After this the algorithm proceeds to wave propagation to represent the temporal order of the features. A wave $Y_i(\mathbf{r},t)$ at time t at position $\mathbf{r}$ starts at time $t=t_i$ at the position of the winner, $\mathbf{r}_i^{win}$. The index i is the number of the feature vector $\mathbf{U}_i$ in the sequence. Wave propagation ends at time t_{i+1} when the next feature vector $\mathbf{U}_{i+1}$ is presented to the map. Now the wave is defined as

$$?_i(\mathbf{r},t) = H\left(ct_i' - d_{win}(\mathbf{r})\right) H\left(d_{win}(\mathbf{r}) - ct_i' + b\right) H(\mathbf{r},t) \qquad (1)$$

where $t'=t-t_i$, b is the wave crest width, $d_{win}(\mathbf{r})$= is the distance of a node from the winner node, H is the Heaviside function, $H(x)=0$ if $x<0$, and $H(x)=1$ otherwise, and $H(\mathbf{r},t)$ is a "history function". The history function restricts wave propagation according to the past winners. In the beginning of a new sequence

$$H(\mathbf{r},t_1) = 1, \text{ for all } \mathbf{r} \qquad (2)$$

The value of the history function is set to zero at positions **r**, which the wave crest has left, i.e.

$$H(\mathbf{r},t) \rightarrow 0 \text{ if } ?i(\mathbf{r},t) \rightarrow 0 \qquad (3)$$

and it remains zero for the rest of the sequence. Thus, a subsequent wave does not propagate into a region where an earlier wave has been during the same sequence.

The described wave propagation is illustrated in Figure 1. The node's probability to win is enhanced in the final wave region according to the equation

$$\mathbf{r}_i^{win} = \arg\min\left(\left\|\mathbf{U}_i - \mathbf{W}_r\right\| - b?_{i-1}(\mathbf{r} - t_i)\right) \qquad (4)$$

As stated earlier, the first winner (i=1) is determined in the standard way, i.e. $Y_0=0$, giving the standard SOM winner node determination. The term b in Eq. (4) determines how much feature similarity or sequence position is emphasized. A choice $b=0$ leads to the standard SOM . If $b>0$, the receptive fields of nodes (i.e. the regions where the node win the competition for the feature vector) with $Y_i=1$ are enlarged. Increasing b increases also the receptive fields until $b > \left\|\mathbf{W}_i - \mathbf{W}_j\right\| \forall i, j$, when the whole feature space is partitioned by the nodes on the wave crest. This means the winner node is always positioned on the crest and the later the feature vector is presented, the farther from the predecessor it will be represented. If the sequence is long, the wave could have left the whole map, i.e. $Y_i(\mathbf{r},t)_0=0$ for all the nodes. In this case the winner is determined according to the standard SOM algorithm.

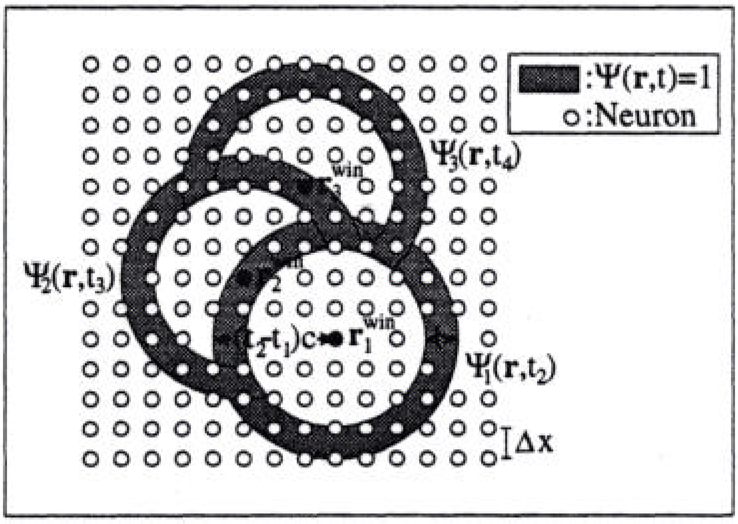

Figure 1. Three waves on the map started by three subsequent winner nodes at positions $\mathbf{r}_i^{win}$.

The wave propagation concept was implemented in digital hardware of 16 microcontrollers. These microcontrollers represented map nodes and they were used in processing of a 4-phoneme sequence "M O T O" . Each phoneme was described by energies of 3 frequency channels resulting to a 3-dimensional feature vector. After normalization the feature vectors lie on a 2-dimensional plane. The results of a 1-dimensional 16 node network are in Figure 2.

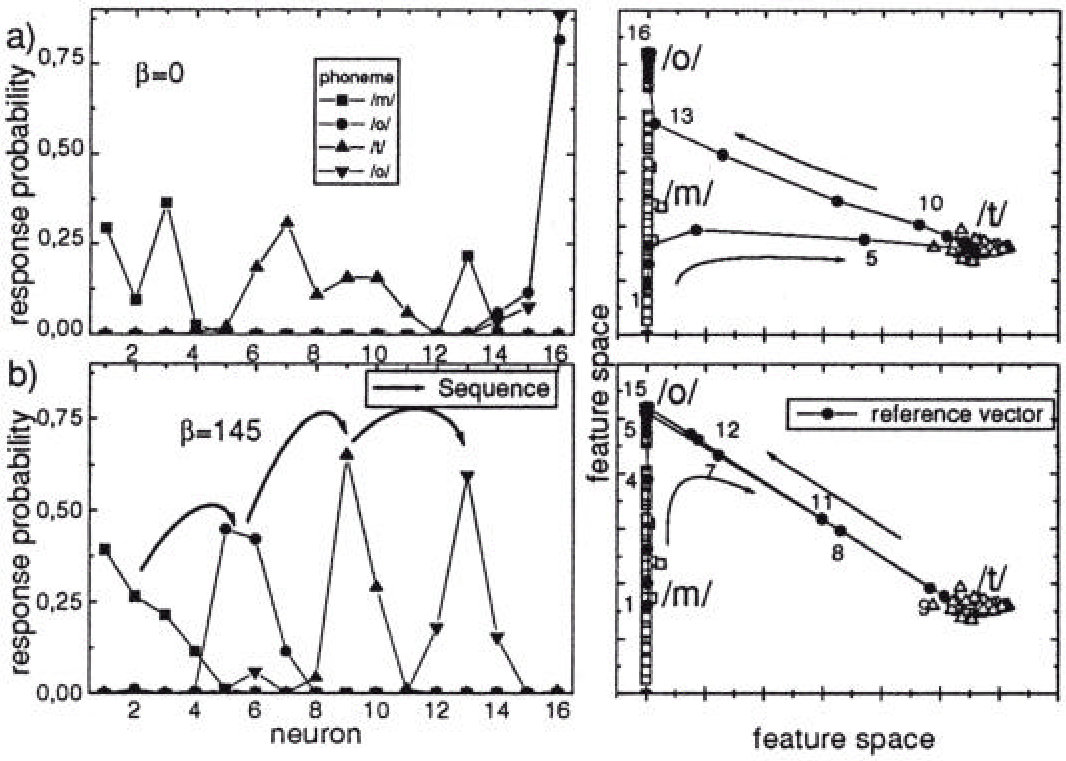

Figure 2. Representation of the phoneme sequence "M O T O" with two different values of b.

3. Neighborhood map of predictors

The papers of Fancourt & Principe [2], [3] describe an off-line technique for unsupervised segmentation and identification of time series. The technique is based on neighborhood map of competitive one step predictors. The map competes for the data and during the learning phase the winning predictor gets the largest parameter update. The other predictors get smaller updates according to their distance from the winner on the neighborhood map.

The winner of the competition for the data is is the one with the smallest memory of squared error:

$$winner(n) = \arg\min_i \left(e_i^2(n)\right) \tag{5}$$

where $e_i^2(n)$ is the memory of ith predictor's squared error at time step n. The memory of squared error is determined with exponentially decaying window

$$e_i^2(n) = l\, e_i^2(n) + (1 - l)e_i^2(n-1) \tag{6}$$

where $e_i^2(n)$ is the instantaneous squared error and l is the memory term($0 < l < 1$). The effective memory depth, i.e. the number of samples, is l^{-1}.

The learning rate of a predictor is

$$\Lambda_{i,j}(n) = e^{\left(-\frac{d_{i,j}^2(n)}{2s^2(n)}\right)} \tag{7}$$

where j is the winning predictor, $d_{i,j}(n)$ is the neighborhood distance from predictor i to j at time n and s is the parameter for neighborhood width. Both the neighborhood width s and the global learning rate g of the map are exponentially annealed:

$$s(n) = s_0 e^{-\frac{n}{t}} \quad \text{and} \quad g(n) = g_0 e^{-\frac{n}{t}} \tag{8}$$

where t is the annealing rate. Thus, the total learning rate for predictor i is

$$h_i(n) = g(n)\Lambda_{i,j}(n) \tag{9}$$

Simulations where run to give a view of the performance of the algorithm. In all of the simulations linear predictors trained with normalized LMS were used. The map is one dimensional continuous map with last node mapped next to the first. The training was done with 50 passes through the entire time series with annealing rate of t=5 passes.

The first simulation was made with a switching FIR process. The process has 8 stationary regions (Figure 3a). The results in Figure 3b are fairly good, except unexpectedly long winning period for predictor 8 and some errors near samples 200 and 475. The latters are due to the outliers in the time series.

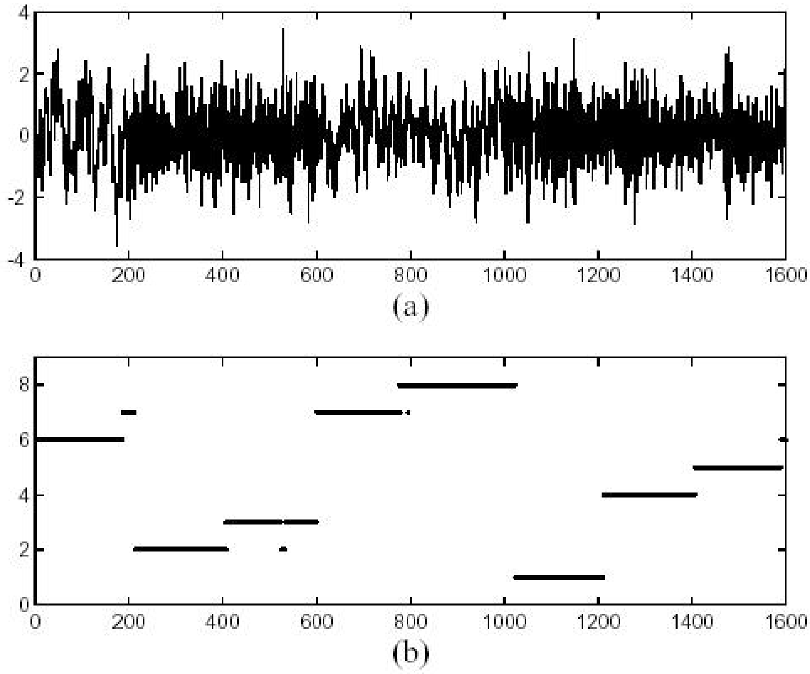

Figure 3. Switching FIR process: a) time series, b) winning predictors after training.

Another simulation was run with speech sample. The sample was word "alone" from Timit database and 7 predictors were used for segmentation. The sample and the results are in the Figure 4. The segmentation achieved is close to the Timit suggested phoneme segmentation shown in the figure. However, the same predictor won both the /a/ and /ow/ phonemes and predictors 1 and 7 won less than 1% of the time. This suggests that the activation distribution of the nodes is not optimal.

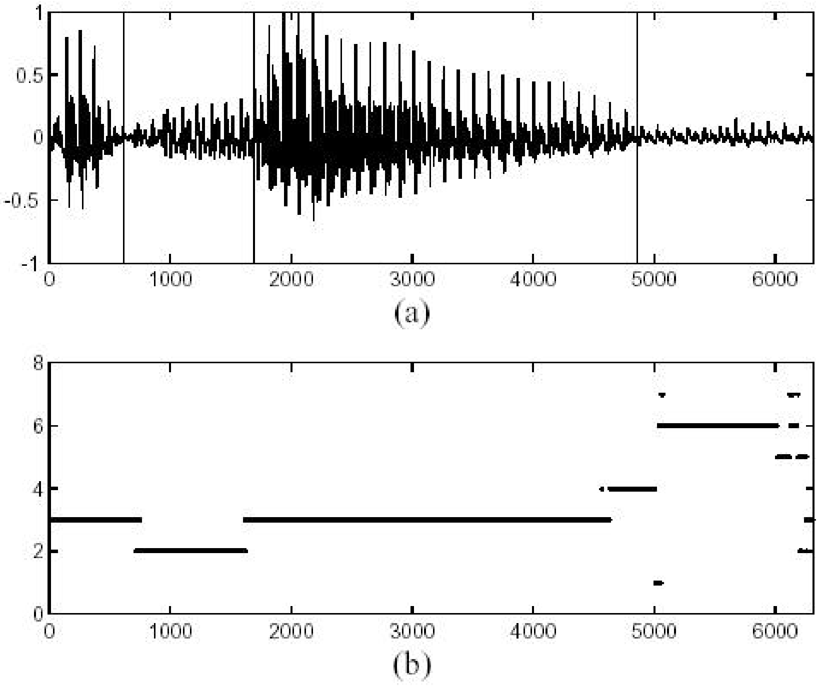

Figure 4. Speech example: a) time series and phoneme segmentation, b) winning predictors after training.

The second paper of Fancourt & Principe [3] also proposes a self-annealing competitive prediction, a way to couple the degree of competition and the memory depth. This eliminates the need for separate annealing of the memory depth and the competition. Moreover, the memory term λ does not need to be experimentally determined anymore like in neighborhood map algorithm. However, the results of the self-annealing algorithm are not as good as with the neighborhood map.

5. Discussion

The wave propagation algorithm of Dobrzewski et al. represents well the temporal aspects of the sample phoneme sequence. However, at the same time this weakens the map's ability to preserve topology. Thus, there is a contradiction between competitive ordering and topographic mapping. This is a clear weakness of the algorithm at least in situations where very different feature vectors succeed each other in the sequence or where similar feature vectors are far from each other in the temporal sense.

Fancourt & Principe's neighborhood map of competing predictors gives encouraging results in the segmentation and identification tasks. The predictors can be trained parallel in their usual way and the only information from the map is the adjustments to the learning rates. One weakness of the algorithm is its need for manual memory depth optimization. In the simulations presented, the memory depth was optimized experimentally, thus increasing the time needed for the training. The self-annealing algorithm described in the second paper of Fancourt & Principe solves the problem of memory depth determination but the results are clearly worse than with the neighborhood map.

References

[1] Dobrzewski, B., Ruwisch, D. & Bode, M. Wave propagation in self-organizing feature maps as means for the representation of temporal sequences. *Proc. Int. Conf. on Artificial Neural Networks* (ICANN'97), Lausanne, Switzerland, pp. 661-666, 1997.

[2] Fancourt C. & Principe J. A neighborhood map of competing one step predictors for piecewise segmentation and identification of time series. *Proc. of the Int. Conf. on Neural Networks* (ICNN'96), Washington, DC, Vol. 4, 1906-1911, 1996.

[3] Fancourt C. & Principe J. Temporal self-organization through competitive prediction. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* (ICASSP'97), Seattle, WA, Vol. 4, pp. 8-12, 1997.