# REPRESENTATION OF SPOKEN WORDS IN A SELF-ORGANIZING NEURAL NET

L.P.J. Veelenturf
University of Twente
Department of Electrical Engineering
P.O. Box 217, 7500 AE Enschede

## ABSTRACT

This paper will deal with an algorithm for a two-dimensional representation of the acoustic signal of spoken words. Having such an algorithm one can use the result for different applications like speech recognition and machine control by voice. If such a map from the domain of continuous speech signals to a two- (or three-) dimensional Euclidean space can be argued to be a reasonable model of the human speech processing system, then one can even investigate with this model psycho linguistic phenomena. The algorithm is in fact a shortcut for a self-organizing artificial neural network as developed by Kohonen

## 1. INTRODUCTION

This paper will deal with an algorithm for a two-dimensional representation of the acoustic signal of spoken words. Having such an algorithm one can use the result for different applications like speech recognition and machine control by voice. If such a map from the domain of continuous speech signals to a two- (or three-) dimensional Euclidean space can be argued to be a reasonable model of the human speech processing system, then one can even investigate with this model psycho linguistic phenomena. The algorithm is in fact a shortcut for a self-organizing artificial neural network as developed by Kohonen [Koh88a].

Our approach has a great resemblance to the "Phonetic Type Writer" as introduced by Kohonen [Koh88b]. The main difference will be the pre-processing of the speech signal. We will give first a description of the neural network and in section 3 the equivalent algorithm. In section 4 we will describe the pre-processing of speech signals and in section 5 the process of learning to represent acoustic signals. Finally, in section 6 we will give some preliminary results of simulations.

## 2. THE SELF-ORGANIZING NEURAL NETWORK

The neural network consists of $n$ units, the neurons, in a $d$-dimensional grid. In general one uses a two-dimensional Euclidean lattice. At time $t$ each neuron is given the same set of $m$ external inputs, represented by a vector $\mathbf{x}(t) = [x_1(t),x_2(t),...,x_m(t)]$. Every neuron $j$ will multiply each input $x_i(t)$ with some factor $w_{ji}(t)$, called the synaptic weight of neuron $j$ for input $x_i(t)$. According to some monotonic increasing and bounded transfer function $n$, the neuron will yield after some delay an output $y(t+1)$ depending among others on the weighted input $\{\Sigma w_{ij}(t).x_j(t)\}$. The output of neuron $j$ multiplied with a so-called fixed lateral weight $\gamma_{kj}$ forms an additional internal input for neuron $k$, this holds for all $k$ and $j$. The lateral weight $\gamma_{kj}$ is positive if the neurons $k$ and $j$ are neigbors and becomes negative if the distance in the lattice of neurons between neuron $k$ and $j$ is large. The distance dependent value of the lateral weight $\gamma_{kj}$ represents the effect of lateral excitation and inhibition as encountered in real neural nets. Depending on the weights $w_{ij}$ some neuron will initially optimal respond to some input vector $\mathbf{x}(t)$. Due to the time delay and the lateral excitation and inhibition a dynamic process will start with as a final result that within a bounded area around the initial most sensitive neuron all neurons will give a high output value.

It is known from neurobiology that the synaptic transfer of information between neurons will increase if there exists a positive correlation between an incoming signal and the output signal of a neuron. This so-called Hebb rule is implemented in the artificial neural net by the following weight adaptation rule:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \varepsilon.\{\mathbf{v}(t) - \mathbf{w}(t)\}.y(t)$$

The application of this rule implies that the area of a neuron with high response to some input vector v($t$) will become next time more sensitive to inputs resembling that particular input v($t$). When the input vectors are taken from an input space V with some probability density distribution then after a learning period, with a sufficient number of sample vectors from V, the neural net will realize a vector quantization of the input space V with the weight vectors $\mathbf{w}_j$ of the $n$ neurons as reference vectors. The point density function of the weight space will be almost the same as the density function of V.

Besides the vector quantization the neural net will yield an ordering of the weight vectors: if the final weight vectors can be totally ordered in a $d$-dimensional ordering, then that ordering of weight vectors is represented by the ordering of the corresponding neurons if the dimension of the neural net is at least equal to the dimension $d$. If for instance the vectors v are 25-dimensional and represent samples with a mask from a two-dimensional picture, then a two-dimensional self-organizing neural net will represent the samples in a two-dimensional neural net yielding an almost photographic copy of the original picture [Vtf89].

## 3. THE SELF-ORGANIZING ALGORITHM

The complete behavior of the neural net can however be simulated by a simple algorithm: Let $\mathbf{V} \subset \mathsf{R}^m$ (with R the set of real numbers) be a finite set of input vectors. Let $\mathbf{L} \subset \mathsf{N}^d$ (with N the set of natural numbers) be a $d$-dimensional lattice of points (the grid of neurons) with some distance measure $d_\mathrm{L}$. With each point $i$ in L there is associated a weight vector $\mathbf{w}_i(t)$. Assume we have n (the number of neurons) weight vectors $\mathbf{w}_i(t) \in \mathsf{R}^m$. Given at step $t$ of the learning process some element v($t$) of V, determine the "winning" weight vector $\mathbf{w}_s$:

$$|\mathbf{v}(t) - \mathbf{w}_s(t)| = \min_{\mathbf{r}} |\mathbf{v}(t) - \mathbf{w}_r(t)|.$$

Adapt every weight vector according to:

$$\mathbf{w}_r(t+1) = \mathbf{w}_r(t) + \varepsilon(t).h(r,s,t).\{\mathbf{v}(t) - \mathbf{w}_r(t)\}$$

with

$\varepsilon(t)$ a monotonically decreasing function of step $t$ and $\varepsilon(0) \leq 1$ (e.g., $\varepsilon(0) = 0.5$)

and

$h(r,s,t)$ a monotone decreasing function of the distance $d_\mathrm{L}(r,s)$ between the points in the lattice associated with the weight vectors $\mathbf{w}_r$ and $\mathbf{w}_s$ and $h(s,s,t)$=1 for all $t$. In addition, $h(r,s,t)$ is decreasing with $t$ for $r \neq s$.

For a proper selection of $\varepsilon(t)$, $h(r,s,t)$ and dimension of the lattice L the final result will be a vector quantization of V realized by the weight vectors w and an ordering of the weight vectors realized by the corresponding neurons.

## 4. THE PRE-PROCESSING

It is assumed that the human ear performs a pre-processing of sound which is optimal for speech recognition. Physiological research has revealed that main operation of the basil air membrane of the cochlea in the inner ear is the spectral decomposition of the speech signal. The same is done in conventional systems for artificial speech recognition like the FFT [DeV82]. The pre-processing can become quite complicated. The pre-processing system of Kohonen for his "Neural Type Writer" consists of eight steps: 1. A noise cancelling microphone. 2. A low pass filter. 3. An analogue to digital conversion with a sampling rate of 13 kHz. 4. A 256 point fast Fourier transform every 9.8 ms using a 256 Hamming window. 5. Logarithmization and filtering. 6. Grouping the 256 spectral component in a 15-component real vector. 7. Subtraction of the average from all components. 8. Normalization of the resulting vector. In our approach we want to reduce the number of artificial steps in the pre-processing. The steps 1, 2 and 3 were the same as Kohonen did (the sampling frequency in our case was 16 kHz. and the AD conversion was 16 bits). The fast Fourier transform was, after some experiments, replaced by the next simplified transform.

Given an interval of 128 samples $s(n)$ of the speech signal, we calculated for seven values of $k$ (frequencies) i.c. $k$=1,2,4,8,16,32,64 the next seven components of the input vector v of the neural network:

$$v_k = \frac{1}{128}\sum_n s(n).[\mathrm{sign}\{\sin\frac{2\pi k}{128}.n\} + \mathrm{sign}\{\cos\frac{2\pi k}{128}.n\}]$$

Each component $v_k$ represents a measure for the presence of a frequency component with angular frequency 2πk/128 in the particular interval of the speech signal. For each interval (shifted over one sample) of the acoustic signal of some spoken sentence we can obtain in this rather straight forward way 7-dimensional vectors as inputs for the self-organizing neural network.

## 5. Learning

We used for learning the acoustic signal of a sentence spoken by 9 different female speakers. The sentence was: "She had your dark suit in greasy wash water all year." Because of the large time length of the sentence we calculated only for intervals shifted over 126 samples of the input vectors v. The neural net was 2-dimensional with size 5x5.

The time dependent learning rate was equal to $\varepsilon(t) = 1/\sqrt{t}$.

The region adaptation function was equal to

$$h(r,s,t) = \mathrm{e}^{-d_L^2(r,s) / \sigma^2(t)}$$

with $\sigma(t) = 25/\sqrt{t}$. The input vectors v were randomly selected from the set of all input vectors obtained from the sentence given above. After 5000 learning steps the adaptation of the net was neglectable and learning was stopped.

## 6. Preliminary Results

In the test phase we shift with a window of 126 samples long over the speech signal of a word and present to the neural network the corresponding sequence of pre-processed vectors v. The words were taken from the sentence mentioned above. At each sample interval there will be one neuron most sensitive to the vector v of a particular speech interval. In this way we obtain a sequence of neurons that are most sensitive to the sequence of interval samples. The sequence of responding neurons gives a two-dimensional representation of a spoken word. In Figure 1 the trace in the neural net for the word "year" is given.

The path in the neural net for the word "year" is quite clear. Other words do not give such a simple result. In this rather small neural net many words result in paths that have jumps and crossings. Although we have an initial demonstration of the possibility to represent words in a self-organizing net, it will be clear that in a neural net with only 25 neurons we can not represent the multitude of characteristic speech intervals, moreover it is very likely that sound intervals have more than four resembling neighbors, so a two-dimensional neural net is not appropriate for full speech representation. A large number of neurons and a higher dimension of the neural net is required for practical applications, like the identification of phonemes by the intersection of different words and the identification of acoustically similar words by tracks in parallel.

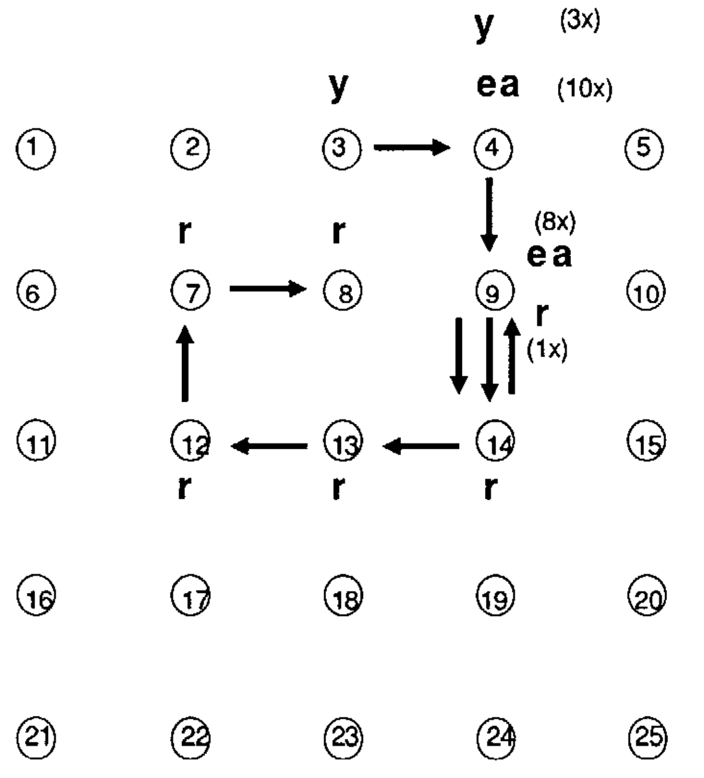

Figure 1

## 7. Conclusion

Our experiment reveals that even with a simple two-dimensional self-organizing neural network of 25 neurons and an elementary form of pre-processing we can obtain a two-dimensional representation of spoken words. More sophisticated pre-processing and larger neural networks of higher dimension requires additional research.

## References

[Koh88a] T. Kohonen. Self-organization and Associative Memory. Springer-Verlag, Berlin, 2nd edition, 1988.

[Koh88b] T. Kohonen. The neural phonetic typewriter. IEEE Trans. on Computer, March 1988, 11-22.

[Vtf89] L.P.J. Veelenturf. Antropomorpic retinotopic pattern recognition by processing global picture samples with a self-organizing neural network. Report University of Twente, km 080-1641, 1989.

[DeV82] P.A. DeVijver and J. Kittler. Pattern Recognition: A Statistical Approach. Prentice Hall, London, 1982.