An Introduction to Mexican Hat Networks

SUBMITTED TO:

Mr. Dilip Singh Sisodia

Asst. Professor

Computer Science & Engg.

NIT,Raipur.

SUBMITTED BY:

Sushant Ranade

08115059

8th Semester

CSE Dept.

Index

Mexican Hat Network [Kohonen, 1989a]

1. Introduction

There are many implementations of Artificial Neural Networks in which the neural networks are required to classify the outputs of their operation into many classes. For example, let us take the following problem into consideration:

Let us take a situation into consideration in which we applied a net that was trained to classify the input signal into one of the output categories, *A, B,* C, *D, E, J,* or *K,* the net sometimes responded that the signal was both a C and a *K,* or both an *E* and a *K,* or both a *J* and a *K.* In circumstances such as this, in which we know that only one of several neurons should respond, we can include additional structure in the network so that the net is forced to make a decision as to which one unit will respond. The mechanism by which this is achieved is called *competition.*

The most extreme form of competition among a group of neurons is called *Winner Take All.* As the name suggests, only one neuron in the competing group will have a nonzero output signal when the competition is completed. A specific competitive net that performs Winner-Take-All competition is the MAXNET.

A more general form of competition is called the MEXICAN HAT, or On-Center-Off-Surround contrast enhancement. In computer simulations of these nets, if full neural implementation of the algorithms is not of primary importance, it is easy to replace the iterative competition phase of the process with a simple search for the neuron with the largest input (or other desired criterion) to choose as the winner.
Mexican Hat network is an example of Fixed Weight Competitive Networks.

2. Fixed Weight Competitive Networks

Many neural nets use the idea of competition among neurons to enhance the contrast in activations of the neurons. In the most extreme situation, often called Winner-Take-All, only the neuron with the largest activation is allowed to remain "on." Typically, the neural implementation of this competition is not specified (and in computer simulations, the same effect can be achieved by a simple, non-neural sorting process).

3. Mexican Hat Networks

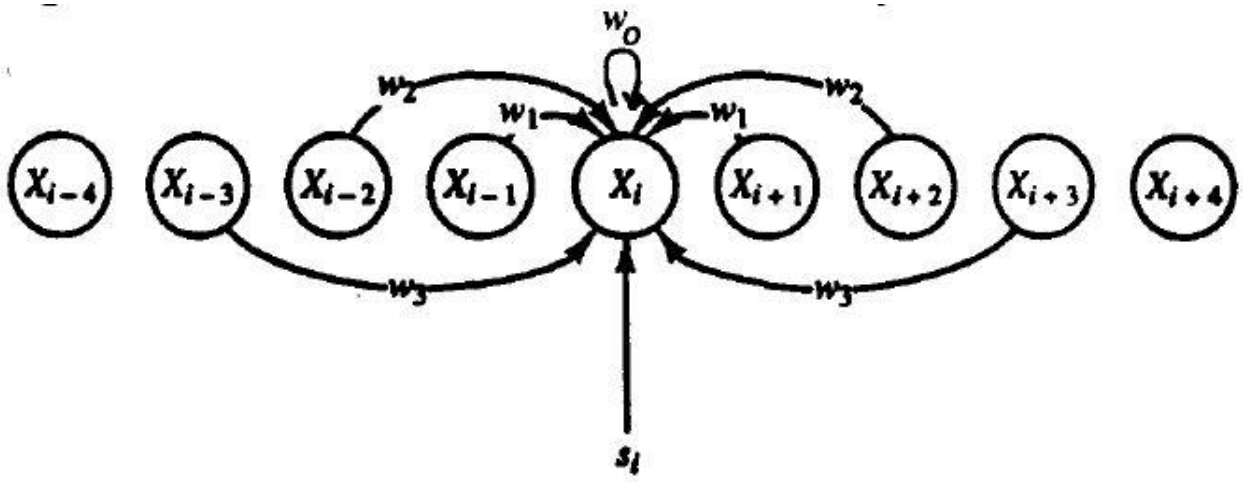
The Mexican Hat network [Kohonen, 1989a] is a more general contrast-enhancing subnet than the MAXNET. Each neuron is connected with excitatory (positively weighted) links to a number of "cooperative neighbors," neurons that are in close proximity. Each neuron is also connected with inhibitory links (with negative weights) to a number of "competitive neighbors," neurons that are somewhat further away. There may also be a number of neurons, further away still, to which the neuron is not connected. All of these connections are within a particular layer of a neural net, so, as in the case of MAXNET, the neurons receive an external signal in addition to these interconnection signals. The pattern of interconnections just described is repeated for each neuron in the layer. The interconnection pattern for unit X_i is illustrated in figure. For ease of description,
the neurons are pictured as arranged in a linear order, with positive connections between unit X_i and neighboring units one or two positions on either side; negative connections are shown for units three positions on either side. The size of the region of cooperation (positive connections) and the region of competition (negative connections) may vary, as may the relative magnitudes of the positive and negative weights and the topology of the regions (linear, rectangular, hexagonal, etc.).
The contrast enhancement of the signal s. received by unit *Xi* is accomplished by iteration for several time steps. The activation of unit *Xi* at time *t* is given by

$$x_i(t) = f[s_i(t) + \sum_k w_k x_{i+k}(t - 1)],$$

where the terms in the summation are the weighted signals from other units (cooperative and competitive neighbors) at the previous time step. In the example illustrated in figure, the weight w_k from unit X_i to unit X_{i+k} is positive for k =-2, -1, 0, 1, and 2, negative for k =-3, and 3, and zero for units beyond these.

3.1 Architecture

The interconnections for the Mexican Hat net involve two symmetric regions around each individual neuron. The connection weights within the closer region weights between a typical unit *X*, and units X_{i+1}, X_{i+2}, X_{i-1}, and X_{i-2}, for example-are positive (and often are taken to have the same value). These weights are shown as W_1 and W_2 in figure. The weights between X_i and units X_{i+3} and X_{i-3} are negative (shown as W_3 in the figure). Unit X_i, is not connected to units X_{i-4} and X_{i+4} in this sample architecture. In the illustration, units within a radius of 2 to the typical unit *X*, are connected with positive weights; units within a radius of 3, but outside the radius of positive connections, are connected with negative weights; and units further than 3 units away are not connected.(Figure is shown on the next page).



Mexican Hat interconnections for unit X_i.

3.2 Algorithm

The algorithm given here is similar to that presented by Kohonen [1989a]. The nomenclature we use is as follows:

R2	Radius of regions of interconnection; X_i is connected to units X_{i+k} and X_{i-k} for i = 1, 2… R2.
R1	Radius of region with positive reinforcement; R1<R2
W_k	Weight on interconnections between Xi and units Xi+k and Xi-k: Wk is positive for $0 \leq k \leq R1$. Wk is negative for $R1 < k \leq R2$.
X	Vector of activation.
X _old	Vector of activations at previous step.
T _max	Total number of iterations of contrast enhancement.
S	External signal.

As presented, the algorithm corresponds to external signal being given only for the first iteration (Step-1) of the contrast enhancing iterations. We have:

Step 0: Initialize parameters T _max, R1, R2 as desired.
Initialize weights:
Wk=C1 for k=0…R1 (C1>0)
Wk=C2 for k=R1+1…R2 (C2<0).
Initialize X _old to 0.

Step 1: Present external signal X:
X = S
Save activations in array X _old (for i=1 … n).
X _old $_i$= X_i
Set iteration counter T=1.

Step 2: While T is less than T _max Repeat Steps 3-7

Step 3: Compute net input (i=1…n):

$$Xi = C1 \sum_{k=-R1}^{R1} X\_old_{\,i+k} + C2 \sum_{k=-R2}^{-R1-1} X\_old_{\,i+k} + C2 \sum_{k=R1+1}^{R2} X\_old_{\,i+k}$$

Step 4: Apply activation function (ramp function from 0 to X _max, Slope=1):
X_i = min (X _max, max (0, X_i)) (i=1…n)

Step 5: Save current activations in X _old
X _old $_i$ = X_i (i=1…n).

Step 6: Increment iteration counter:
T=T+1

Step 7: Test stopping condition:
If T < T _max, continue; otherwise stop.

In a computer implementation of this algorithm, one simple method of dealing with the units near the ends of the net, i.e., for i close to I or close to *n*, which receive input from less than the full range of units i – *R2* to i+*R*1 is to dimension the array *X _old* from 1 – *R2* to *n* + R1, (rather than from 1 to *n).* Then, since only the components from I to *n* will be updated, the formulas in Step 3 will work correctly for all units.

The positive reinforcement from nearby units and negative reinforcement from units that are further away have the effect of increasing the activation of units with larger initial activations and reducing the activations of those that had a smaller external signal. This is illustrated in the following example.

3.3 Application

The following example illustrates the use of Mexican Hat Network for a simple net with seven units. The activation function for this net is:

$$f(x)=\begin{cases} 0 \; if \; x = 0 \\ x \; if \; 0 \le x \le 2 \\ 2 \; if \; x > 2 \end{cases}$$

Step 0. Initialize parameters:

$R_1 = 1;$

$R_2 = 2;$

$C_1 = 0.6;$

$C_2 = -0.4.$

Step 1. *(t = 0).*

The external signal is (0.0, 0.5, 0.8, 1.0, 0.8, 0.5, 0.0), so

$\mathbf{x} = (0.0, 0.5, 0.8, 1.0, 0.8, 0.5, 0.0).$

Save in x_old:

$\mathbf{x\_old} = (0.0, 0.5, 0.8, 1.0, 0.8, 0.5, 0.0).$

Step 2. *(t = 1).*

The update formulas used in Step 3 are listed as follows for reference:

$$x_1 = 0.6\, x\_old_1 + 0.6\, x\_old_2 - 0.4\, x\_old_3$$

$$x_2 = 0.6\, x\_old_1 + 0.6\, x\_old_2 + 0.6\, x\_old_3 - 0.4\, x\_old_4$$

$$x_3 = -0.4\, x\_old_1 + 0.6\, x\_old_2 + 0.6\, x\_old_3 + 0.6\, x\_old_4 - 0.4\, x\_old_5$$

$$x_4 = -0.4\, x\_old_2 + 0.6\, x\_old_3 + 0.6\, x\_old_4 + 0.6\, x\_old_5 - 0.4\, x\_old_6$$

$$x_5 = -0.4\, x\_old_3 + 0.6\, x\_old_4 + 0.6\, x\_old_5 + 0.6\, x\_old_6 - 0.4\, x\_old_7$$

$$x_6 = -0.4\, x\_old_4 + 0.6\, x\_old_5 + 0.6\, x\_old_6 + 0.6\, x\_old_7$$

$$x_7 = -0.4\, x\_old_5 + 0.6\, x\_old_6 + 0.6\, x\_old_7.$$

Step 3. *(t = 1).*

$$x_1 = 0.6(0.0) + 0.6(0.5) - 0.4(0.8) = -0.2$$

$$x_2 = 0.6(0.0) + 0.6(0.5) + 0.6(0.8) - 0.4(1.0) = 0.38$$

$$x_3 = -0.4(0.0) + 0.6(0.5) + 0.6(0.8) + 0.6(1.0) - 0.4(0.8) = 1.06$$

$$x_4 = -0.4(0.5) + 0.6(0.8) + 0.6(1.0) + 0.6(0.8) - 0.4(0.5) = 1.16$$

$$x_5 = -0.4(0.8) + 0.6(1.0) + 0.6(0.8) + 0.6(0.5) - 0.4(0.0) = 1.06$$

$$x_6 = -0.4(1.0) + 0.6(0.8) + 0.6(0.5) + 0.6(0.0) = 0.38$$

$$x_7 = -0.4(0.8) + 0.6(0.5) + 0.6(0.0) = -0.2.$$

Step 4.

$$\mathbf{x} = (0.0, 0.38, 1.06, 1.16, 1.06, 0.38, 0.0).$$

Steps 5–7. Bookkeeping for next iteration.

Step 3. *(t = 2).*

$$x_1 = 0.6(0.0) + 0.6(0.38) - 0.4(1.06) = -0.196$$

$$x_2 = 0.6(0.0) + 0.6(0.38) + 0.6(1.06) - 0.4(1.16) = 0.39$$

$$x_3 = -0.4(0.0) + 0.6(0.38) + 0.6(1.06) + 0.6(1.16) - 0.4(1.06) = 1.14$$

$$x_4 = -0.4(0.38) + 0.6(1.06) + 0.6(1.16) + 0.6(1.06) - 0.4(0.38) = 1.66$$

$$x_5 = -0.4(1.06) + 0.6(1.16) + 0.6(1.06) + 0.6(0.38) - 0.4(0.0) = 1.14$$

$$x_6 = -0.4(1.16) + 0.6(1.06) + 0.6(0.38) + 0.6(0.0) = 0.39$$

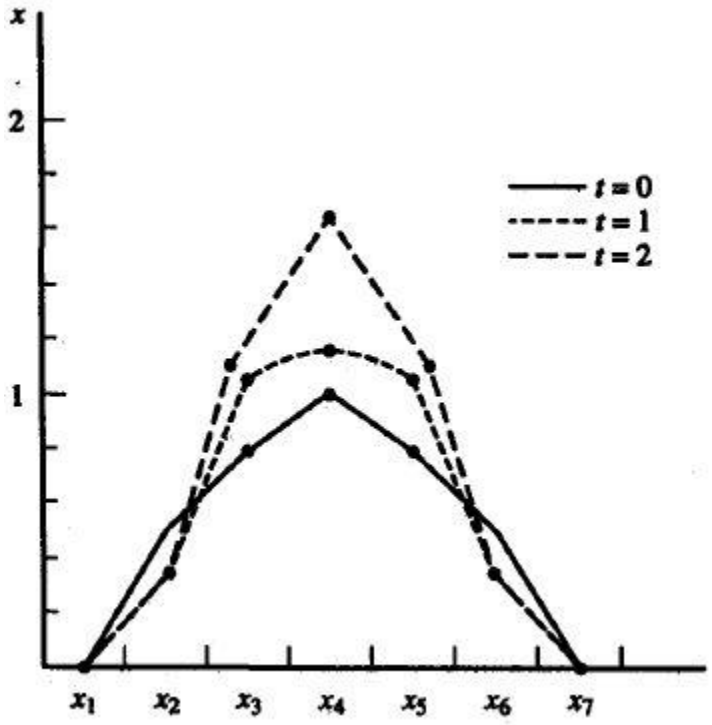
$$x_7 = -0.4(1.06) + 0.6(0.38) + 0.6(0.0) = -0.196$$

Step 4.

$$\mathbf{x} = (0.0, 0.39, 1.14, 1.66, 1.14, 0.39, 0.0).$$

Steps 5–7. Bookkeeping for next iteration.

The pattern of activations is shown in the following figure:



Questions:

Q.1 How is Mexican Hat Network different from the MAXNET?

Q.2 What are the merits and demerits of using Mexican Hat Networks?

Q.3 What is the role of C1 and C2 in the algorithm?

Q.4 How does the value of T_ max alters the amount of learning of the Mexican Hat Network?

Q.5 What is meant by On-Center-Off-Surround contrast enhancement?

Bibliography

1. Fundamentals of Neural Networks by Laurene Fausett.
2. Neural Networks- A Comprehensive Foundation by Simon Haykin.
3. Kohonen [1989a].