Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program

605-447/625-438

Dr. Mark Fleischer

Module 12.2: Competitive Learning--MAXNET

In This Module We Will Cover:

- Competitive Learning
  - The Hamming Net
  - The MAXNET algorithm
  - Self-Organizing Maps
  - Data values ‘compete’ in some fashion

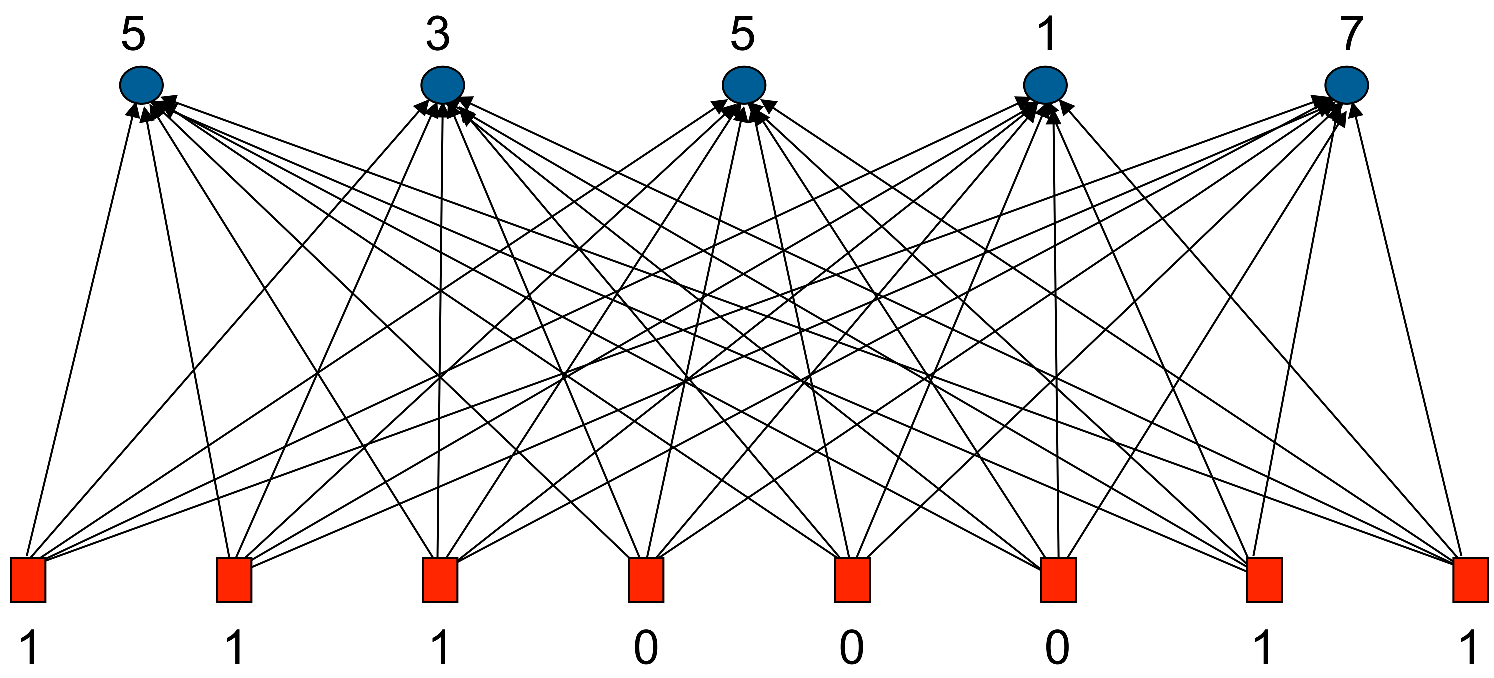
Using the Weights

- Each of the output nodes calculates the value of H based on its weight vector and the input vector.
- The node with the greatest value is the one most similar to the input!
- So ….?

The Hamming/MAXNET Network

5 Exemplars:

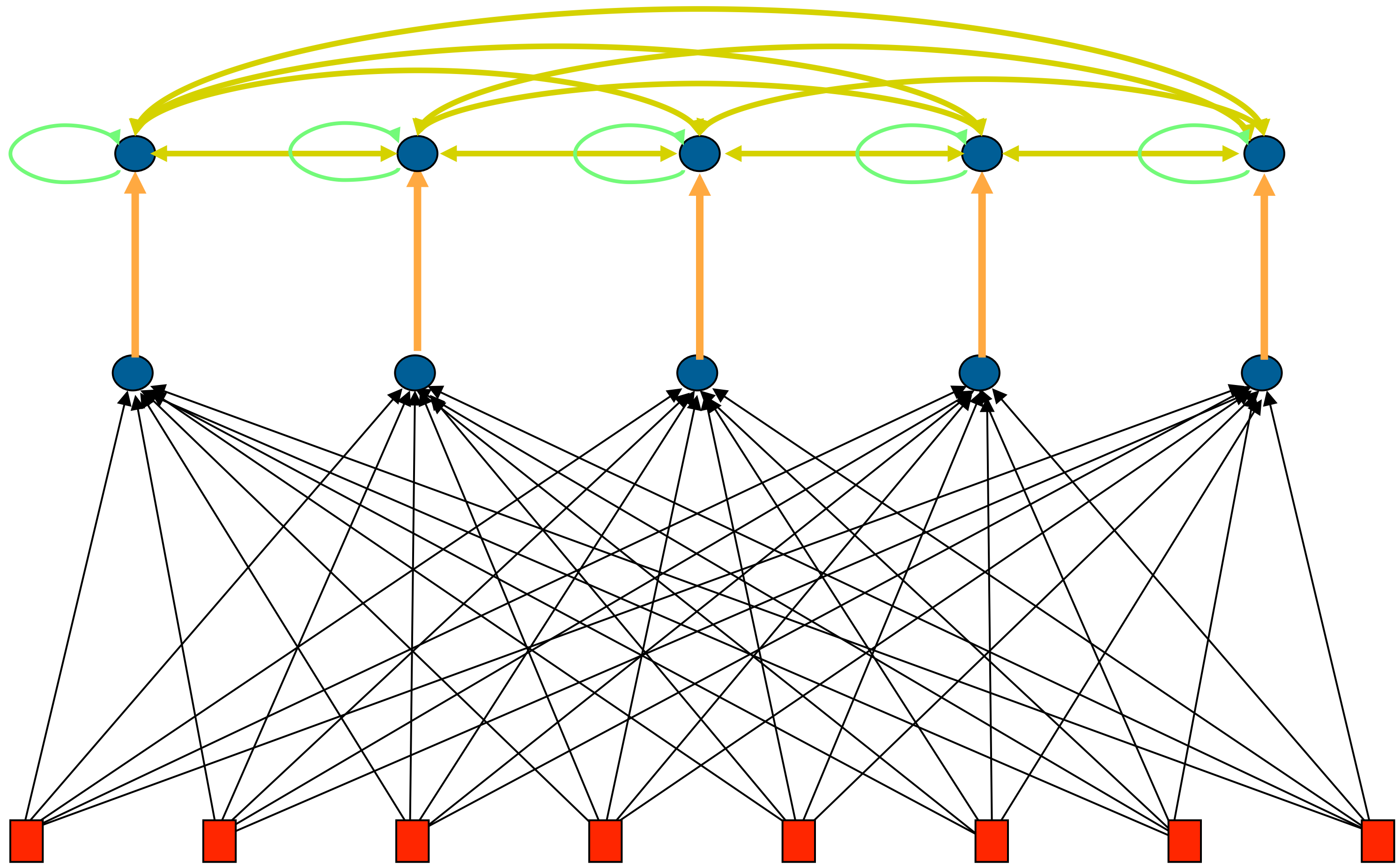
1:	1 0 1 0 1 0 1 0
2:	0 0 0 0 1 1 1 1
3:	1 1 1 1 0 0 0 0
4:	0 0 1 1 1 1 0 0
5:	1 1 0 0 0 0 1 1



The Hamming/MAXNET Network

- We want the network to “tell us” which node has the highest value of H.
- These nodes will have to fight it out!
- Competitive Learning!
- Add MAXNET to the Hamming Net.

The Hamming/MAXNET Network



The Hamming/MAXNET Network

- The MAXNET is a fully connected Hopfield type network.
- Weights w_{jk} are set by:

$$w_{jk} = \begin{cases} 1 & \text{if } j = k \\ -\varepsilon & \text{otherwise} \end{cases}$$

where $\varepsilon < 1/M$

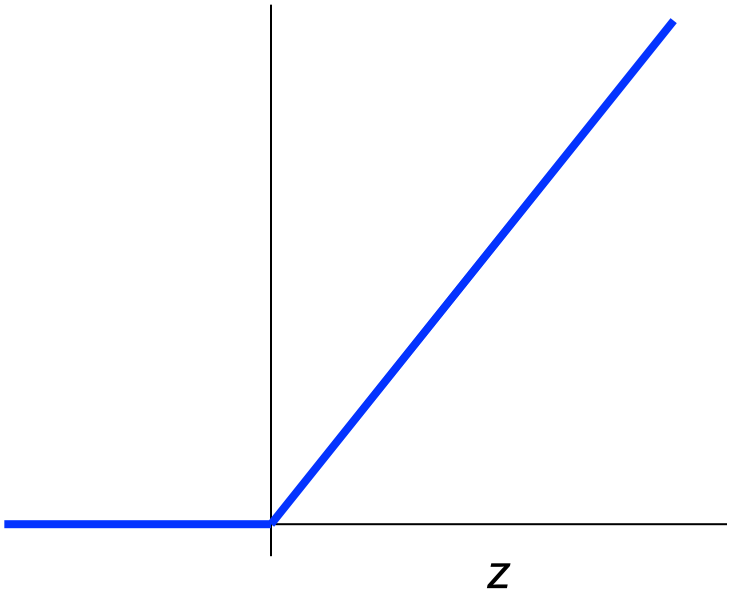
MAXNET

Each node calculates its activity value as usual and uses the hard-limiting function for its activation function:

$$A_j(t) = \sum_{k=1}^{M} w_{jk} x_k(t)$$

$$x_j(t+1) = f_t(A_j(t)) \quad \text{where}$$

$$f_t(z) = \begin{cases} az & \text{for } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



MAXNET

1. Calculate H using Hamming Net.
2. Present values of H to MAXNET.
3. Iterate MAXNET until convergence.

$$A_j(t) = \sum_{k=1}^{M} w_{jk} x_k(t)$$

$$x_j(t+1) = f_t(A_j(t))$$

$$= f_t\left[x_j(t) - \varepsilon \sum_{\substack{k=1 \\ k \neq j}}^{M} x_k(t) \right]$$

MAXNET

Time 0:	5	3	5	1	7

A for Node 1 = 5 – (1/6)(3 + 5 + 1 + 7) = 2.3333 Decreased by 2 2/3

A for Node 2 = 3 – (1/6)(5 + 5 + 1 + 7) = 0

A for Node 3 = 5 – (1/6)(5 + 3 + 1 + 7) = 2.3333

A for Node 4 = 1 – (1/6)(5 + 3 + 5 + 7) = -2.3333

A for Node 5 = 7 – (1/6)(5 + 3 + 5 + 1) = 4.6666 Decreased by 2 1/3!

Time 1:	2.333	0	2.333	0	4.666

MAXNET

Time 1:	2.333	0	2.333	0	4.666

A for Node 1 = 2.333 – (1/6)(0 + 2.333 + 0 + 4.666) = 1.16666

A for Node 2 = 0 – (1/6)(2.333 + 2.333 + 0 + 4.666) = -1.16666

A for Node 3 = 2.333 – (1/6)(2.333 + 0 + 0 + 4.666) = 1.16666

A for Node 4 = 0 – (1/6)(2.333 + 0 + 2.333 + 4.666) = -1.5555

A for Node 5 = 4.666 – (1/6)(2.333 + 0 + 2.333 + 0) = 3.8888

Time 2:	1.1666	0	1.1666	0	3.888

MAXNET

The keys to understanding it:

$$A_j(t) = \sum_{k=1}^{M} w_{jk} x_k(t)$$

$$x_j(t+1) = f_t(A_j(t))$$

$$= f_t\left[x_j(t) - \varepsilon \sum_{\substack{k=1 \\ k \neq j}}^{M} x_k(t) \right]$$

$$f_t(z) = \begin{cases} az & \text{for } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

MAXNET

Considerations:

- It can be **proven** that MAXNET always converges and find the node with maximum value so long as $\varepsilon < 1/M$.
- It has **advantages** over Hopfield net. It implements the optimum minimum error classifier when bit errors are random and independent.
- Tends to require **fewer** connections than in Hopfield. E.g., with 100 inputs and 10 classes, only 1100 connections are needed whereas with Hopfield with 100 inputs, 10,000 connections are needed. This difference increases as the number of inputs increase.

Summary

- The Hamming Distance measure for discrete vectors was modified to define a metric that measures how well a discrete vector matches another discrete vector.
- Defined an iterative scheme such that a node that has the greatest activity value will be the only node in a set of nodes that has a positive value.
- Competitive Learning.
- Can be used in Self-Organized Maps. Stay tuned!