Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program

605-447/625-438

Dr. Mark Fleischer

Module 12.1: The Hamming Network

What We've Covered So Far

- Examined approaches to unsupervised learning methods
  - Recurrent networks: Hopfield networks, Boltzmann Machines
  - BAMs and RBMs
  - Data itself 'trains' the network

In This Module We Will Cover:

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

A Basic Problem in Communications

- In Hopfield Net, the network converged to an exemplar from a ‘noisy’ version of the exemplar.
- Let’s try a different approach based on ‘classification’.

What to do with a noisy exemplar?

- Compare it to all possible patterns and pick the one it is 'closest' to.
- For binary information, we use the 'Hamming distance'.
- Recall the notion of distance from the material on metric spaces.

Hamming Distance

- For two binary strings, the Hamming Distance is the number of corresponding bits (vector elements) that are different.
- A Hamming Distance of zero implies the two strings are the same.
- Example:
  - x = {0 0 1 1 0 1 0 1}
  - x′ = {0 1 1 1 0 1 0 0}

$$(x, y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{otherwise} \end{cases}$$

- HD = $\sum_{i=1}(x_i, x'_i)$ = ?

Using the Hamming Distance

- Suppose we want a *larger* number to correspond to better matching?
  - i.e., score $\propto$ similarity

- Define:

$$H(\mathbf{x},\mathbf{x}') = N - \sum_{i=1}^{N} (x_i, x_i')$$

- We could use this to determine the nearest exemplar to determine the best match.

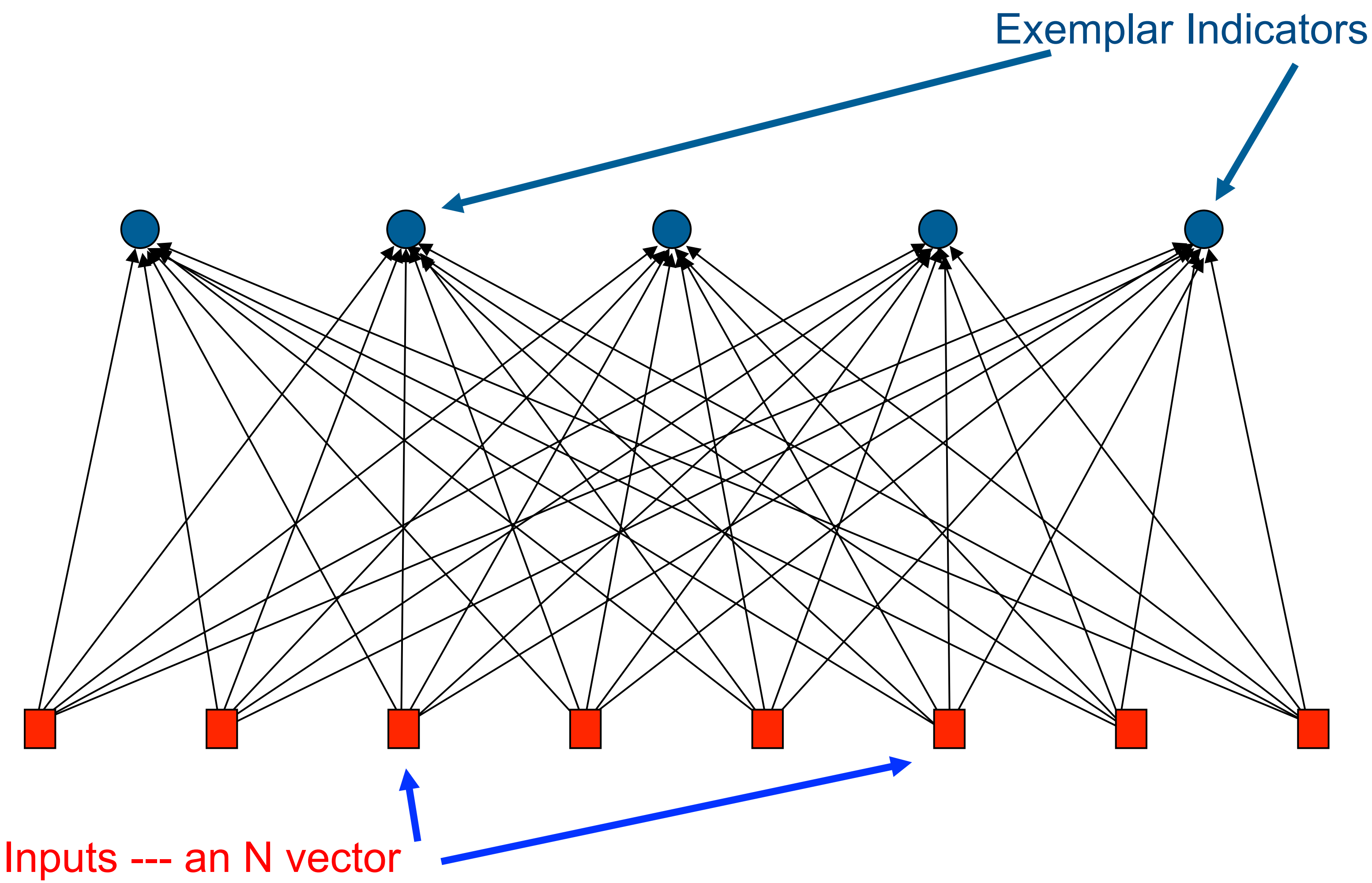
The Hamming Network

- This is more interesting.
- Let a network decide which exemplar is the best match to a network input.
- Let a node designate a particular exemplar which 'fires' a particular node when the network is presented with a noisy input that is closest to it.
- Use a competitive learning approach.

The Hamming Network

- Suppose we have M exemplars of vectors each with N elements.
- We want only one of the M neurons to fire corresponding to the exemplar closest to the noisy input.

The Hamming Network



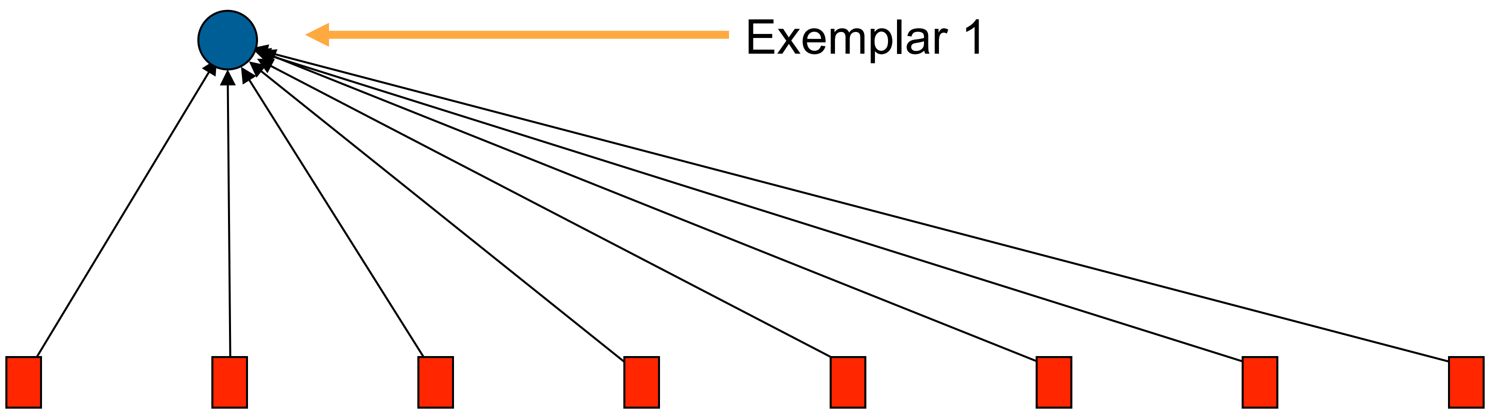
The Hamming Network

- Each output has N weight connections --- one for each input.
- We need to set the weights so that one exemplar that an input is closest to will have the highest value of H.

Setting Weights

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 Network

- Set weights w_{ij} for input i to exemplar j according to the exemplar pattern. One approach is …

$$w_{ij} = \frac{x_i^j}{2}, \quad \theta_j = \frac{N}{2}$$

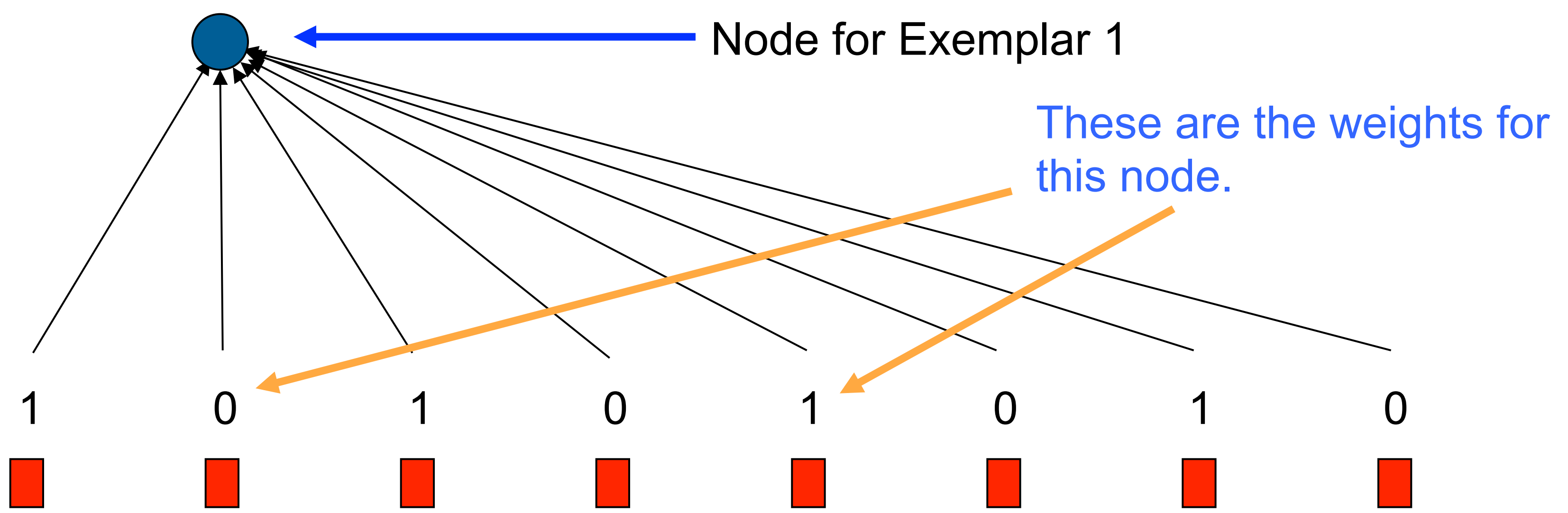
where x_i^j is the i^{th} element of exemplar j.

Another approach is to simply use the exemplar pattern itself and then calculate the value of H.

Setting Weights

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



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 ….?