# Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program

605-447/625-438

Dr. Mark Fleischer



Module 9.2: Binary Associative Memories

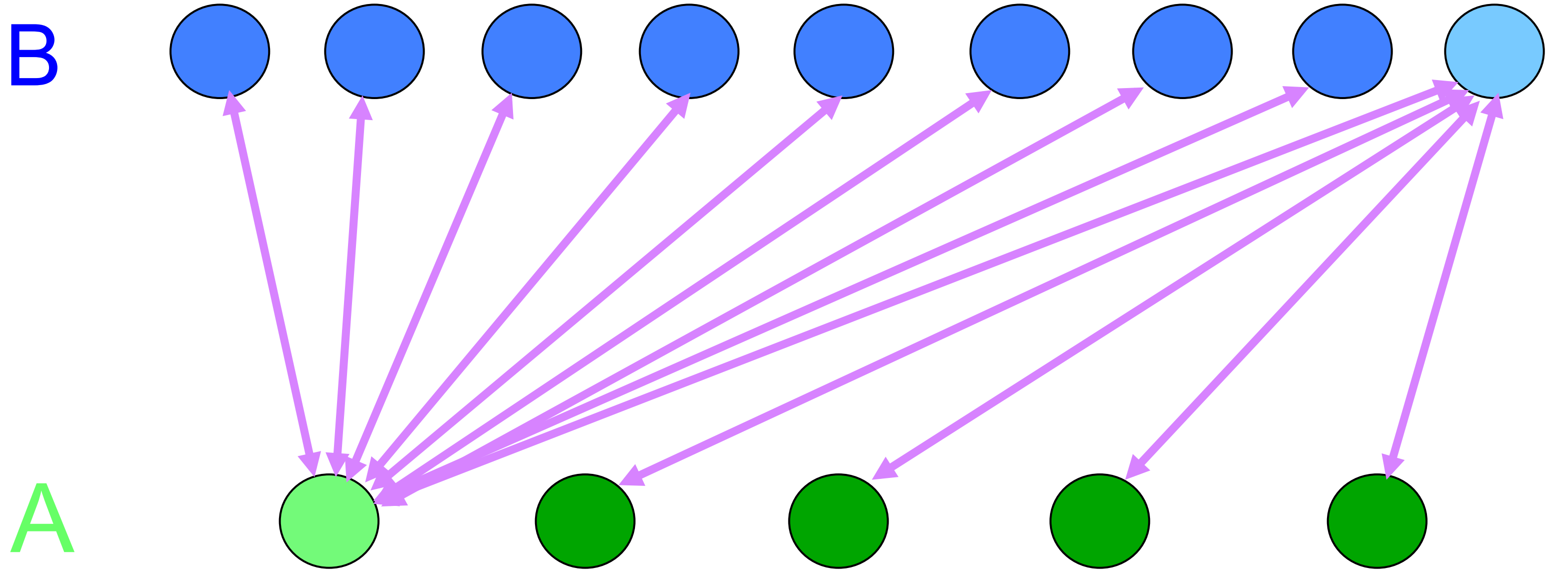
# In this sub-module…

- We will learn about Binary Associative Memories (BAMs)
  - Based on *Adaptive Resonance Theory*
  - Another example of *unsupervised learning*
  - Restricted form of a Hopfield network
- We will also learn about
  - Concepts such as *Feature Detectors*
  - Matrix/vector analysis of BAMs.

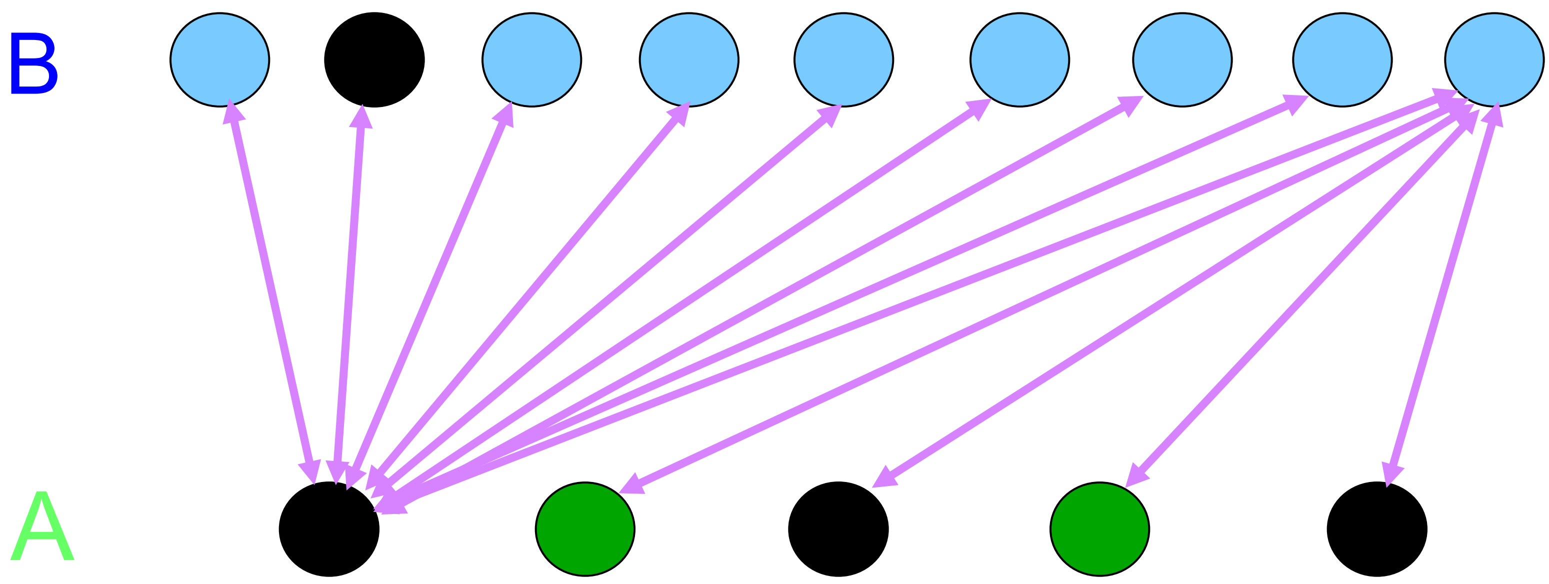
# Binary Associative Memories

- Based on a bi-partite graph
- All nodes from one layer connect to all nodes in a second layer.
- No nodes in the same layer connect to each other.
- Connections are bi-directional.
- Same weights in each direction (more general examples don't have this.)

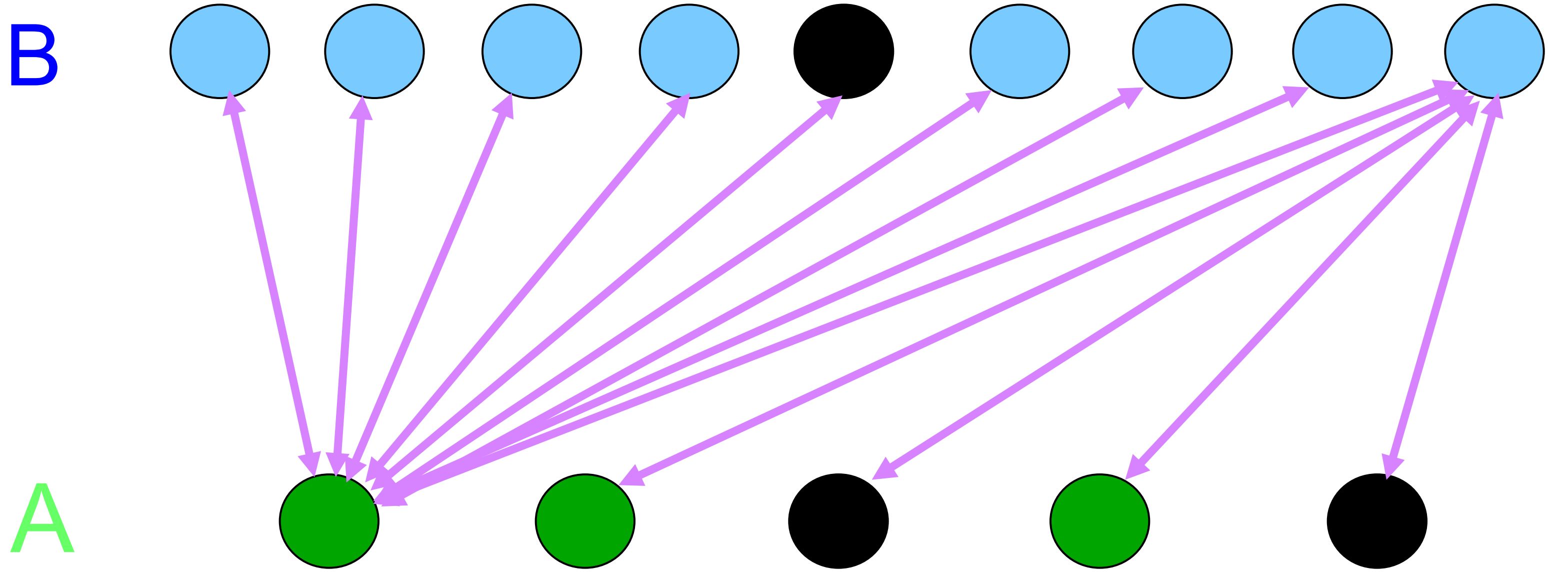
# Binary Associative Memories



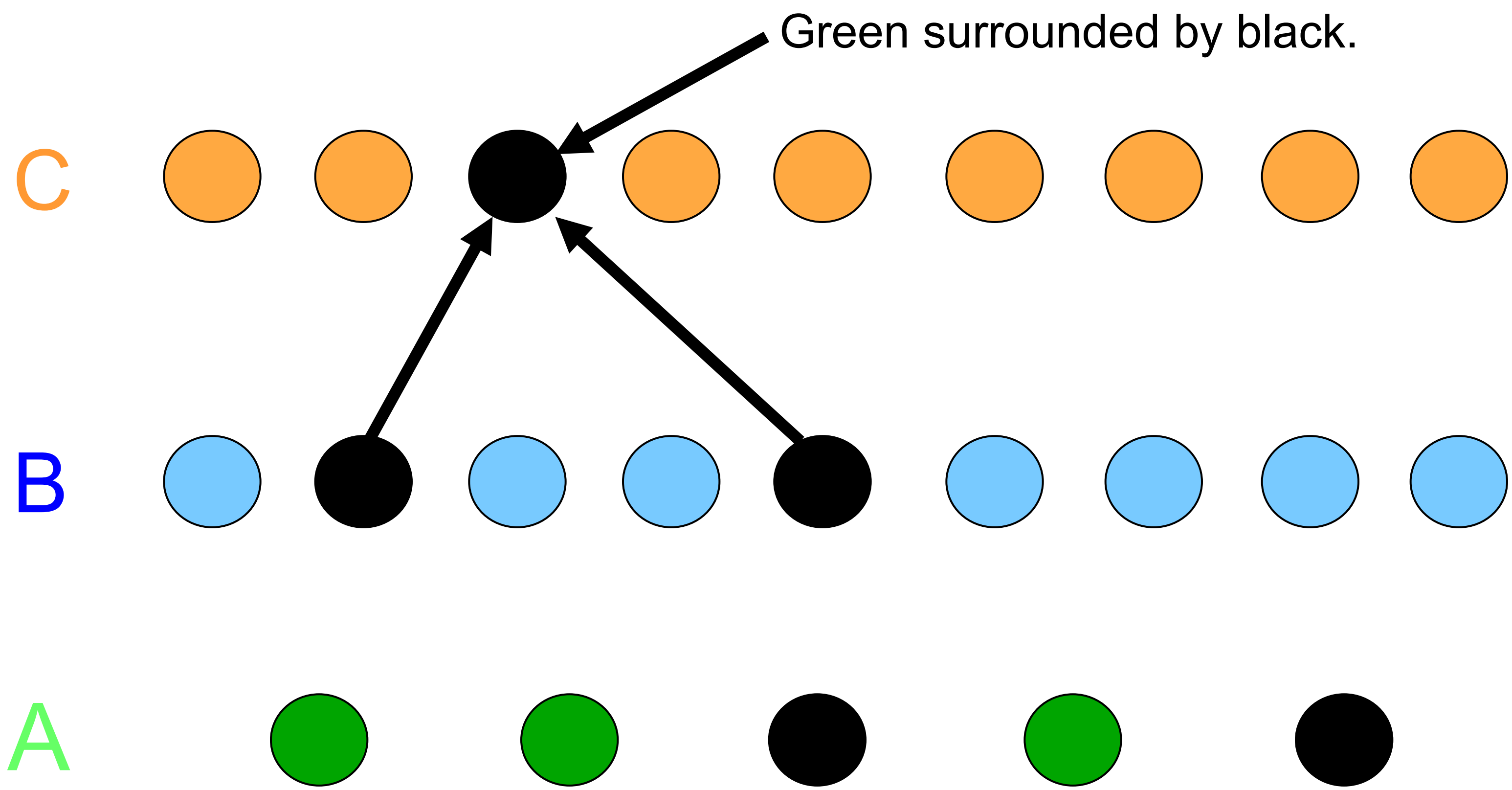
# Binary Associative Memories as Feature Detectors



# Binary Associative Memories as Feature Detectors



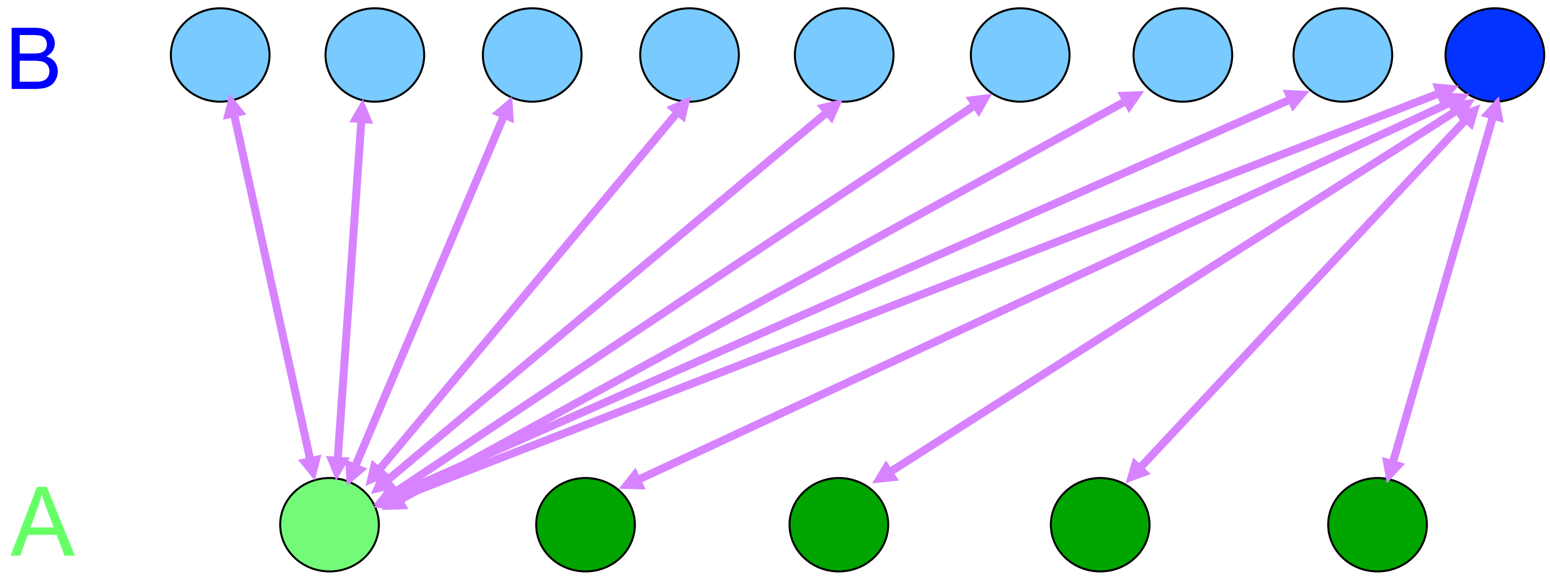
# Binary Associative Memories
# Features of Features



# Binary Associative Memories

- More layers, more feature generalizations, more abstractions.
- More layers, more versatility, more weights to train.
- For now we'll only consider two layers.

# Binary Associative Memories




We can use this topology to train the network with two sets of **associated** exemplars.

# Binary Associative Memories

$$A_1 \longleftrightarrow B_1$$

$$A_2 \longleftrightarrow B_2$$

$$\vdots \qquad\qquad \vdots$$

$$A_n \longleftrightarrow B_n$$

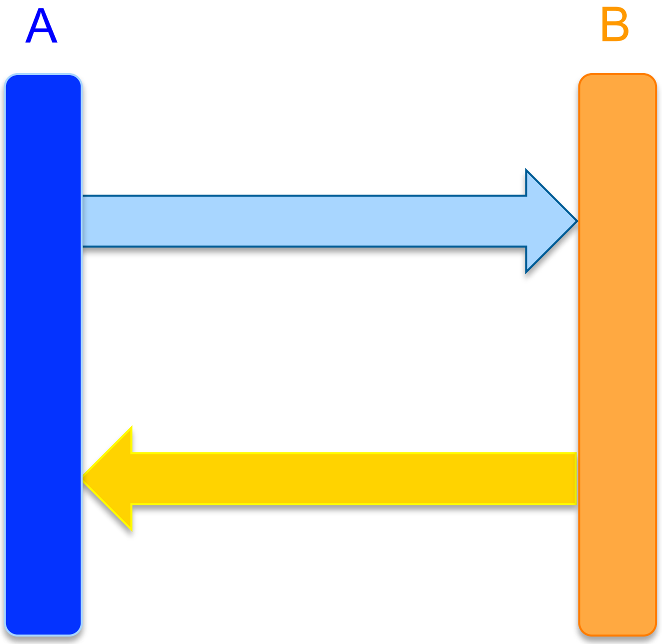
Goal: Noisy $A_1$ produces a correct $B_1$ which then produces a correct $A_1$.

$$\widetilde{A}_1 \rightarrow B_1 \rightarrow A_1$$

# Binary Associative Memories

- Present a noisy A as input to the A nodes.
- The A nodes produce outputs and are presented to the B nodes.
- The B nodes produce outputs and are presented back to the A nodes.

# Binary Associative Memories



# Binary Associative Memories

An example:

$$\mathbf{A}_1^{\mathbf{T}} = \begin{pmatrix} 1, & -1, & 1, & -1, & 1, & 1 \end{pmatrix}$$

$$\mathbf{A}_2^{\mathbf{T}} = \begin{pmatrix} 1, & 1, & 1, & -1, & -1, & -1 \end{pmatrix}$$

$$\mathbf{B}_1^{\mathbf{T}} = \begin{pmatrix} 1, & 1, & -1, & 1 \end{pmatrix}$$

$$\mathbf{B}_2^{\mathbf{T}} = \begin{pmatrix} 1, & -1, & 1, & 1 \end{pmatrix}$$

Want to associate $A_1 \Leftrightarrow B_1$ and $A_2 \Leftrightarrow B_2$

# Binary Associative Memories

Create a weight matrix ala Hopfield thusly:

$$\mathbf{W}_{6\times 4} = \mathbf{A}_1\mathbf{B}_1^{\mathrm{T}} + \mathbf{A}_2\mathbf{B}_2^{\mathrm{T}}$$

$$= \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -1 & 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 1 & 1 \end{pmatrix}$$

# Binary Associative Memories

$$\mathbf{W}_{6\times4} = \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & 2 \\ -2 & 0 & 0 & -2 \\ 0 & 2 & -2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$

$$\mathbf{W}_{6\times4} = \mathbf{A}_1\mathbf{B}_1^{\mathbf{T}} + \mathbf{A}_2\mathbf{B}_2^{\mathbf{T}}$$

[1 × 6] × [6 × 4] = [1 × 4]

# Binary Associative Memories

$$\hat{\mathbf{A}}_1^{\mathrm{T}}\mathbf{W} = \begin{pmatrix} -1, & -1, & 1, & -1, & 1, & 1 \end{pmatrix} \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & 2 \\ -2 & 0 & 0 & -2 \\ 0 & 2 & -2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$

$$\begin{pmatrix} 4, & 4, & -4, & 4 \end{pmatrix} = \hat{\mathbf{b}}^{\mathrm{T}}$$

$$f_h\left(\hat{\mathbf{b}}^{\mathrm{T}}\right) = \begin{pmatrix} 1, & 1, & -1, & 1 \end{pmatrix} = \mathbf{B}_1^{\mathrm{T}}$$

# Binary Associative Memories

Just some integer value!

$$\hat{\mathbf{A}}_1^T\mathbf{W} = \hat{\mathbf{A}}_1^T\mathbf{A}_1\mathbf{B}_1^T + \hat{\mathbf{A}}_1^T\mathbf{A}_2\mathbf{B}_2^T$$

How do we analyze this multiplication?
Why does this work the way it does?

So what do these integers evaluate to?

# Binary Associative Memories

When $\hat{\mathbf{A}}_1^{\mathbf{T}}$ is not too different from $\mathbf{A}_1^{\mathbf{T}}$, then most of the vector elements will be the same and $\hat{\mathbf{A}}_1^{\mathbf{T}}\mathbf{A}_1$ will be a positive number while $\hat{\mathbf{A}}_1^{\mathbf{T}}\mathbf{A}_2$ will tend to be ... ?

$$(1)\quad f_h(x) = \begin{cases} 1 & x \geq 1 \\ 0 & -1 < x < 1 \\ -1 & x \leq -1 \end{cases}$$

where $n$ is the vector length

$$(2)\quad f_h(x) = \begin{cases} 1 & x \geq n/2 \\ 0 & -n/2 < x < n/2 \\ -1 & x \leq -n/2 \end{cases}$$