WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF MATHEMATICS
AND INFORMATION SCIENCE

# Neural Networks

**Lecture 12**

# ***Associative memory***

# Associative memories

The massively parallel models of associative or content associative memory have been developed.
Some of these models are: Kohonen, Grossberg, Hamming and widely known Hopfield model.
The most interesting aspect of the most of these models is that they specify a learning rule which can be used to train network to associate input and output patterns.

# Associative memories

The associative network is a computational model emphasizing local and synchronous or asynchronous control, high parallelism, and redundancy. Such a network is a connectionist architecture and shares some common features with the Rosenblatt's Perceptron. However, that is much more powerful and flexible than the Perceptron.

# Associative memory model

The model has its origin both in the Hamming and Grossberg models.
The network model is composed of 3 layers or slabs: and input layer, an intermediate layer, and an output layer. The intermediate layer is a modified totally interconnected memoryless Grossberg slab with recurrent shunting on-center off-surround subnets, whose purpose is to achieve a majority vote so that only one neuron from this level, the one with the highest input value, will send its output to the next layer.

# Associative memory model

**The similarities to Grossbergs' model:**
interconnections between input layer and intermediate layer
**The similarities to Hammings' model**
interconnections (feedback) in the intermediate layer.

The connections between the input layer and intermediate layer contain all the information about one stored vector. The network is implementing the nearest-neighbor algorithm.

# Associative memory model

The number of elements in the intermediate layer defines the number of stored patterns..

All feedback connections within the intermediate layer are based on the rule of **lateral inhibition.**

The network is performing a

winner-takes-all

operation.

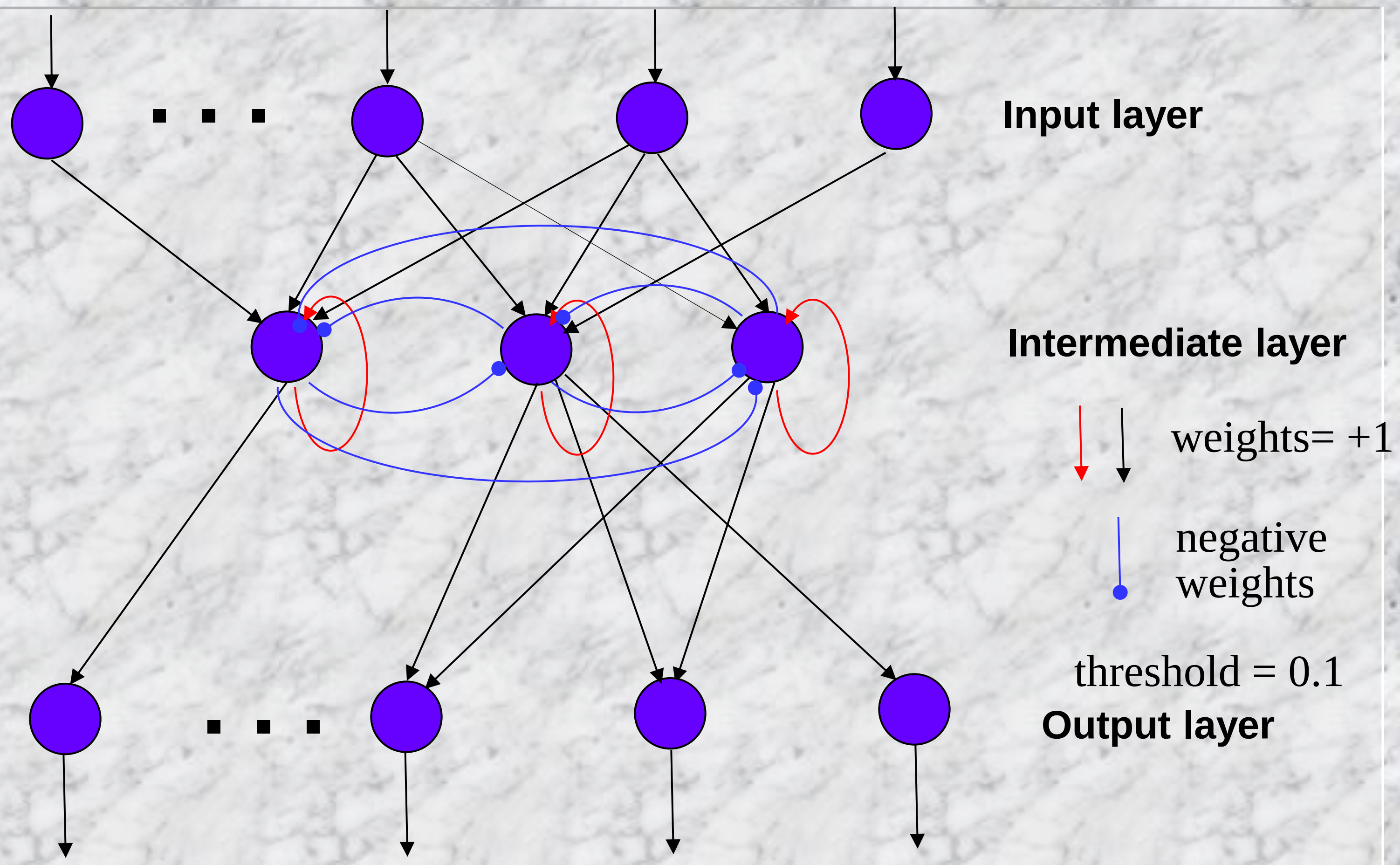
The elements of input signals (and stored vectors) are the binary values 0 and 1.

$$X = [x_1, x_2, x_3, ..., x_n] \qquad x_i \in \{0,1\}$$

# Associative memory model

The input and output elements (neurons) are only nodes whose purpose is to connect the inputs and outputs respectively to the intermediate slab. The network can be programmed to function as an autoassociative content-addressable memory or as symbolic substitution system which yields an arbitrary defined output for any input – it depends from the connections between the intermediate slab and the output layer.

# Associative memory model



# Associative memory model

**Programming the network**

The interconnections (weights) between the input elements and each intermediate neuron are independent to each other. Each intermediate element has its weights programmed to one input signal and these connections are left unchanged while the other neurons are programmed.

Adding or removing a new pattern does not influence to the existing network structure and weights.

# Associative memory model

The connection weights between the elements of the input layer and $j^{th}$ element of intermediate slab are:

- if the $i^{th}$ element of the input vector is equal to zero

$$w_i^j = 0$$

- if the $i^{th}$ element of the input vector is equal to one

$$w_i^j = \frac{1}{b_j}$$

where $b_j$ is the number of non-zero elements in the $j^{th}$ input vector to be stored.

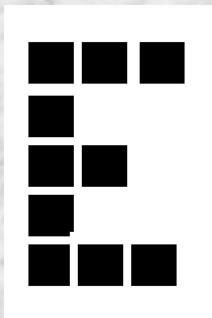
# Associative memory model

This procedure normalizes the total input to each element of the intermediate slab to the interval <0;1>, and takes not account the ***relative*** number of stored elements equal to the input elements, instead of the ***absolute*** number. It allows to distinguish between signals if one is included in another one.

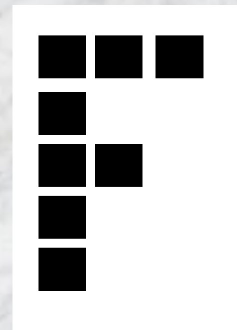
# Associative memory model

**Example:**

pattern ❶



pattern ❷



weights of intermediate slab element where the pattern is recorded

$$w_i^j = \frac{1}{10}$$

$$w_i^j = \frac{1}{8}$$

# Associative memory model

1. In the input signal is ❶, the output from both elements is equal to one.
2. If the input signal is ❷, the output signal from element 1 is equal to 0.8 hence from element 2 is equal to 1.0

The ambiguous output signal in the first case can be solved by the proper network structure.

# Associative memory model

This learning procedure is repeated for each input vector, each time with a new intermediate neuron.

The total number of different vectors that can be stored with this prescription in the net with $n$ – elements in the input layer is

$$\sum_{k=1}^{n} \binom{n}{k} = 2^{n} - 1$$

# Associative memory model

Each neuron in the intermediate slab is connected to all other neurons of this slab. The weight on the self feedback loop is equal to one, and all the other values depend on the correlation between stored vectors. The weight between the output of $j^{th}$ neuron and input of the $k^{th}$ neuron is given by

$$\boldsymbol{w}(\boldsymbol{k},\boldsymbol{j}) = \frac{1 + \boldsymbol{cor}(\boldsymbol{k},\boldsymbol{j})\boldsymbol{w}^{\boldsymbol{k}}}{2(\boldsymbol{M}-1)}$$

where $\boldsymbol{cor}(\boldsymbol{k},\boldsymbol{j})$ is correlation (inner product) between $k^{th}$ and $j^{th}$ stored vectors.

$\boldsymbol{w}^{\boldsymbol{k}}$ is one of the identical positive weight from the input slab to the $k^{th}$ neuron,

$\boldsymbol{M}$ is equal to the number of neurons in the intermediate slab with non-zero inputs.

# Associative memory model

The denominator ensures that the total lateral inhibition for the element with the greatest value is smaller that its input.
This procedure realizes the rule winner-takes-all.
The intermediate slab selects the maximum input, and drives all the other intermediate neurons to zero. If more then one intermediate neuron has the same maximum value, the slab will select the one that is less correlated to the remaining stored vectors.

# Associative memory model

The structure of connections in the intermediate slab **is not symmetrical**

$$w(k,j) \neq w(j,k), \text{ hence } w^j \neq w^k$$

If two or more neurons will have the same input signal, and the outputs may not be discriminated by the criterion, then the slab will be unable to distinguish between them and the outputs will be driven to zero or will be a superposition of the twp or more outputs.

# Associative memory model

**<u>Retrieval of stored vectors</u>**

At the input layer the unknown signal is applied and the network has to „recognize” it.

Of the stored vectors are orthogonal, any full of or partial input corresponding to one stored vector would cause only one neuron in the intermediate slab to have a non-zero output in the first iteration. When the stored vectors are not orthogonal, a certain number of neurons will be excited.

# Associative memory model

Let $\boldsymbol{f}$ is the unknown input signal

The elements of the vector $\boldsymbol{X}$ define the total input to the elements of the intermediate layer

$$\boldsymbol{X} = \boldsymbol{f} * \boldsymbol{W}^1$$

$W^1$ is the matrix of connections between the input layer and intermediate layer (the columns are equal to the input weights $w_k^j$ of each stored vector.

# Associative memory model

The output to of the first iteration is equal to

$$G = W^2 * X^T$$

where $W^2$ is square matrix of connections between elements of the intermediate slab

$$W^2 = \begin{bmatrix} 1 & -w(1,2) & -w(1,3) & \dots & -w(1,n) \\ -w(2,1) & 1 & -w(2,3) & \dots & -w(2,n) \\ \dots & \dots & \dots & \dots & \dots \\ -w(n,1) & -w(n,2) & -w(n,3) & \dots & 1 \end{bmatrix}$$

# Associative memory model

the iterative formula

$$\boldsymbol{G}(\boldsymbol{t}+1)=\boldsymbol{W}^{2}*\boldsymbol{G}(\boldsymbol{t})=\left(\boldsymbol{W}^{2}\right)^{\boldsymbol{t}}\left(\boldsymbol{f}*\boldsymbol{W}^{1}\right)^{\boldsymbol{T}}$$

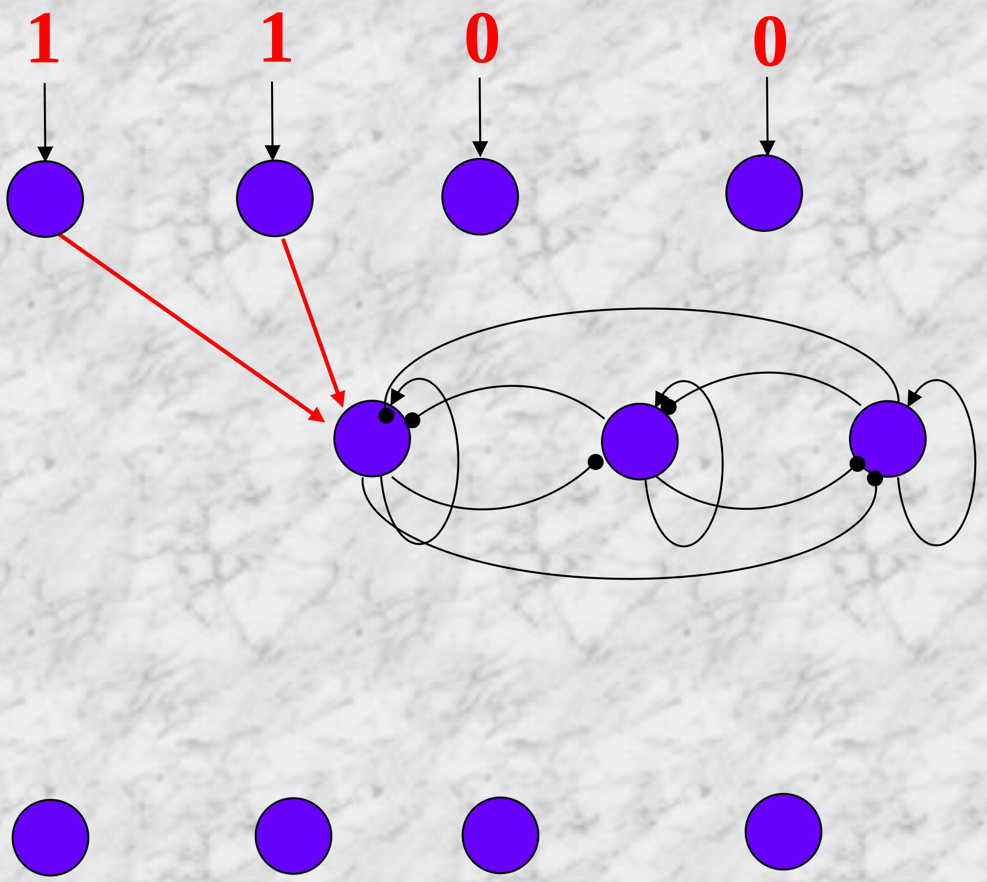
the output values are calculated by formula

$$\boldsymbol{Y}=\boldsymbol{W}^{3}*\boldsymbol{G}$$

$\boldsymbol{W}^{3}$ matrix of connections between the intermediate slab and the output layer; for the associative memory

$$\boldsymbol{W}^{3}=\boldsymbol{W}^{1}$$

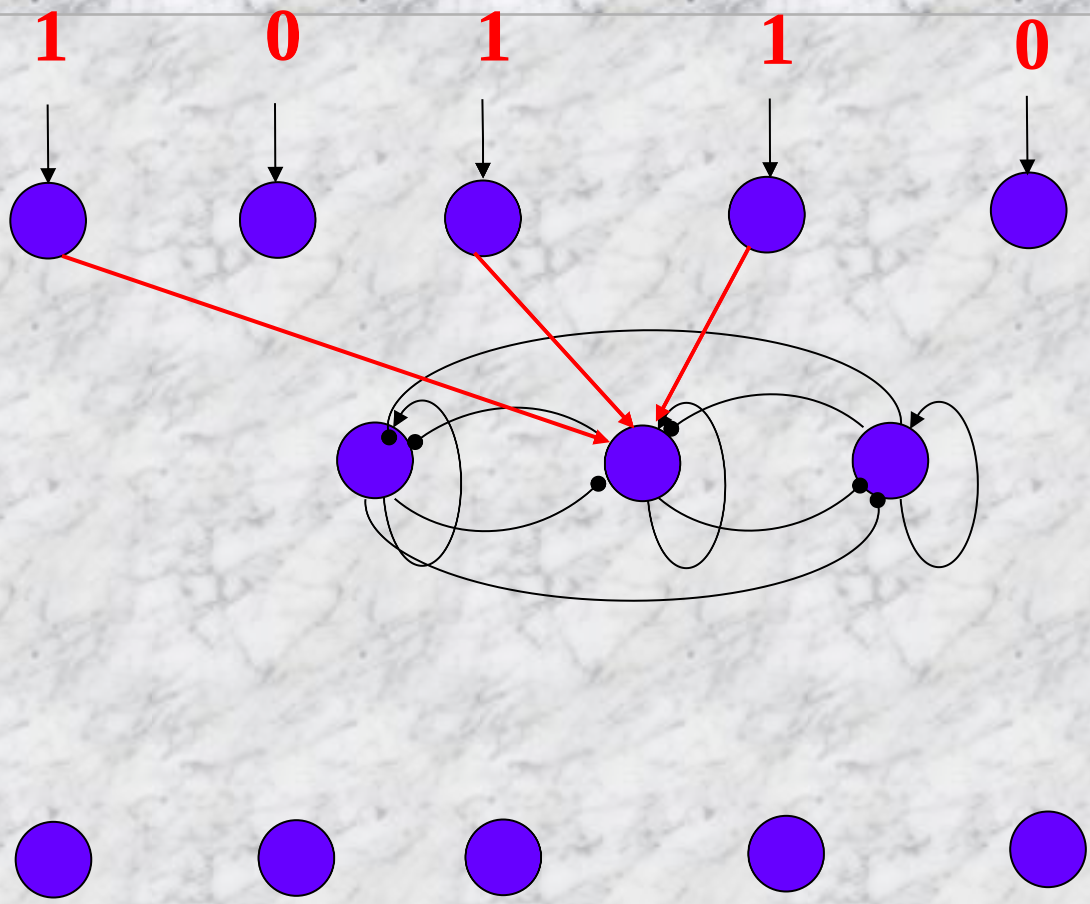
# Associative memory model




$V^1$=[1 1 0 0 0]

$$w_i^1 = \begin{cases} 0 & \text{for } i = 3,4,5 \\ 0.5 & \text{for } i = 1,2 \end{cases}$$

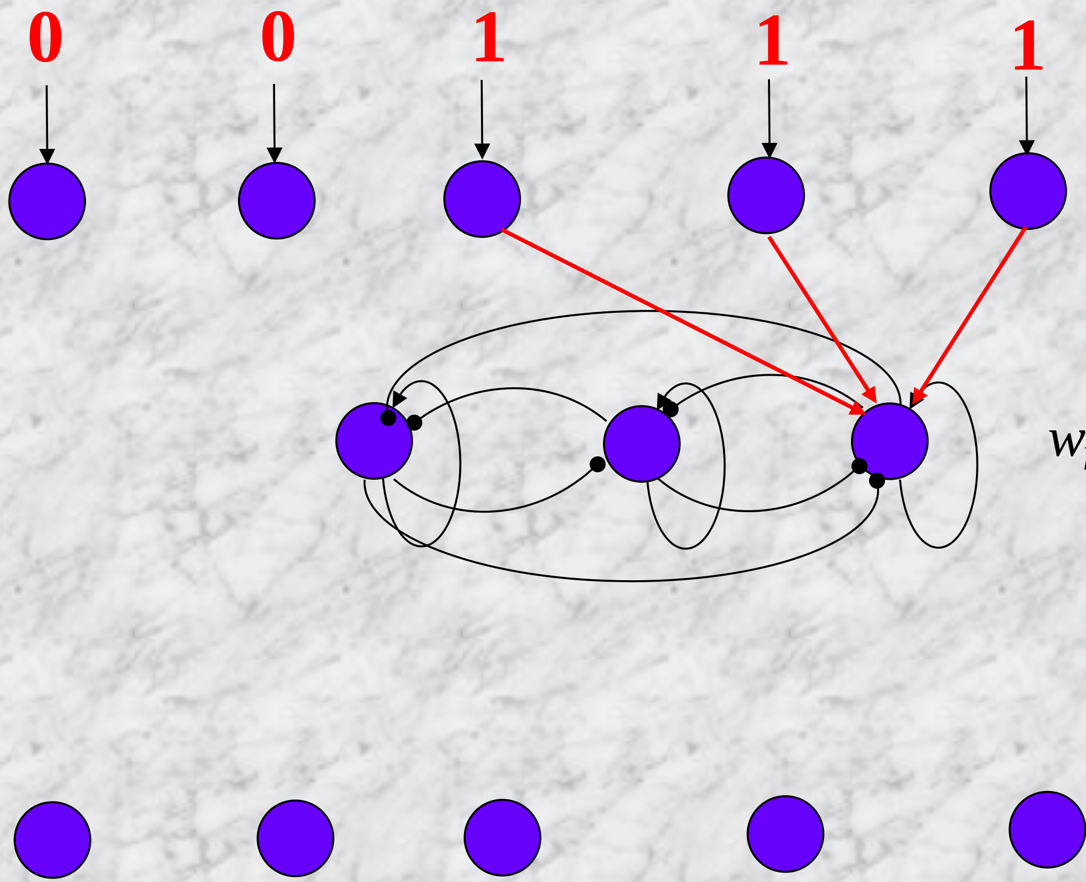
# Associative memory model




$$V^2=[1\ 0\ 1\ 1\ 0]$$

$$w_i^2 = \begin{cases} 0 & \text{for } i = 2,5 \\ 1/3 & \text{for } i = 1,3,4 \end{cases}$$

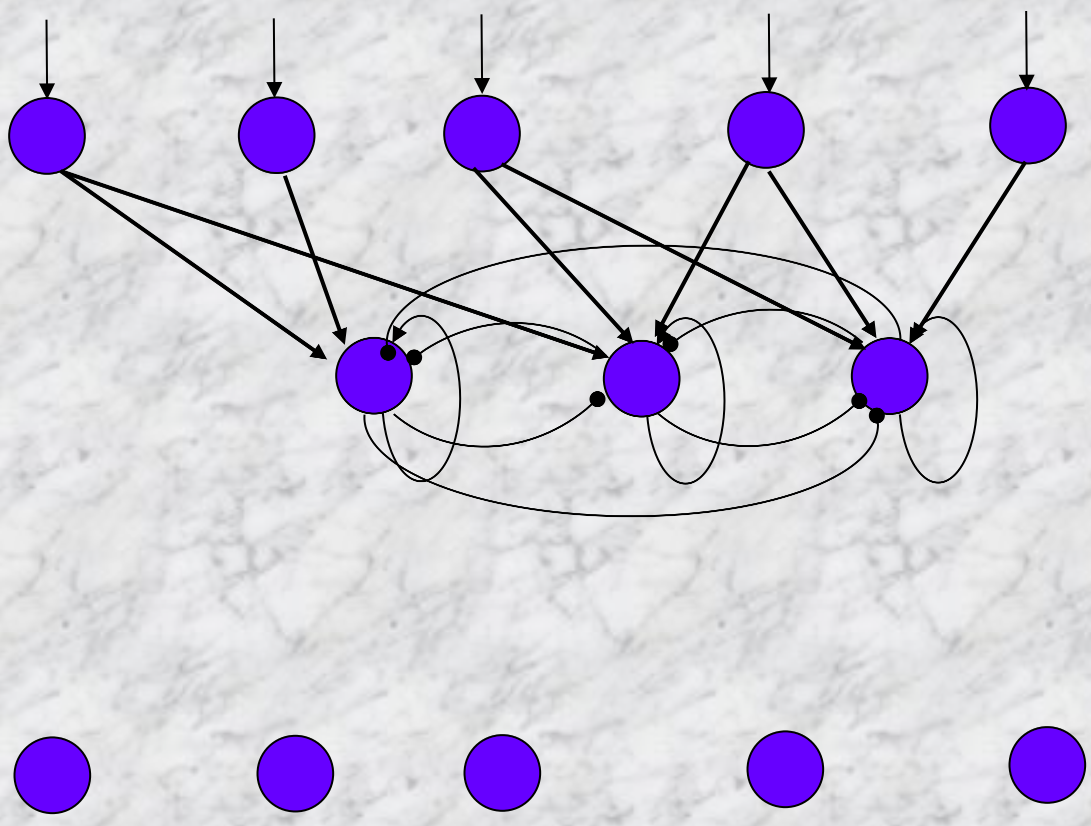
# Associative memory model




$$V^3=[0\ 0\ 1\ 1\ 1]$$

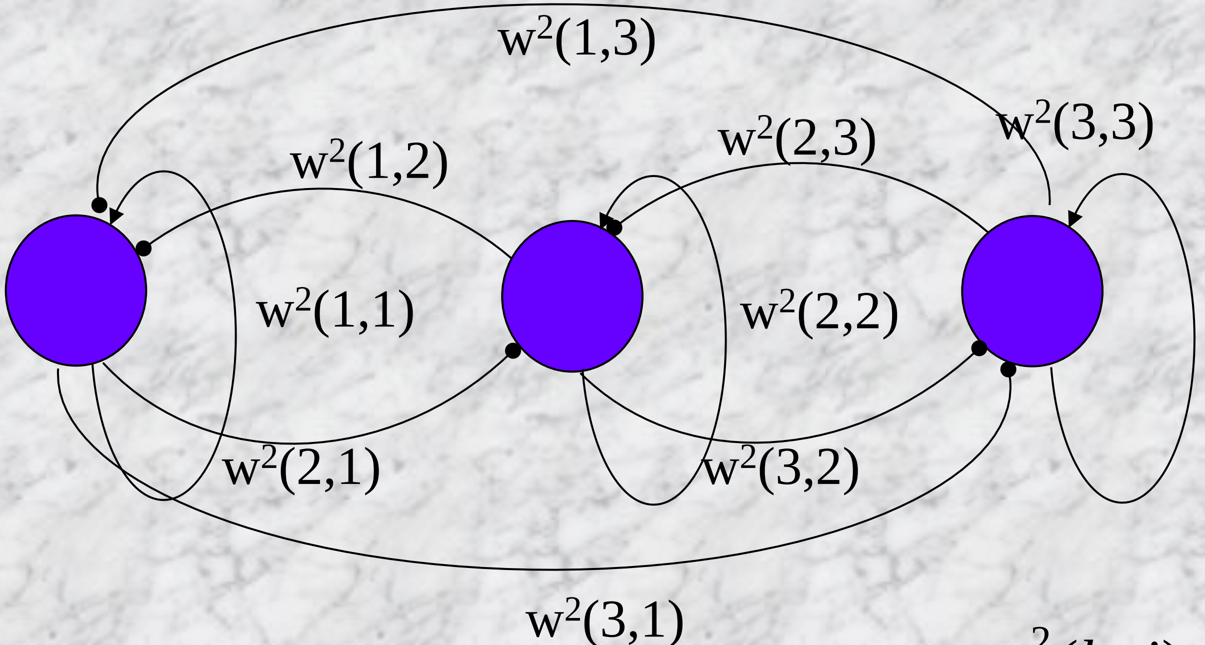
$$w_i^2 = \begin{cases} 0 & \text{for } i = 1,2 \\ 1/3 & \text{for } i = 3,4,5 \end{cases}$$

# Associative memory model



$$W^1 = \begin{bmatrix} 1/2 & 1/3 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 \\ 0 & 0 & 1/3 \end{bmatrix}$$

# Associative memory model



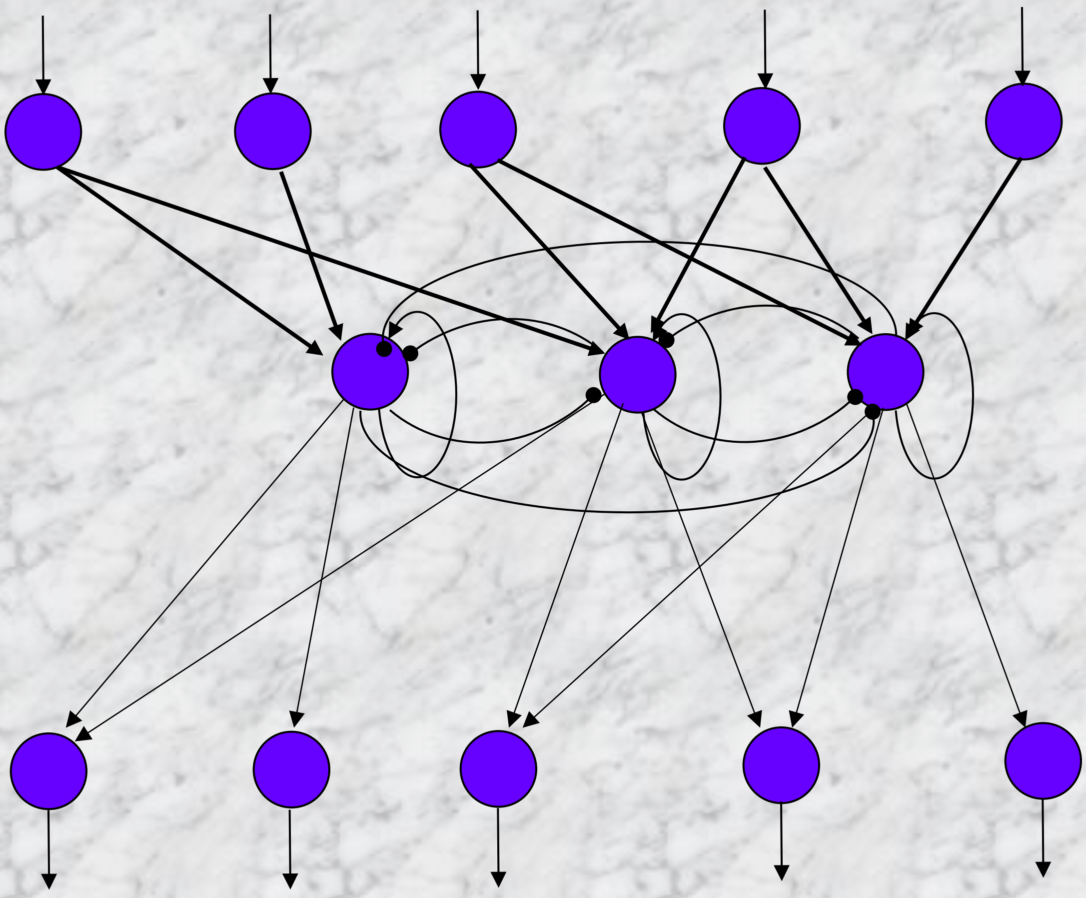

$$w^2(j,j)=1$$

$$w^2(k,j)=-\frac{1+cor(k,j)w^k}{2(n-1)}$$

$$\boldsymbol{w^k}=\boldsymbol{w_j^k}>0$$

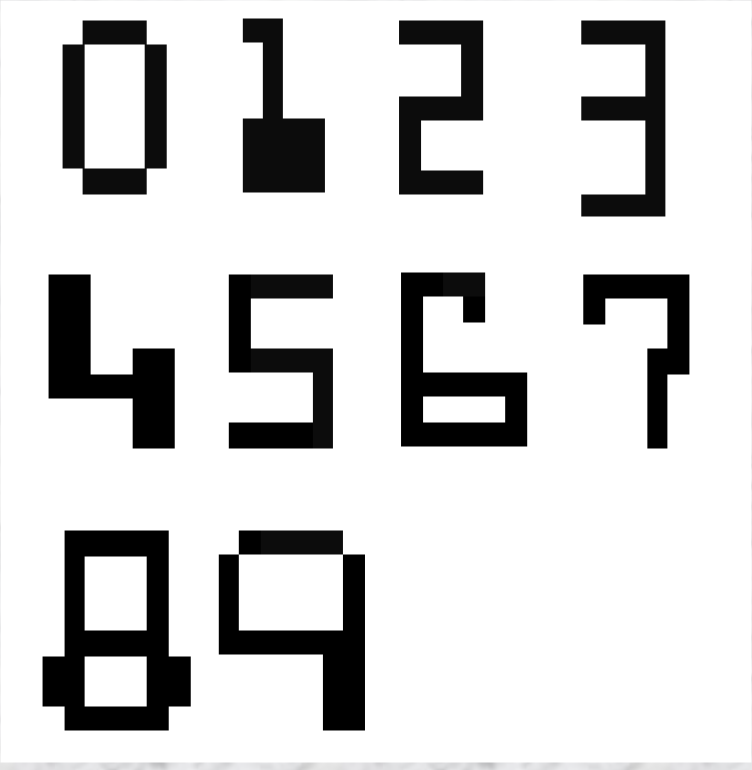
$$\boldsymbol{W}^2=\begin{bmatrix}1 & -3/4 & -1/2\\ -2/3 & 1 & -2/3\\ -1/2 & -2/3 & 1\end{bmatrix}$$

# Associative memory model



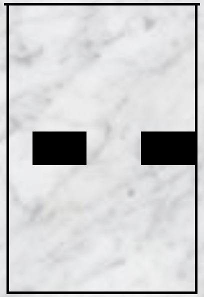
# Associative memory model

2D patterns stored in the network (9x6)

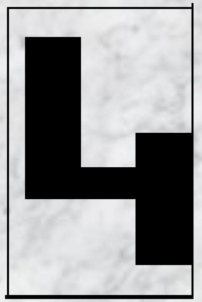


# Associative memory model

**Input signal** | **Output signal**



**after 3 iterations**



**after 4 iterations**