Biologically Inspired Computing: Neural Computation

Lecture 5

Patricia A. Vargas

Lecture 5

I. Lecture 4 – Revision
II. Artificial Neural Networks (Part III)
    I. Recurrent Artificial Networks
        I. Hopfield Network
        II. GasNet models

Artificial Neural Networks

- Learning Paradigms

$$w(t+1) = w(t) + \Delta w(t)$$

I. Supervised Learning (with a teacher)

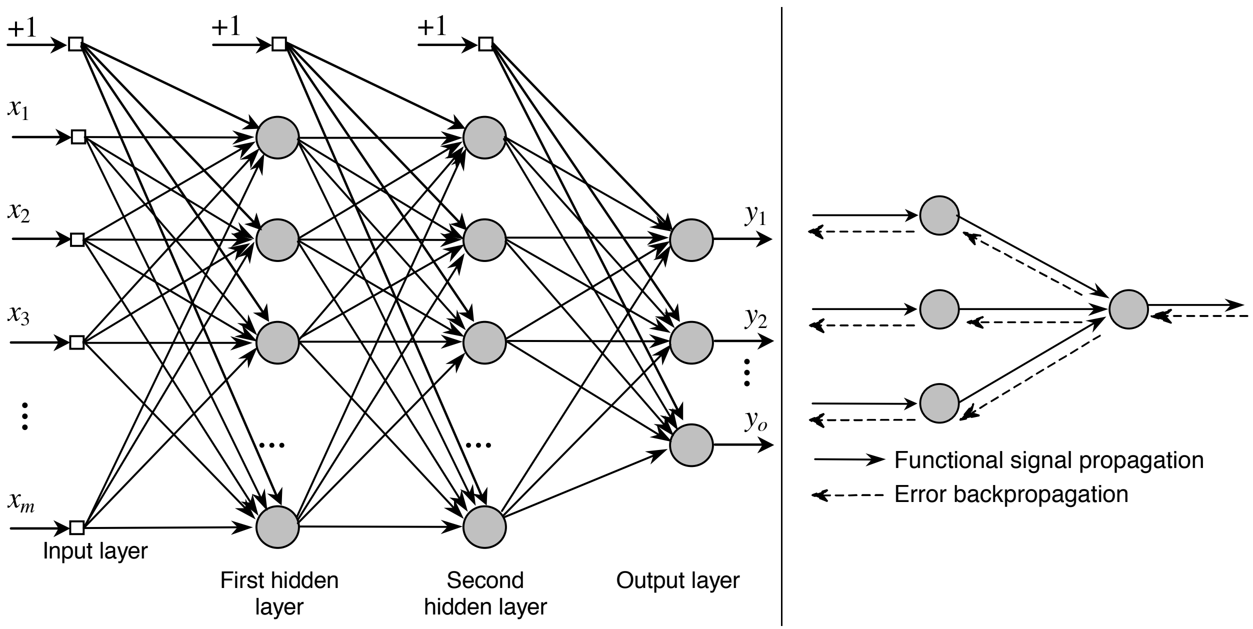
II. Unsupervised Learning

III. Reinforcement Learning

Artificial Neural Networks

Training the Multi-Layer Perceptron via the Back-Propagation Algorithm:

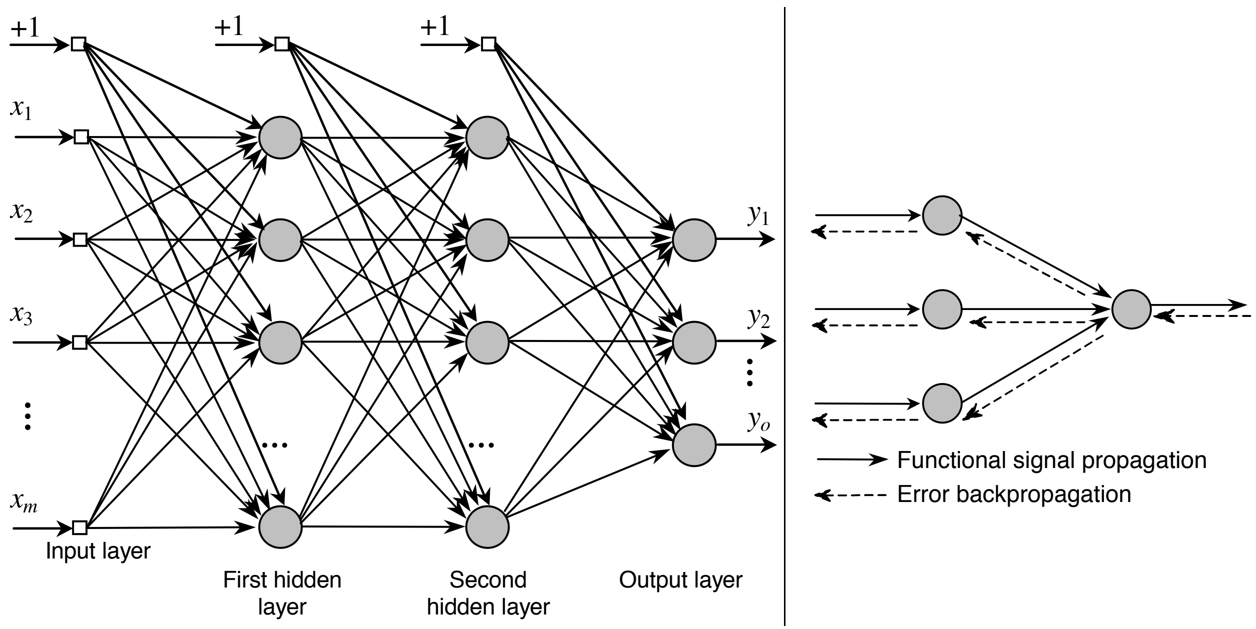
1. Feed inputs forward through network
2. Determine error at outputs
3. Feed error backwards towards inputs
4. Determine weight adjustments
5. Repeat for next input pattern
6. Repeat until all errors acceptably small



Artificial Neural Networks

The Back-Propagation Algorithm

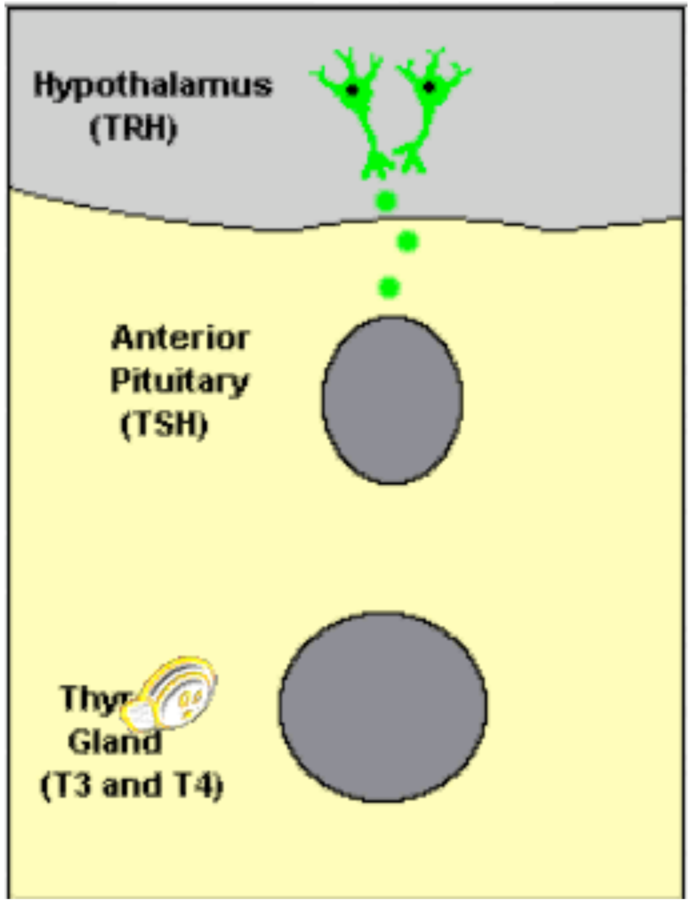
- Batch Learning
- On-Line Learning



Artificial Neural Networks

Recurrent Neural Networks

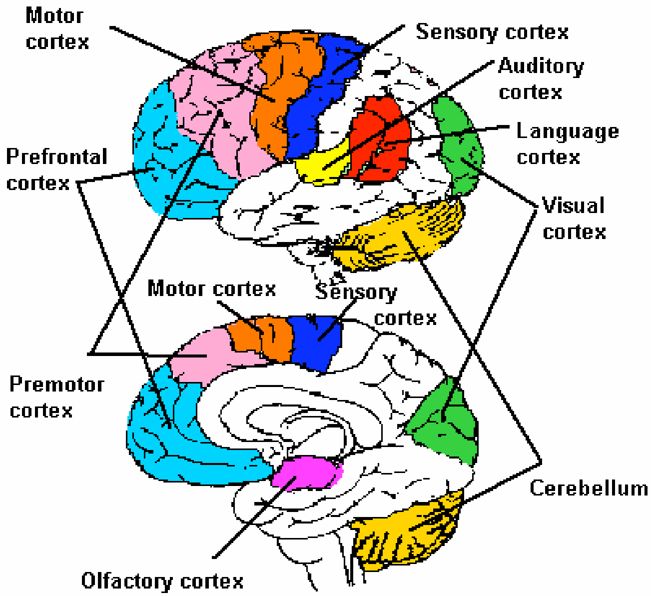
- Feedback Mechanisms

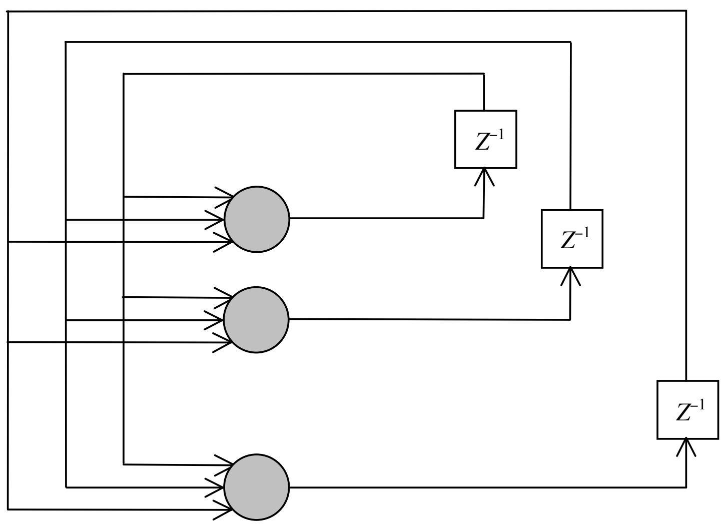


Artificial Neural Networks

Recurrent Neural Networks

- Feedback mechanisms allow:
  - internal representations (memories)
  - retrieval of noisy learned patterns
- Associative Memory
  - is memory by association
  - also called content-addressable memory



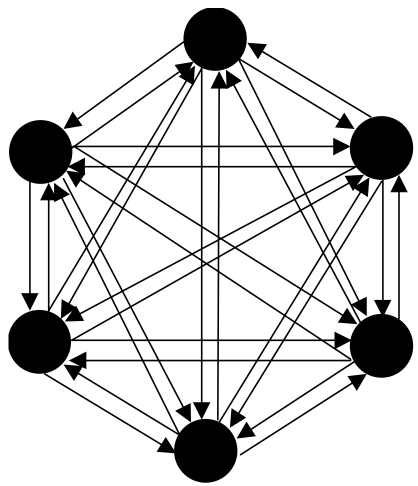


Artificial Neural Networks

Recurrent Neural Networks

I. Hopfield Neural Network

- Due to John Hopfield (1982)
- Did much to restore the credibility of ANNs following Minsky& Papert'sbook
- Hopfield's key contribution was to provide an analysis of the network he devised in terms of the energy of the system
- Hopfield Neural Networks are associative memory devices

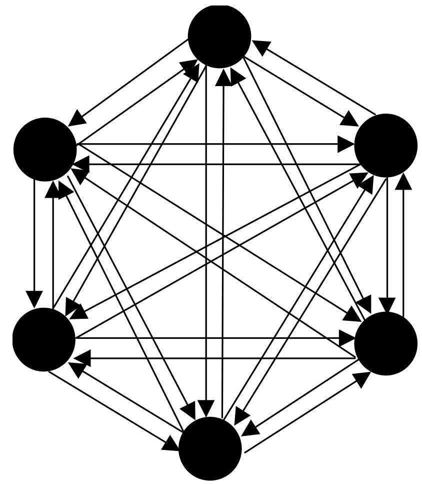


Artificial Neural Networks

Recurrent Neural Networks

I. Hopfield Neural Network

- Each node is connected to every other node in the network
- But there is no self-connection
- Symmetric weights on connections ($w_{i,j} = w_{j,i}$)
- Node activations are either *-1* or *+1*
- Execution involves iteratively re-calculating the activation of each node until a "stable-state" is achieved – energy minimization concept



Artificial Neural Networks

Recurrent Neural Networks

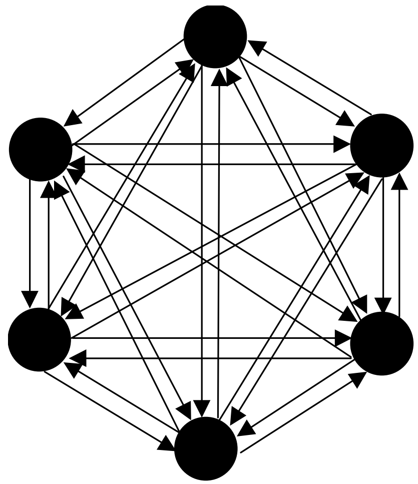
I. Hopfield Neural Network

- Training performed in one pass:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{n} p_i^k p_j^k$$

where:

- w_{ij} is the weight between nodes *i* & *j*
- *N* is the number of nodes in the network
- *n* is the number of patterns to be learnt
- p_i^k is the value required for the i-th node in pattern *k*

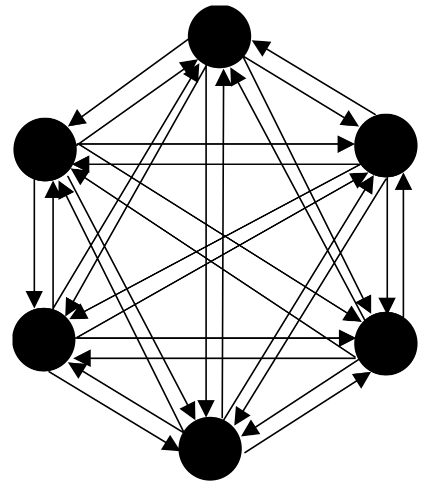


Artificial Neural Networks

Recurrent Neural Networks

I. Hopfield Neural Network

- Execution is performed iteratively:



$$s_i = sign\left(\sum_{j=1}^{N} w_{ij} s_j\right)$$

where:

- *s_i is the activation of the i-th node*

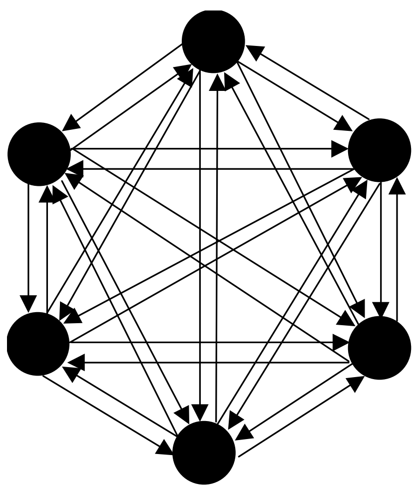
Artificial Neural Networks

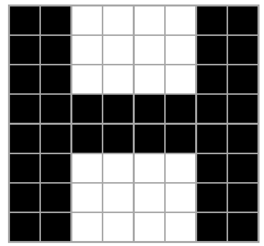
Recurrent Neural Networks

I. Hopfield Neural Network

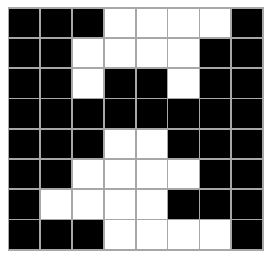
$$w_{ij} = \frac{1}{N} \sum_{k=1}^{n} p_i^k p_j^k$$

$$s_i = sign\left(\sum_{j=1}^{N} w_{ij} s_j \right)$$





64 pixel image of an "H"



Same image with 10 pixels altered
(I.e. approximately 16% noise added)

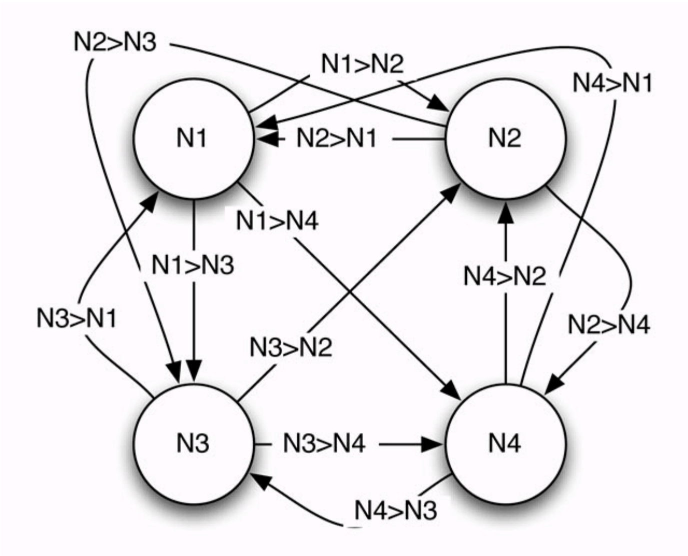
http://facstaff.cbu.edu/~pong/ai/hopfield/hopfieldapplet.html

Artificial Neural Networks

Recurrent Neural Networks

I. Hopfield Neural Network

- Example:
  - Recalling the pattern "0101"

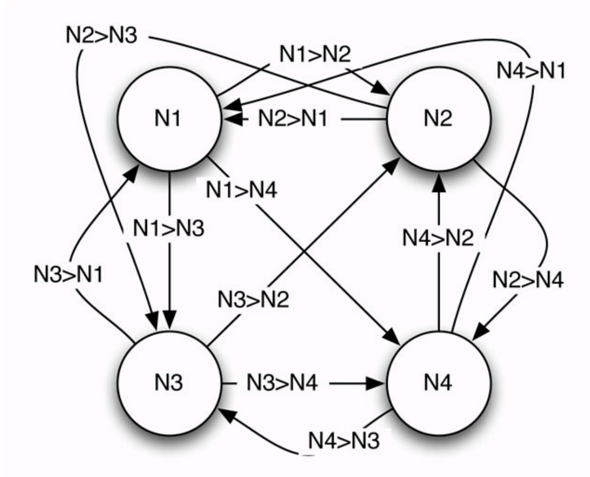


Artificial Neural Networks

I. Hopfield Neural Network

Recalling the pattern “0101”

TRAINING



- Step 1: create the weight matrix

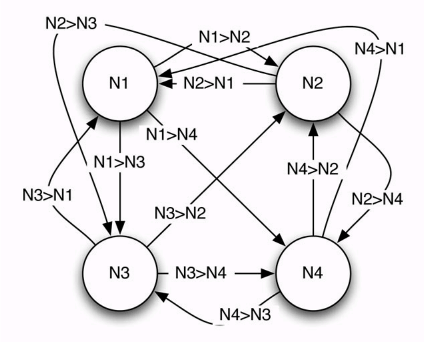
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Step 2: create the contribution matrix for the pattern “0101”

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Artificial Neural Networks

I. Hopfield Neural Network



Recalling the pattern “0101”

- Step 3: add the two matrices

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

- Step 4: set the diagonal to zero

$$\begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{bmatrix}$$

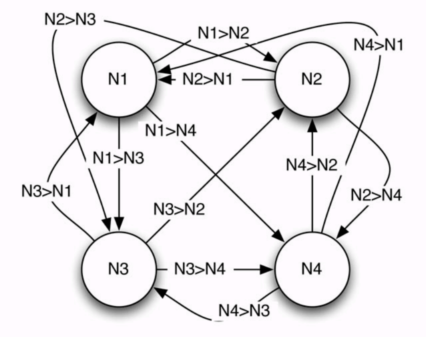
Artificial Neural Networks

I. Hopfield Neural Network

Recalling the pattern "0101"

EXECUTION

- Step 1: present the pattern "0101" to the each neuron (node) in the network



$$\begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{bmatrix}$$

- Step 2: Calculate the activation of each neuron

$$s_i = sign\left(\sum_{j=1}^{N} w_{ij} s_j \right)$$

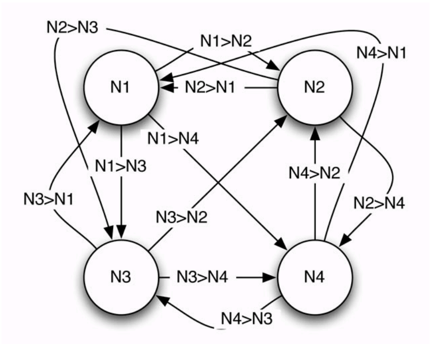
Artificial Neural Networks

I. Hopfield Neural Network

Recalling the pattern “0101”

EXECUTION

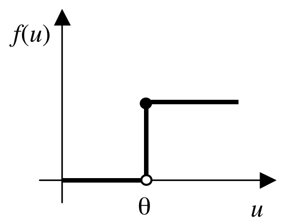
- Step 2: Calculate the activation of each neuron



$$\begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{bmatrix}$$

$$s_i = sign\left(\sum_{j=1}^{N} w_{ij} s_j \right)$$

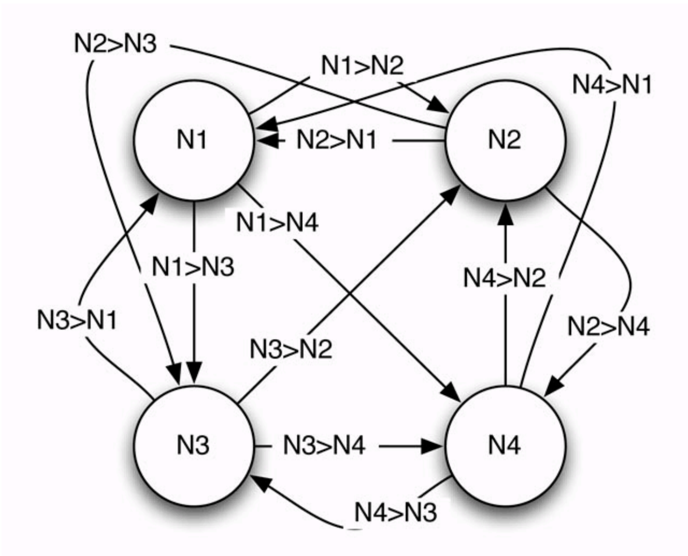
N1 = -1 + -1 = -2	N1 activation result is –2; will not fire	(0)
N2 = 0 + 1 = 1	N2 activation result is 1; will fire	(1)
N3 = -1 + -1 = -2	N3 activation result is –2; will not fire	(0)
N4 = 1 + 0 = 1	N4 activation result is 1; will fire	(1)



- Our Hopfield Network recalled the pattern “0101”

Artificial Neural Networks

What about the pattern “1010” ?



NEXT Lecture…

Recurrent Neural Networks

II. GasNets Models: Biological Inspiration

synaptic signalling

+

non-synaptic chemical signalling

nitric oxide (NO)
carbon monoxide (CO)

Garthwaite et al. (1988)
in Nature, 336, pp. 385-388

Lecture 5

I. Lecture 4 – Revision
II. Artificial Neural Networks (Part III)
    I. Recurrent Neural Networks
        I. Hopfield Network

Lecture 6

What's next?

Artificial Neural Networks

(Part IV)