NEURAL NETWORKS

LECTURE 1

dr Zoran Ševarac

sevarac@gmail.com

FON, 2014.

AIMS

- Brief introduction to neural networks concepts and architectures
- Capabilities and possible applications
- Java Neural Network Framework-om Neuroph
- Recognize typical neural network problems and implement neural network based solutions
- Neuroph framework development

BRIEF OVERVIEW

Lectures

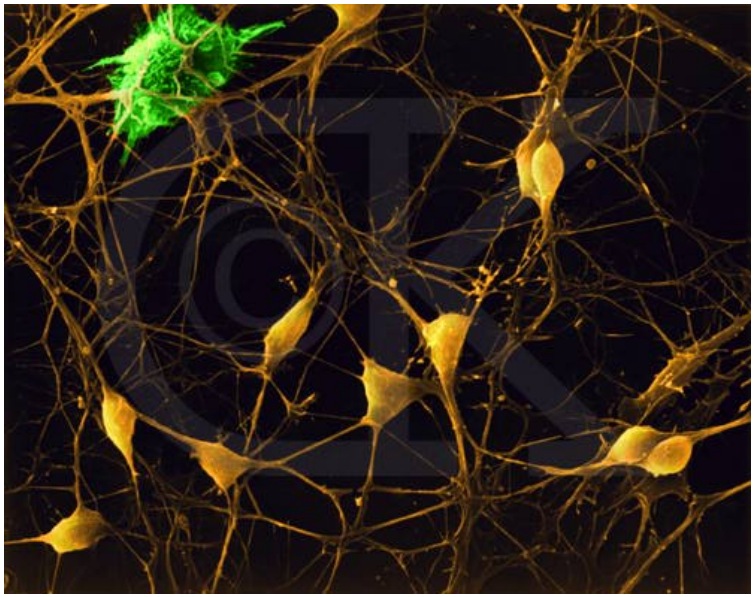
1. Basic concepts, architecture and components
2. Multi Layer Perceptron and Backpropagation
3. Problem solving using neural networks
4. Backpropagation improvements

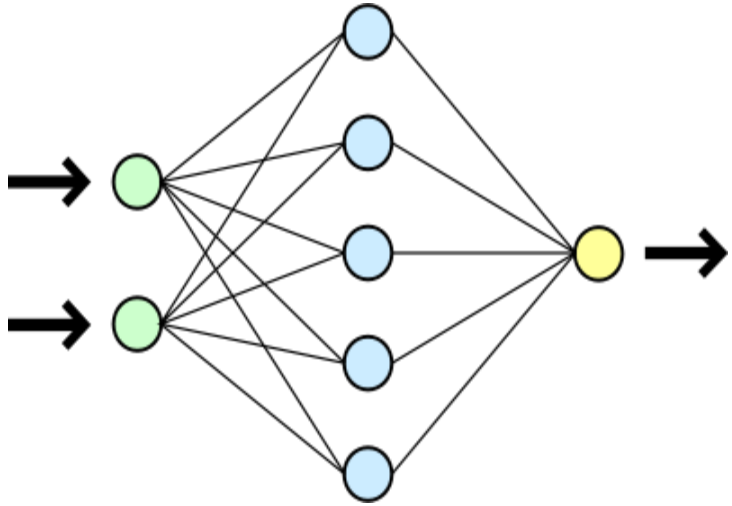
Labs

1. and 2. Neural network framewrok Neuroph and basic usage for image recognition and OCR
3. and 4. Neuroph framework architecture and extensions

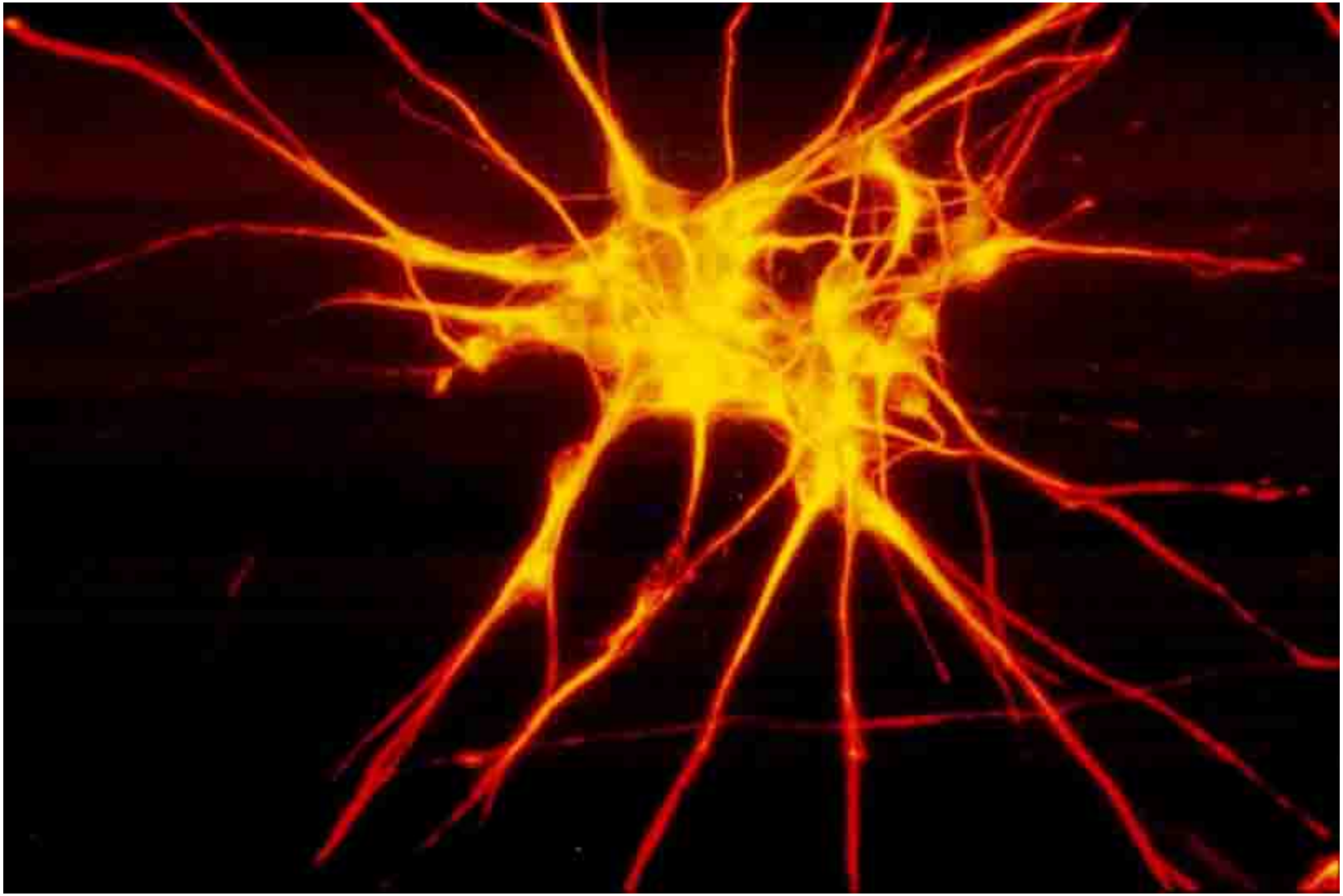
WHAT IS NEURAL NETWORK

- Brain inspired mathematical models
- Biological and artificial NN

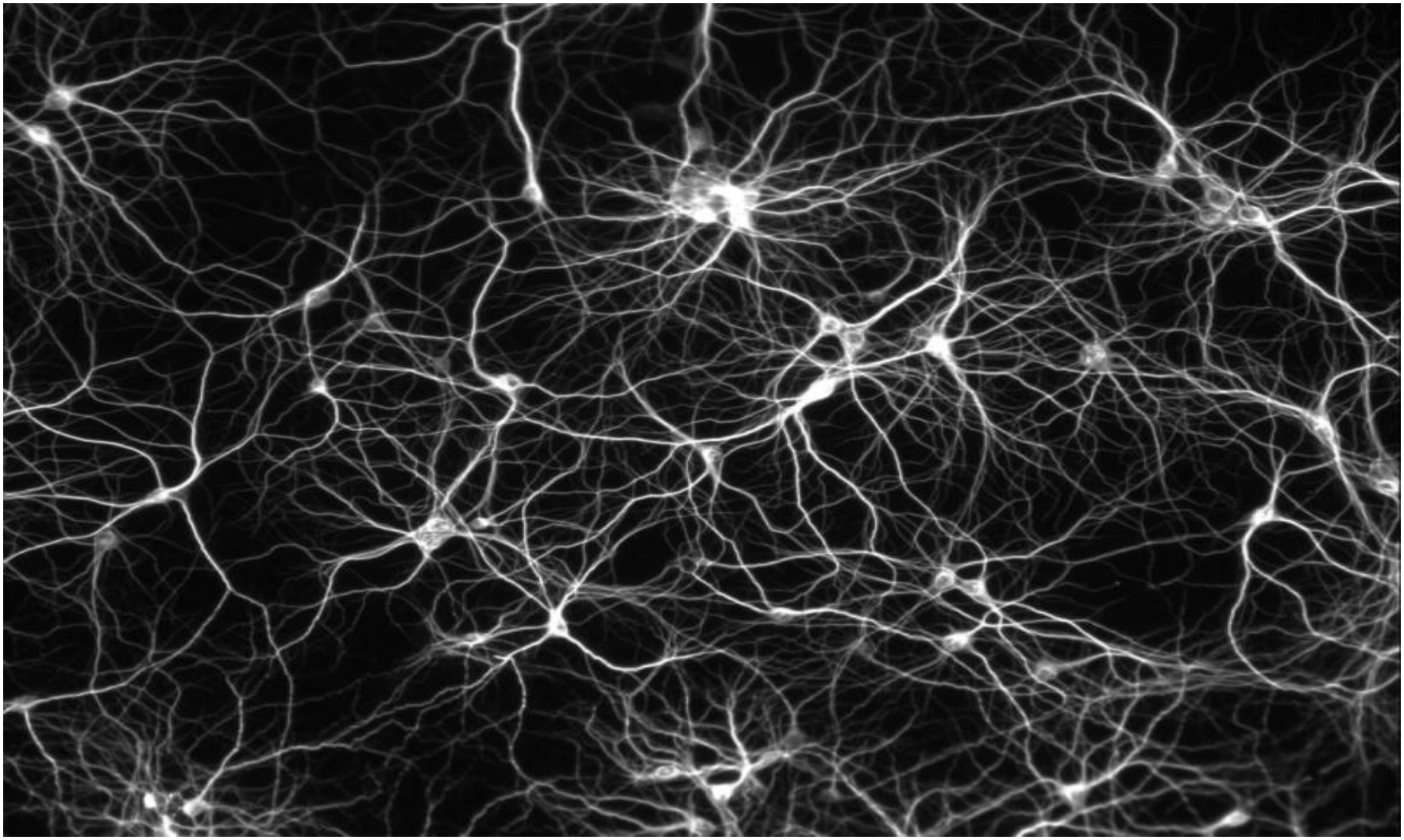




BIOLOGICAL NEURON



Biological Neural Network

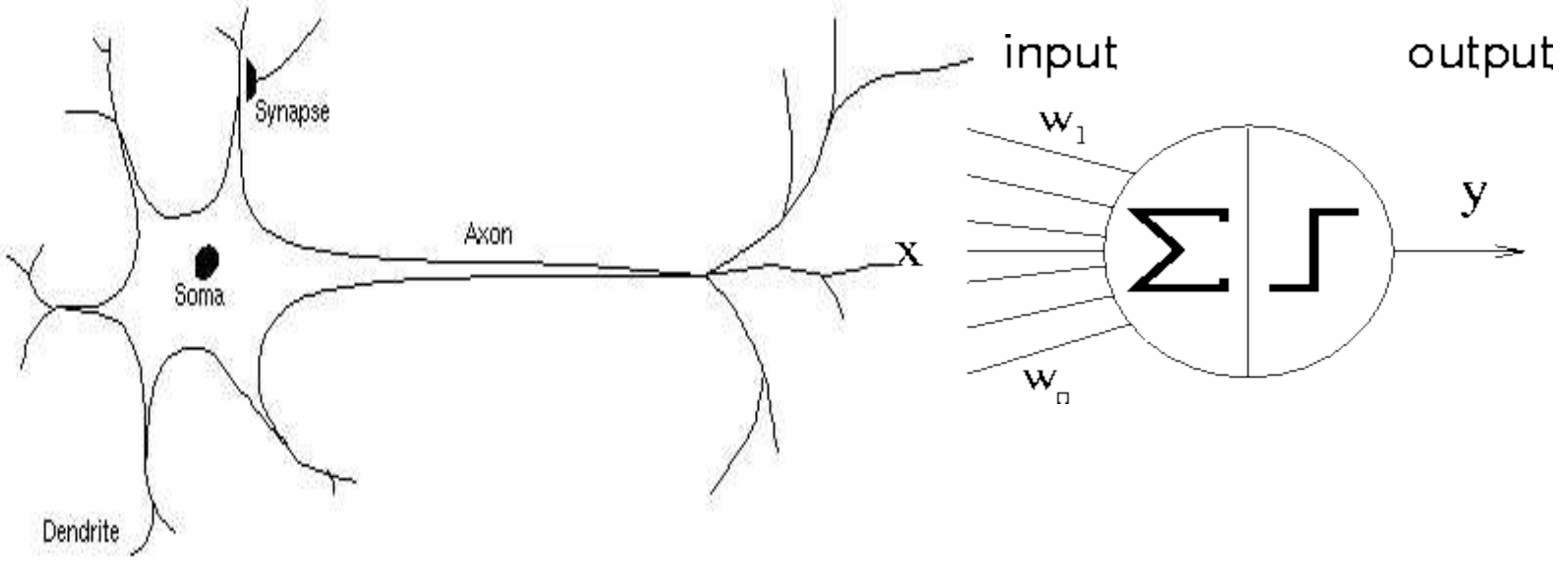


BRAIN AND ANN

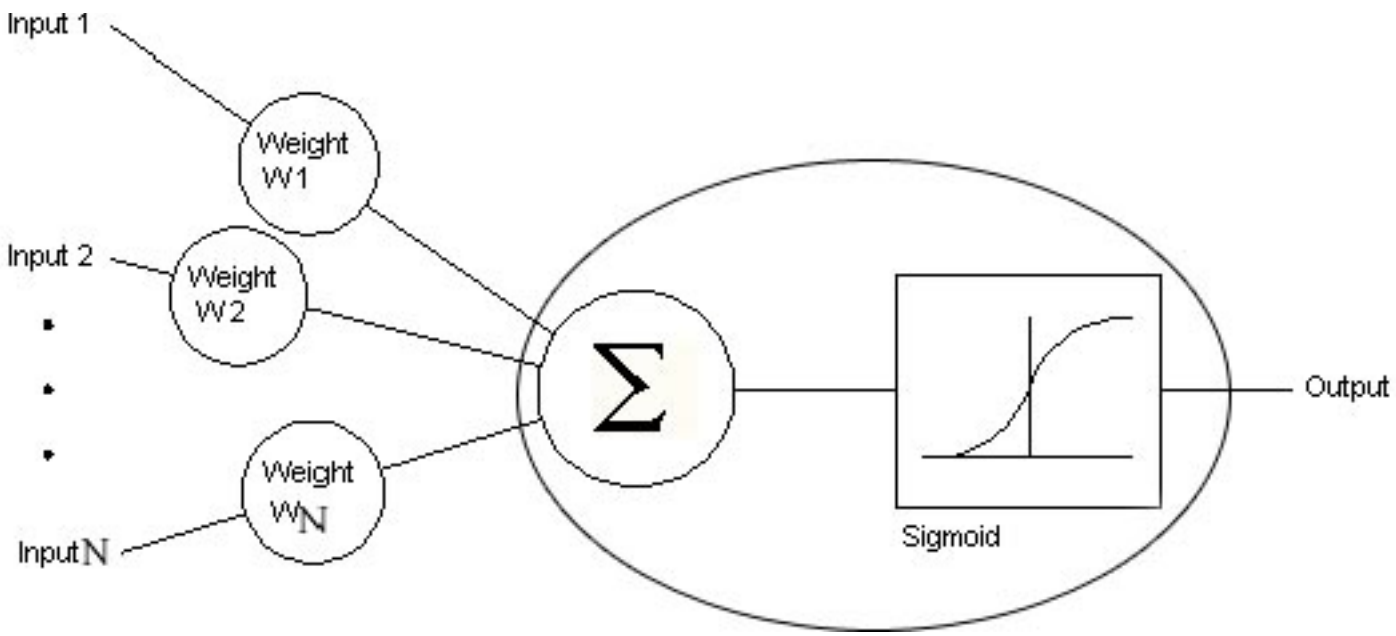
- Brain:
  - 10^{10} neurons
  - 10^{13} connections
  - speed at milisec
  - Fully parallel
- ANN:
  - 20 000 neurons
  - Speed at nanosec
  - Simulating parallel computation

Biological and artificial neuron

- Basic parts: body(soma), dendrites(inputs), axons (outputs), synapses (connections)



Artificial Neuron



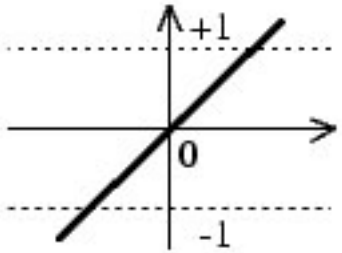
$$\text{output} = f(w_1 in_1 + \ldots + w_n in_n)$$

Basic parts of artificial neuron

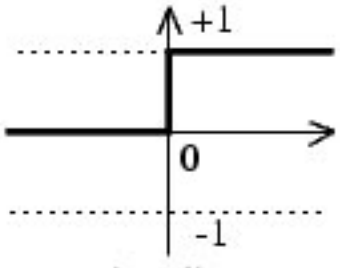
- Input summing function
- Transfer function
- Weighted inputs
- Output

TRANSFER FUNCTION

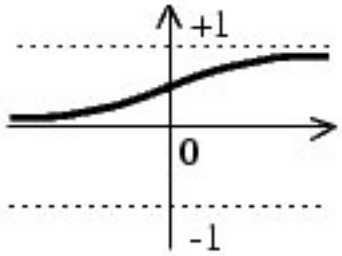
Linear



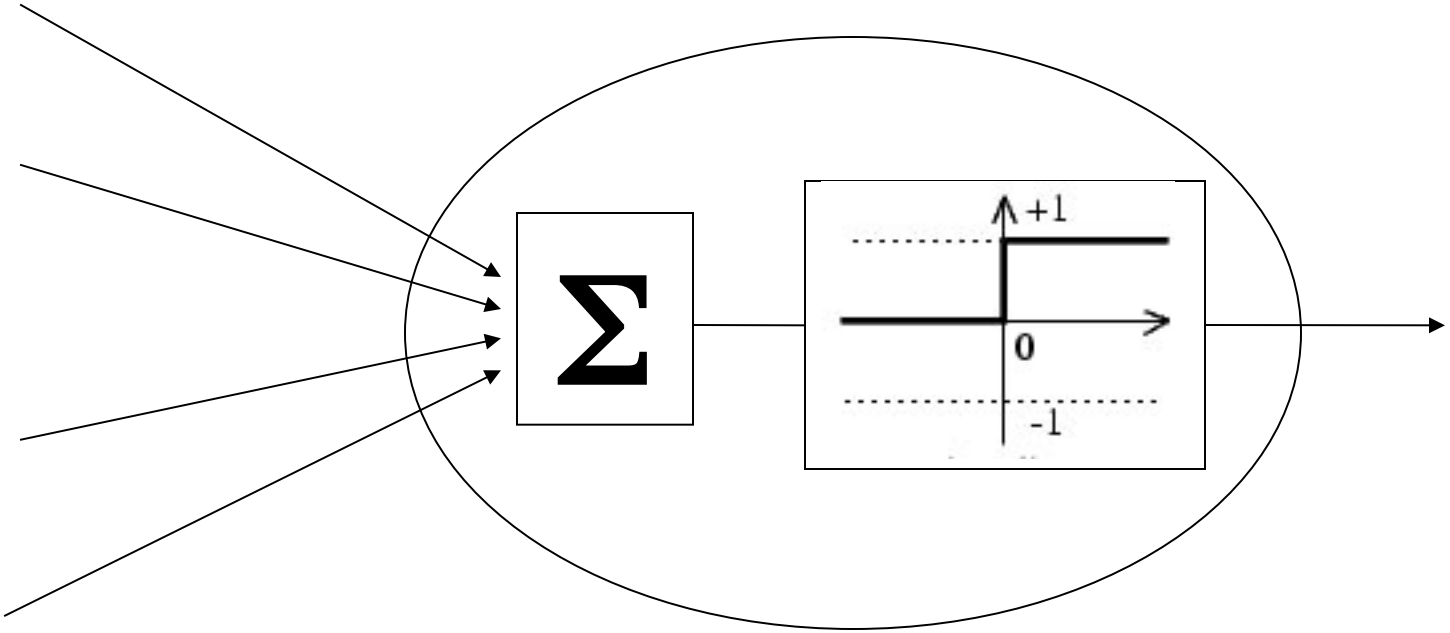
Step



Sigmoid

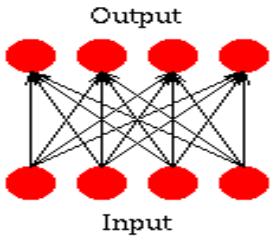


McCulloch Pits Neuron
Threshold Logic Unit

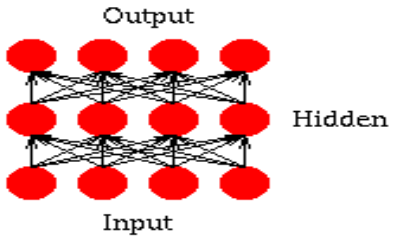


$y = \text{STEP}(w_1u_1 + \ldots + w_nu_n)$

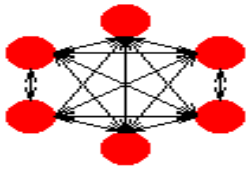
NEURAL NETWORK ARCHITECTURES



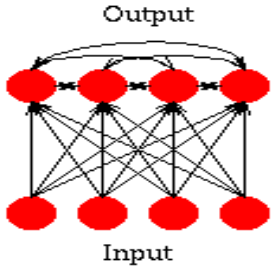
Single Layer Feedforward



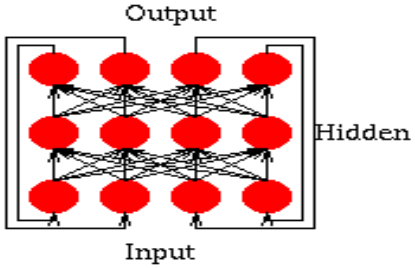
Multi Layer Feedforward



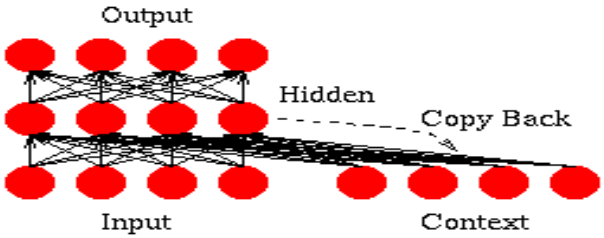
Fully Recurrent Network



Competitive Network



Jordan Network



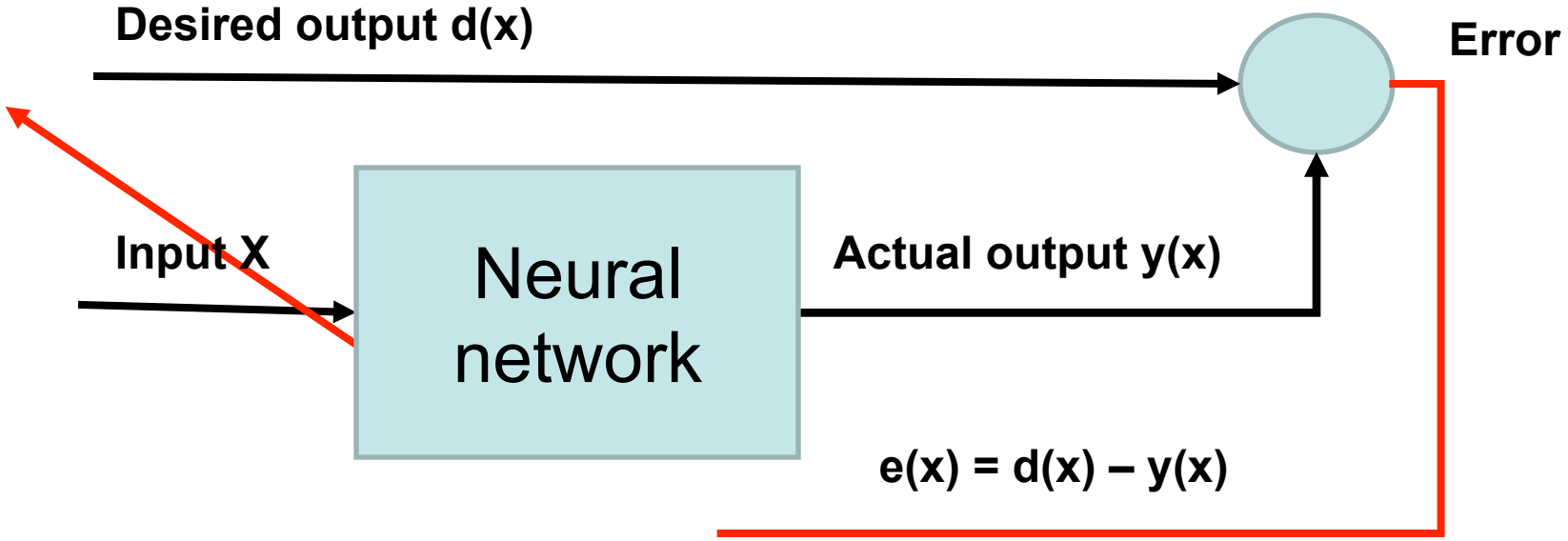
Simple Recurrent Network

Neural network training/ learning

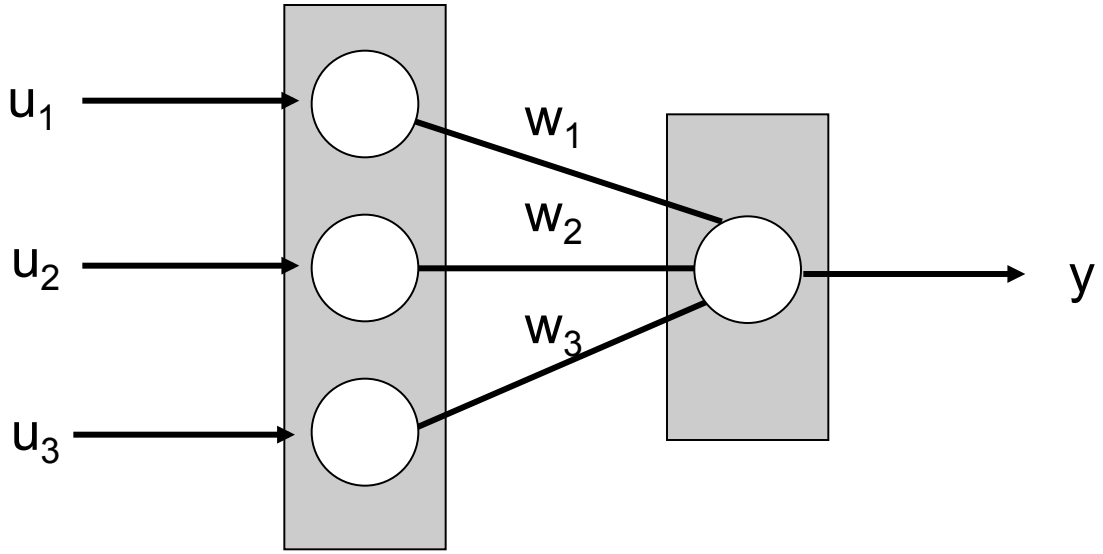
- **Learning procedure:** adjusting connection weights, until network gets desired behaviour
- *Supervised Learning*
- *Unsupervised Learning*

SUPERVISED LEARING

Basic principle: iterative error minimization



ADALINE



Linearna transfer function

Linear combination of inputs

$y = w_1u_1+w_2u_2+\ldots+w_nu_n,$

Learning algorithm: Least Mean Squares

LMS LEARNING

LMS rule equations:

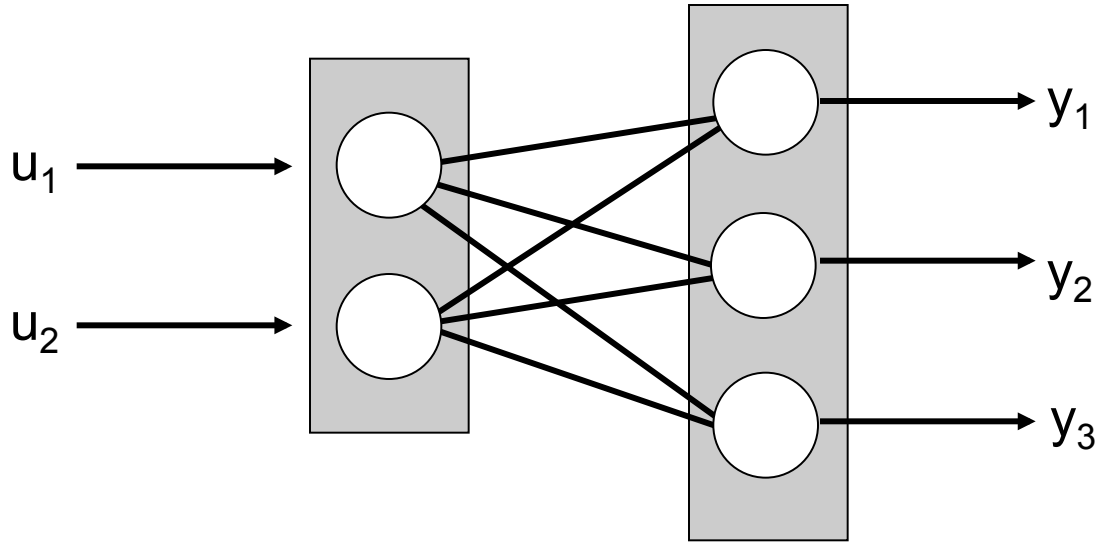
(1) output neuron error for pattern *p*

$$\varepsilon_p = d_p - y_p$$

(2) weight change according to error

(3) total network error for all patterns from training set (training stops when this error is below some predefined value)

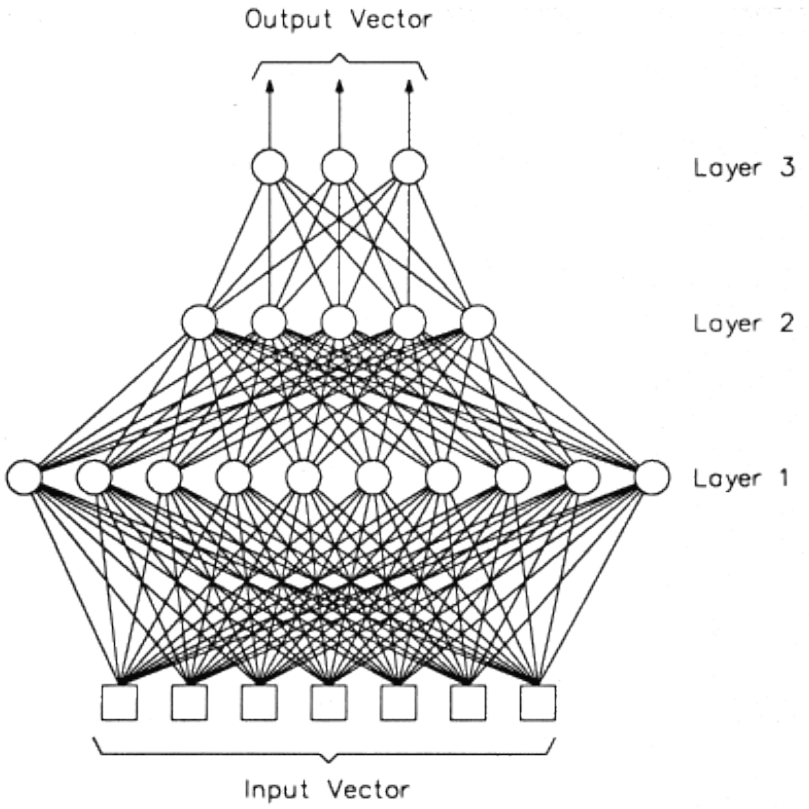
PERCEPTRON



- Step transfer function
- Perceptron learning - first algorithm for learning nonlinear systems
- Only for linear separable problems

Multi Layer Perceptron

- Basic perceptrn extended with one or more layers of hidden neurons between input and output layer
- Differentiable neuron transfer functions (tanh, sigmoid)
- Using Backpropagation algorithm for learning, which is bassed on LMS algorithm
- Able to solve complex problems



Backpropagation algorithm

- For Multi Layer Perceptron training – able to adjust weights in hidden layers
- Supervised learning algorithm based on LMS algorithm
- Multi Layer Perceptron with Backpropagation is universal aproximator

Backpropagation algorithm

- Formula for adjusting hidden neurons

$$w_{ji}(k+1) = w_{ji}(k) + \mu f'(net_j(k)) \left(\sum_a \varepsilon_a(k) f'(net_a(k)) w_{aj}(k) \right) u_{ji}$$

Backpropagation settings

- Max error
- Max iterations
- Learning rate
- Momentum
- Batch mode

Backpropagation algoritam

- Local minimum
- Stopping conditions:
  - Max iterations
  - Max error
  - Error stall
  - MSE for specific test set

LINKS AND BOOKS

- http://is.fon.rs/neuronske_mreze
- http://neuroph.sourceforge.net
- http://opensource.fon.bg.ac.rs

- Neural Networks - A Systematic Introduction , free online book
- Introduction to Neural Computation