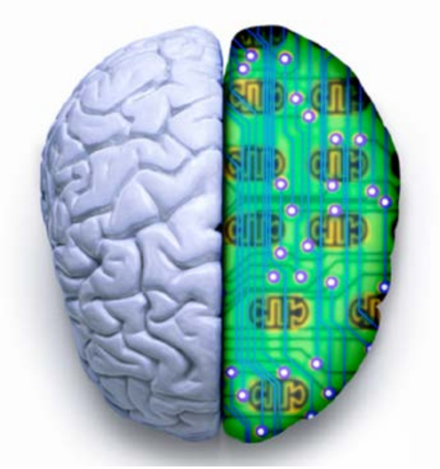
Advanced Artificial Intelligence CM4107 (Week 6)



Neural and Bio-Inspired Intelligence

Dr. Yann Savoye
School of Computing Science and Digital Media
Robert Gordon University

Module Information

- **Assessment:**

  - Coursework (2 components)
    - ~~Component 1: literature review~~
    - Component 2: paper implementation

  - No mid-term or final written exam.

    All deadlines are strong:
    - *It will not be possible to upload material after the deadline.*
    - *No deadline extension will be granted. No excuse.*
    - *Only the content submitted via the Moodle will be mark.*

Coursework

- **Submission of the Coursework Part 2:**

Deadline - Monday, December 11th, 2017 23:00:

Activity 3 and Activity 4

- *Prolog Programming (code in Prolog)*
- *Paper Implementation (code Java or C++)*

The coursework Part 2 has been released and you should start to work on it since today.

Overview

- Part I - Bio-Inspired Computing
- Part II – Artificial Neural Network

Part I – Bio-Inspired Computing

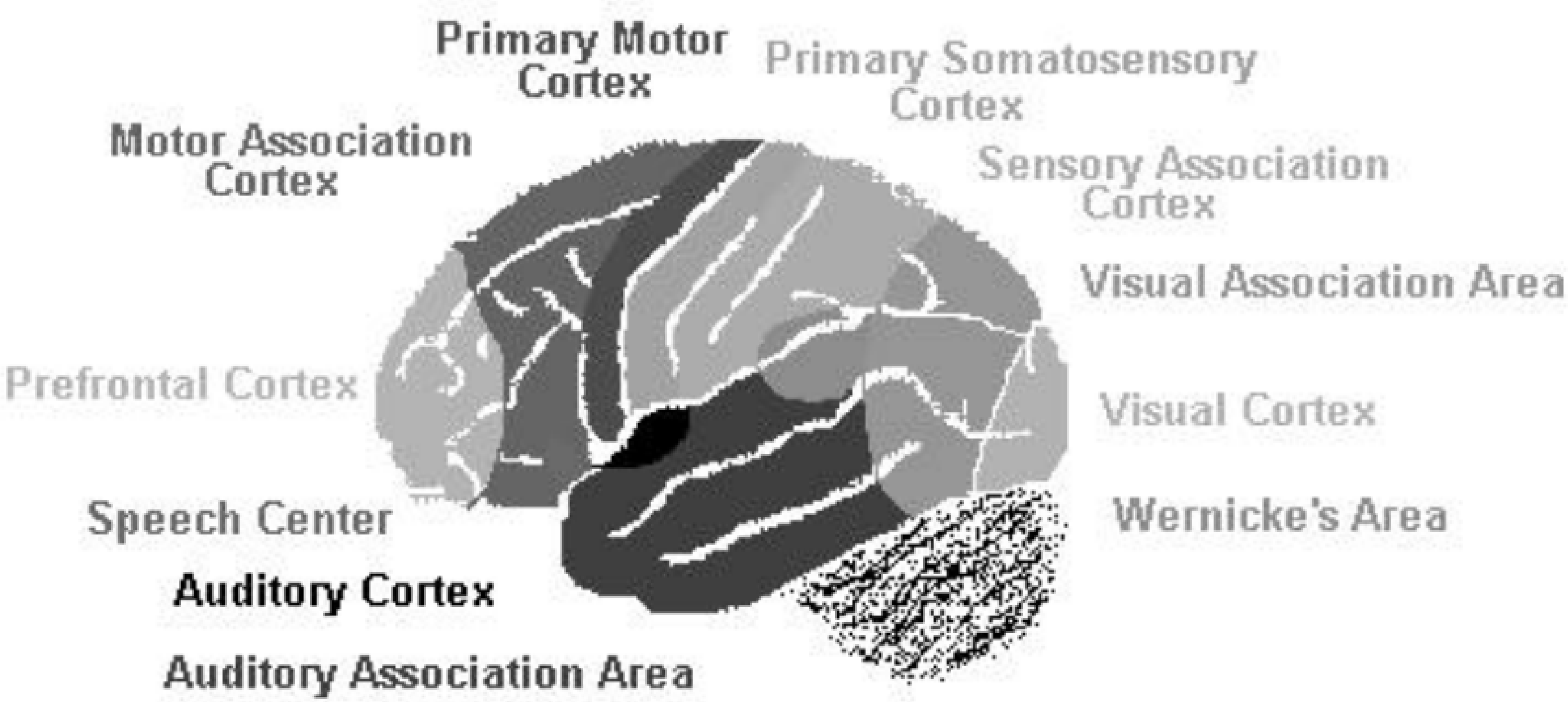
Imitation of Life

- Important to study the living organization
- Bio-inspired computing
- What can be abstracted in a different medium?

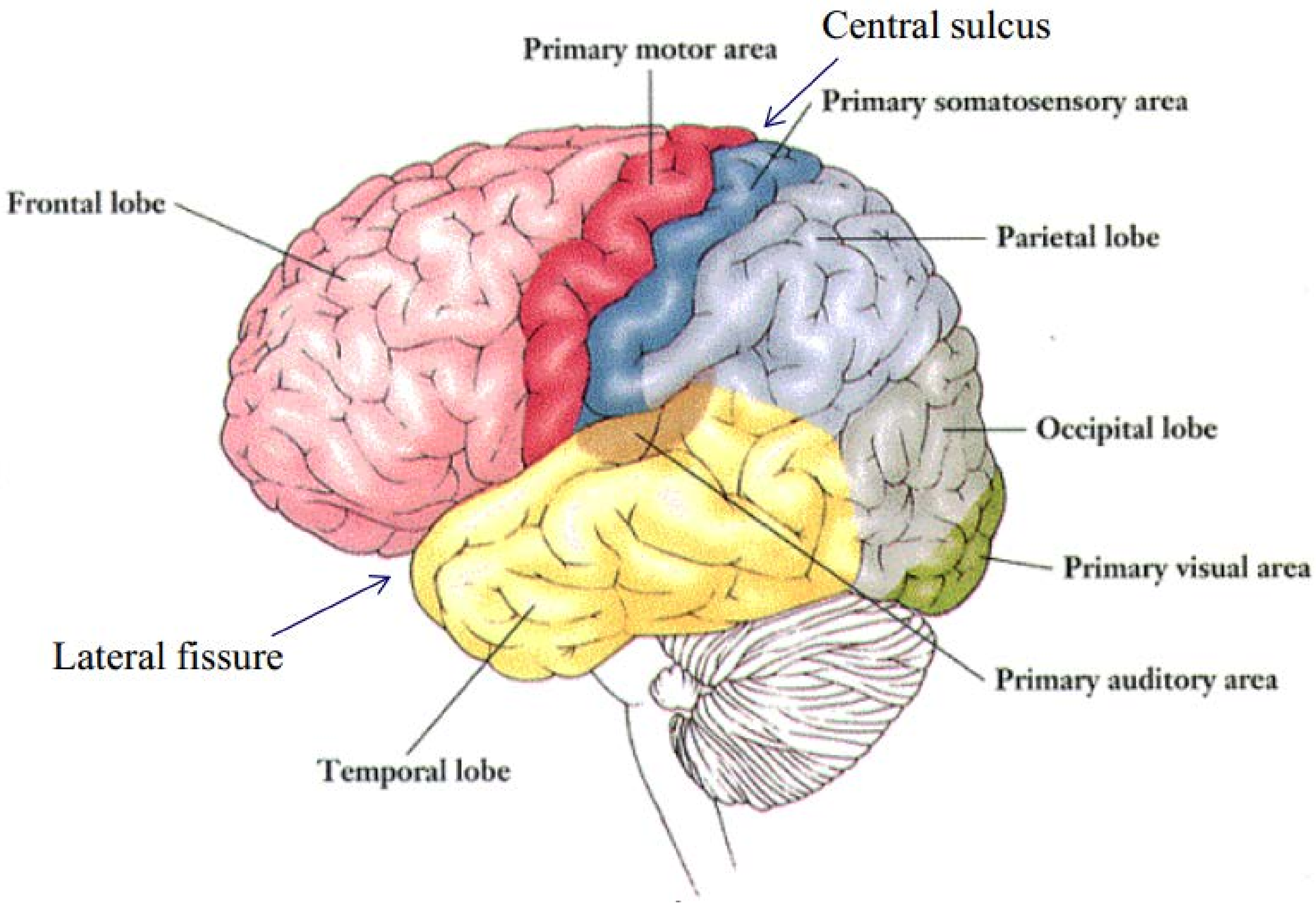
- Major Mechanisms in Nature
- ***Evolution:*** Biological systems change during generations.
- ***Development/growth***: By cell division into organisms.
- ***Learning***: Individuals undergo learning through their lifetime.
- ***Collective behaviour:*** Immune systems, flocks of birds, fishes.

Human Brain

Each **sensory input** is mapped into a corresponding area of the **cerebral cortex.** The cortex is a self-organizing **computational map** in the human brain.



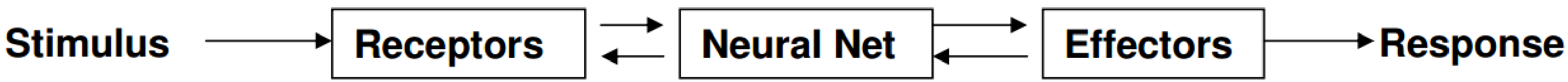
Human Brain



Central sulcus
Primary motor area
Primary somatosensory area
Frontal lobe
Parietal lobe
Occipital lobe
Primary visual area
Lateral fissure
Primary auditory area
Temporal lobe

The Nervous System

The human nervous system can be broken down into three stages that can be represented in block diagram form as:



- **The receptors** convert stimuli from the external environment into electrical impulses that convey information to the neural net (brain).

- The effectors convert electrical impulses generated by the neural net into responses as system outputs.

- The neural net (brain) continually receives information, perceives it and makes appropriate decisions.

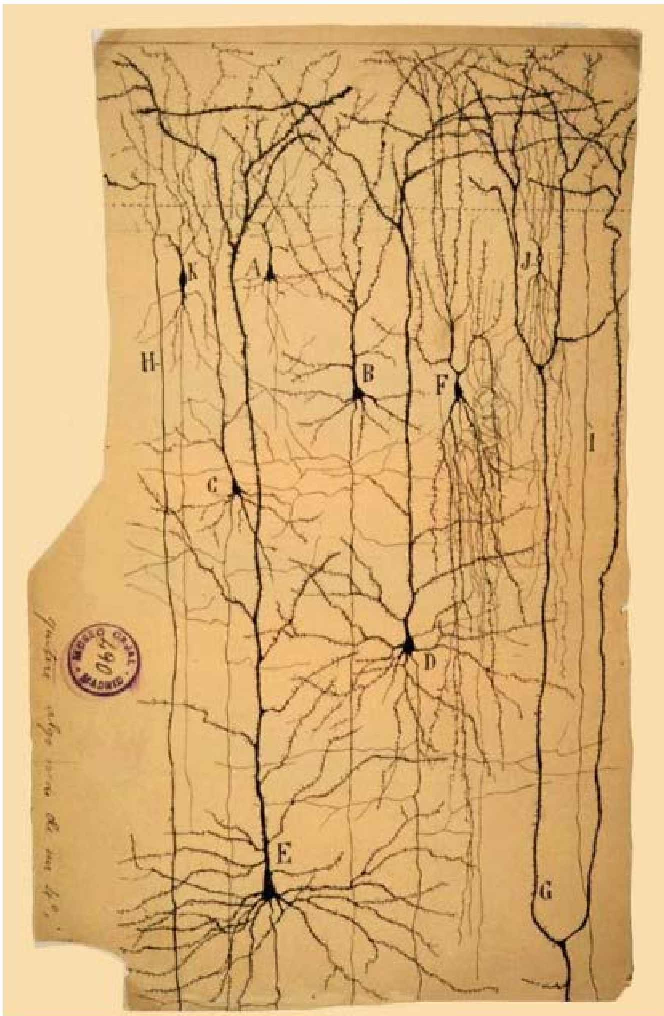
Nervous System

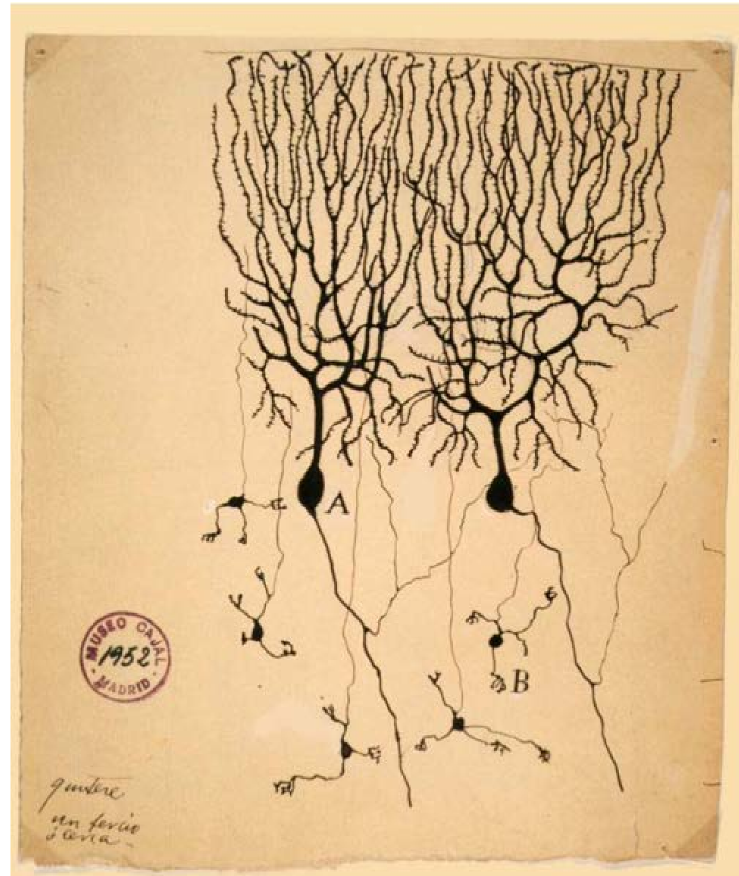
In the brain there are both **small-scale** and **large-scale** anatomical **organizations**, and different functions take place at lower and higher levels.

There is a hierarchy of interwoven **levels of organization:**

1. Behaviour
2. Systems
3. Microcircuits
4. **Neurons**
5. Dendrites
6. Synapses
7. Molecules
8. Genes

The Biological Neurons

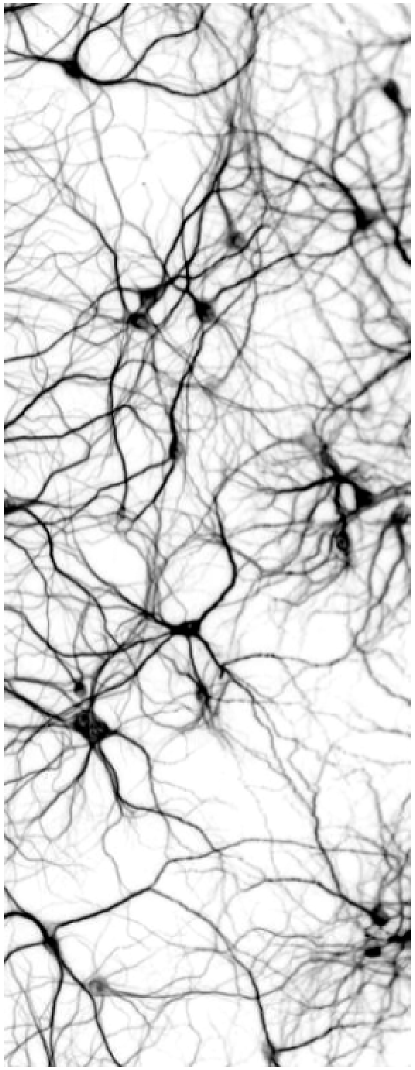




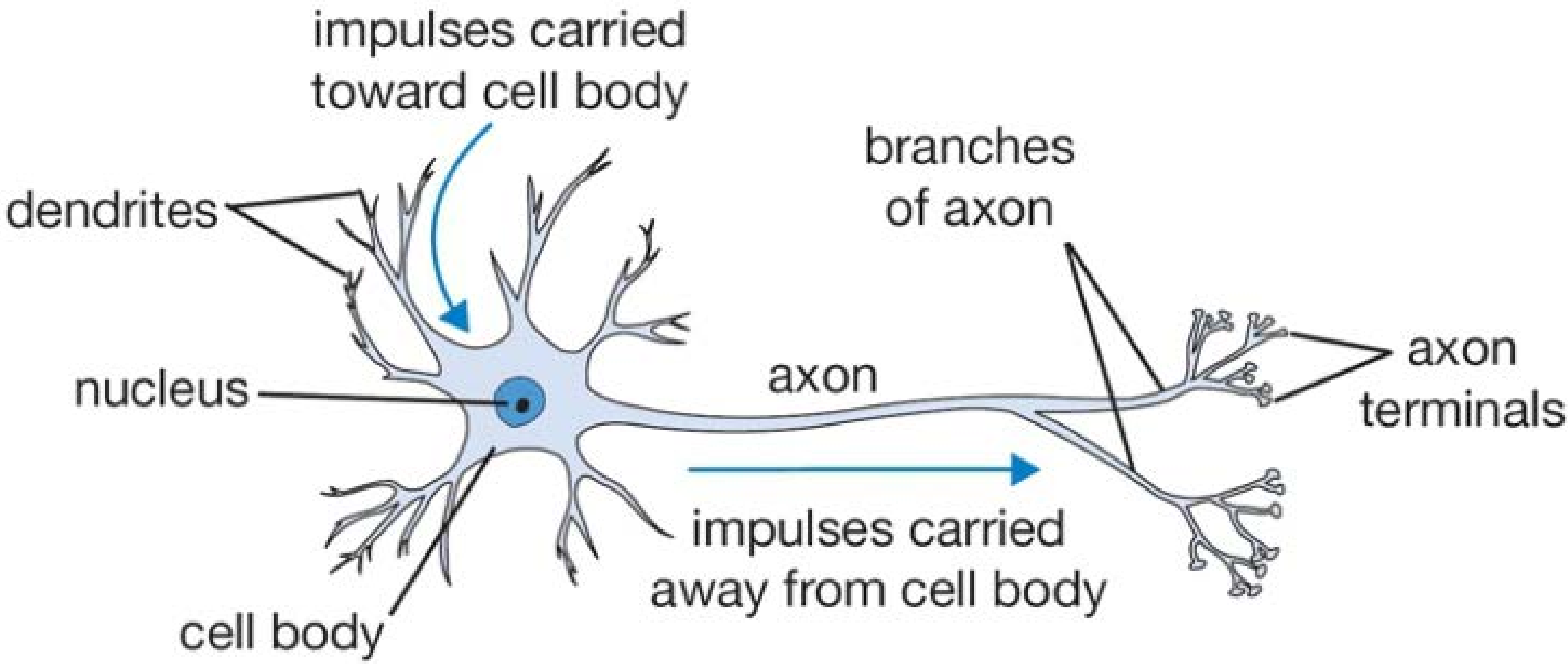
"Pigeon cerebellum. A, Purkinje cell, B, granule cell
Modified from a photograph taken from the original
(14X15.5 cm). Drawn on sheet/paper. P.Y. 1899. S.R.
y Cajal Institute - CSIC - Madrid, Spain.

(scientific illustrations of ramon y Cajal)

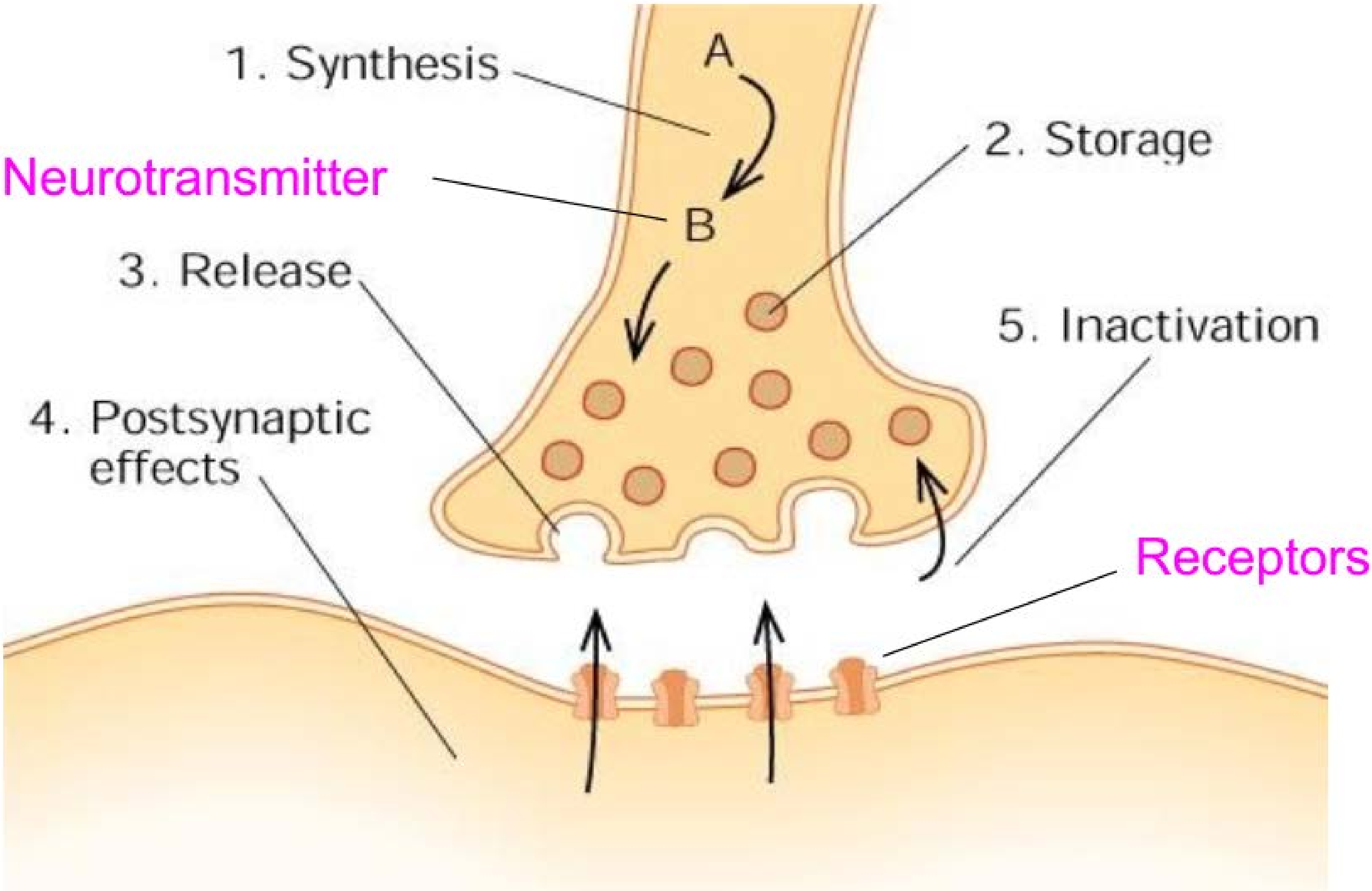
The Biological Neurons



The Biological Neurons



Synapse Activation



Part II – Artificial Neural Network

What are neural networks?

Artificial neural networks (ANNs) are not as complex as the brain, but imitate it in the following sense:

1. The building blocks of ANNs are simple **computational device** (capable of summing and thresholding incoming signals).
2. These devices are **highly interconnected**.
3. Information is processed locally at **each neuron.**
4. The **strength of a synapse** is modified by learning.
5. The **topological connections** between the neurons as well as the connection strengths determine the function of the ANN.
6. **Memory is distributed** in the **synapse**
7. **ANNs** are inherently **massively parallel.**

What are neural networks?

- **Neural Networks** are networks of neurons, for example, as found in real (i.e. biological) brains

- A Neural Network is an **interconnected assembly** of **simple processing elements**, units or nodes, whose functionality is loosely based on the animal neuron.

- The processing ability of the network is stored in the inter-unit connection strengths, or **weights**, obtained by **a process of adaptation to, or learning** from, a set of training patterns.

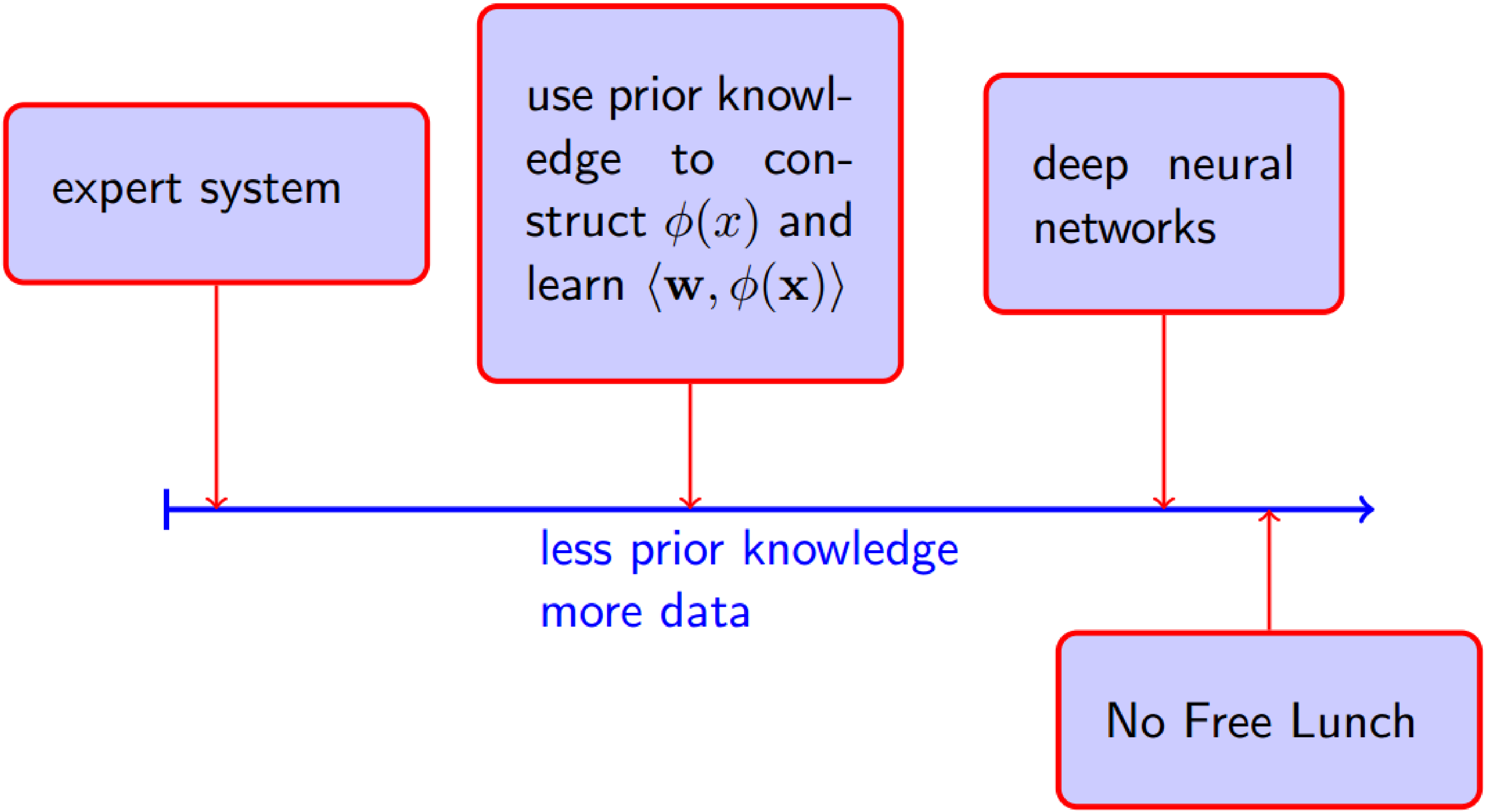
What are neural networks?

An **Artificial neuron** is characterized by:

1. **Architecture (**connection between neurons)
2. **Training or learning** (determining weights on the connections)
3**. Activation function**

- The **signals** are **transmitted** by the means of **connection links.**
- The **link** possesses an **associated weight**, which is multiplied along with the incoming signal (net input) for any typical neural net.
- The **output signal** is obtained by **applying activations** to the net input.

Less Prior Knowledge, More Data



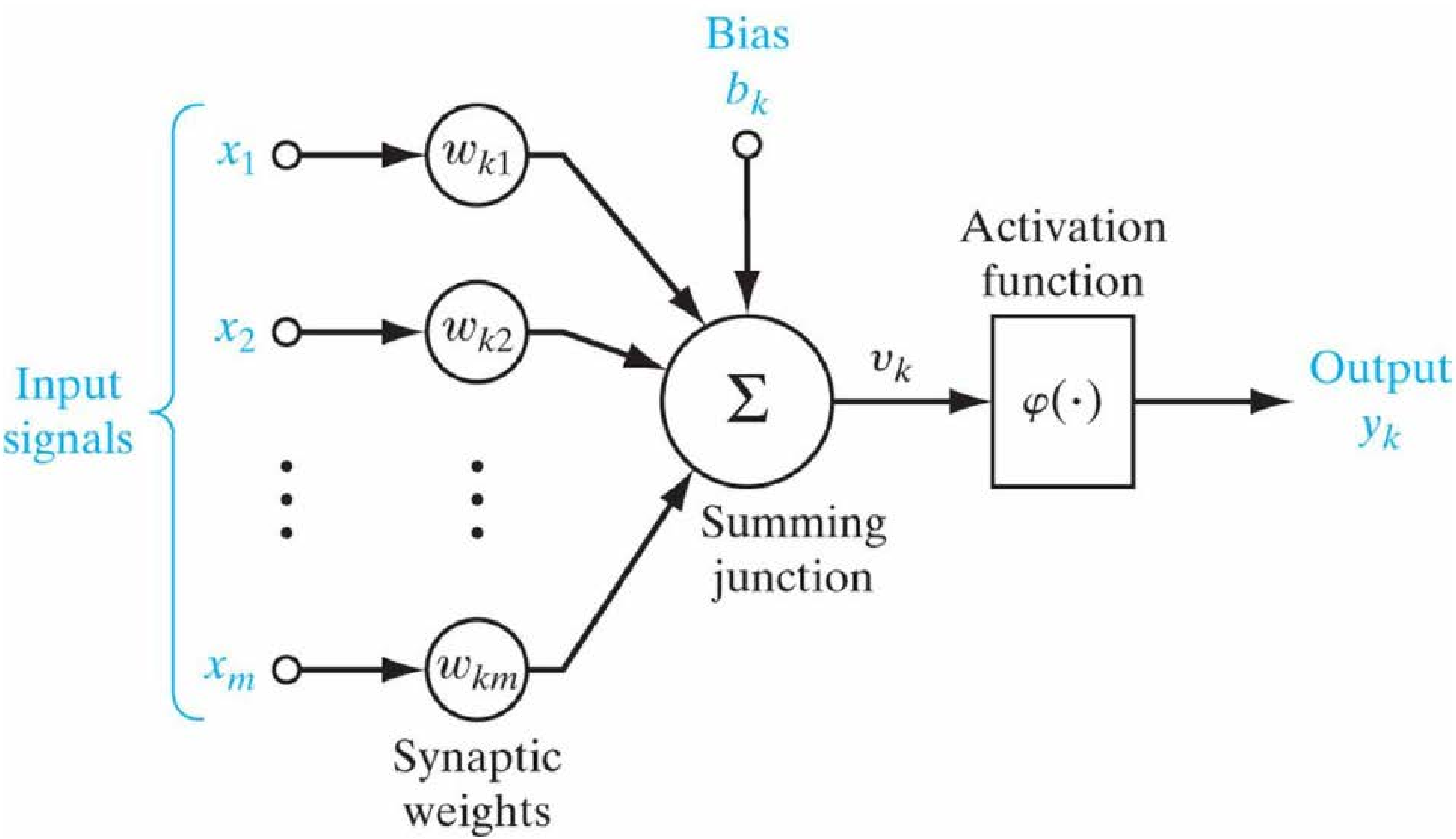
Neural Network

McCulloch and Pitts threw some ideas together and came up with the idea of an artificial neural network (1943)

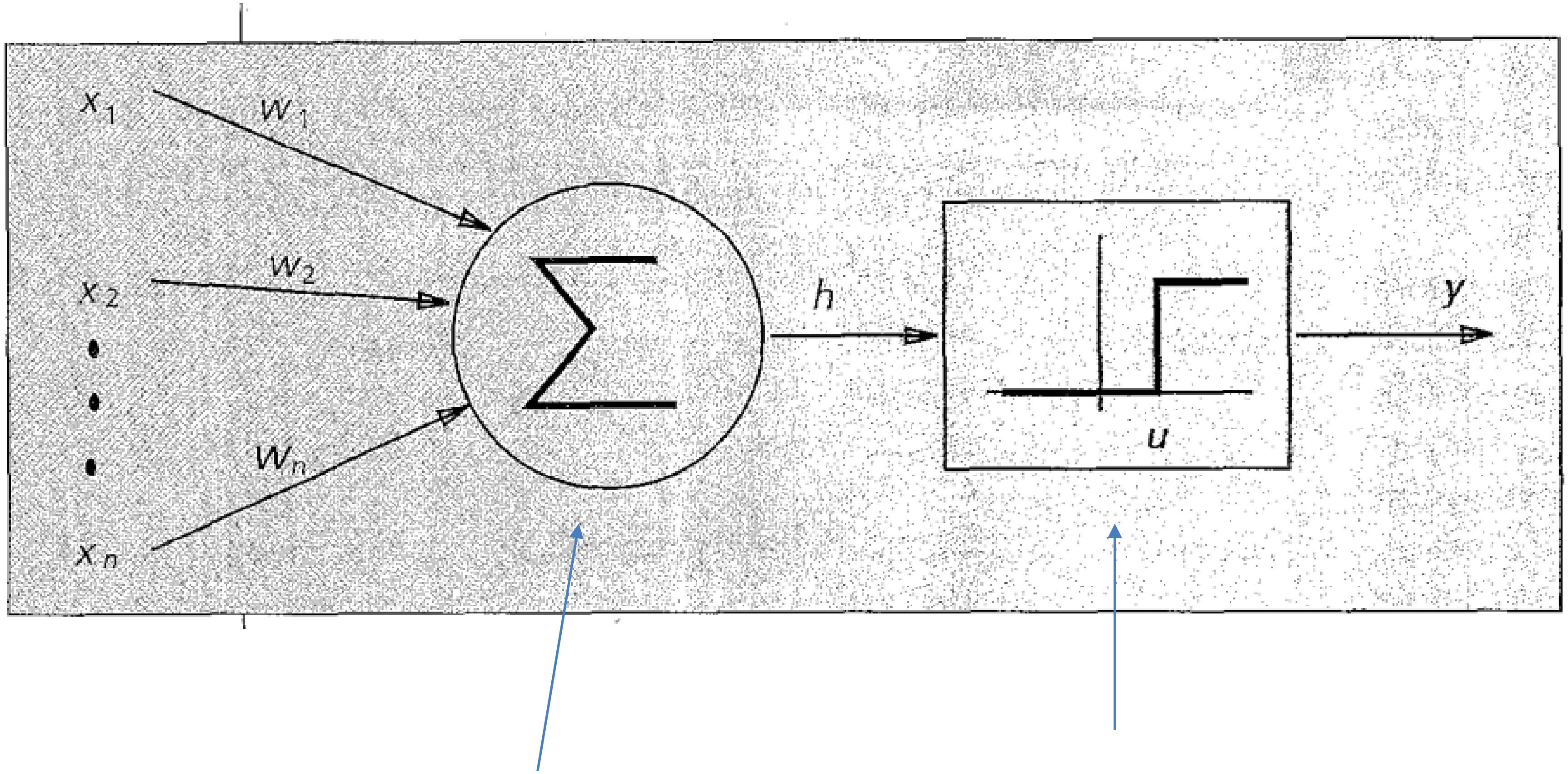
Hebb later demonstrated an updating rule for the Weights between neurons that allowed the neural network to learn (1949).

Minsky and Edmonds built the first neural network computer (1951)

The McCulloch-Pitts Neuron



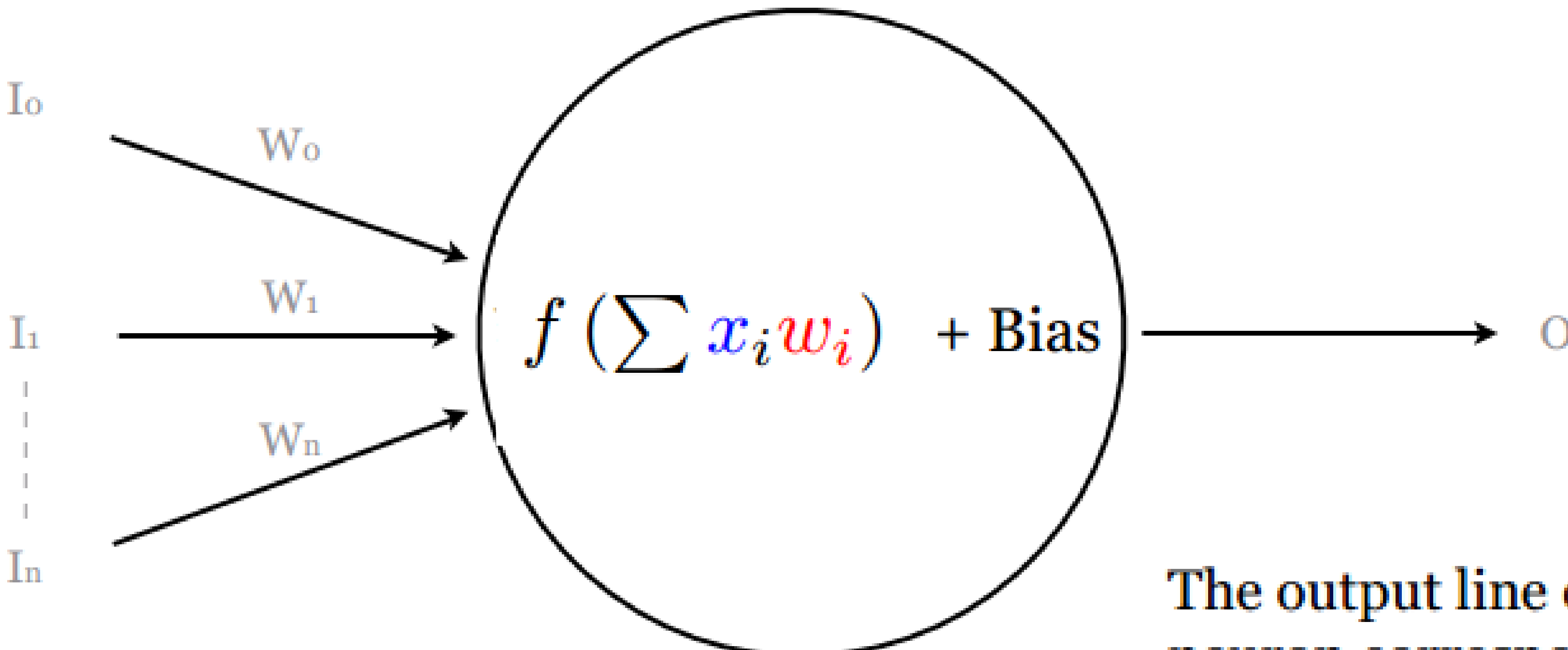
Neurons Model



Weighted summing **Activation function**

Weighted Summing

The bias determines the artificial neuron's inhibitory or excitatory effect.



The input lines I to the neuron correspond to biological dendrites.

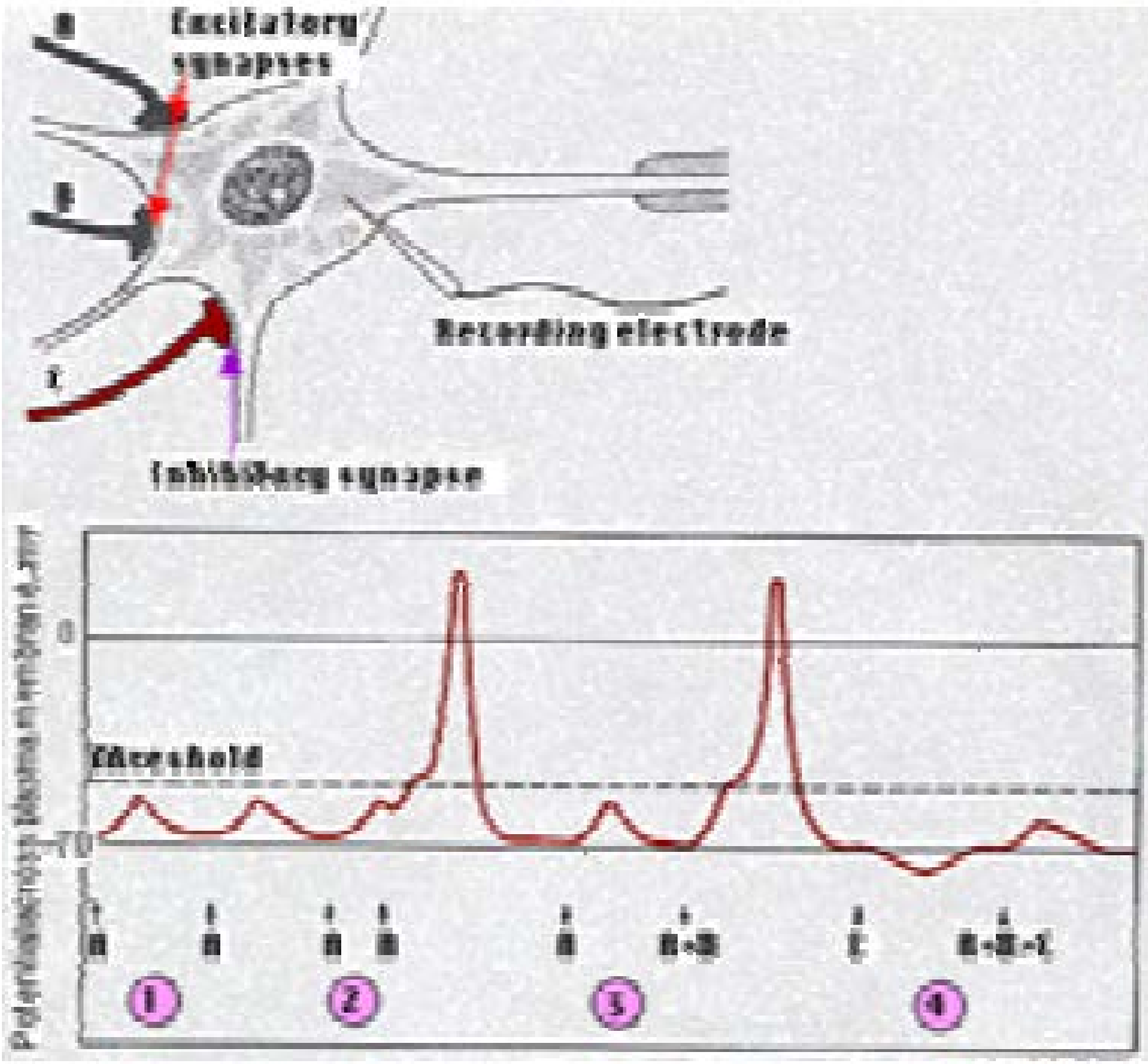
The output line of the neuron corresponds to a biological axon.

Weight Summing

- Denote the m input values by $x_1, x_2, \ldots, x_m$.
- Each of the m inputs (synapses) has a weight $w_1, w_2, \ldots, w_m$.
- The input values are multiplied by their weights and summed

$$v = w_1 x_1 + w_2 x_2 + \cdots + w_m x_m = \sum_{i=1}^{m} w_i x_i \quad + \text{Bias}$$

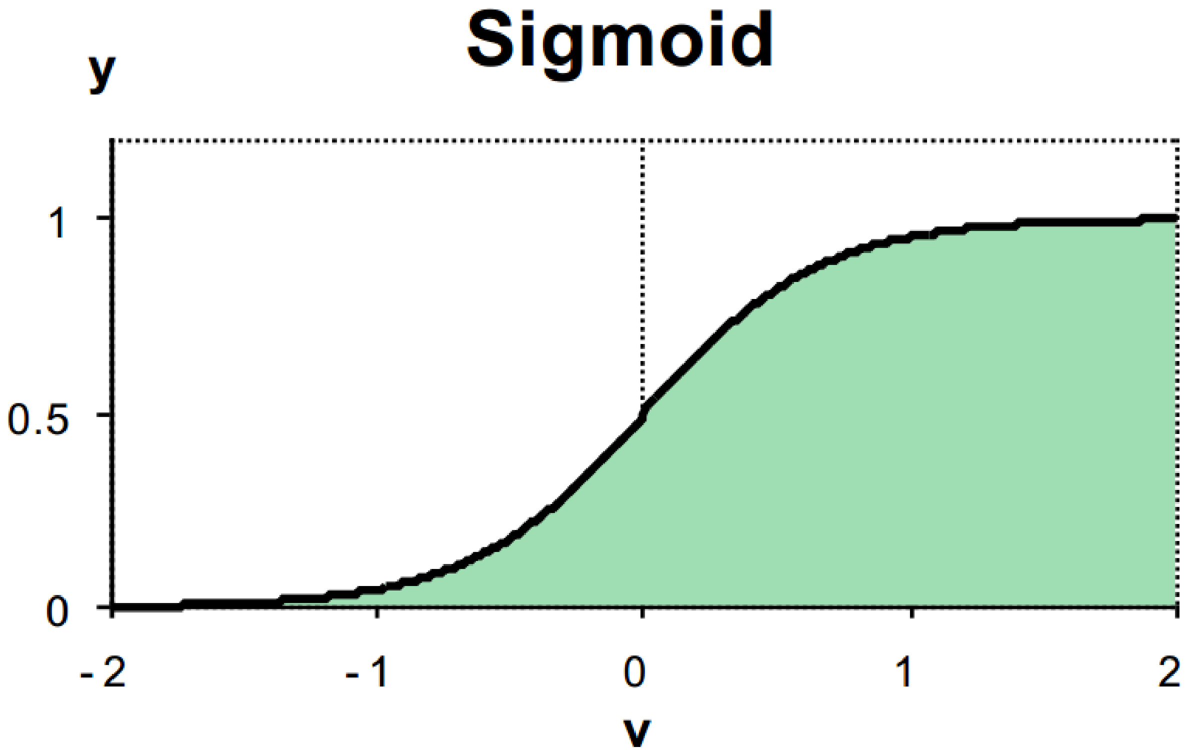
Activation


Excitatory synapses
Recording electrode
Inhibitory synapse
Threshold
0
-70
1
2
3
4

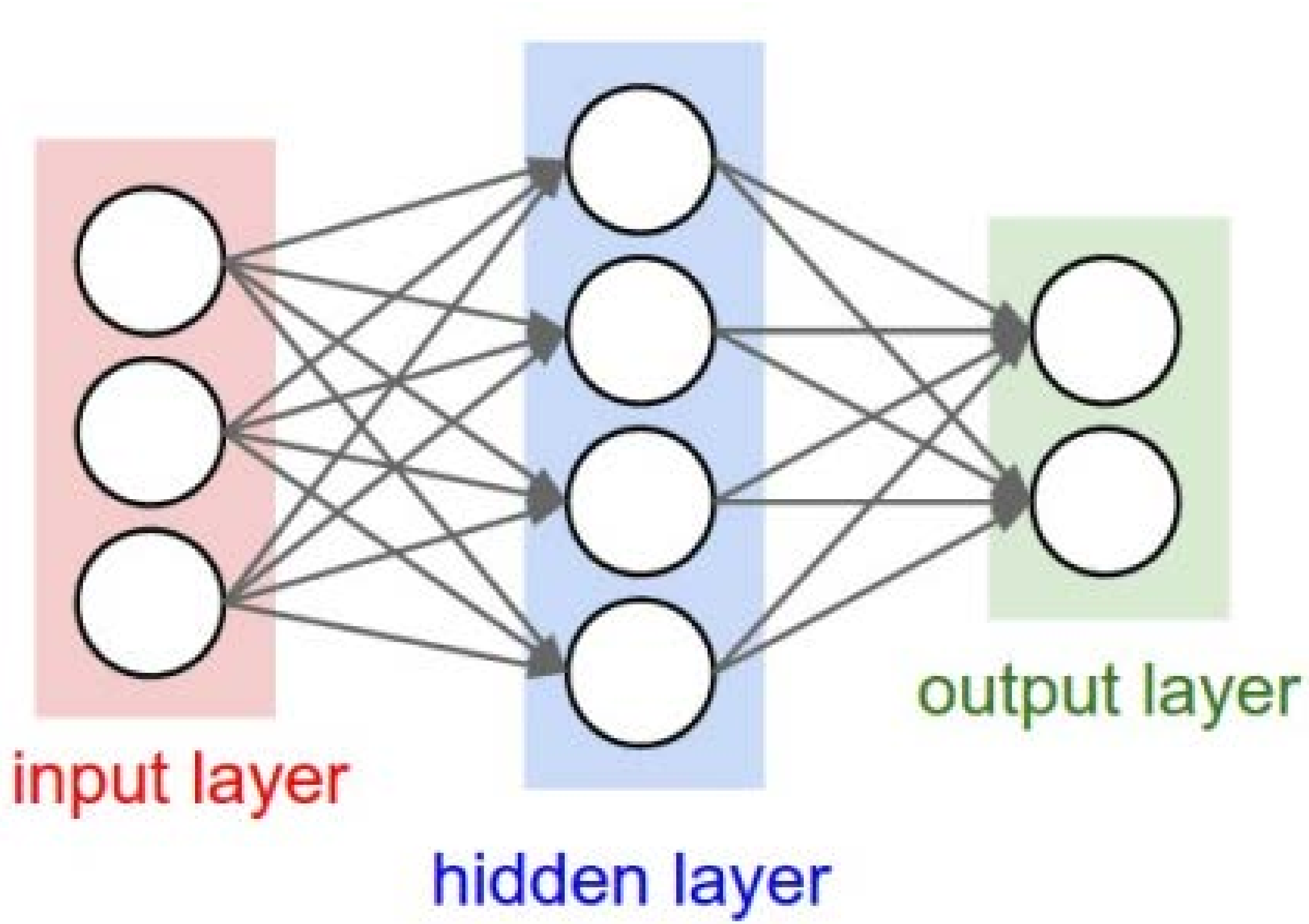
Activation Functions

An **activation function** is applied to the output of a neuron to obtain smooth changes in output.

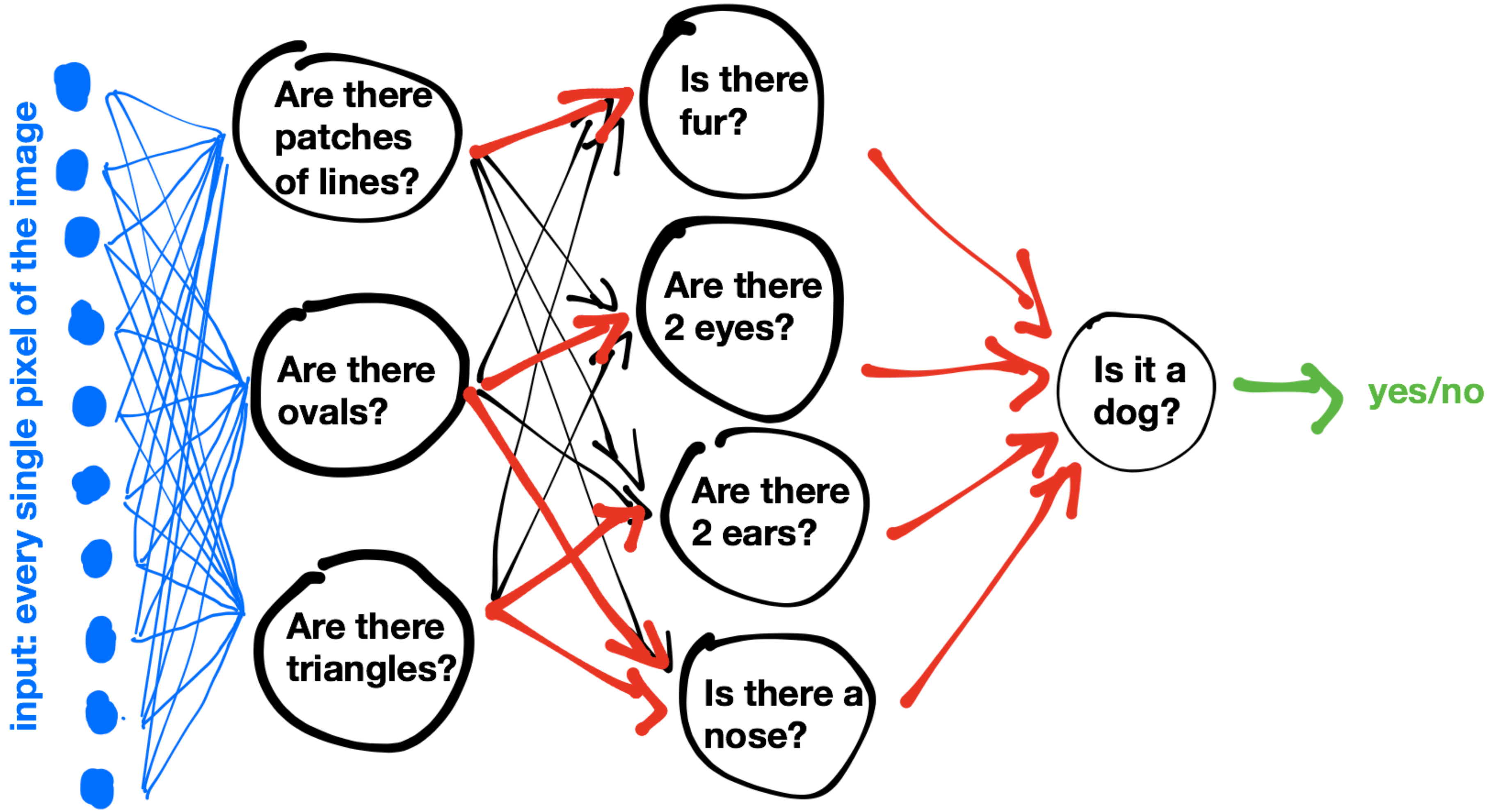
$$f(v) = \frac{1}{1 + e^{-v}}$$



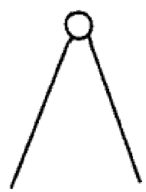
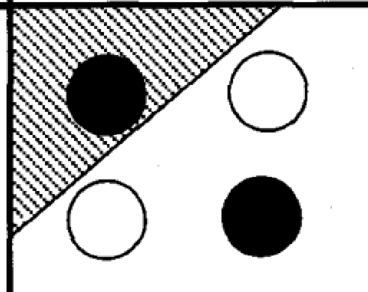
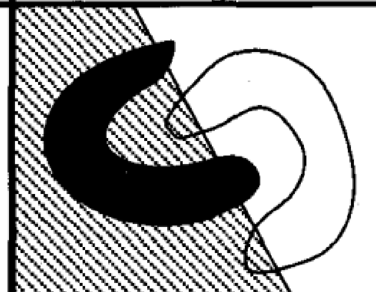
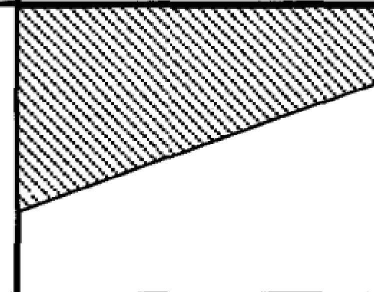
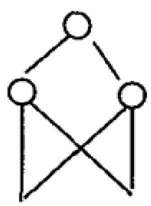
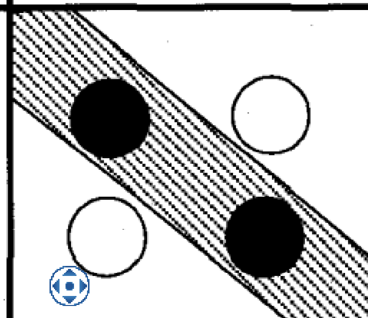
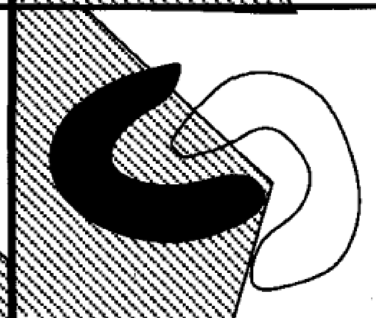
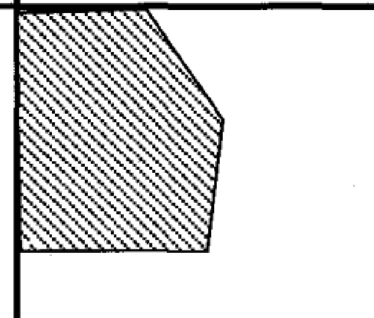
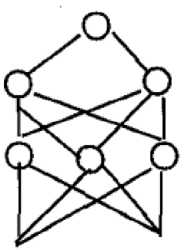
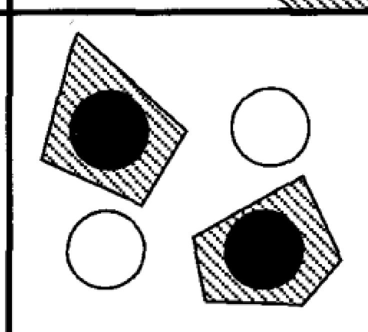
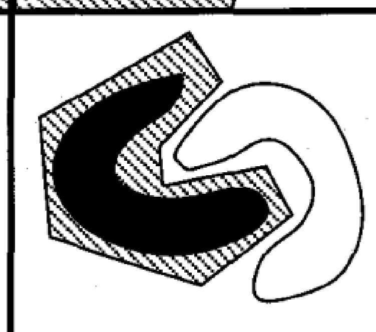
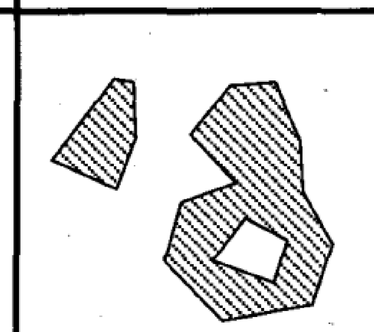
Neural Network Architecture



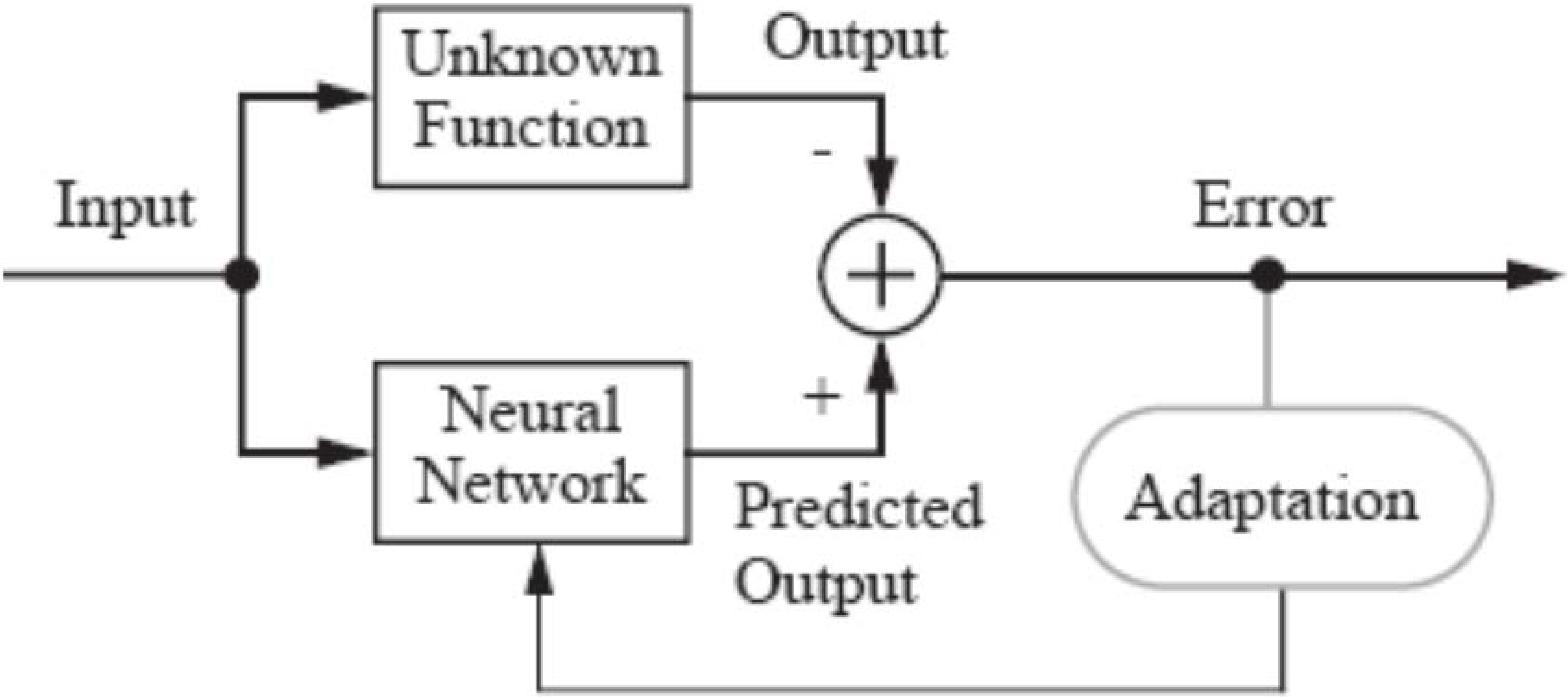
Learning in a Neural Network



Network Structure

Structure	Description of decision regions	Exclusive-OR problem	Classes with meshed regions	General region shapes
 Single layer	Half plane bounded by hyperplane			
 Two layer	Arbitrary (complexity limited by number of hidden units)			
 Three layer	Arbitrary (complexity limited by number of hidden units)			

Function Approximator

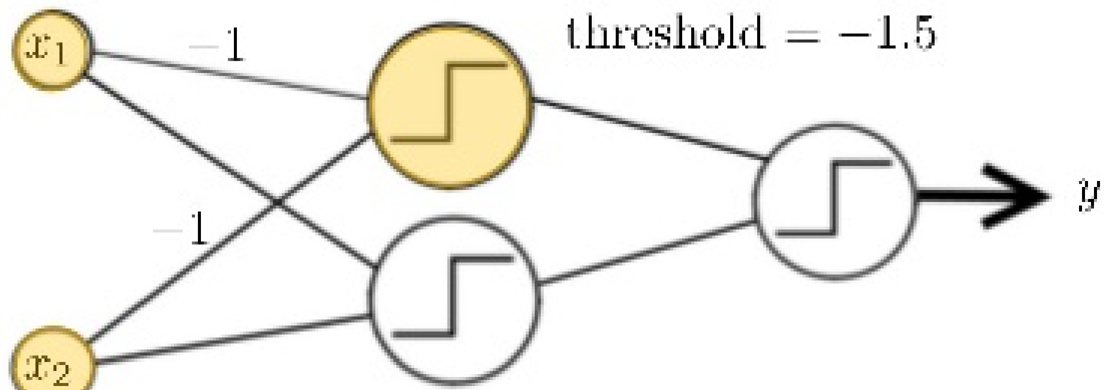


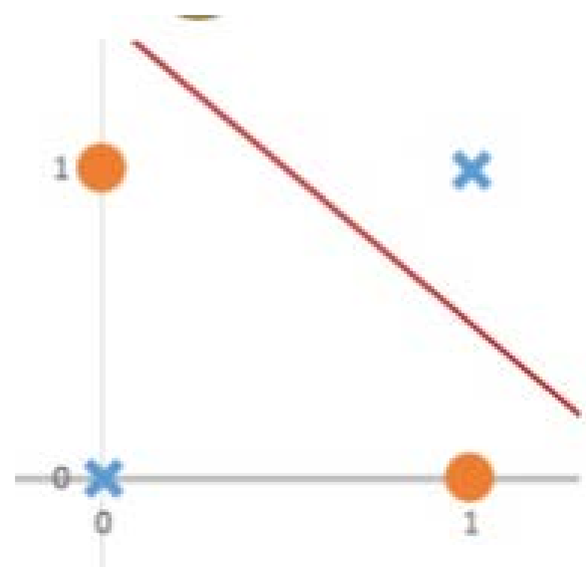
Approximate Function

XOR Problem

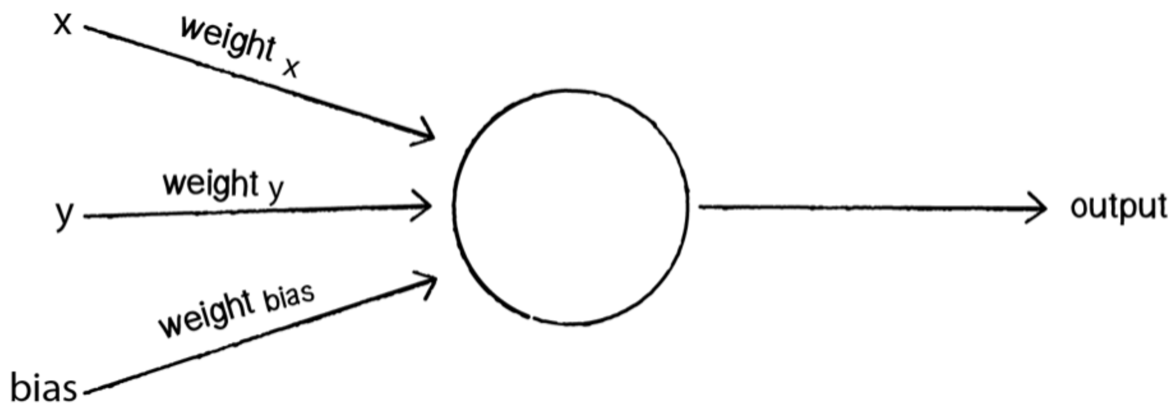
Training Data

x_1	x_2	t
0	0	0
1	0	1
0	1	1
1	1	0



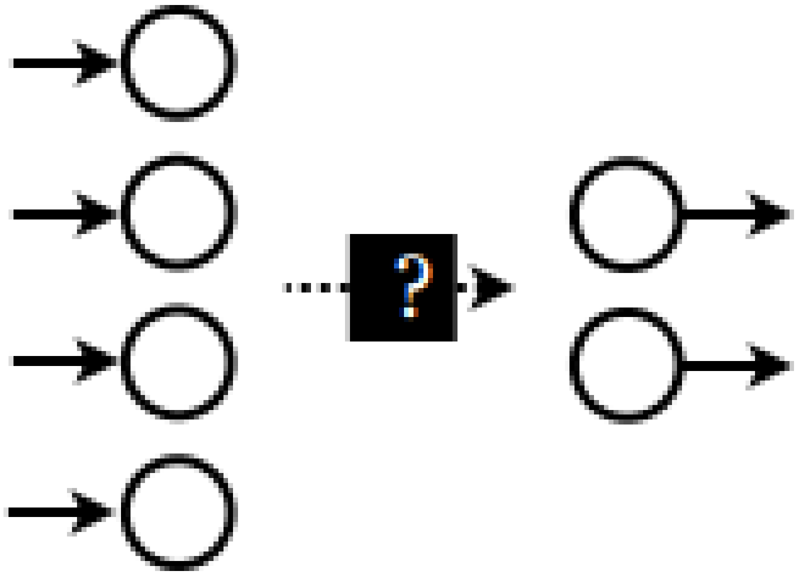


Forward Propagation




```
int feedforward(float[] inputs) {
  float sum = 0;
  for (int i = 0; i < weights.length; i++) {
    sum += inputs[i]*weights[i];
  }
  return activate(sum);
}
```

Learning Mechanism



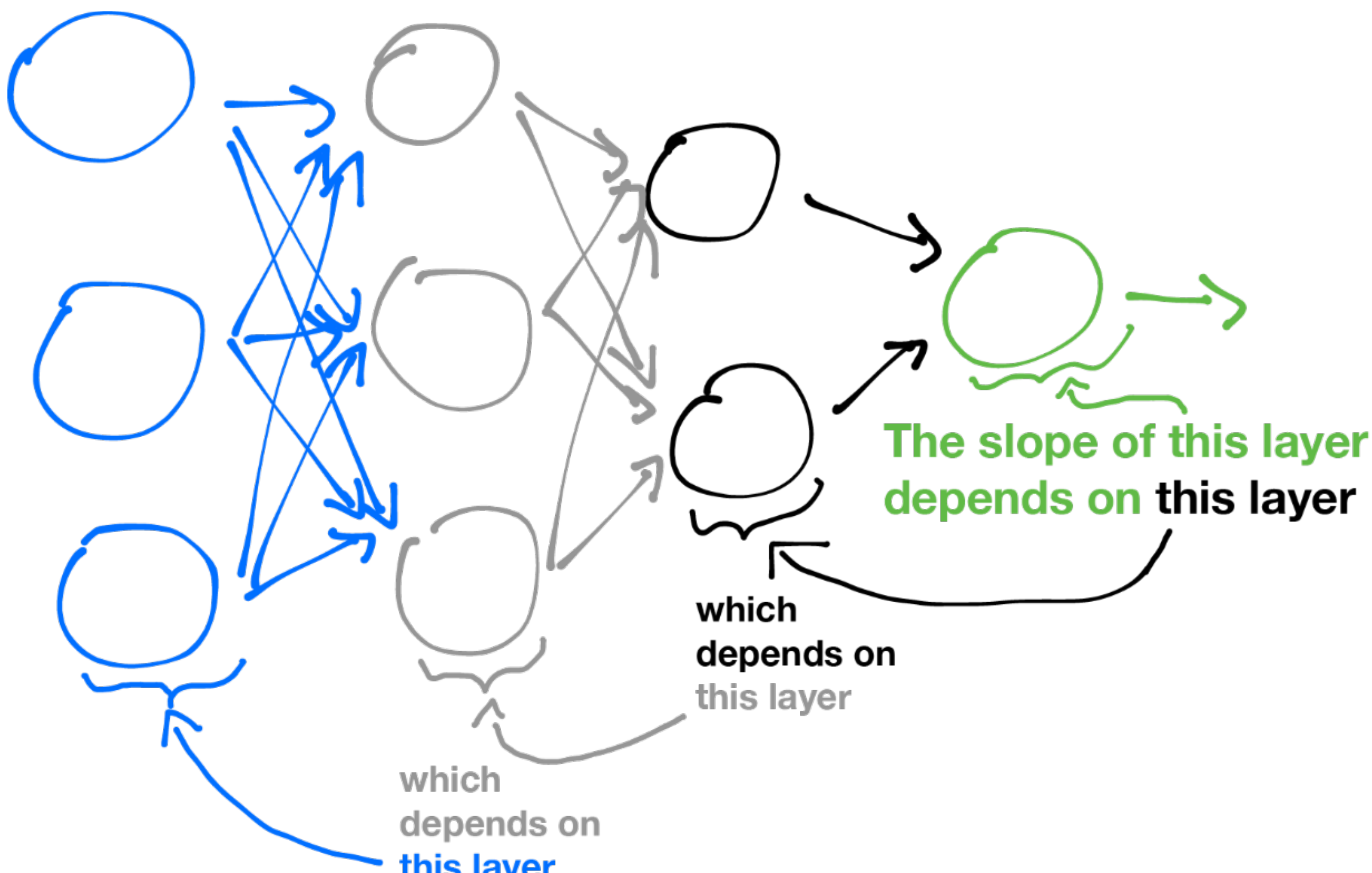
Learning involves **adjusting the weights** of the neural network so the **desired mapping** between inputs and outputs is achieved.

Neural Network Training

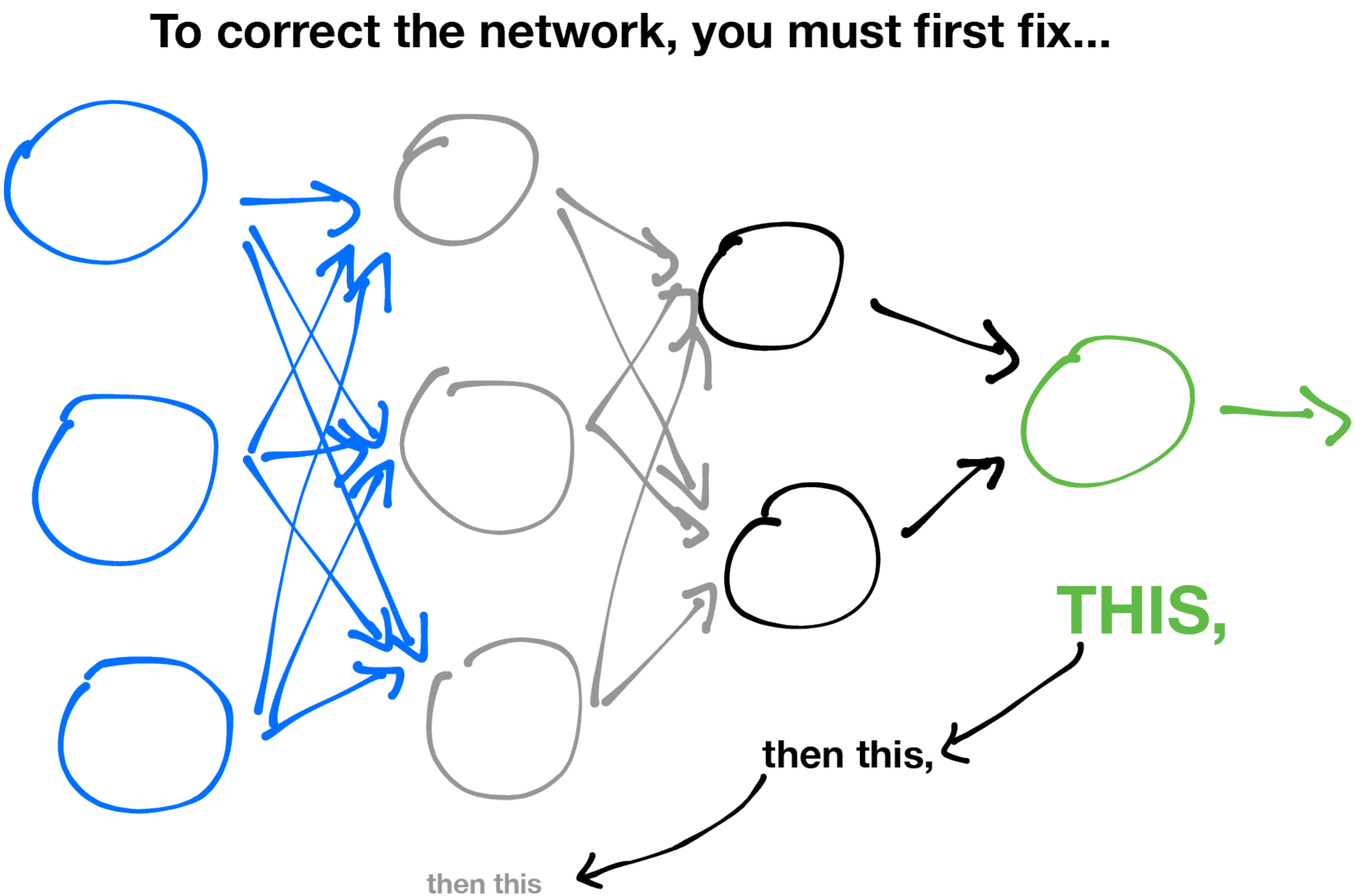
- The **network** must be **trained**:
- Test **data is fed** into the network **via its inputs**.
- The network’s responses are read from its outputs.
- The **connection weights are adjusted** after each test to improve the response of the network as desired.
- After training, **real data is fed into the network** and its responses are used to control the behaviour of some system.

Backpropagation in a Neural Network

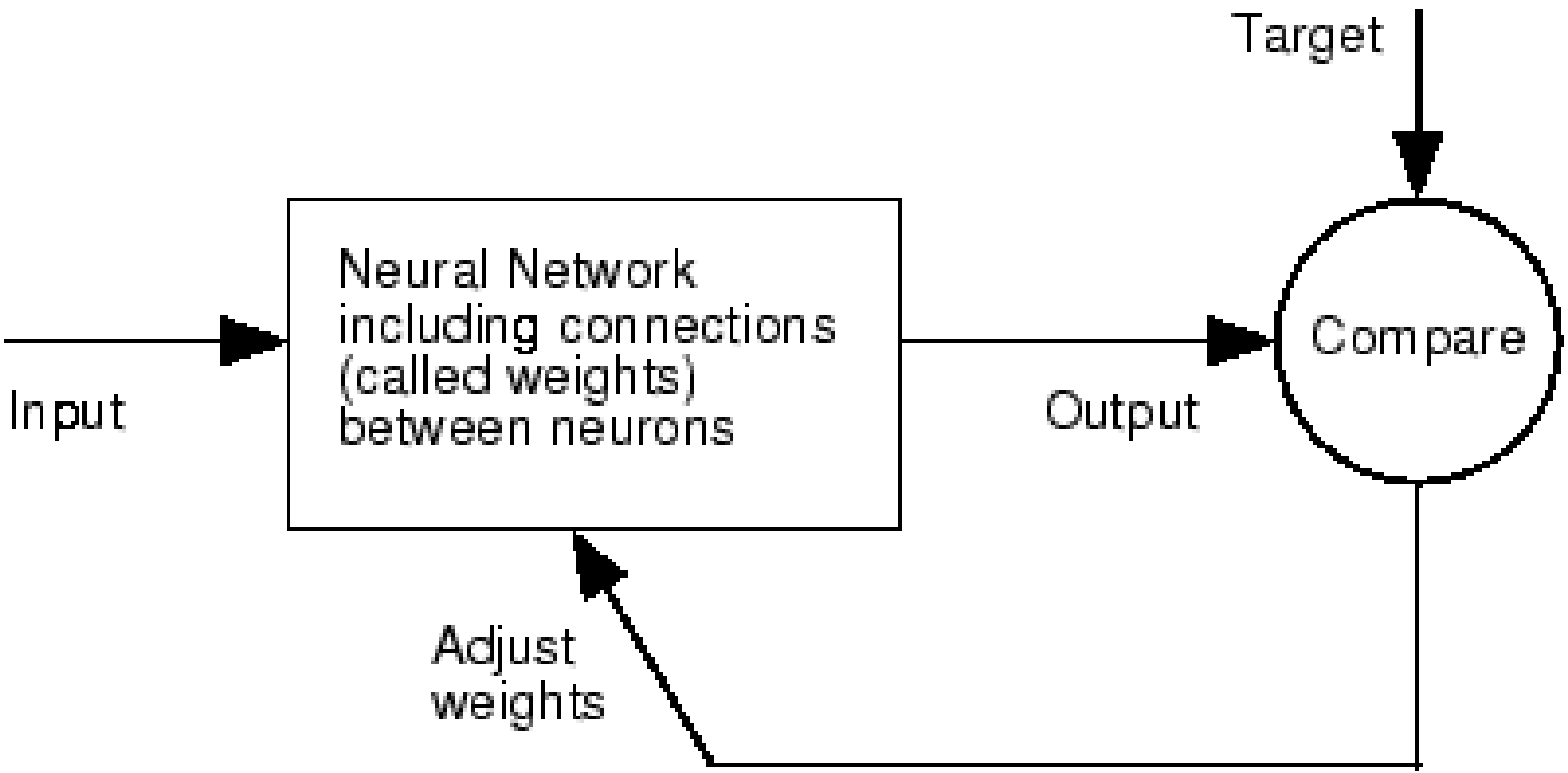
Backpropagation is a common method for **training** a neural network. The goal of backpropagation is to **optimize the weights** so that the neural network can learn how to correctly map arbitrary inputs to outputs.



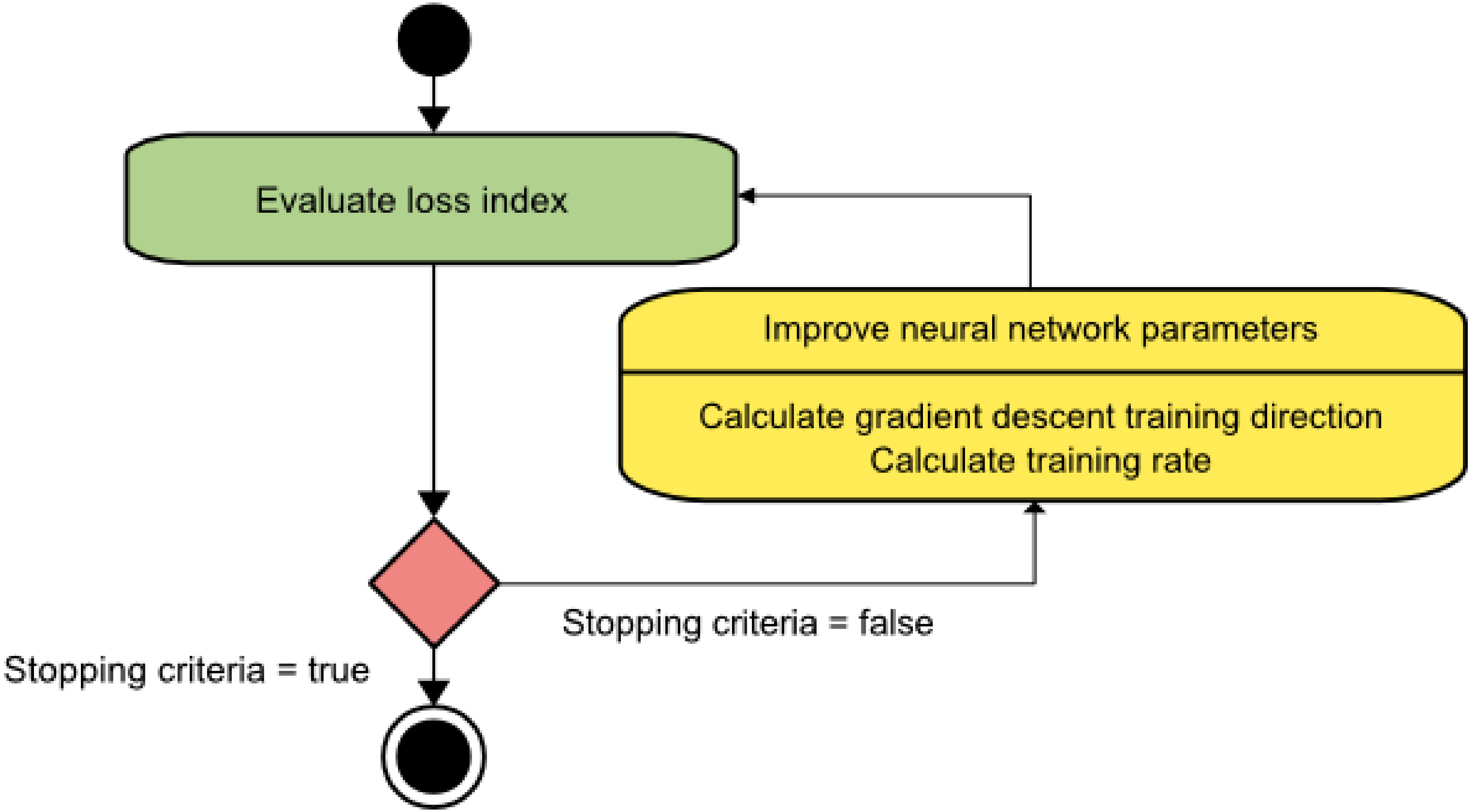
Backpropagation in a Neural Network



Training in a Neural Network



Training in a Neural Network



Training in a Neural Network

```
float c = 0.01;
```

A new variable is introduced to control the learning rate.

```
void train(float[] inputs, int desired) {
```

Step 1: Provide the inputs and known answer. These are passed in as arguments to train().

```
  int guess = feedforward(inputs);
```

Step 2: Guess according to those inputs.

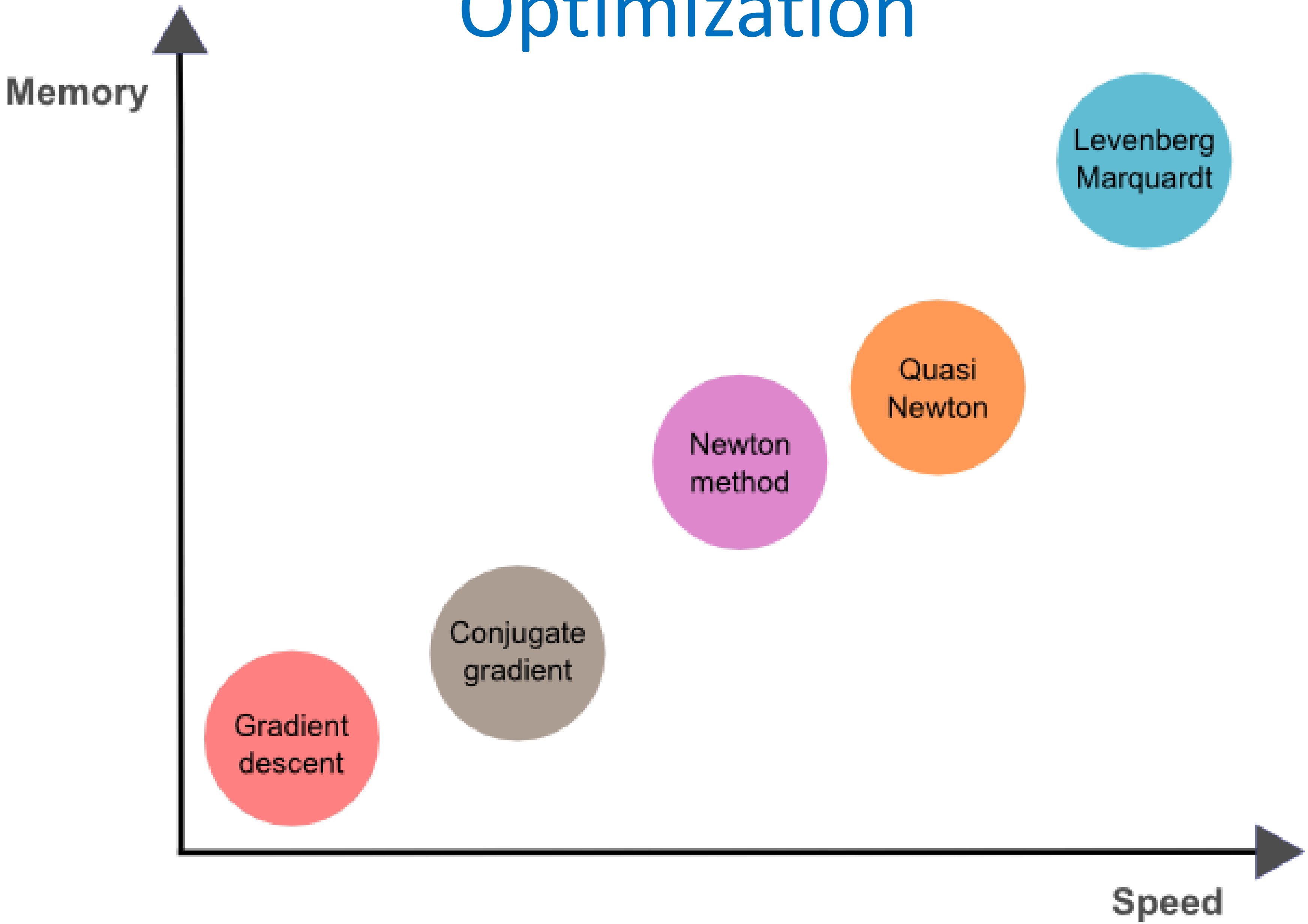
```
  float error = desired - guess;
```

Step 3: Compute the error (difference between answer and guess).

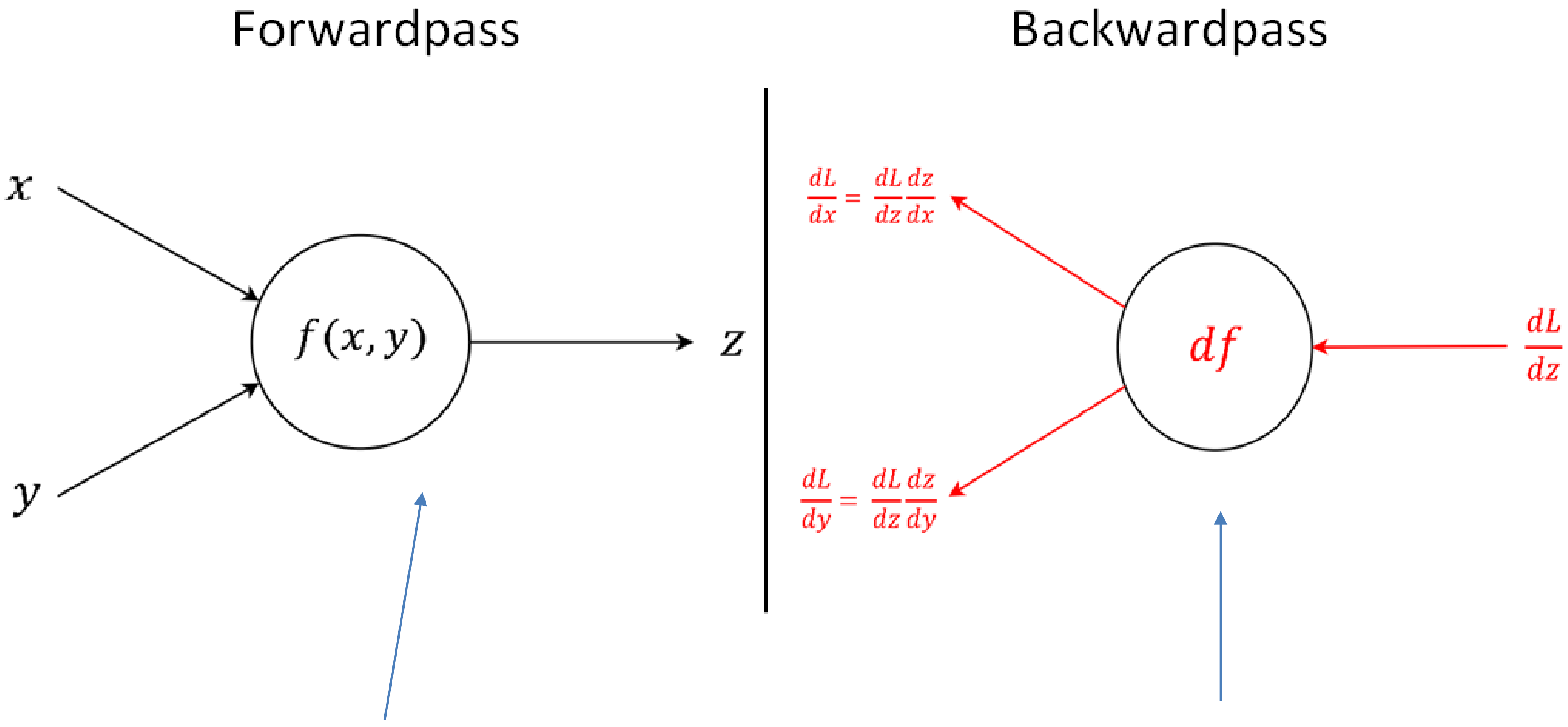
```
  for (int i = 0; i < weights.length; i++) {
    weights[i] += c * error * inputs[i];
  }
}
```

Step 4: Adjust all the weights according to the error and learning constant.

Optimization



Forward vs Backward Pass



From input to output

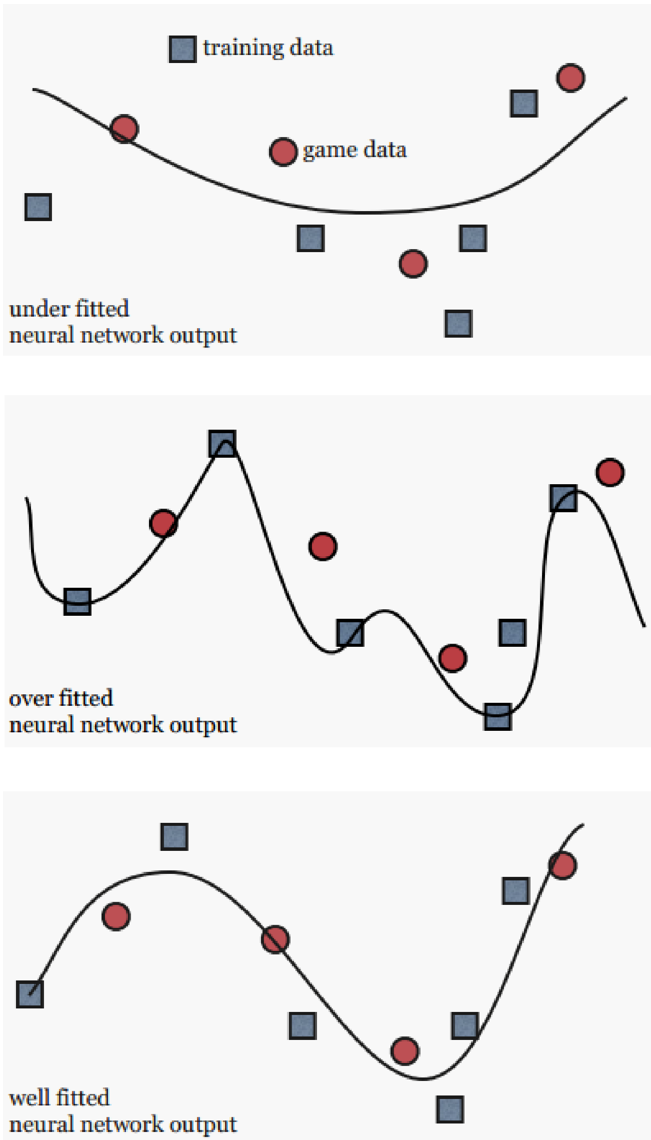
From output to input

Real-World Applications

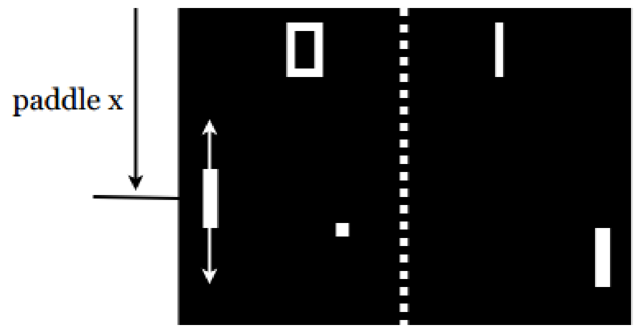
Real-world applications of Artificial Neural Network Applications include:

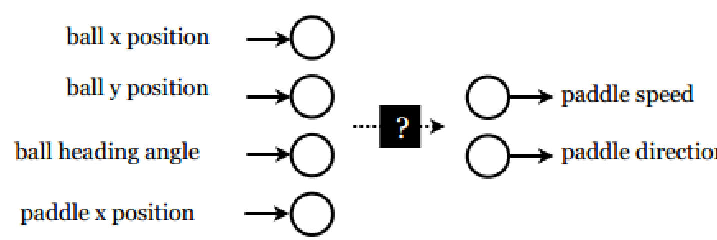
- ***Function approximation*** (modelling)
- ***Pattern classification***
- ***Object recognition*** (e.g. character recognition)
- ***Financial modelling*** (predicting the stock market)
- ***Time series prediction*** (climate, weather)
- ***Computer games*** – (chess, backgammon)
- ***Speech recognition***
- ***Character recognition***
- ***Face Recognition***

ANN Pong Controller



ANN Pong controller training





Advantages of Neural Network

- Can be **applied to many problems,** as long as there is some data.
- Can be applied to problems, for which **analytical methods do not yet exist**
- If **there is a pattern**, then neural networks should quickly **work it out**, even if the data is ‘noisy’.
- **Always gives some answer** even when the input information is not complete.
- Networks are **easy to maintain**.

Limitation of Neural Network

- Like with any data-driven models, they cannot be used if there is **no or very little data available**.
- There are **many free parameters**, such as the number of hidden nodes, the learning rate, minimal error, which may greatly influence the final result.
- Not good for **arithmetics** and **precise calculations.**
- Neural networks **do not provide explanations.**
- If there are many nodes, then there are too many weights that are **difficult to interprete.**

Computer vs Brain

Quantity	Electronic Computer (CMOS)	Human Brain
Mass	$\ll 1$ kg	1 kg
Volume	$< 10^{-6}$ m^3	10^{-3} m^3
# Units	10^{10} gates	10^{10} neurons, 10^{14} synapses
Gate Density	10^{16} gates/m^3	10^{17} synapes/m^3
Gate Dimensions	10^{-6} m	10^{-5} to 10^{-9} m
Period	10^{-9} sec	10^{-2} sec.
Signal Amplitude	2 V	50 mV
Pulse Duration	10^{-9} s	10^{-3} s
Signal Velocity	2×10^8 m/s	10^2 m/s
Energy Dissipation	50 W	10 W
Precision	10^{-12}	10^{-4}
Failure Rate	$< 10^{-9}$/s	1/s
Fan Out Capacity	10	10^4

(cf., John von Neumann, *The Computer and the Brain*, Yale Univ. Press, New Haven, 1958.)

Lab Activities

- Activity : Starting Activity 3 of the coursework (2hours)

References

[1] - Machine Learning - Dan Roth
[2] - Introduction to Neural - Networks - Swingler and Graham
[3] - Artificial Neural Networks - Negnevitsky
[4] - Kohonen Self-Organization Map - Kaustubh Chokshi
[5] - Neural Networks CS105: Great Insights in Computer Science - Michael L. Littman
[6] - Introduction to Neural Networks - Boltzmann Machines - DeLiang Wang
[7] - Self-Organizing Maps - Sven Kruger
[8] - Neural Network - A.C.C Coolen
[9] - Neural Networks - C. R. Dyer
[10] - Backpropagation and Neural Networks - Fei Li and Andrej Karpathy and Justin Johnson
[11] - Introduction to Computing with Neural Nets - Pushpak Bhattacharyya
[12] - Feed-Forward Neural Networks - Roman V Belavkin
[13] - Self-Organising Maps - Roman V Belavkin
[14] - Introduction to Neural Networks - Hopfield Network for Associative Memory
[15] - Introduction to Neural Networks - Wang
[16] - Biologically-Inspired Computing - Luis Rocha
[17] - Introduction to Neural Networks - DeLiang Wang
[18] - Neural Computation - Robert R. Snapp
[19] - Introduction to Artificial Neural Networks - K. Ming Leung
[20] - Self Organizing Maps: Fundamentals - John A. Bullinaria

References

[21] - A Basic Introduction To Neural Networks - Josef Burge
[22] - Neural Networks Theory - Matus Telgarsky Scribes: Xintong Wang, Editors: Yuan Zhuang
[23] - Biologically inspired computing - Jim Torrese
[24] - The Self-Organizing Map - Gisbert Schneider
[25] - An Introduction to Artificial Neural Networks - Alan Dorin
[26] - Introduction to Machine Learning - Shalev-Shwartz
[27] - Artificial Neural Networks - Stephen Lucci
[28] - Introduction to Neural Networks - Taylor B. Arnold
[29] - Introduction to Neural Networks - Linear Regression
[30] - Introduction to Neural Networks - Multilayer Perceptrons - Ohio State university
[31] - Multi-Layer Artificial Neural Networks - Simon Colton
[32] - Multilayer Neural Network - Jeff Robble, Brian Renzenbrink, Doug Roberts
[33] - Artificial Neuron Network - Anil Jain
[34] - Perceptron for Pattern - Classification - K. Ming Leung
[35] - Self-Organizing Maps (SOM) - Laurenz Wiskott
[36] - Simple Neural Networks for Pattern Classification - K. Ming Leung
[37] - An Introduction to Self-Organizing Maps - Christoph Brauer
[38] - Ryszard Tadeusiewcz "Sieci neuronowe", Kraków 1992