Artificial Neural Networks (ANNs) Introduction

Sabah Sayed

Department of Computer Science

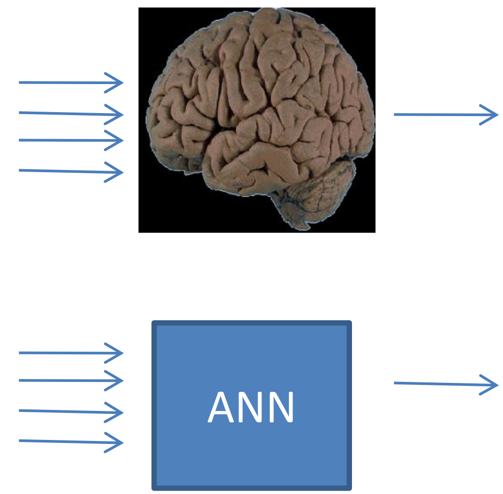
Faculty of Computers and Artificial Intelligence

Cairo University

Egypt

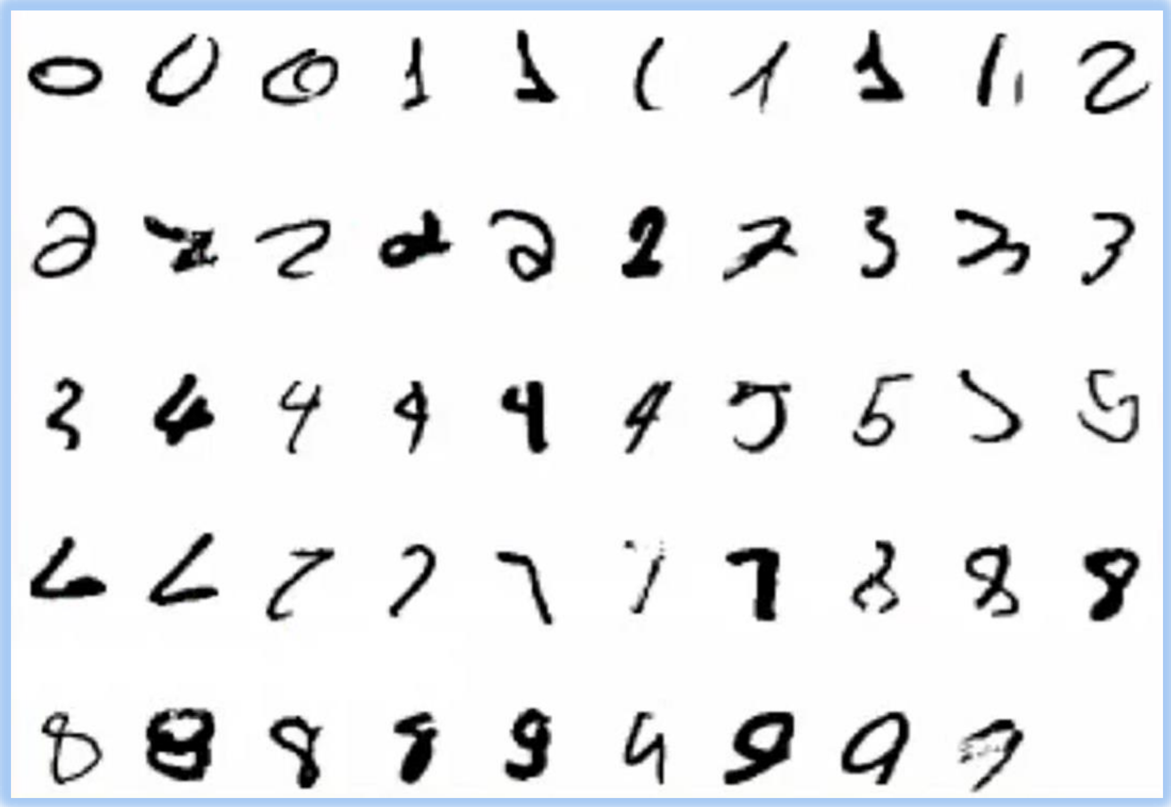
Why we need ANNs?

- There are some advanced operations (like face recognition, hand writing recognition … etc) that we can do using our brain, but we don't know how it is exactly/internally done.
- We used to successfully do these operations after sufficient amount of training and practice.
- Unfortunately, it is very hard to try writing such a program that can recognize faces or hand-written text.
- Here, the role of the ANNs comes. It mimics the training activities that are internally done using the ***Nerve System*** inside the human brain!
- Training occurs through learning from many already-known past examples.
- A well-trained ANN will be able to yield correct results for new unknown problem instances.



Example: Hand-writing Recognition

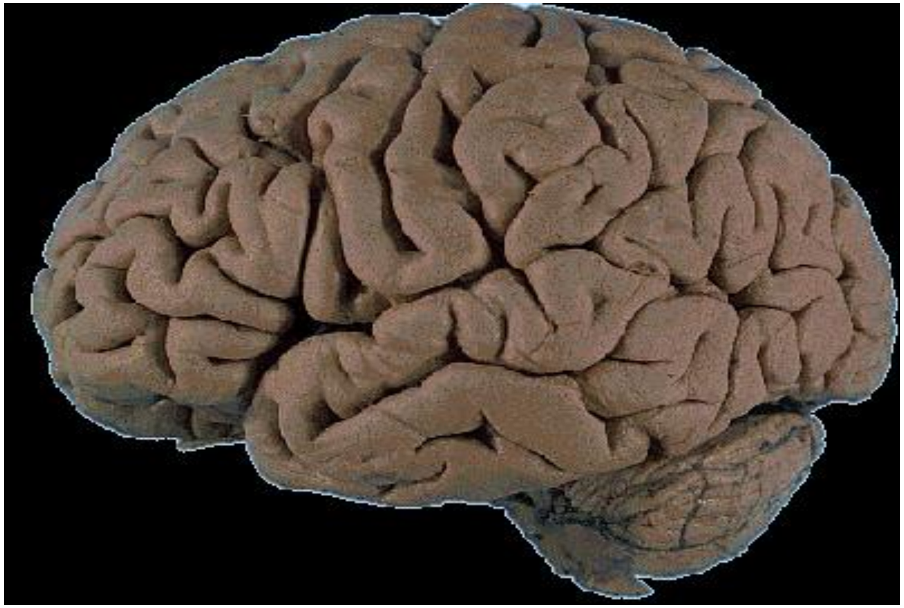
- It is very hard to develop a program that tells what makes a 2



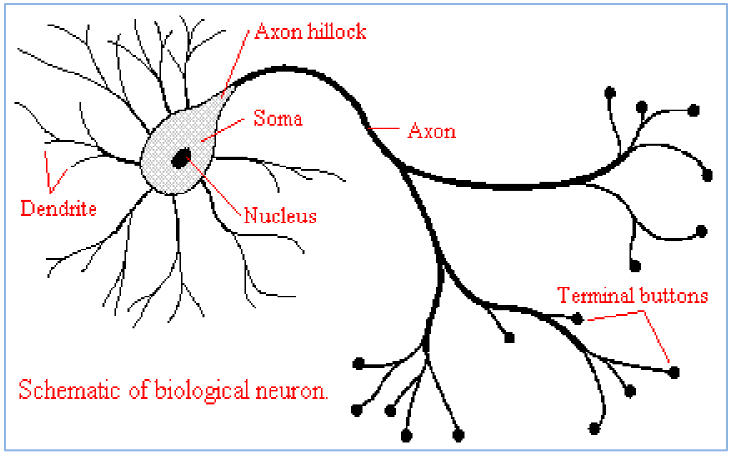
The Brain

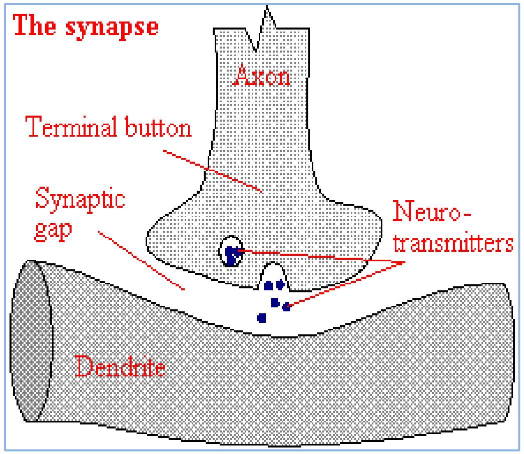
The Brain as an Information Processing System

The human brain contains about **10 billion** nerve cells, or **neurons**. On average, each neuron is connected to other neurons through about **10 000 synapses**.



The Biological Neuron



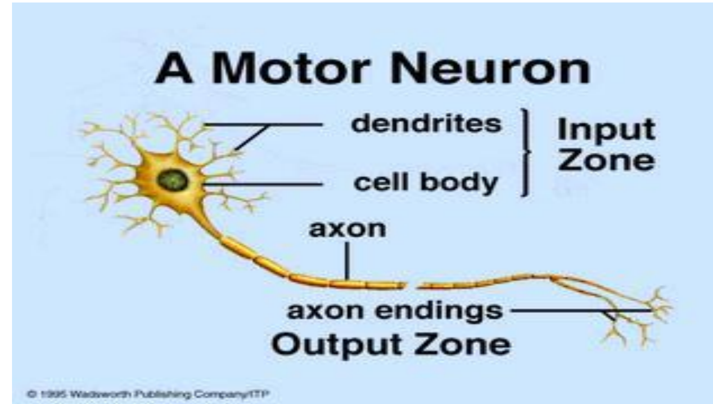


- The brain is a collection of about 10 billion interconnected neurons. Each *neuron* is a cell that uses *biochemical reactions* to ***receive***, ***process*** and ***transmit*** information.

- Each terminal button is connected to other neurons across a small gap called a *synapse*.

Neurons and Synapses

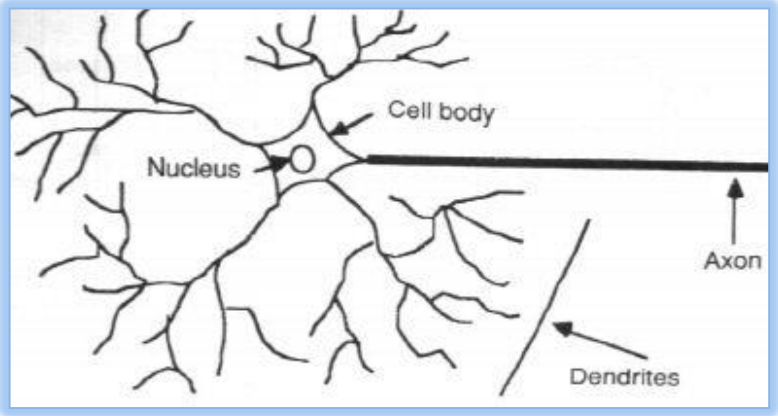
- The basic computational unit in the nervous system is the nerve cell, or neuron. A neuron has:
    - Inputs (Dendrites)
    - Cell body
    - Output (Axon)

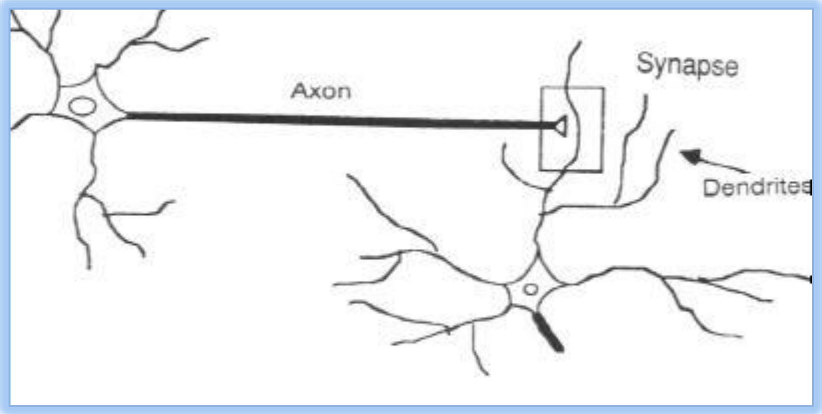


- A neuron receives input from other neurons (typically many thousands).
- Inputs sum (approximately).
- Once input exceeds a critical level, the neuron discharges a spike - an electrical pulse that travels from the cell body, down the axon, to the next neuron(s) (or other receptors).

- The axon endings (Output Zone) almost touch the dendrites or cell body of the next neuron.
- Transmission of an electrical signal from one neuron to the next is effected by neuro-transmittors, chemicals which are released from the first neuron and which bind to receptors in the second.
- This link is called a synapse.
- The extent to which the signal from one neuron is passed on to the next depends on many factors, e.g.:
    - The amount of neuro-transmittor available,
    - The number and arrangement of receptors,
    - Amount of neuro-transmittor reabsorbed, etc.

How the Human Brain learns



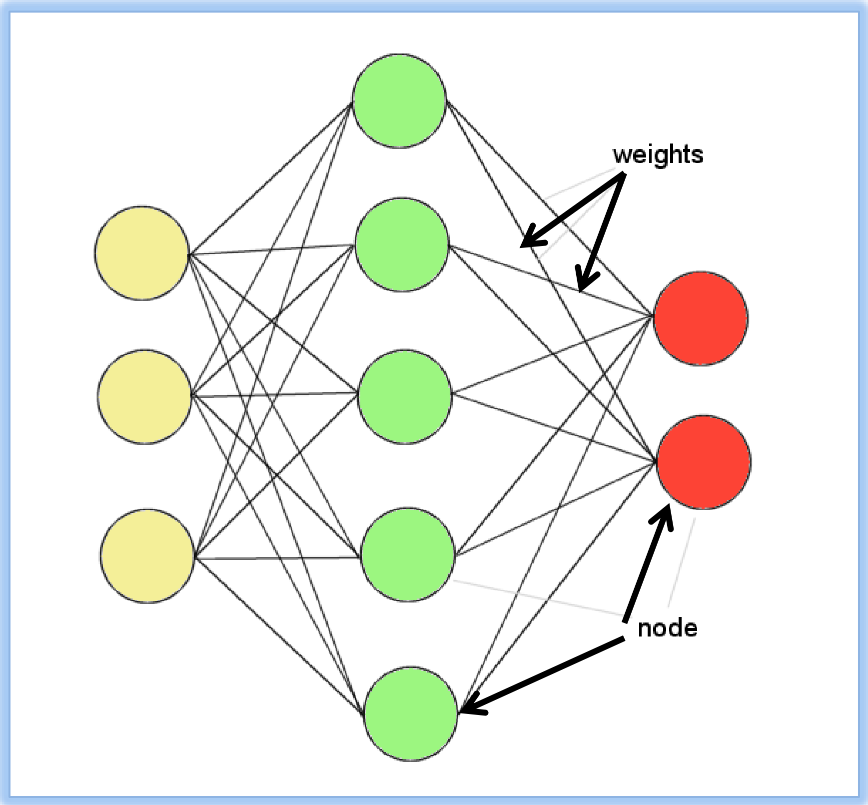


- In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*.
- The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches.
- At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons.

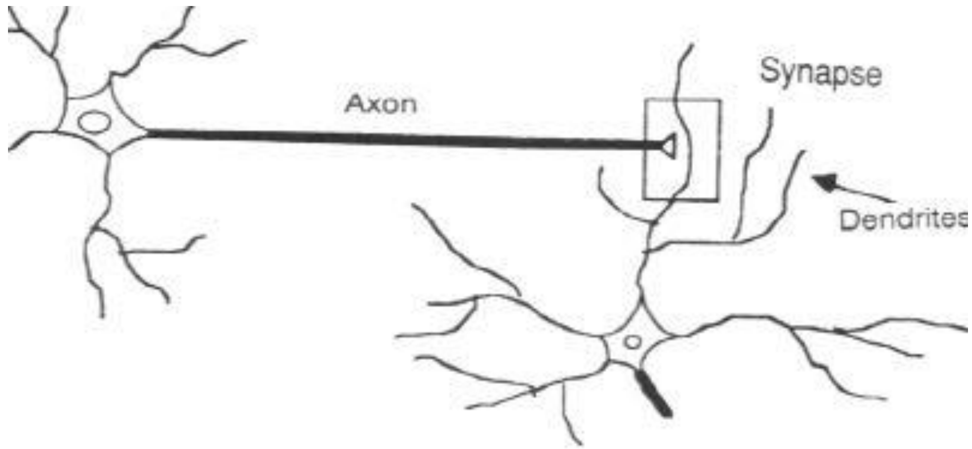
ANNs

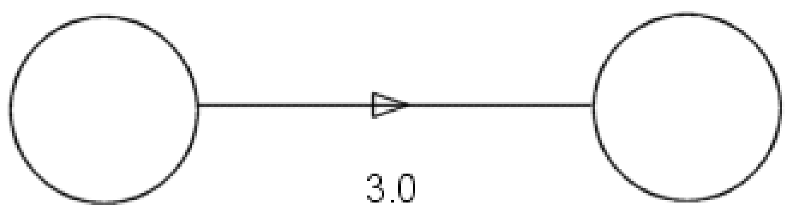
- ANNs incorporate the two fundamental components of biological neural networks:

1. Neurons (nodes)
2. Synapses (weights)

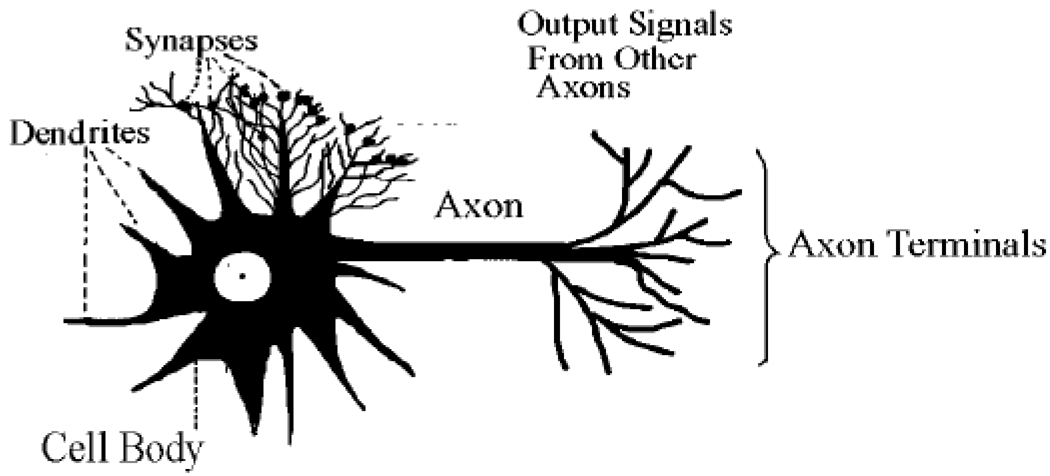


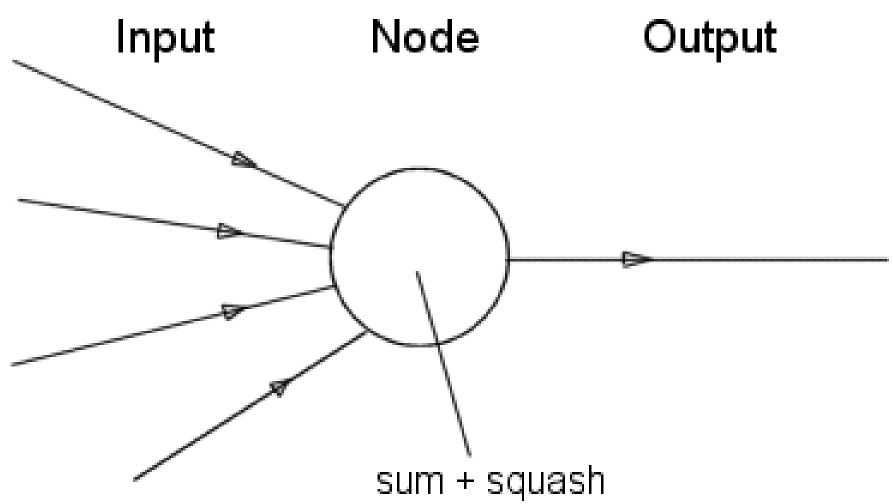
Axon/Synapse vs. Weight



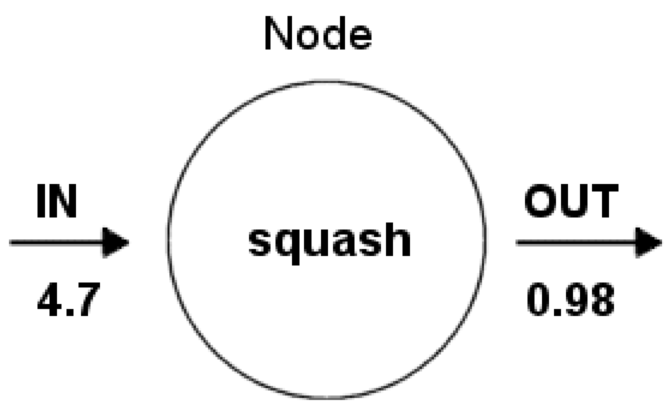


Neuron vs. Node

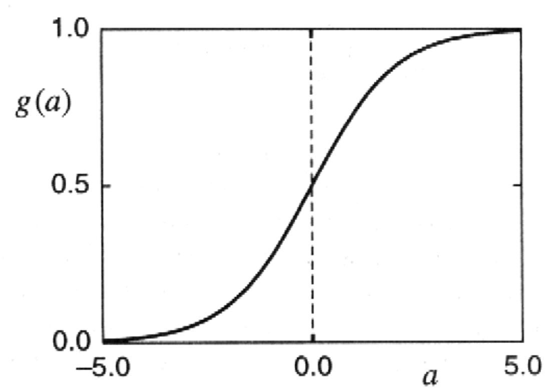




Structure of a node

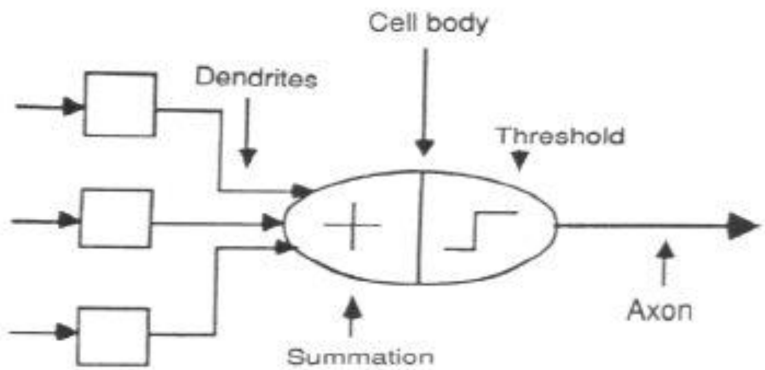


Squashing/Activation function limits node output:



A Neuron Model

- When a neuron receives *excitatory input* that is sufficiently large compared with its *inhibitory input*, it sends a spike of electrical activity down its axon.
- *Learning* occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.



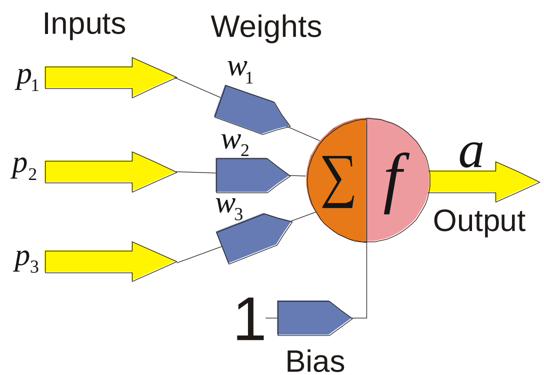
- We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections.
- We then typically program a computer to simulate these features.

Definition of Neural Network

- A Neural Network is a system composed of many simple processing elements operating in parallel which can acquire, store, and utilize experiential knowledge.
- **Neural network:** *information processing paradigm inspired by biological nervous systems, such as our brain*
- Structure: large number of highly interconnected processing elements (*neurons*) working together
- Like people, they learn *from experience* (by example)

The Key Elements of Neural Networks

- Neural computing requires a number of neurons, to be connected together into a neural network. Neurons are arranged in layers.



$$a = f(p_1w_1 + p_2w_2 + p_3w_3 + b) = f\left(\sum p_iw_i + b\right)$$

- Each *neuron* within the network is usually a *simple processing unit* which takes one or more inputs and produces an output.
- At each neuron, every input has an associated weight which modifies the strength of each input.
- The neuron simply adds together all the inputs and calculates an output to be passed on.

ANN

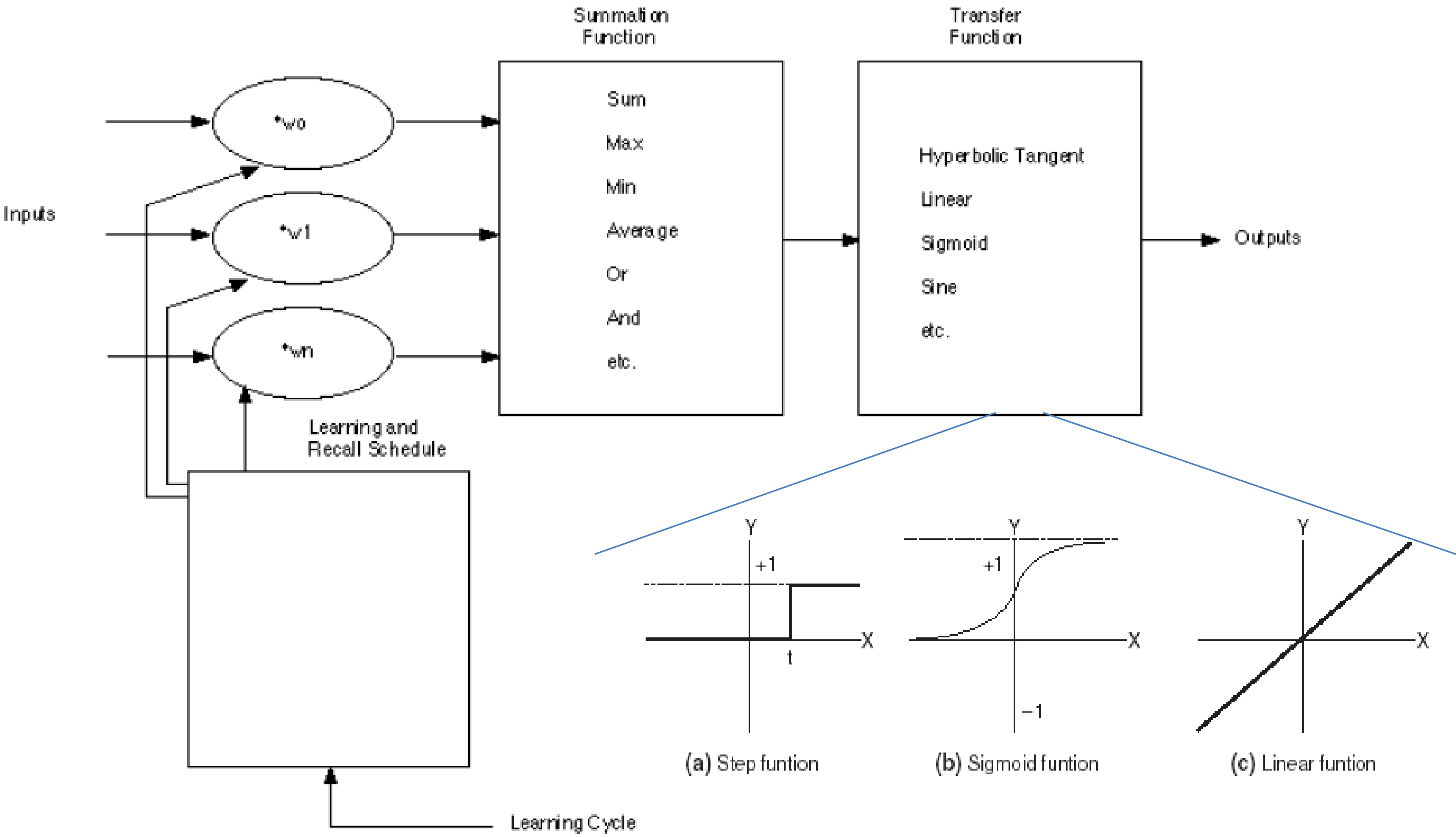
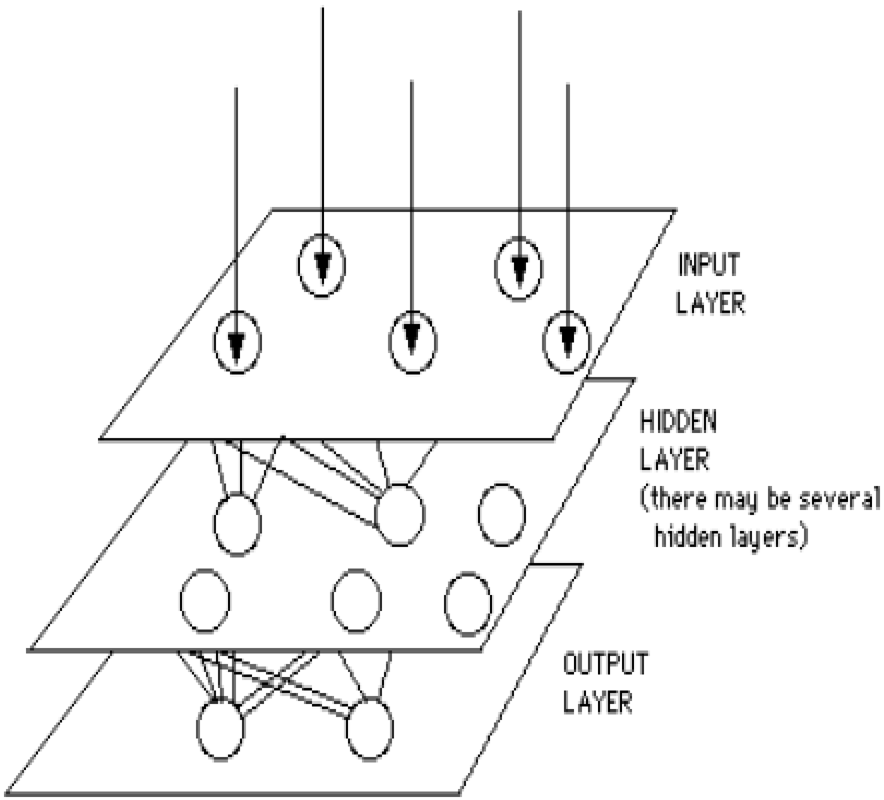
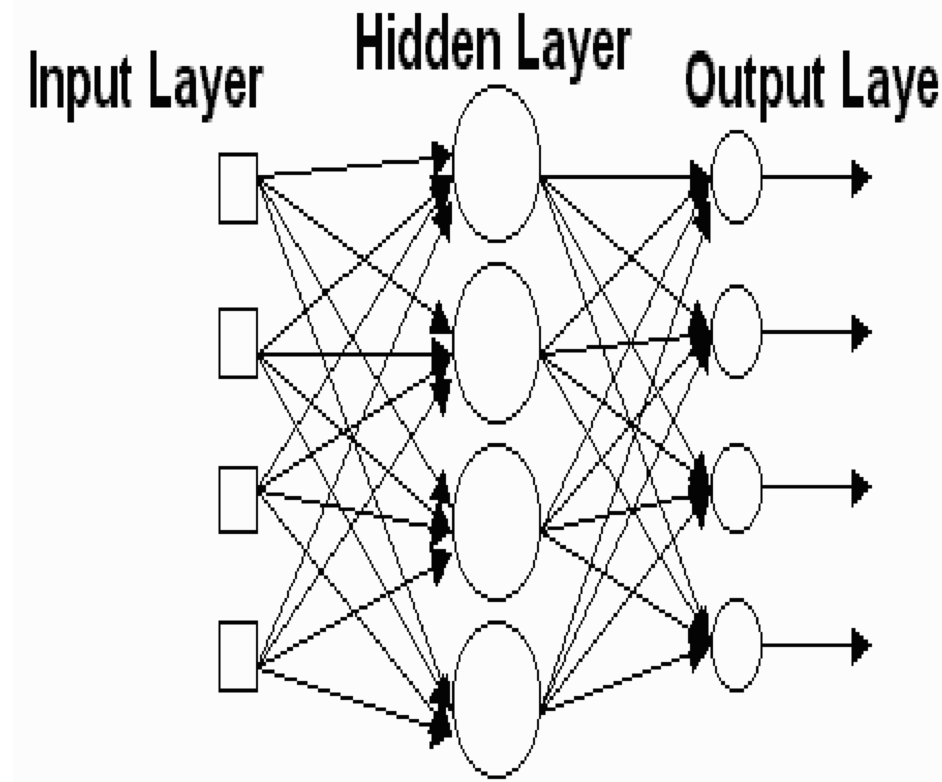
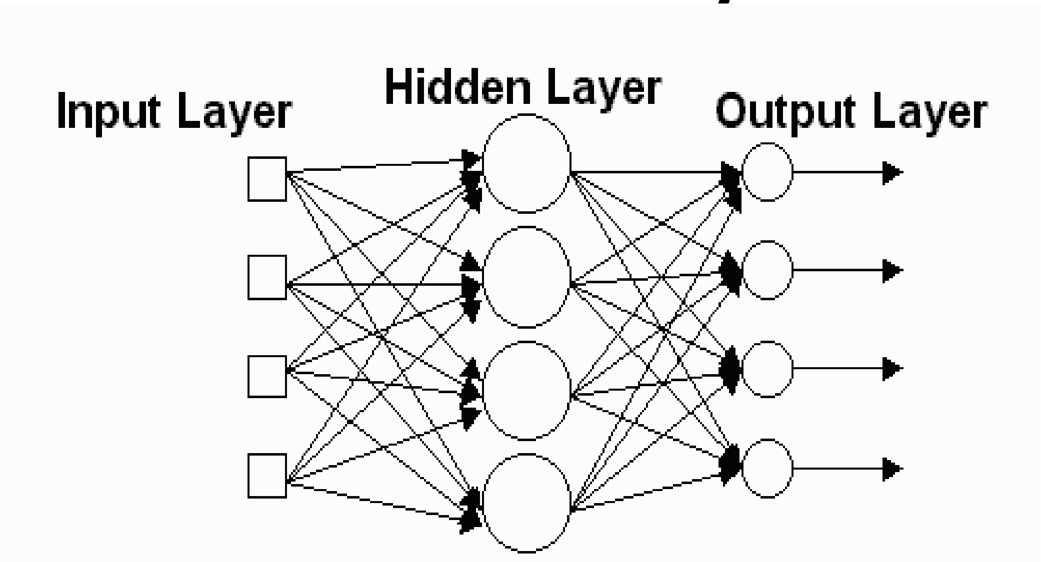

Summation Function
Transfer Function
Sum
Max
Min
Average
Or
And
etc.
Hyperbolic Tangent
Linear
Sigmoid
Sine
etc.
Inputs
*wo
*w1
*wn
Outputs
Learning and Recall Schedule
Learning Cycle
Y
+1
X
t
−1
(a) Step funtion
(b) Sigmoid funtion
(c) Linear funtion

Diagram of a simple ANN





Network Layers

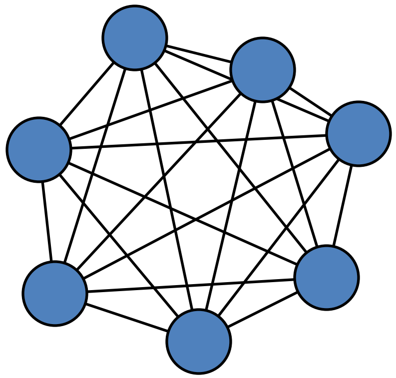


- **Input Layer** - The activity of the input units represents the raw information that is fed into the network.

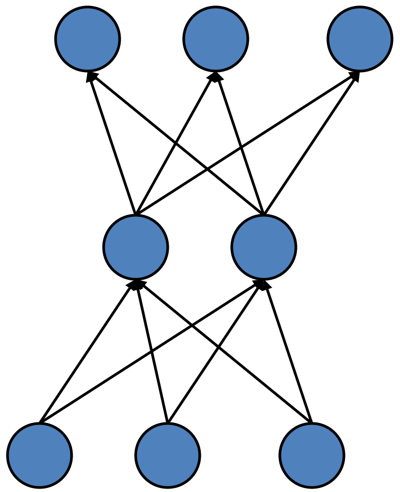
- **Hidden Layer** - The activity of each hidden unit is determined by:
  - The activities of the input units and
  - The weights on the connections between the input and the hidden units.

- **Output Layer** - The behavior of the output units depends on:
  - The activity of the hidden units and
  - The weights between the hidden and output units.

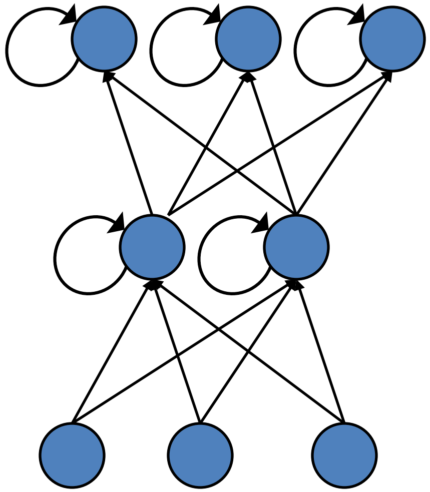
Topologies of Neural Networks



completely connected



feedforward (directed, a-cyclic)



recurrent (feedback connections)

History

- 1943: McCulloch–Pitts “neuron”
  - Started the field
- 1962: Rosenblatt’s perceptron
  - Learned its own weight values; convergence proof
- 1969: Minsky & Papert book on perceptrons
  - Proved limitations of single-layer perceptron networks
- 1982: Hopfield and convergence in symmetric networks
  - Introduced energy-function concept
- 1986: Backpropagation of errors
  - Method for training multilayer networks
- Present: Probabilistic interpretations, Bayesian and spiking networks

Artificial Neural Networks vs. Computers

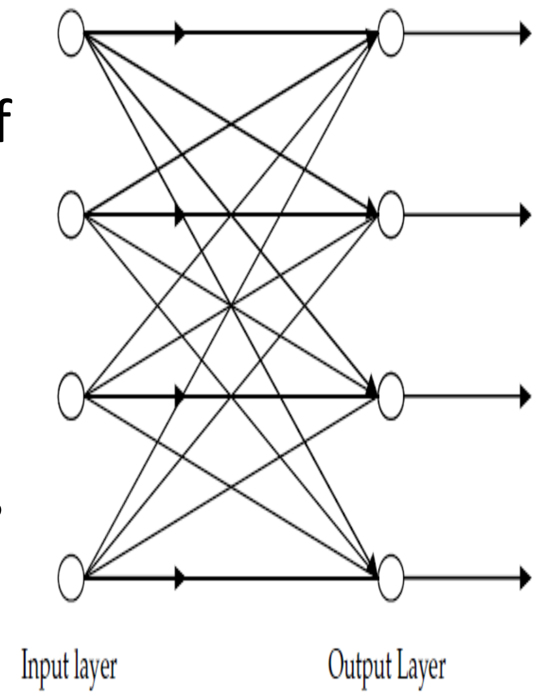
- Computers have to be explicitly programmed:
  - Analyze the problem to be solved.
  - Write the code in a programming language.
- Neural networks learn from examples:
  - No requirement of an explicit description of the problem.
  - No need for a programmer.
  - The neural network adapts itself during a ***training period***, based on examples of similar problems even without a desired solution to each problem.
  - After sufficient training, the neural network is able to relate the problem data to the solutions, inputs to outputs, and it is then able to offer a viable solution to a brand new problem.
  - Able to generalize or to handle incomplete data.

The ANN applications

- ***Classification, the aim is to predict the class of an input vector***
- ***Pattern matching, the aim is to produce a pattern best associated with a*** given input vector
- ***Pattern completion, the aim is to complete the missing parts of a given*** input vector
- ***Optimization, the aim is to find the optimal values of parameters in an*** optimization problem
- ***Control, an appropriate action is suggested based on given an input*** *vectors*
- ***Function approximation/times series modeling, the aim is to learn the*** functional relationships between input and desired output vectors;
- ***Data mining, with the aim of discovering hidden patterns from data*** (knowledge discovery)
- **Noise Reduction**

Different types of Neural Networks

- **Feed-forward Neural Network: (FFNN)**
  - It allows signals to travel one way only; from input to output.
  - There is no feedback (loops) i.e. the output of any layer does not affect that same layer.
  - It tends to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition.
  - This type of organization is also referred to as bottom-up or top-down.

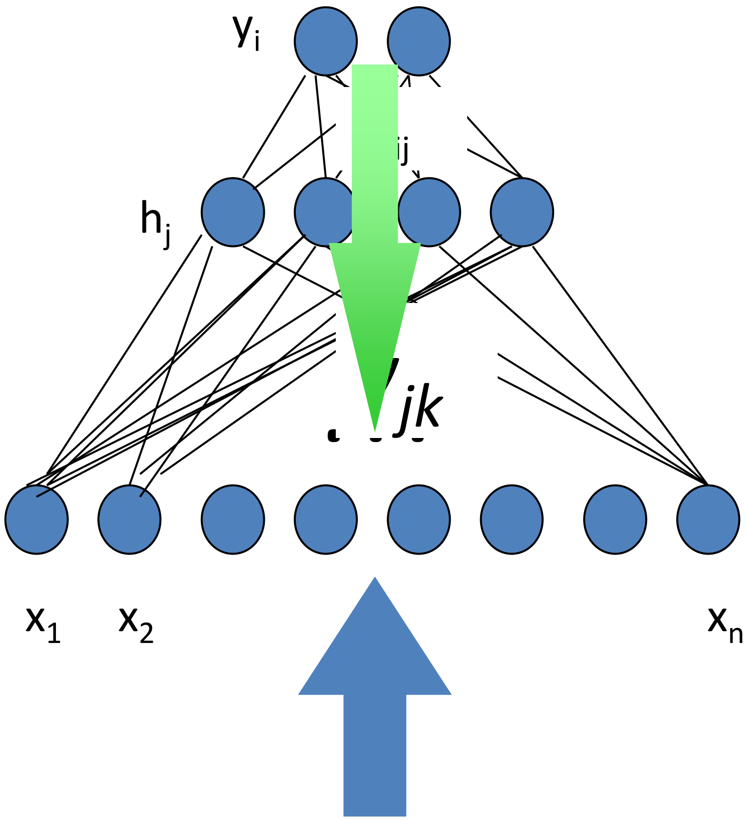


Single Layer FFNN

- ➢ Single layer FFNN
- ➢ Multilayer FFNN

Different types of Neural Networks

- **Feedback Neural Networks**
  - Feedback networks can have signals traveling in both directions by introducing loops in the network.
  - Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point.
  - They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.
  - Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.



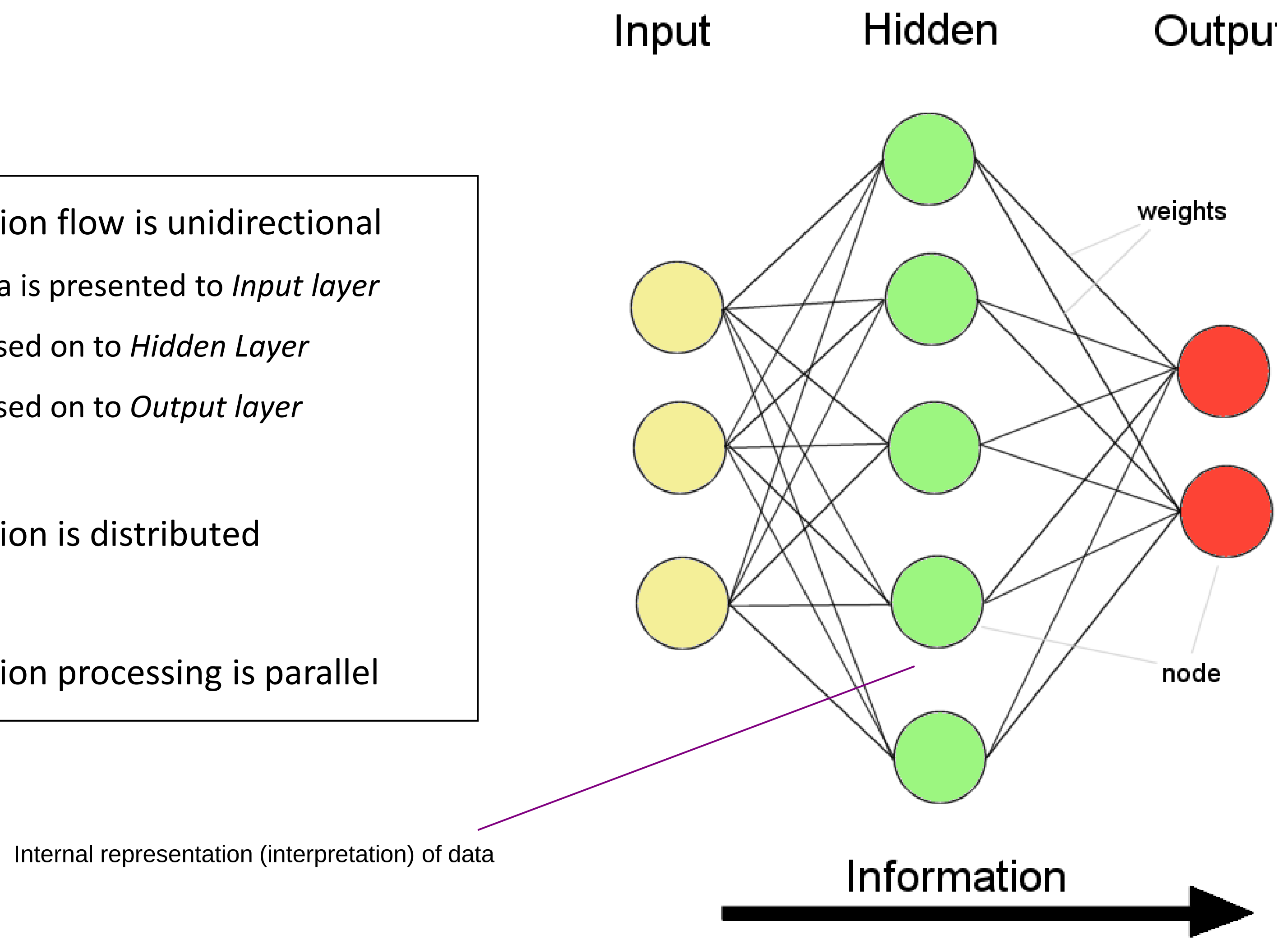
Feed-forward Neural Networks (FFNNs)

Information flow is unidirectional

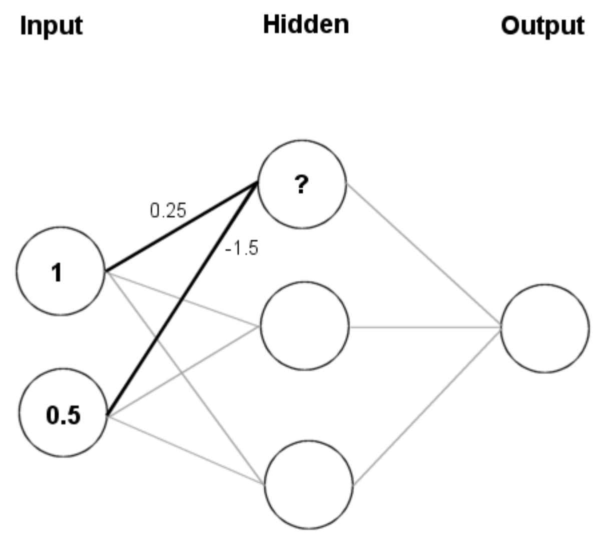
- Data is presented to *Input layer*
- Passed on to *Hidden Layer*
- Passed on to *Output layer*

Information is distributed

Information processing is parallel



Feeding data through the Network



(1 × 0.25) + (0.5 × (-1.5)) = 0.25 + (-0.75)   = - **0.5**

Squashing:

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \qquad \frac{1}{1+e^{0.5}} = 0.3775$$

Phases and Data of an ANN

- Phases:
  - Learning (training)
    - Training can be done using **Back-propagation** technique (will be seen later).
    - Training is done using a big part of known historical (labeled) data.
      - The dataset contains both inputs and their corresponding output(s).
    - This phase is *time consuming*
  - Testing
    - Testing is done using the **Feed-forward** technique.
    - A small part of the historical data is used in this phase.
  - Execution
    - The **Feed-forward** technique is used.
    - New unlabeled data (inputs with unknown outputs) are fed to the NN.
- Labeled Data:
  - Data is usually given in one or more consistent dataset(s)
  - Data is divided into two parts:
    - One part for the training/learning phase
    - Another part for the testing phase

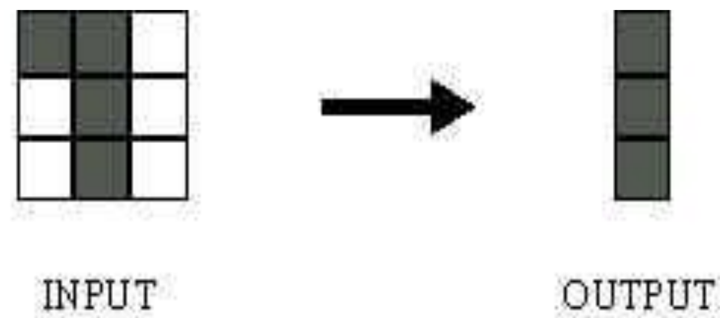
Example: Pattern Recognition

- An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward neural network that has been **trained** accordingly.
- During training, the network is trained to associate outputs with input patterns.
- When the network is used, it identifies the input pattern and tries to output the associated output pattern.
- The power of neural networks comes to life when a pattern that has no output associated with it, is given as an input.
- In this case, **the network gives the output that corresponds to a taught input pattern that is least different from the given pattern**.

Pattern Recognition (cont.)

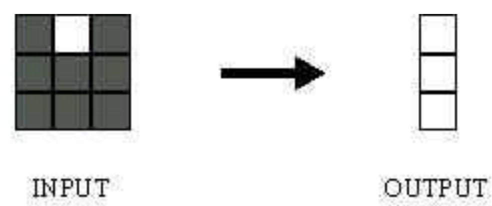
- Suppose a network is trained to recognize the patterns T and H. The associated patterns are all black and all white respectively as shown above.

Pattern Recognition (cont.)



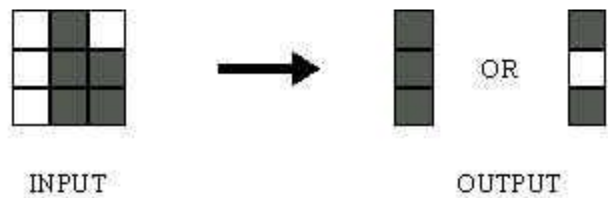
Since the input pattern looks more like a ‘T’, when the network classifies it, it sees the input closely resembling ‘T’ and outputs the pattern that represents a ‘T’.

Pattern Recognition (cont.)



The input pattern here closely resembles 'H' with a slight difference. The network in this case classifies it as an 'H' and outputs the pattern representing an 'H'.

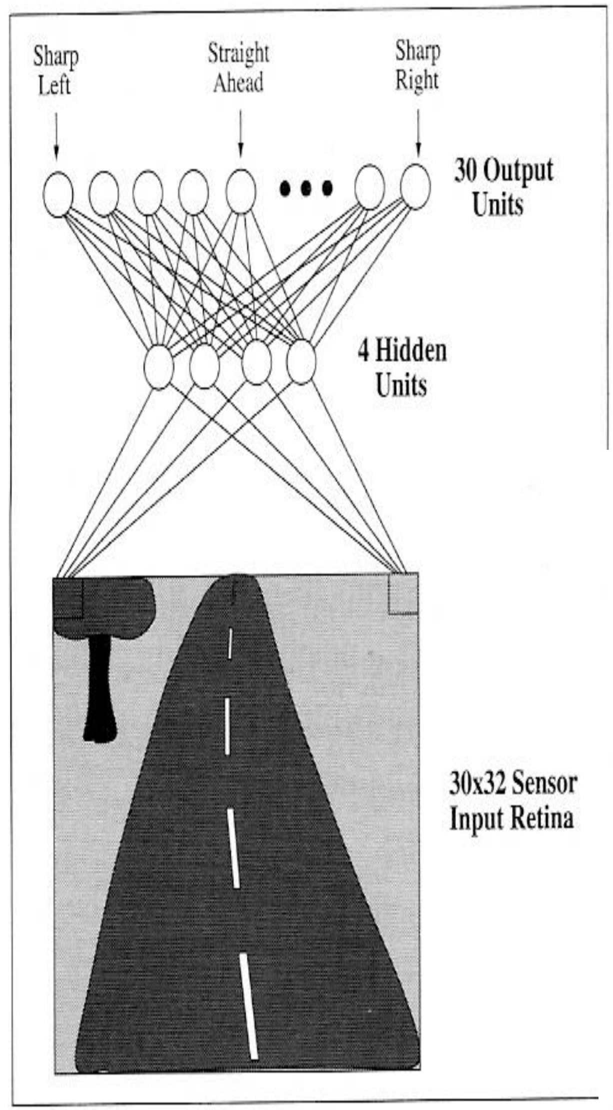
Pattern Recognition (cont.)

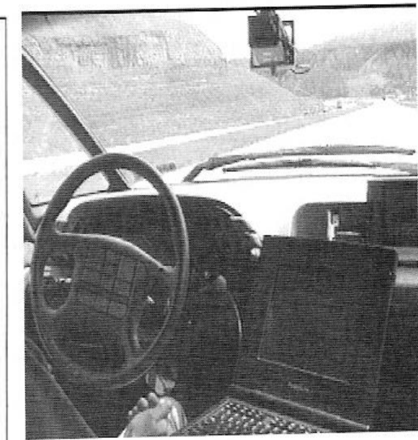


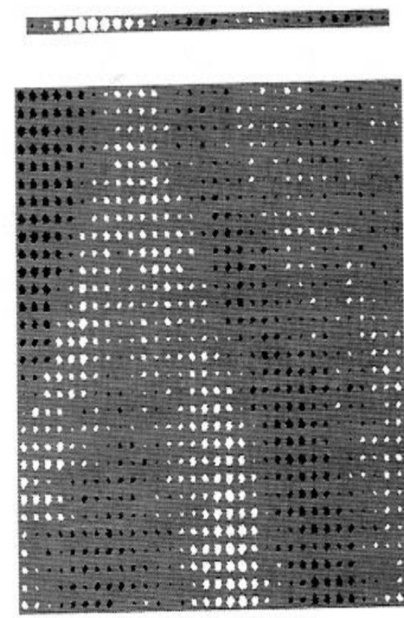
- Here the top row is 2 errors away from a ‘T’ and 3 errors away from an H. So the top output is a black.
- The middle row is 1 error away from both T and H, so the output is random.
- The bottom row is 1 error away from T and 2 away from H. Therefore the output is black.
- Since the input resembles a ‘T’ more than an ‘H’ the output of the network is in favor of a ‘T’.

Example

- ALVINN uses a learned ANN to steer an autonomous vehicle driving at normal speeds on public highways
  - Input to network: 30x32 grid of pixel intensities obtained from a forward-pointed camera mounted on the vehicle
  - Output: direction in which the vehicle is steered
  - Trained to mimic observed steering commands of a human driving the vehicle for approximately 5 minutes







Issues to be discussed

- Initializing the weights.
- Use of a learning algorithm.
- Set of training examples.
- Encode the examples as inputs.
- Convert output into meaningful results.