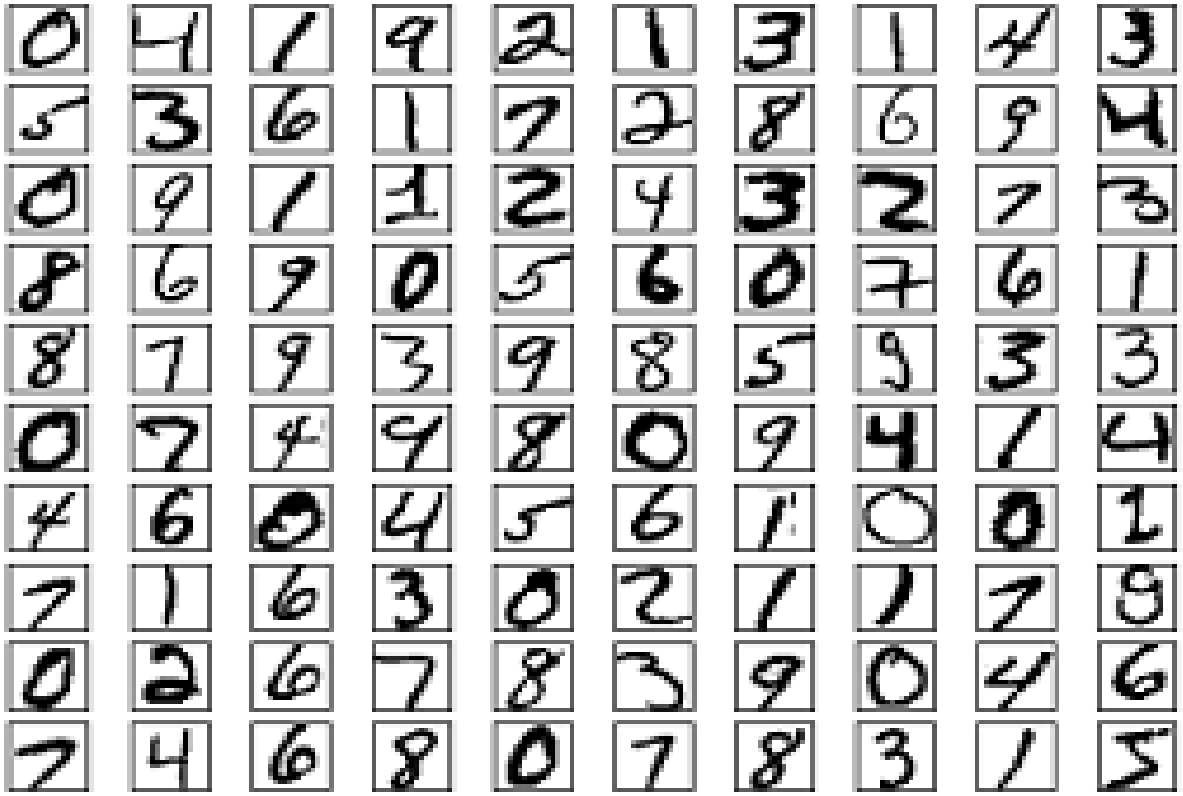
Artificial Neural Network

Neural Network

- Consider the following sequence of handwritten digits:

- Most people effortlessly recognize those digits as 504192.
- What happens if you attempt to write a program to recognize digits like those above????

Neural Network



Training Examples

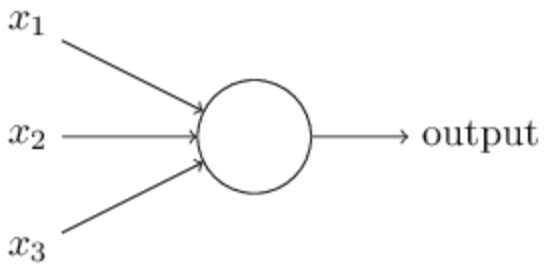
But this short program can recognize digits with an accuracy over 96 percent, without human intervention.

Neural Network

- A neural network is a model of reasoning based on the human brain.
- The human brain consists of interconnected set of nerve cells or basic information processing unit called neuron.

Perceptron

- *perceptron* – A simple ANN which consists of a single neuron.



- A perceptron takes several binary inputs, *x*1, *x*2, …, and produces a single binary output:

Perceptron

- *weights*, *w*1, *w*2,…, real numbers expressing the importance of the respective inputs to the output.
- The neuron's output, 0 or 1, is determined by whether the weighted sum $\sum_j w_j x_j$ is less than or greater than some *threshold value*.
- Just like the weights, the threshold is a real number which is a parameter of the neuron. To put it in more precise algebraic terms:

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \qquad (1)$$

- A way you can think about the perceptron is that it's a device that makes decisions by weighing up evidence.

Perceptron

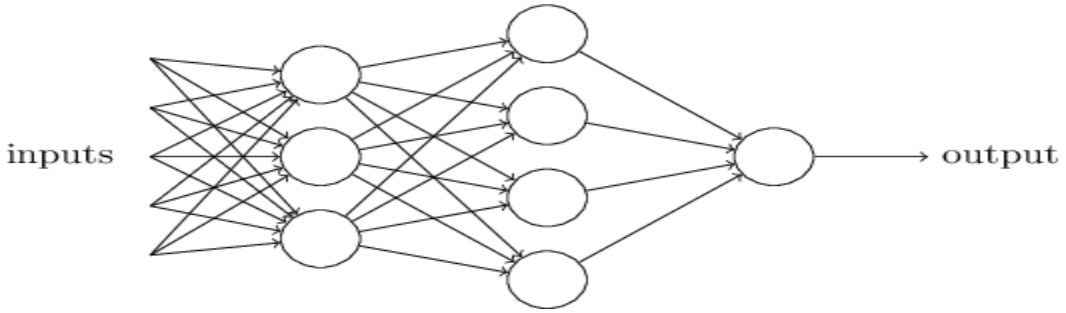
- Suppose the weekend is coming up, and you've heard that there's going to be a pizza festival in your city. You like pizza, and are trying to decide whether or not to go to the festival. You might make your decision by weighing up three factors:
- Is the weather good?
- Do your friends want to go too?
- Is the festival near public transit? (You don't own a car).
- We can represent these three factors by corresponding binary variables $x1$,$x2$, and $x3$.
- $x1=1$ if the weather is good, and $x1=0$ if the weather is bad.
- Similarly, $x2=1$ if your friends want to go, and $x2=0$ if not.
- And similarly again for $x3$ (public transit).

Perceptron

- Say, $w1=6$ for the weather, and $w2=2$ and $w3=2$ for the other conditions.
- The larger value of $w1$ indicates that the weather matters a lot to you, much more than whether your friends join you, or the nearness of public transit.
- Finally, suppose you choose a threshold of 5 for the perceptron.
- With these choices, the perceptron implements the desired decision-making model.
- It makes no difference to the output whether your friends want to go, or whether public transit is nearby.
- By varying the weights and the threshold, we can get different models of decision-making.

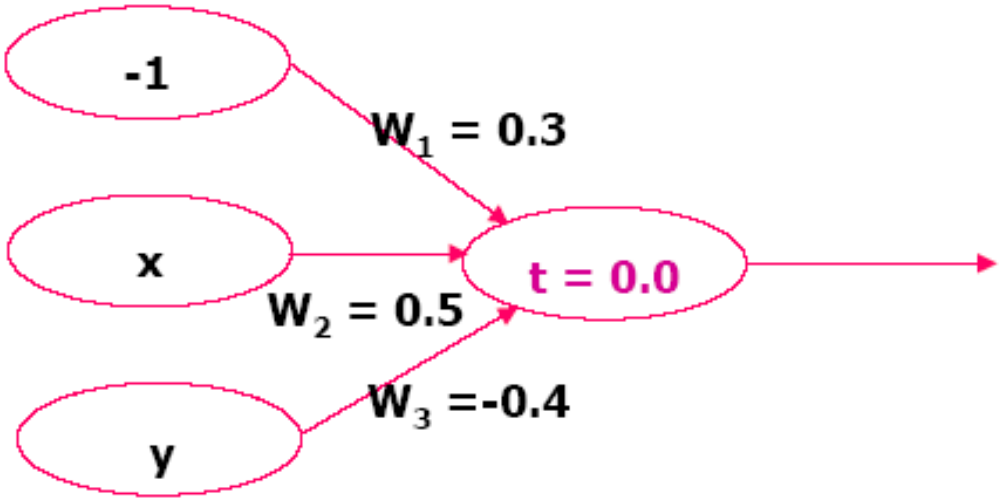
Multilayer Perceptrons

- In this network, the first column of perceptrons - what we'll call the first *layer* of perceptrons - is making three very simple decisions, by weighing the input evidence.



- Each of the perceptrons in 2nd layer is making a decision by weighing up the results from the first layer of decision-making.
- In this way a perceptron in the second layer can make a decision at a more complex and more abstract level than perceptrons in the first layer.
- And even more complex decisions can be made by the perceptron in the third layer.

How perceptron learn?



For AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

I_1	I_2	I_3	Summation	Output
-1	0	0	(-1*0.3) + (0*0.5) + (0*-0.4) = -0.3	0
-1	0	1	(-1*0.3) + (0*0.5) + (1*-0.4) = -0.7	0
-1	1	0	(-1*0.3) + (1*0.5) + (0*-0.4) = 0.2	1
-1	1	1	(-1*0.3) + (1*0.5) + (1*-0.4) = -0.2	0

How perceptron learn?

- Make small adjustment in the wights to reduce the difference between actual and desired output.
- The initial weights are randomly assigned in between [-0.5, 0.5]

Notation for ANN

Target value: When training the net is required to provide with the desired output as well as input. In case of AND function the target value will be 1 for [1,1]

Output (y): The output value from the neuron

I_i: Inputs being presented to the neuron

w_j: Weight from the input neuron to the output neuron

LR: The learning rate. This dictates how quickly the network converge. It is set by a matter of experimentation. It is typically 0.1.

- Error = Target – Output
- If Error < > 0 then
- ΔW_j = new W_j - old W_j = LR * I_j * Error
- new W_j = old W_j + ΔW_j

Learning Algorithm Example

- To Learn AND.
- Given that initial weight 0,0.4, threshold =0.3, Learning rate=0.25,

w_1	w_2	θ	x_1	x_2	$\Sigma w_i x_i$	y	t	Δw_1	Δw_2
0.0	0.4	0.3	0	0	0	0	0	0	0

Artificial Neural Network

Contents

What is ANN?

A neural network can be defined as a model of reasoning based on the human brain.

The brain consists of a densely interconnected set of nerve cells (information processing units) called neurons.

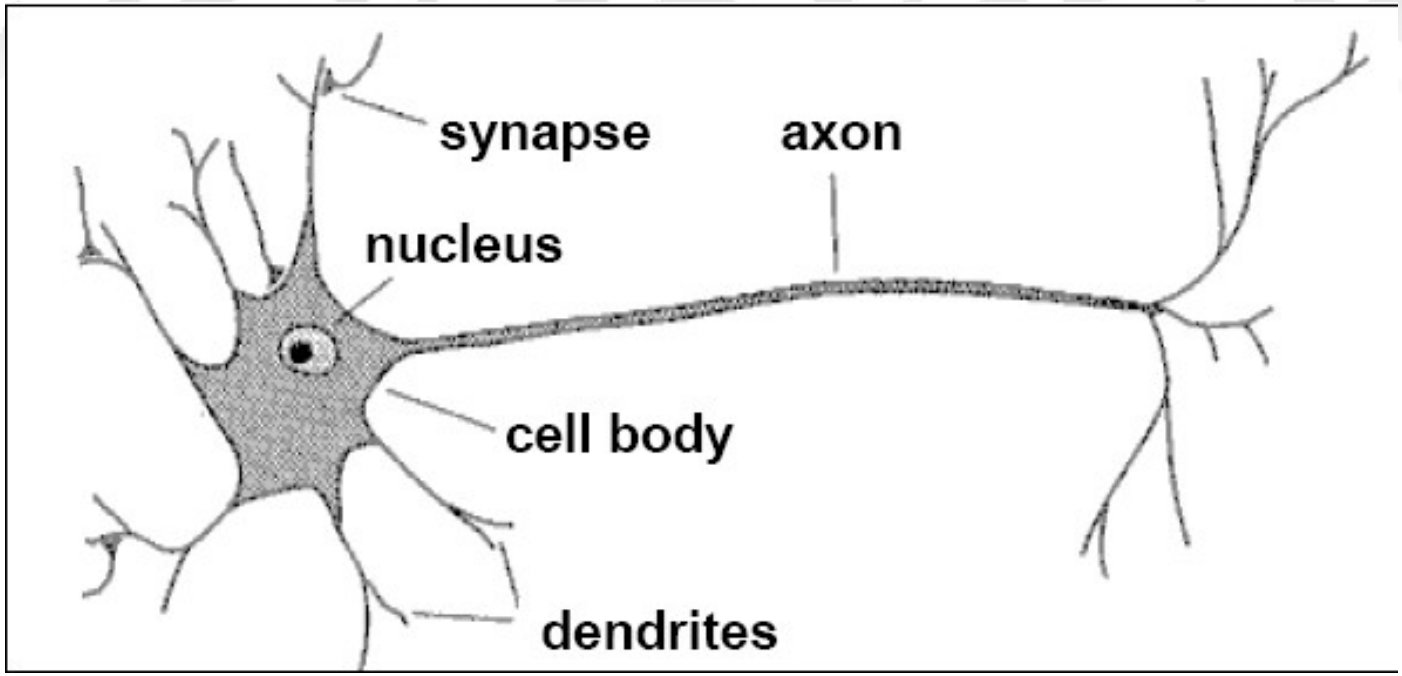
Human brain has 10 billion neurons and 60 trillion connections.

ANN's are a type of artificial intelligence that attempts to imitate the way a human brain works.

The approach is beginning to prove useful in certain areas that involve recognizing complex patterns,

such as voice recognition and image recognition.

Biological Neuron



A neuron has a cell body, a branching input structure and a branching output structure (the axon)
Axons connect to dendrites via synapses.
Electro-chemical signals are propagated from the dendrites input, through the cell body, and down the axon to other neurons

Structure of Neuron

A neuron only fires if its input signal exceeds a certain amount (the threshold) in a short time period.

Synapses vary in strength

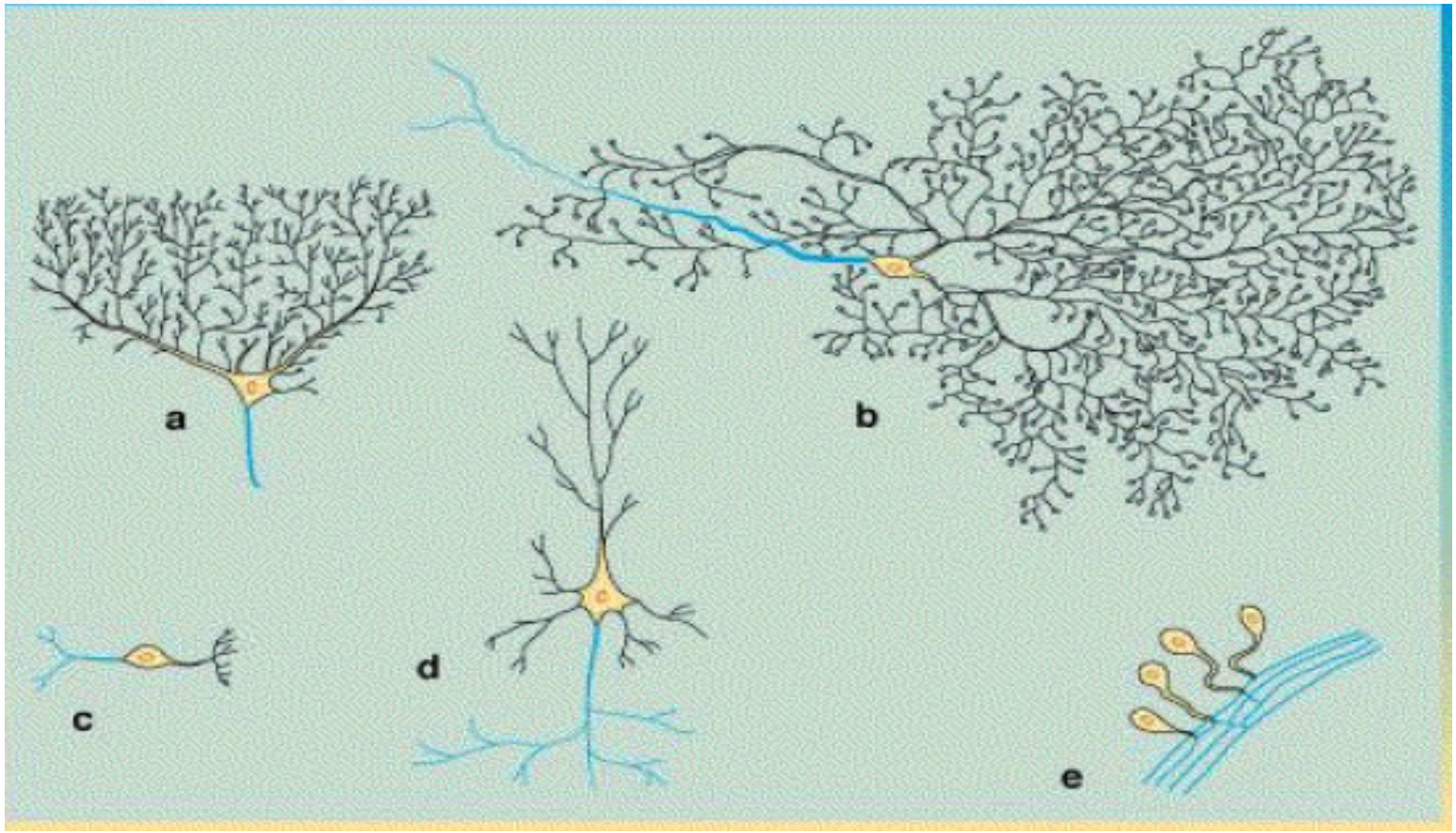
Good connections allowing a large signal

Slight connections allow only a weak signal.

Synapses can be either excitatory or inhibitory.

Types of Neuron

Neuron comes in many shapes and sizes



Models of Neuron

Neuron is an information processing unit

A set of synapses or connecting links

–characterized by weight or strength

An adder

–summing the input signals weighted by synapses

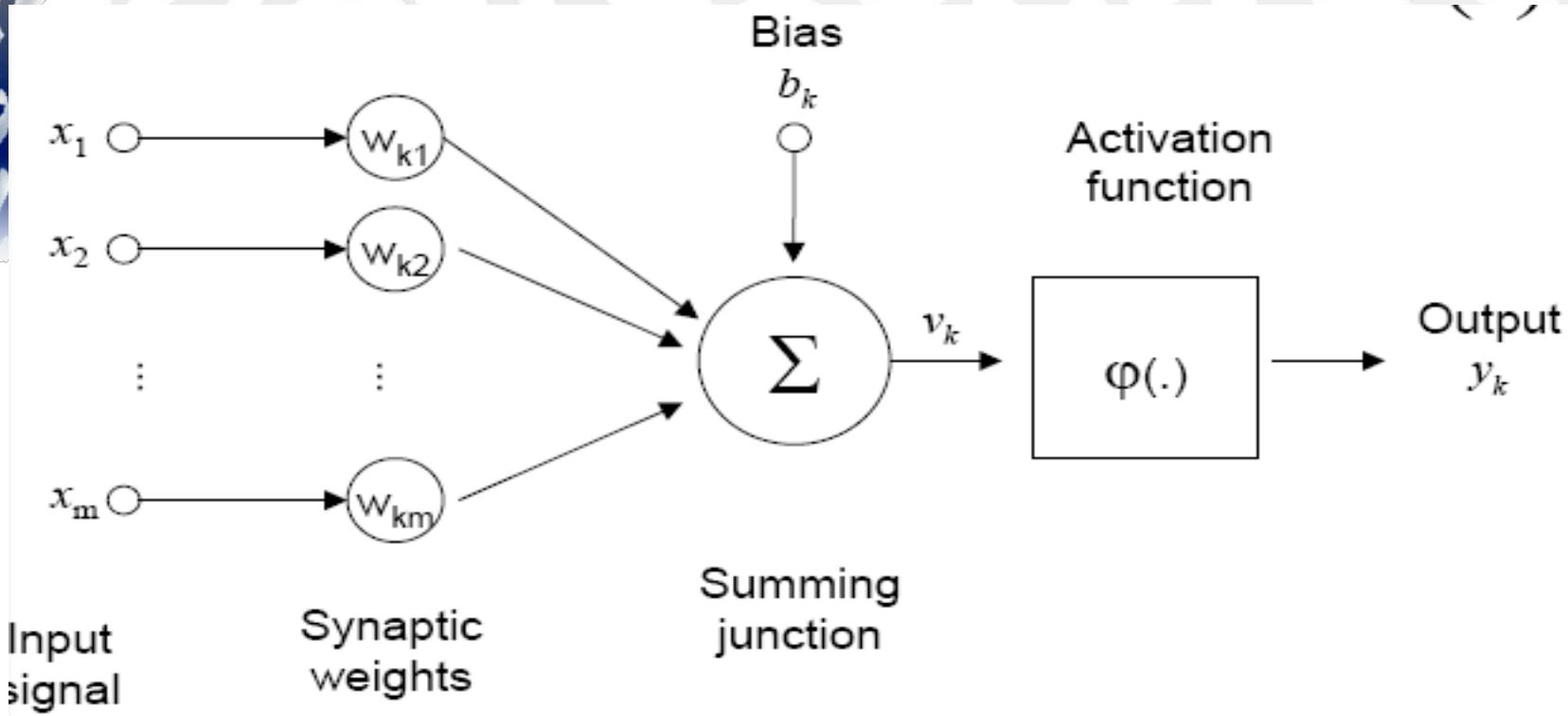
–a linear combiner

An activation function

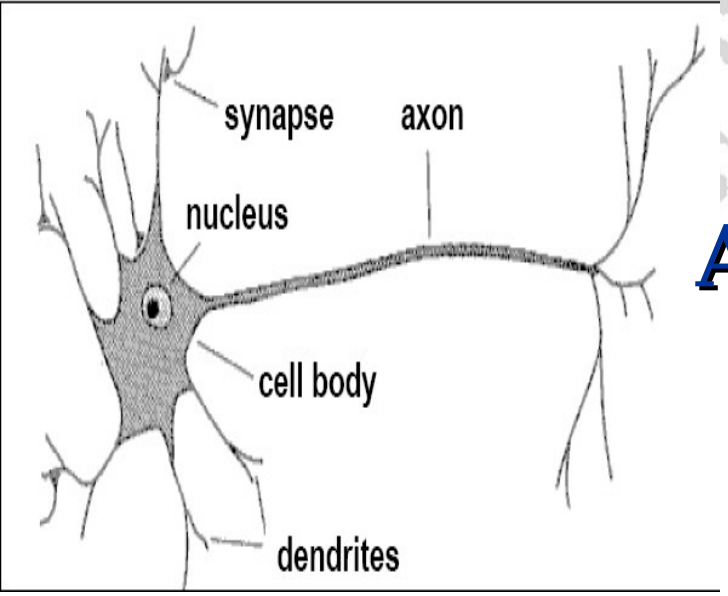
–also called squashing function

-squash (limits) the output to some finite values

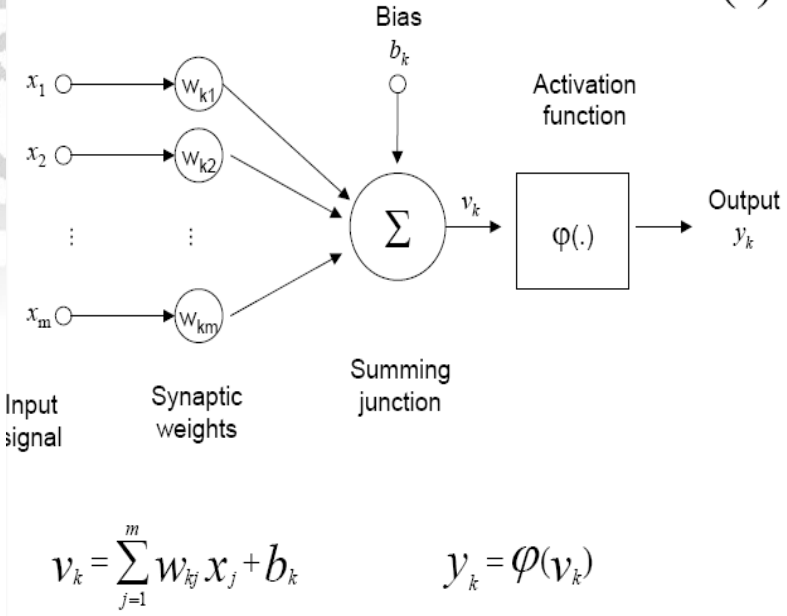
Model of Neuron



$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \qquad y_k = \varphi(v_k)$$



Analogy



$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \qquad y_k = \varphi(v_k)$$

- Inputs represent synapses
- Weights represent the strengths of synaptic links
- Wire presents dendrite secretion
- Summation block represents the addition of the secretions
- Output represents axon voltage

Explanation

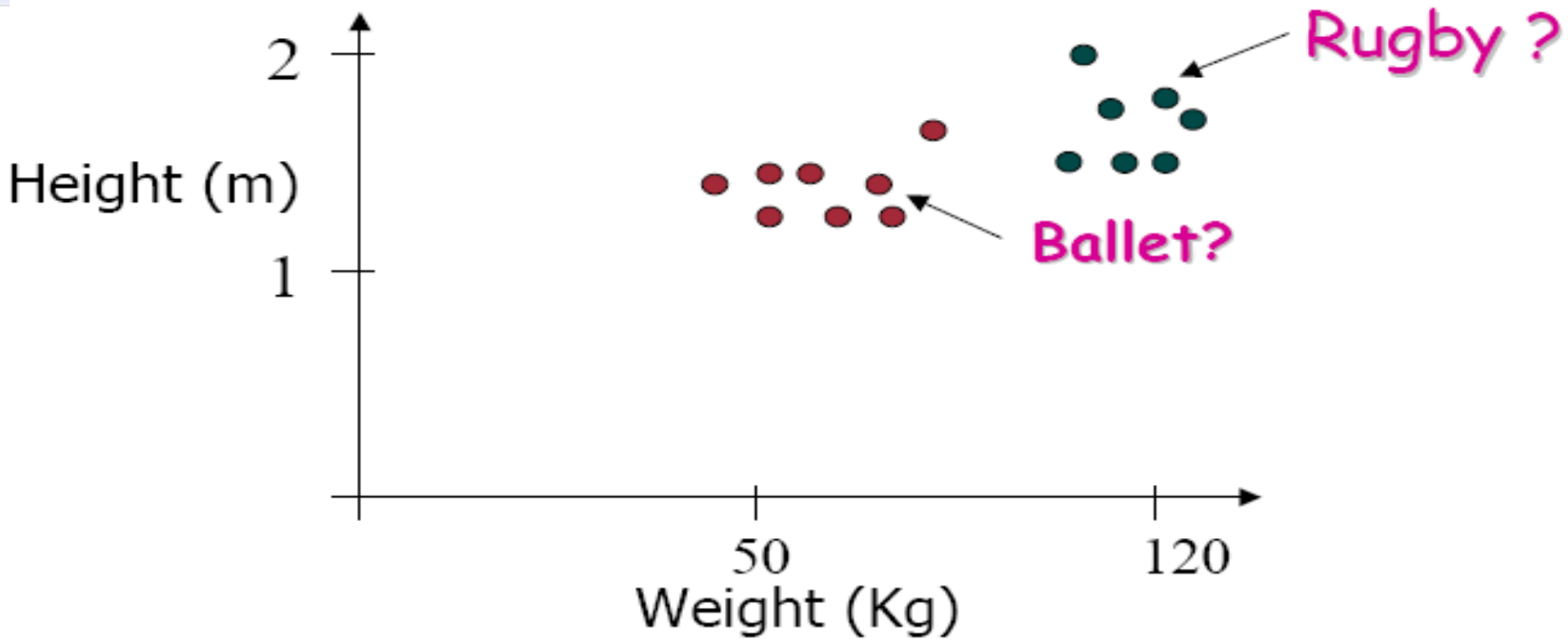
Neural Networks use a set of processing elements (or nodes) loosely analogous to neurons in the brain.

These nodes are interconnected in a network that can then identify patterns in data as it is exposed to the data.

In a sense, the network learns from experience just as people do (case of supervised learning).

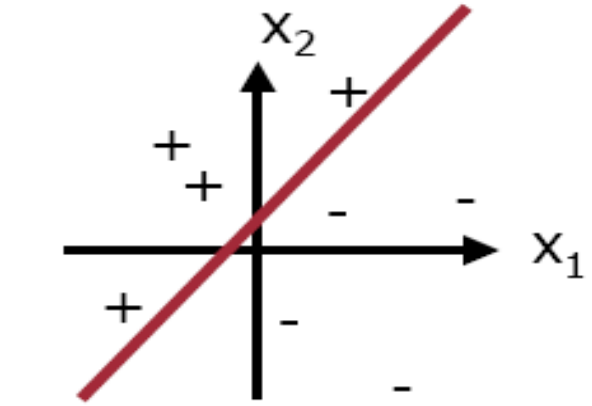
This distinguishes neural networks from traditional computing programs, that simply follow instructions in a fixed sequential order.

Decision Boundary

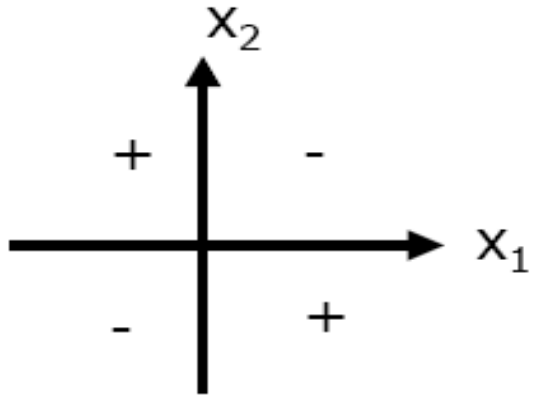


Decision Boundary

Divide feature space by drawing a hyper-plane across it

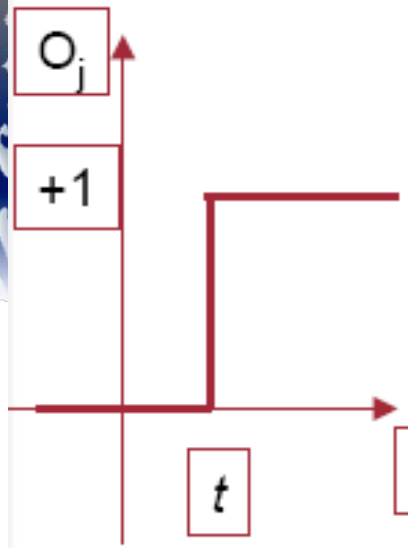


Linearly separable



Non-Linearly separable

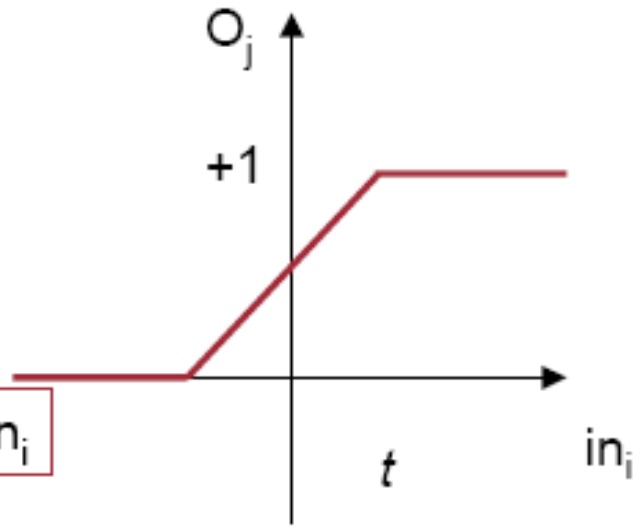
Activation Function



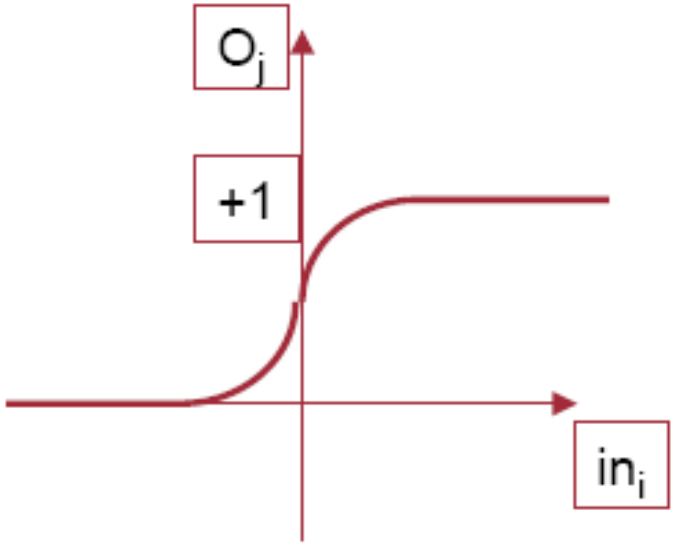
The hard-limiting Threshold Function

Corresponds to the biological paradigm: either fires or not

$$O(in) = \begin{cases} 1 & in > t \\ 0 & in \le t \end{cases}$$



Piecewise-linear Function

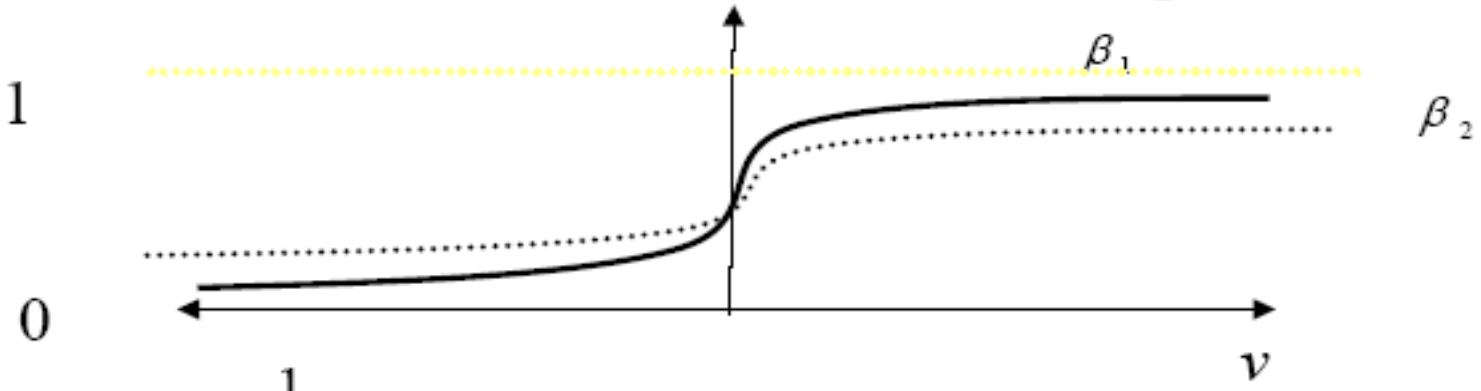


Sigmoid Function (differentiable) ('S'-shaped curves)

$$\varphi(v) = \frac{1}{1+\exp(-av)}$$

a is slope parameter

Sigmoid function



$$\varphi_\beta(v) = \frac{1}{1+\exp(-\beta v)}$$

$$if \quad (i) \quad \beta v \to \infty \quad then \quad \varphi_\beta(v) \to 1$$

$$(ii) \quad \beta v \to -\infty \quad then \quad \varphi_\beta(v) \to 0$$

$$(iii) \begin{cases} \beta \to \infty \\ and \quad v \quad fixed \end{cases} \quad then \quad \varphi_\beta(v) \to 1(v)$$

1(v) is the modified Heaviside function

How Does the Neuron Determine its Output?

The neuron computes the weighted sum of the input signals and compares the result with a threshold value of, T_h

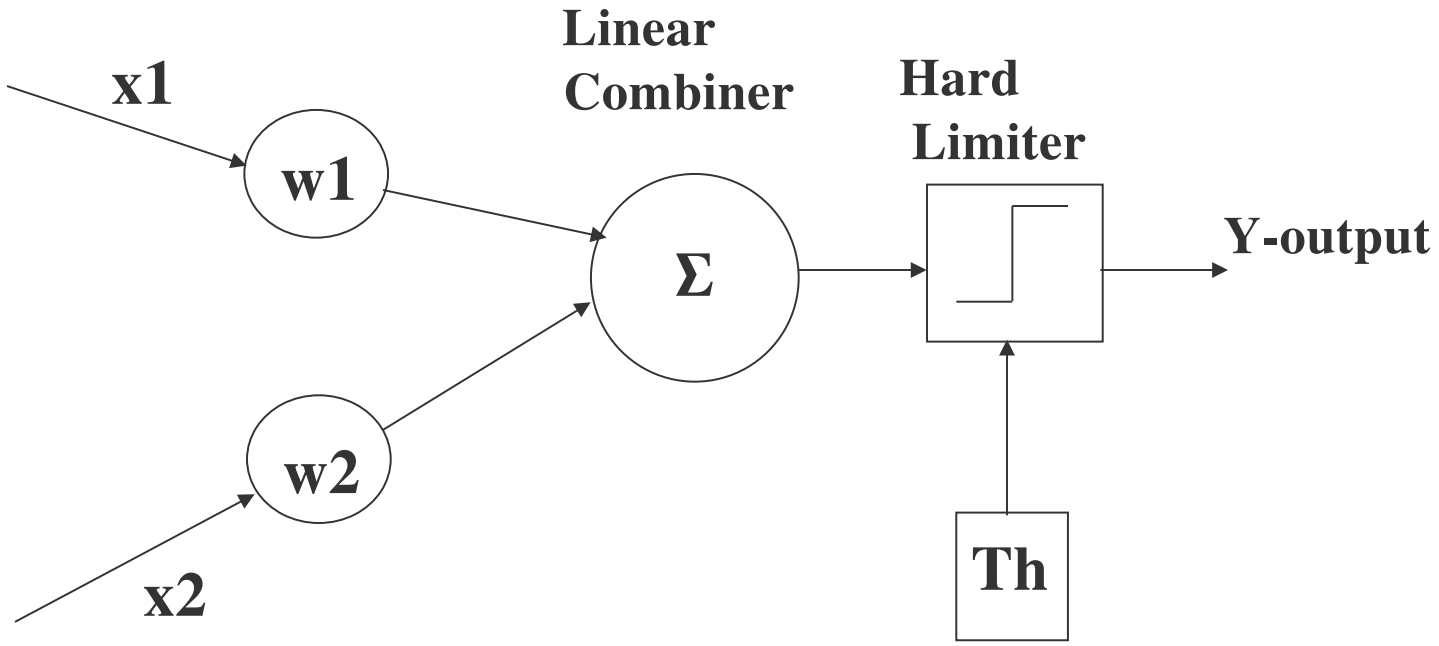
If the net weighted input is less than the threshold the neuron output is –1.

If the net weighted input is greater than or equal to the threshold, the neuron becomes activated and its output attains a value +1

(This type of activation function is called a sign function)

Example of NN: The Perceptron

Single neuron with adjustable synaptic weight and a hard limiter.



$$\sum_{i=1}^{n} x_i w_i \quad Th \quad 0$$

Step & sign activation function called hard limit functions.

How Does the Perceptron Learn?

Step 1: Initialization

Set the initial weights $w_1, w_2, \ldots w_n$ and Threshold-Th

Step 2: Activation

Active the perceptron by applying inputs $x_1(p)$, $x_2(p)$….. $x_n(p)$ and desired output $Y_d(p)$. Where p iteration, n number of inputs

Step 3: Weight Training

Update the weight of the perceptron.

$$w_i(p \quad 1) \quad w_i(p) \qquad x_i(p) \quad e(p)$$

Step 4: Iteration

Increase iteration p by one, go back to step 2 and repeat the process.

Train a Perceptron to Perform Logical AND operation (1)

Epoch	Inputs		Y(d)	Initial Weight		Output Y	Error e	Final Weight	
	x1	x2	Yd	w1	w2			w1	w2
1	0	0	0	0.3	-0.1	0	0	0.3	-0.1
	0	1	0	0.3	-0.1	0	0	0.3	-0.1
	1	0	0	0.3	-0.1	1	-1	0.2	-0.1
	1	1	1	0.2	-0.1	0	1	0.3	0.0
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	-1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0
Threshold=0.2, Learning rate = 0.1									

Train a Perceptron to Perform Logical AND operation (2)

Epoch	Inputs		Y(d)	Initial Weight		Output Y	Error	Final Weight	
	x1	x2	Yd	w1	w2			w1	w2
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	-1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	-1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
Threshold=0.2, Learning rate = 0.1									

Train a Perceptron to Perform Logical AND operation (3)

Epoch	Inputs		Y(d)	Initial Weight		Output Y	Error	Final Weight	
	x1	x2	Yd	w1	W2			w1	w2
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
Threshold=0.2, Learning rate = 0.1									

Recommended Textbooks

[Negnevitsky, 2001] M. Negnevitsky " Artificial Intelligence: A guide to Intelligent Systems", Pearson Education Limited, England, 2002.

[Russel, 2003] S. Russell and P. Norvig Artificial Intelligence: A Modern Approach Prentice Hall, 2003, Second Edition

[Patterson, 1990] D. W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", Prentice-Hall Inc., Englewood Cliffs, N.J, USA, 1990.