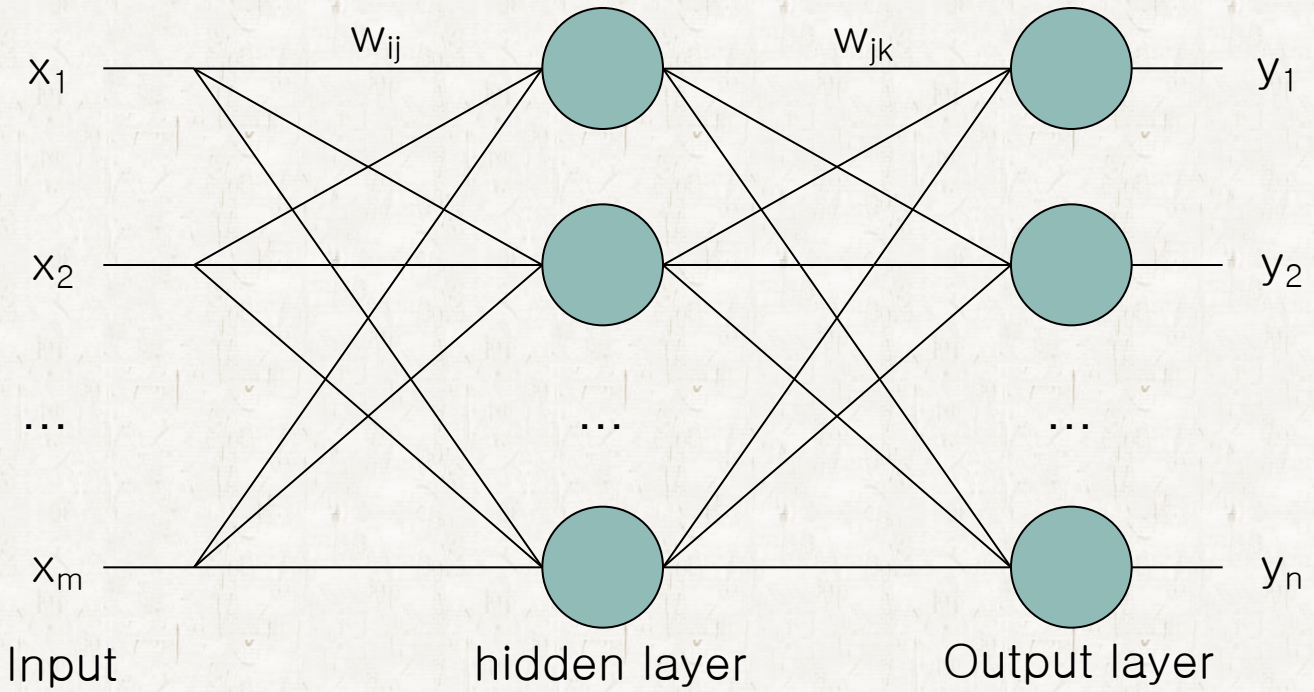
Neural Networks

J.H. Lee, Dept. of Software, Sungkyunkwan Univ.

Introduction (1)

- Neural Network



J.H. Lee, Dept. of Software, Sungkyunkwan Univ.

Introduction (2)

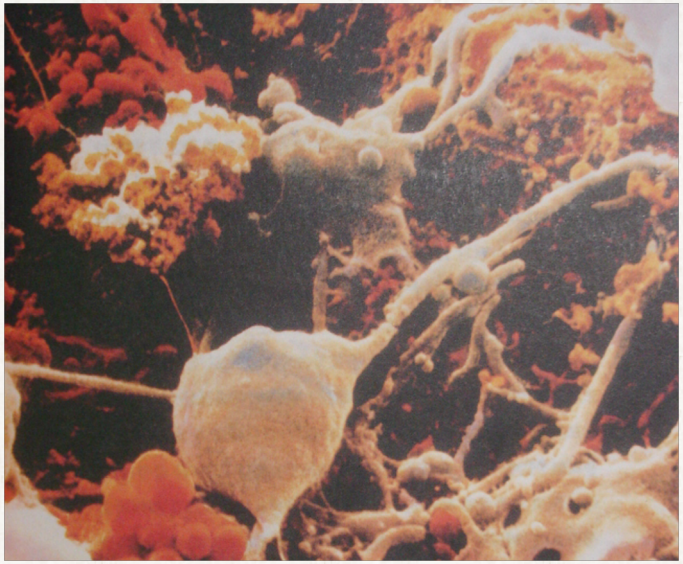
- Artificial Neural Network
  - AI tools based on biological brains
  - It can learn anything!!
- Types of Artificial Neural Network
  - Multilayer perceptron
  - Kohonen's Self-Organizing Neural Networks
  - ..
- Other names of Multilayer Perceptron
  - Feed-forward Neural Network
  - Multilayer Feed-forward Neural Network

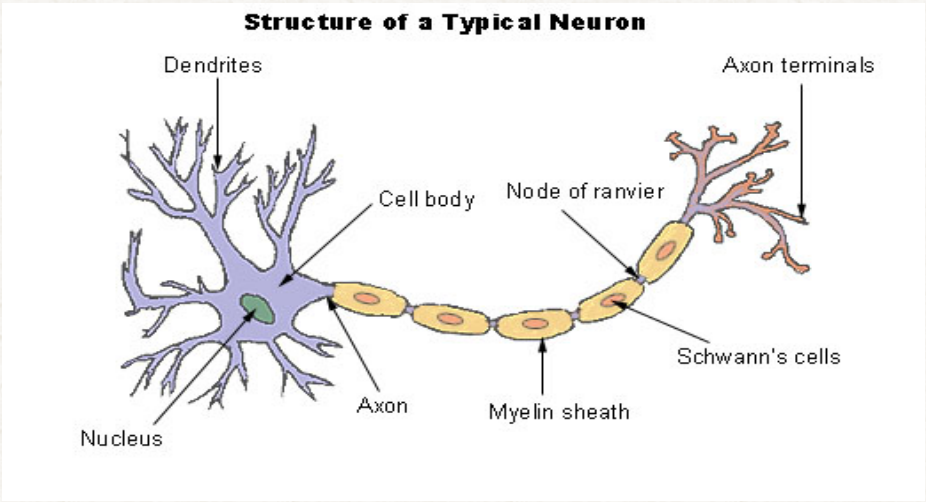
Introduction (3)

- Brain
  - There are about 10^{11} neurons (brain cells)

Introduction (4)

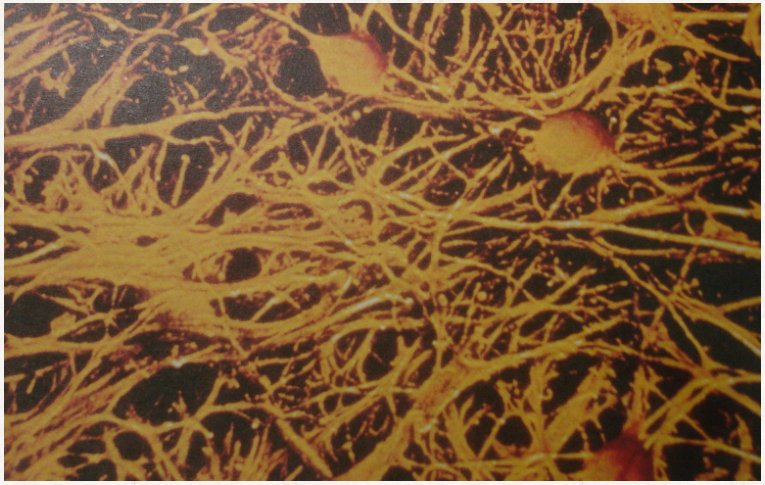
- Neurons
  - Basic components of brain
  - Shape
    - Cell body: main process unit
    - Dendrite: input gates (numerous dendrites per neuron)
    - Axon: output gates (one per neuron)

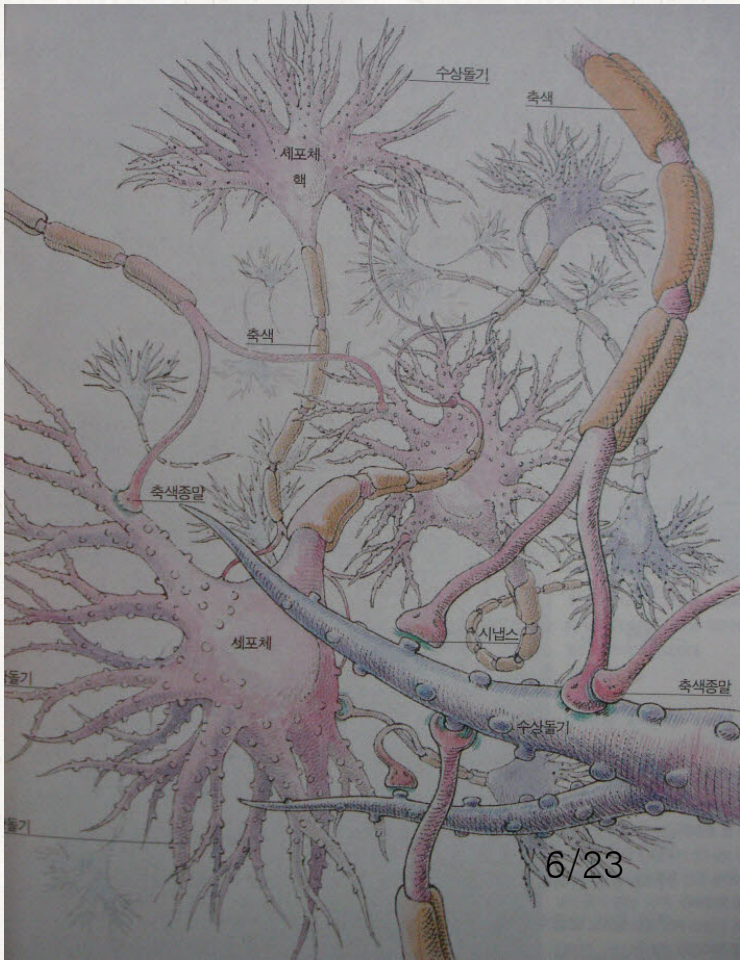




Introduction (5)

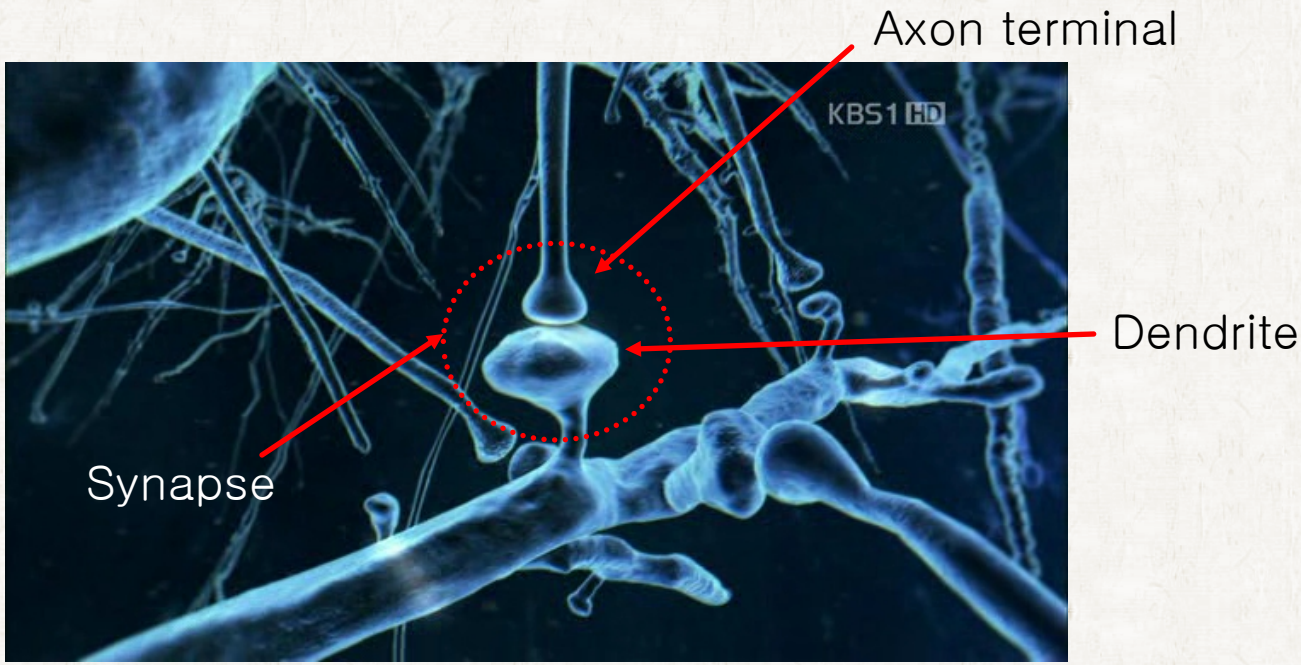
- Neurons in brain
  - Every neuron connects to 10^3 to 10^4 other neurons
  - A brain is a network of neurons
  - Neurons just transmit signals to neighboring neurons!!





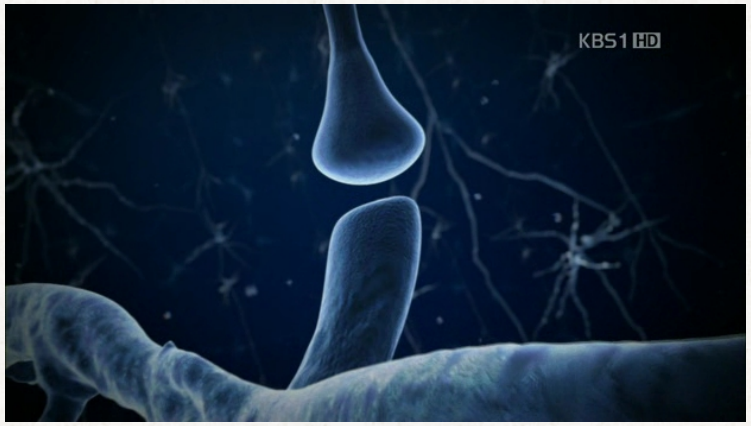
Introduction (6)

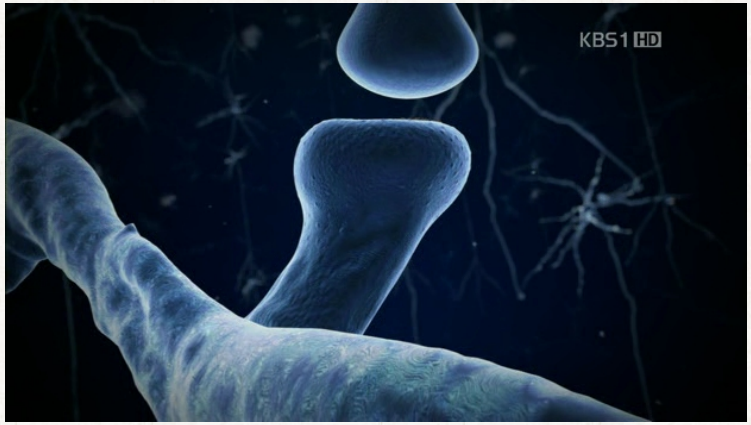
- Connection between neuron
  - Structure
    - Synapse : Connection spot
    - Axon terminal : release neurotransmitter
    - Dendrite : receive neurotransmitter



Introduction (7)

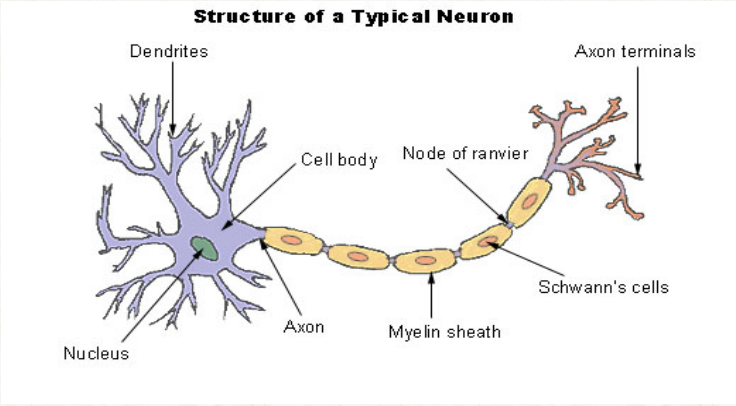
- Connection between neuron
  - Every connection does not has the same effect
  - Each connection has different strength
    - The more receptor a dendrite has (mushroom shape), the better it receives neurotransmitter

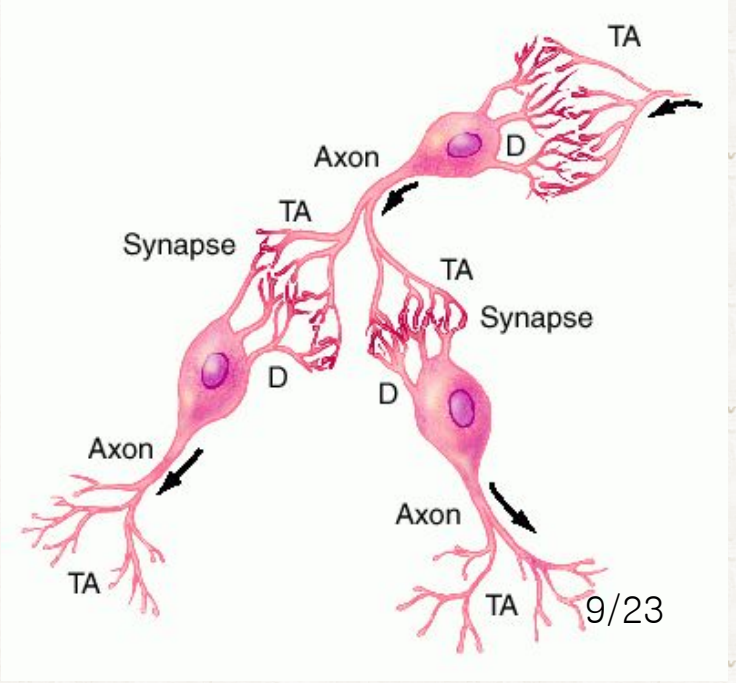




Introduction (8)

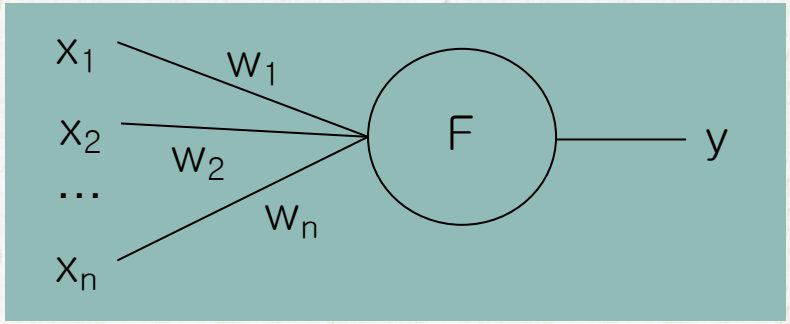
- Function of neurons
  - Input
    - Input signals coming from dendrites
    - Signals are amplified
  - Reservoir
    - Cell body reservoirs the signals
  - Output
    - If the amount of reserved signals in body cell is larger than a threshold, cell body releases a signal through axon

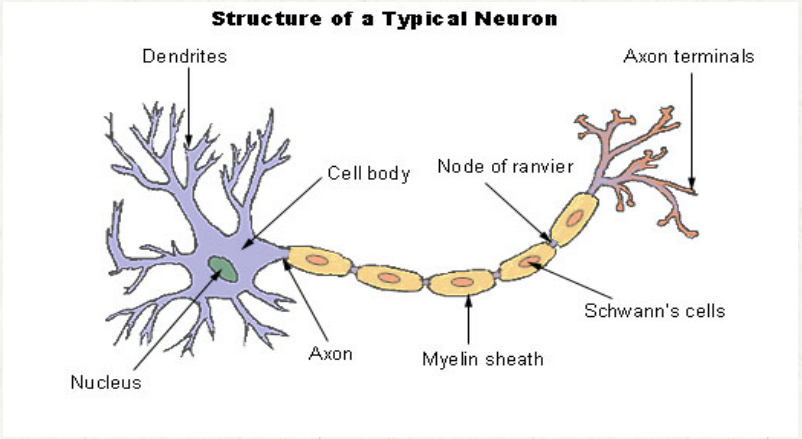




Simple Mathematical Model (1)

- Simple representation of neurons





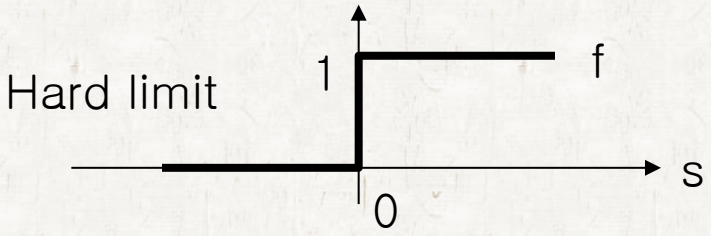
- x : dendrites (input)
- w : amount of receptors in each dendrite (connection strength)
- F : cell body
- y : axon (output)

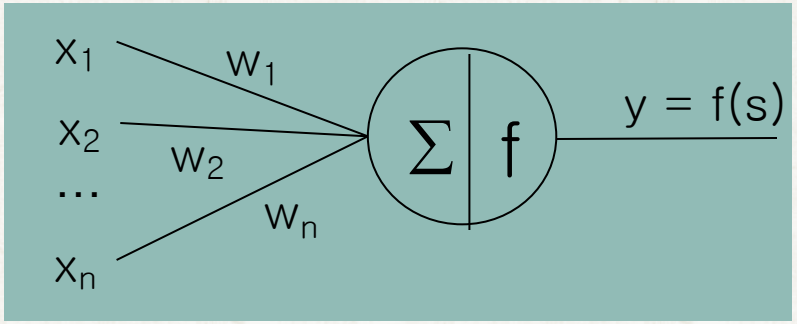
Simple Mathematical Model (2)

- Simple mathematical model of neurons-con'd
  - First function: Weighted summation of inputs

$$s = x_1w_1 + x_2w_2 + \ldots + x_nw_n$$

  - Second function: Non-linear threshold

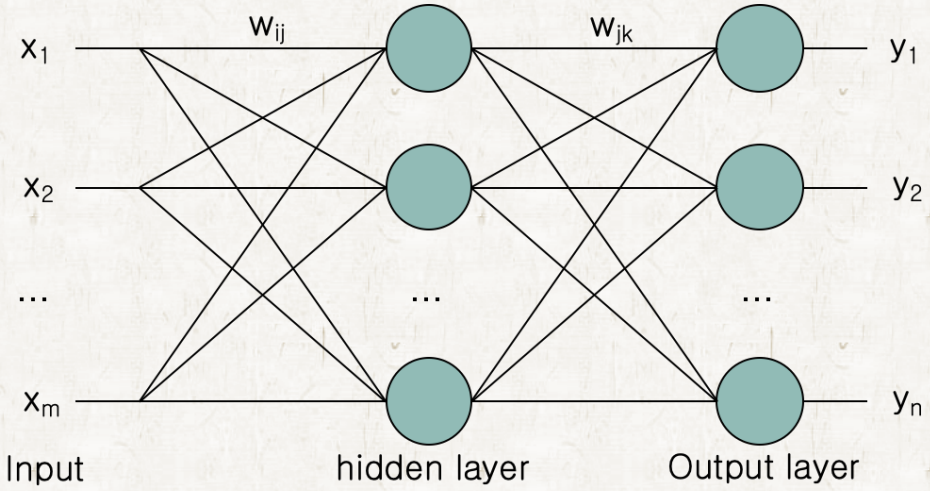


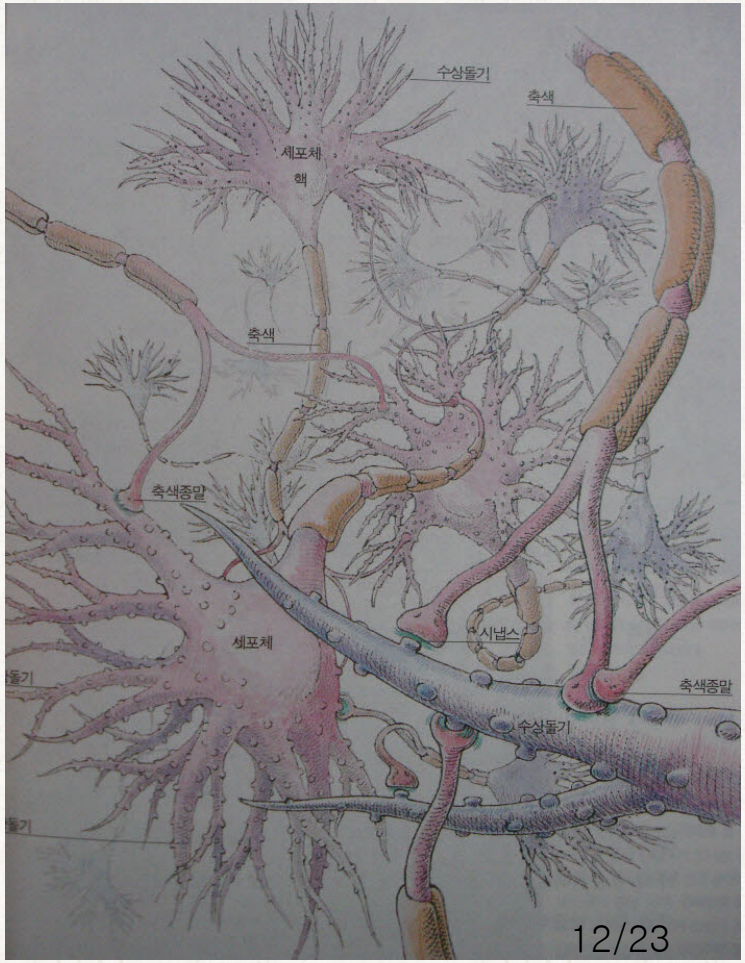


$$y = \begin{cases} 1 & \sum_{i=1}^{n} x_i w_i > 0 \\ 0 & otherwise \end{cases}$$

Simple Mathematical Model (3)

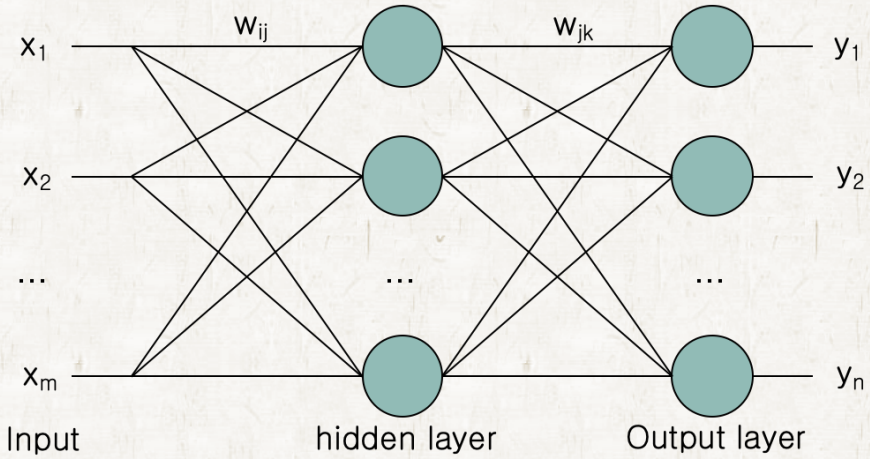
- Simple mathematical model of brains
  - Brain is a network of neurons
  - So, let's simply connects artificial neurons and call it artificial neural network





Simple Mathematical Model (4)

- Simple mathematical model of brains-con'd



- What a stupid it is!!

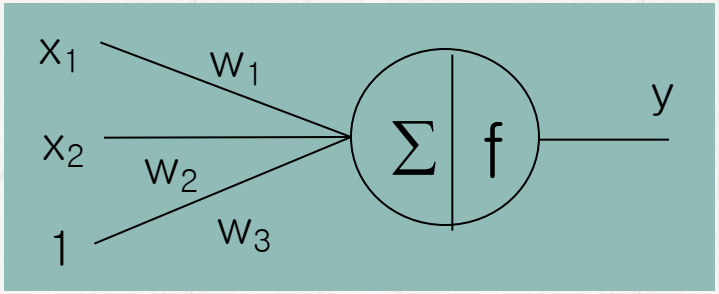
$$y = \begin{cases} 1 & \sum_{i=1}^{n} x_i w_i > 0 \\ 0 & otherwise \end{cases}$$

- What can it do?
  - Everything a Pentium can do!!

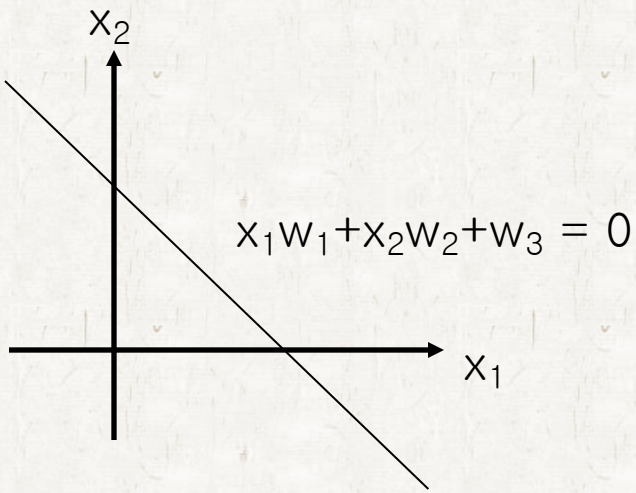
What a Perceptron Can Do

Perceptrons can solve linearly separable problems!!

- What a perceptron does



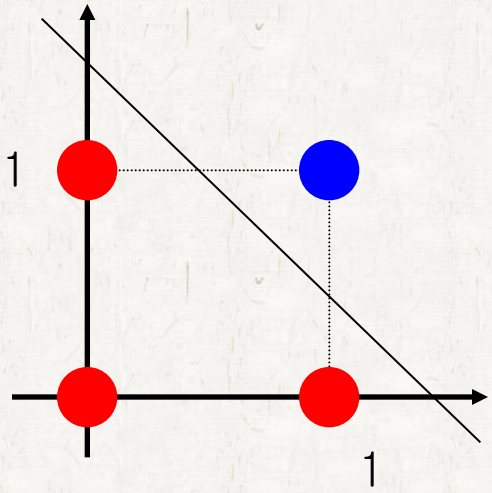
$$y = \begin{cases} 1 & \sum_{i=1}^{n} x_i w_i > 0 \\ 0 & otherwise \end{cases}$$

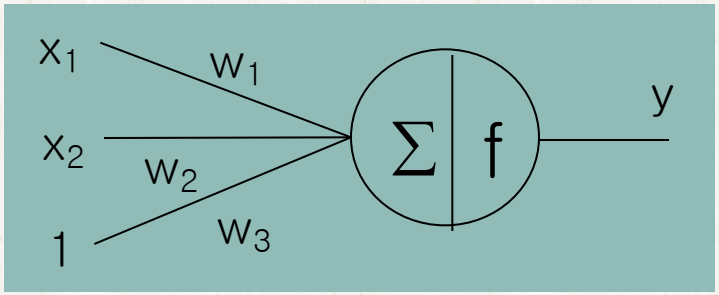


If an input is above the line
    output 1
else
    output 0

What a Perceptron Can Do? (2)

- What a perceptron can do
  - And operation



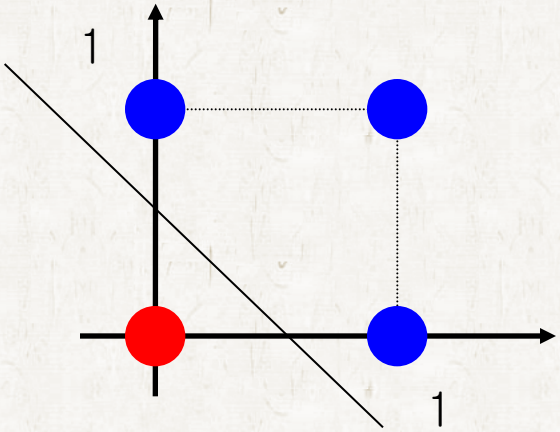


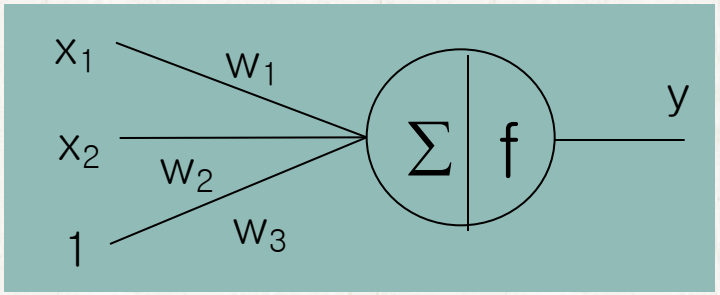
w1=1.0, w2=1.0, w3=−1.5

x_1	x_2	Σ	y
0	0	−1.5	0
0	1	−0.5	0
1	0	−0.5	0
1	1	0.5	1

What a Perceptron Can Do? (3)

- What a perceptron can do – con'd
  - OR operation



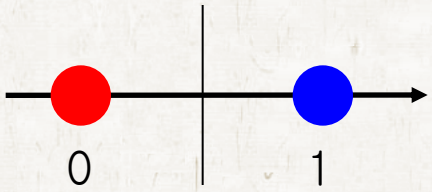


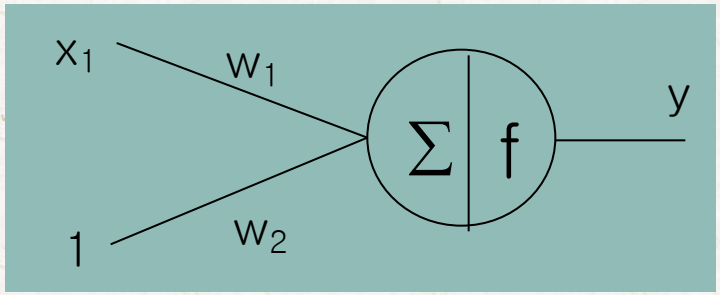
w1=1.0, w2=1.0, w3=−0.5

x_1	x_2	Σ	y
0	0	−0.5	0
0	1	0.5	1
1	0	0.5	1
1	1	1.5	1

What a Perceptron Can Do? (4)

- What a perceptron can do – con'd
  - NOT operation



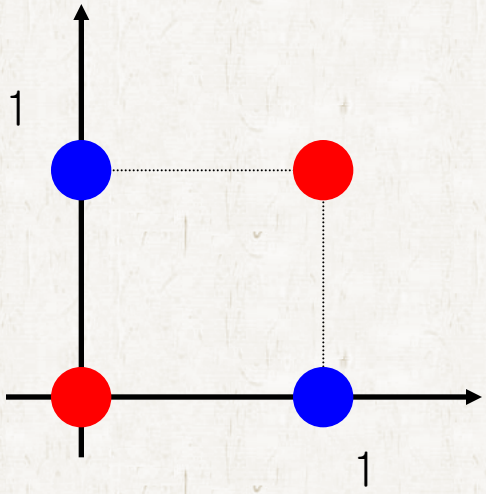


w1=−1.0, w2=0.5

x_1	Σ	y
0	0.5	1
1	−0.5	0

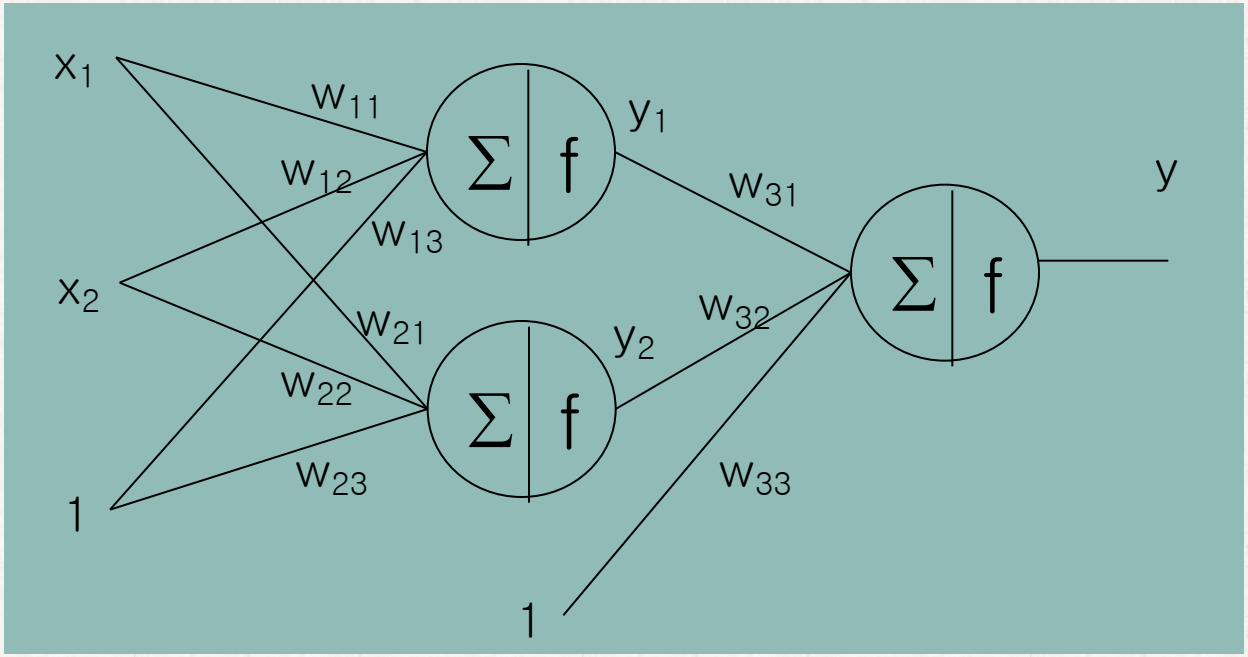
What a Neural Network Can Do? (1)

- What a neural network can do
  - A neural network can solve non-linearly separable problems
  - Example: XOR operation



What a Neural Network Can Do? (2)

- What a neural network can do– con'd
  - XOR operation



What a Neural Network Can Do? (3)

- What a neural network can do– con'd
  - XOR operation

$w_{11}=1.0,\ w_{12}=1.0,\ w_{13}=-1.5$

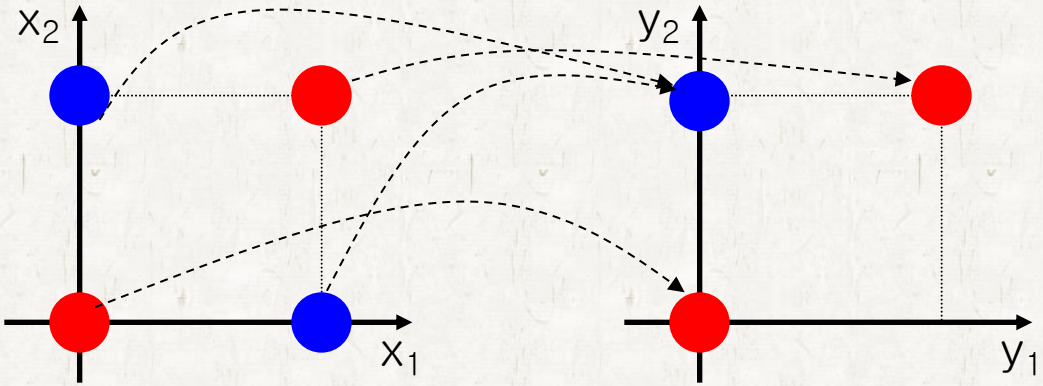
x_1	x_2	Σ	y_1
0	0	−1.5	0
0	1	−0.5	0
1	0	−0.5	0
1	1	0.5	1

$w_{21}=1.0,\ w_{22}=1.0,\ w_{23}=-0.5$

x_1	x_2	Σ	y_2
0	0	−0.5	0
0	1	0.5	1
1	0	0.5	1
1	1	1.5	1

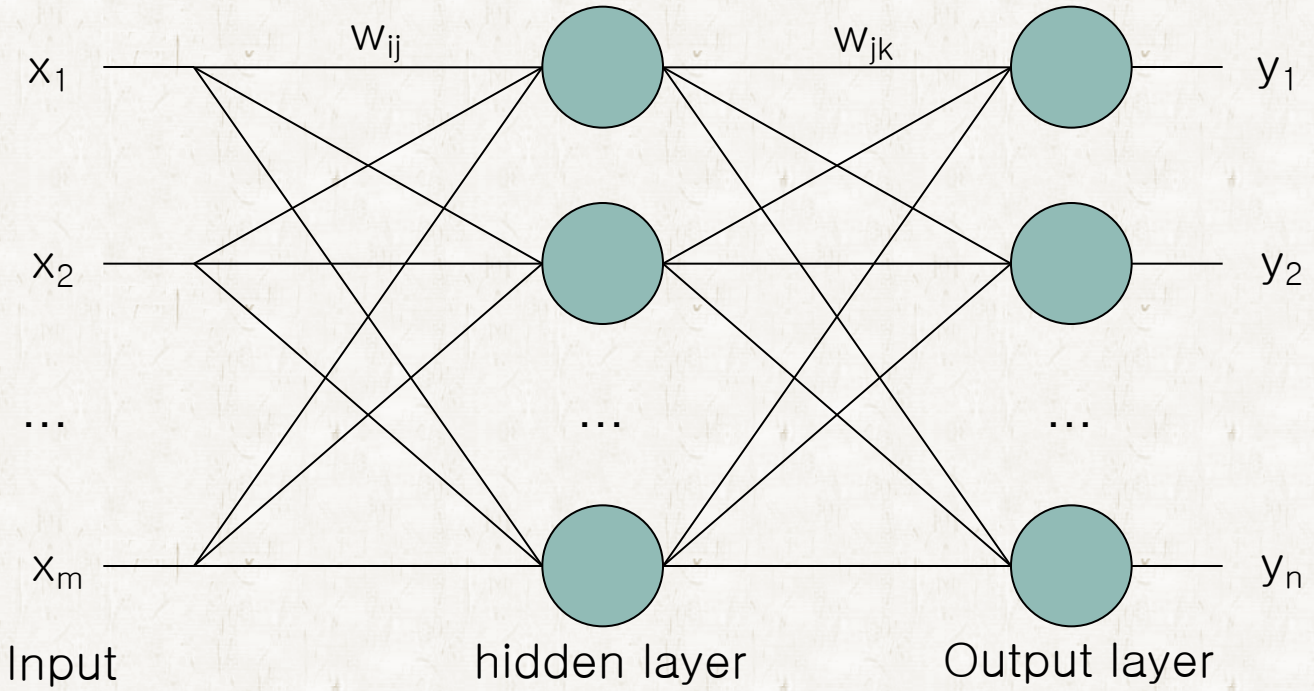
$w_{31}=-1.0,\ w_{32}=1.0,\ w_{33}=-0.5$

y_1	y_2	Σ	y
0	0	−0.5	0
0	1	0.5	1
0	1	0.5	1
1	1	−0.5	0



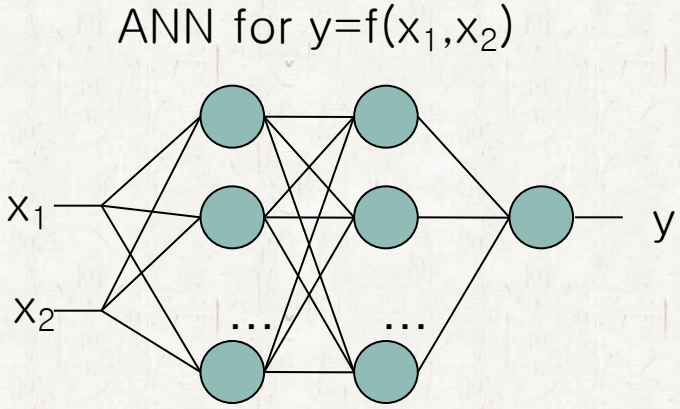
Multilayer Perceptron (1)

- Structure of Multilayer Perceptron
  - Here, we focus on a special type of neural networks
  - Layered structures



Multilayer Perceptron (2)

- Structure of Multilayer Perceptron – con'd
  - Input layer
    - Simply pass the input values to the next layer
    - # of nodes = # of inputs
  - Hidden layer
    - There can be several hidden layers
    - # of nodes should be given
  - Output layer
    - # of nodes = # of outputs



Multilayer Perceptron (3)

- What a multilayer perceptron can do – con'd
  - Anything digital computers can do
    - Boolean function: 2-layer perceptron
    - Continuous function: 2-layer perceptron
    - Arbitrary function: 3-layer perceptron
  - We can build a multi-layer perceptron which satisfies given input-output pairs