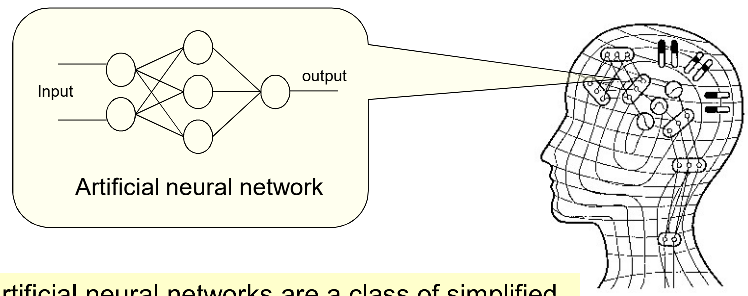
Artificial Neuron Models

A Simplified Brain Model



萩原将文「ニューロ・ファジイ・遺伝的アルゴリズム」、産業図書より

Artificial neural networks are a class of simplified brain models, imitating certain aspects of information processing in the brain, in a highly simplified way.

- Learning from experience (samples)
- Parallel, distributed computing

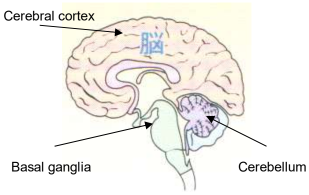
Brain and Computer

Comparison from a view point of information processing

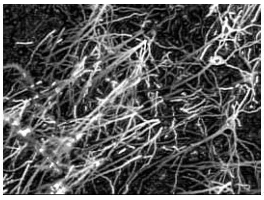
	Processing elements	Processing speed	Style of computation	Fault tolerant	learns	Intelligent, conscious
	synapses	100 Hz	Parallel, distributed	yes	yes	usually
	transistors	10^9 Hz	Serial, centralized	no	A little	Not (yet)

As a discipline of Artificial Intelligence, Neural Network attempt to bring computers a little closer to the brain's capabilities by imitating certain aspects of information processing in the brain, in a highly simplified way.

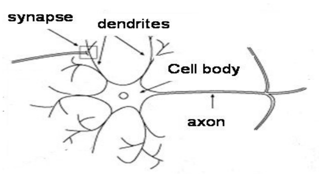
Neural Networks in Brain



The brain is not homogeneous. At the largest anatomical scale, it can be distinguished as cerebral cortex, basal ganglia, and cerebellum.



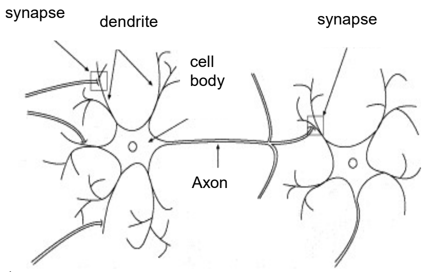
The human brain contains about 10 billion neurons. These neurons form very dense, complex local networks.



A typical neuron consists of *cell body*, *axon*, *dendrites* and *synapse*. With the synapses, a neuron connects with other neurons.

Behavior of a Typical Neuron

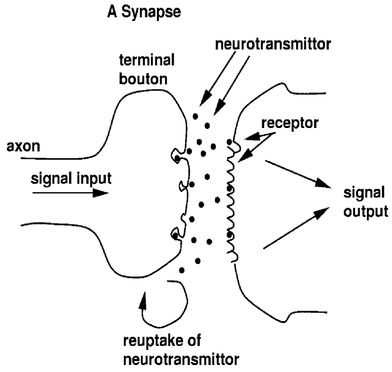
Depolarization



A neuron receives input from other neurons. Inputs sum (approximately). Once input exceeds a critical level, the neuron discharges a spike - an electrical pulse, to the next neuron(s) (or other receptors). This spiking event is also called depolarization, and is followed by a refractory period, during which the neuron is unable to fire.

Synapse Learning

Synapse learning

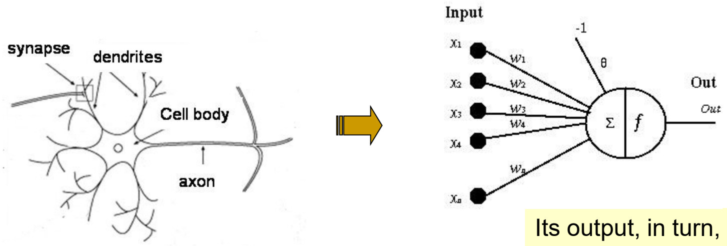


Transmission of an electrical signal from one neuron to the next is effected by neurotransmittors, chemicals which are released from the first neuron and which bind to receptors in the second. This link is called a synapse.

The extent to which the signal from one neuron is passed on to the next depends on many factors, e.g. the amount of neurotransmittor available, the number and arrangement of receptors, amount of neurotransmittor reabsorbed, etc.

McCulloch-Pitts Neuron Model

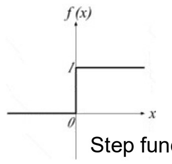
Proposed by McCulloch and Pitts in 1943



Its output, in turn, can serve as input to other units.

$$net = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n$$

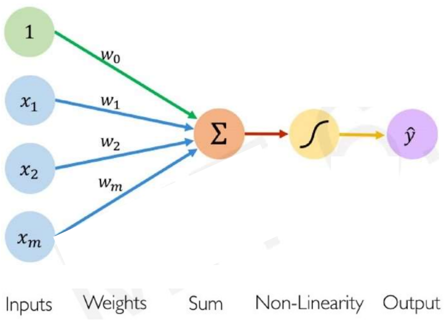
$$out = f(net - \theta)$$



Step function

Neuron model
Linear regression model + nonlinear unit

The Perceptron: Forward Propagation



Output

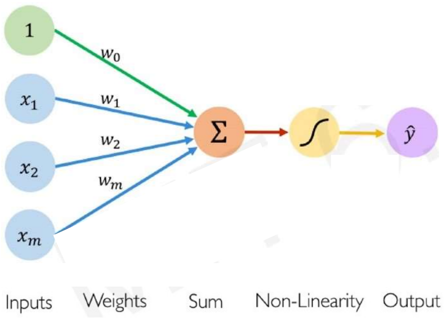
Linear combination of inputs

$$\hat{y} = g\left(w_0 + \sum_{i=1}^{m} x_i w_i\right)$$

Non-linear activation function

Bias

The Perceptron: Forward Propagation

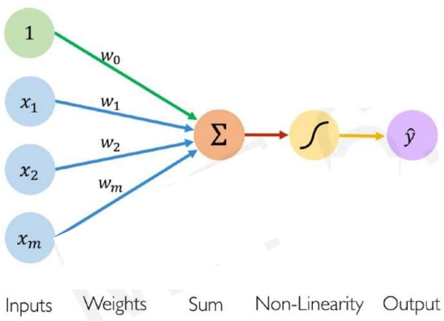


$$\hat{y} = g\left(w_0 + \sum_{i=1}^{m} x_i \, w_i \right)$$

$$\hat{y} = g \left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

where: $\boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$ and $\boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$

The Perceptron: Forward Propagation

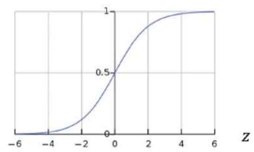


Activation Functions

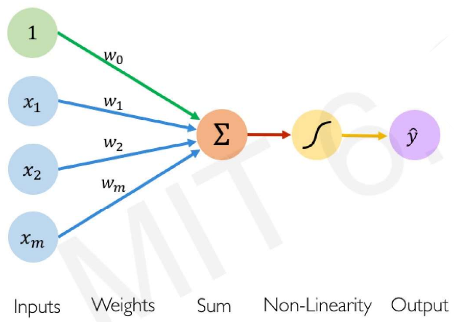
$$\hat{y} = g \left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



The Perceptron: Forward Propagation

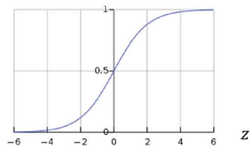


Activation Functions

$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

- Example: sigmoid function

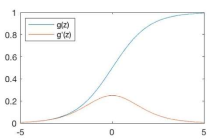
$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



Activation Functions

Common Activation Functions

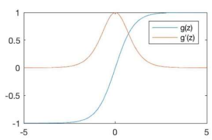
Sigmoid Function



$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

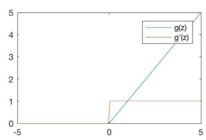
Hyperbolic Tangent



$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

Rectified Linear Unit (ReLU)



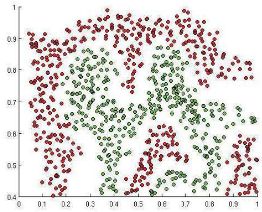
$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

NOTE: All activation functions are non-linear

Importance of Activation Functions

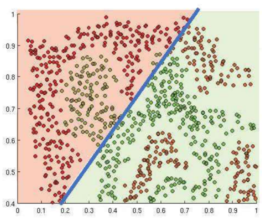
*The purpose of activation functions is to **introduce non-linearities** into the network*



What if we wanted to build a Neural Network to distinguish green vs red points?

Importance of Activation Functions

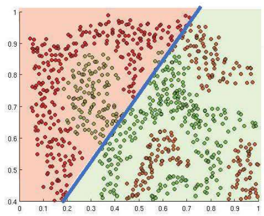
*The purpose of activation functions is to **introduce non-linearities** into the network*



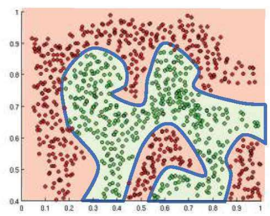
Linear Activation functions produce linear decisions no matter the network size

Importance of Activation Functions

The purpose of activation functions is to ***introduce non-linearities*** *into the network*



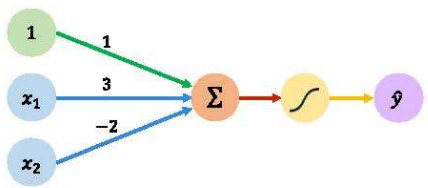
Linear Activation functions produce linear decisions no matter the network size



Non-linearities allow us to approximate arbitrarily complex functions

An Classification Example

The Perceptron: Example



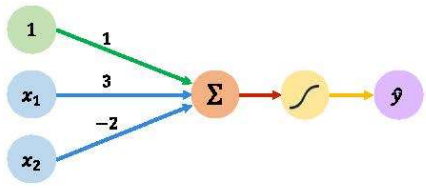
We have: $\theta_0 = 1$ and $\boldsymbol{\theta} = \begin{bmatrix}3\\-2\end{bmatrix}$

$$\hat{y} = g(\theta_0 + X^T\theta)$$

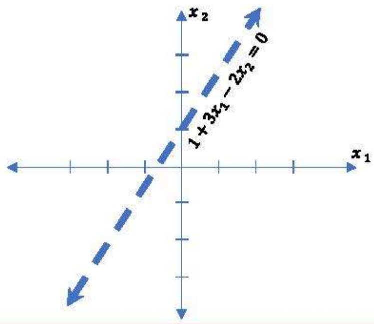
$$= g\left(1 + \begin{bmatrix}x_1\\x_2\end{bmatrix}^T \begin{bmatrix}3\\-2\end{bmatrix}\right)$$

$$\hat{y} = g(\underbrace{1 + 3x_1 - 2x_2}_{\text{This is just a line in 2D!}})$$

The Perceptron: Example

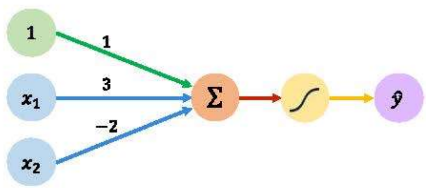


$\hat{y} = g(1 + 3x_1 - 2x_2)$

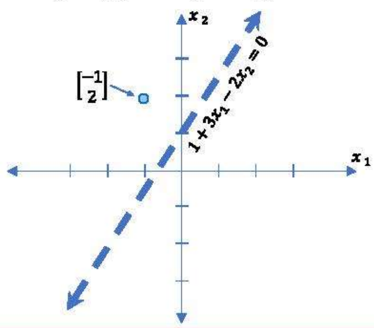


17

The Perceptron: Example



$\hat{y} = g(1 + 3x_1 - 2x_2)$

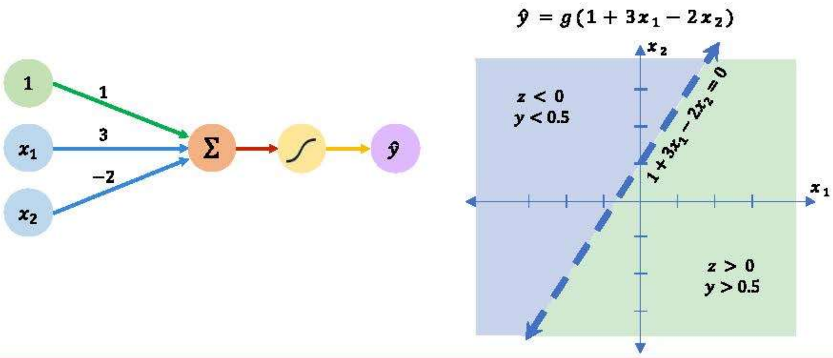


Assume we have input: $\boldsymbol{X} = \begin{bmatrix}-1\\2\end{bmatrix}$

$$\hat{y} = g\big(1 + (3 * -1) - (2 * 2)\big)$$
$$= g(-6) \approx 0.002$$

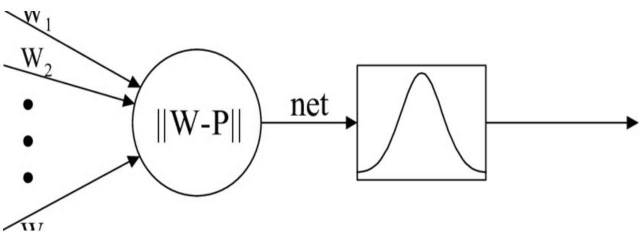
18

The Perceptron: Example



Other Neuron Models

- RBF neuron model



Similarity or distance measure