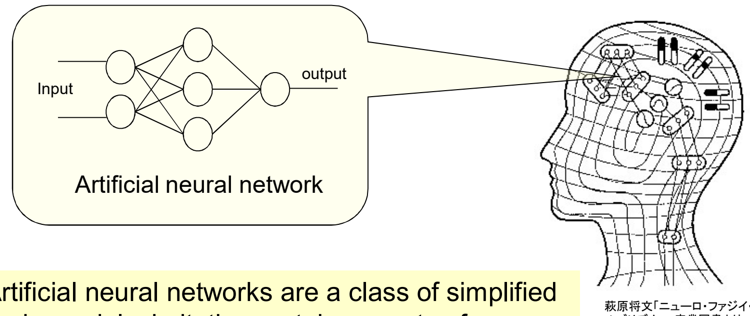
Neural Networks

A Simplified Brain Model



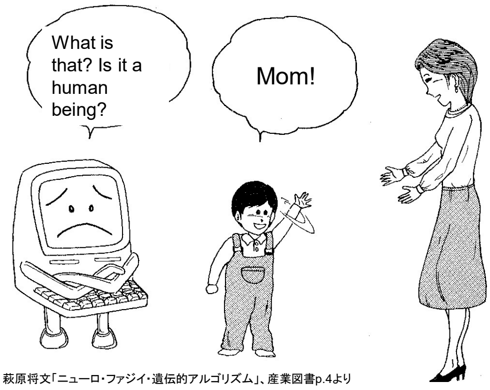
萩原将文「ニューロ･ファジイ･遺伝的アルゴリズム」、産業図書より

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

Why neural networks, why now?

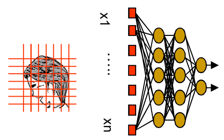
- Human and Computer



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p.4より

Pattern recognition that even a small child can easily do is difficult for a computer.

Face detection using NN



P=11

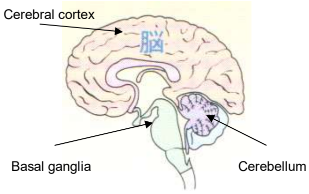
Brain and Computer

Comparison from a view point of information processing

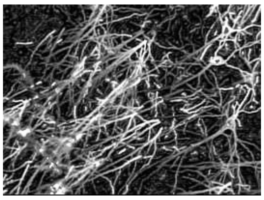
	Processing elements	Processing speed	Style of computation	Fault tolerant	learns	Intelligent, conscious
	synapses	100 Hz	Parallel, distributed	yes	yes	usually
	transistors	10^9 Hz	Serial, centralized	no	A little	Not (yet)

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.

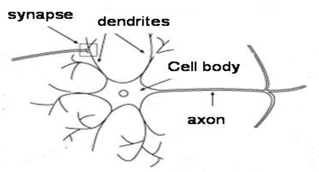
Neural Networks in Brain



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



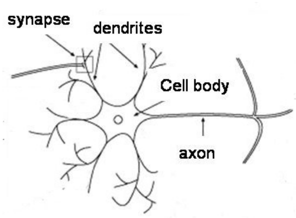
The human brain contains about 10 billion neurons. These neurons form very dense, complex local networks.

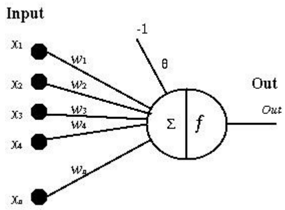


A typical neuron consists of *cell body*, *axon*, *dendrites* and *synapse*. With the synapses, a neuron connects with other neurons.

McCulloch-Pitts Neuron Model

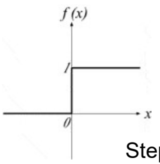
Proposed by McCulloch and Pitts in 1943





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

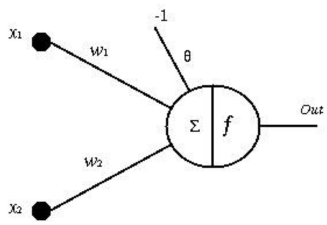
$$net = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n$$
$$out = f(net - \theta)$$



Step function

An Example: NN realizes AND function

A two-input neuron model



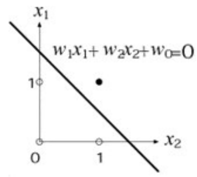
Let $w_1 = 1.0,\ w_2 = 1.0,\ \theta = 1.5$

$f(x) = \text{step}(x)$

then $Out = \text{step}(w_1 x_1 + w_2 x_2 - \theta)$

The neuron model realizes logical AND, which is a linear separable problem.

x_1	x_2	Out
0	0	0
0	1	0
1	0	0
1	1	1

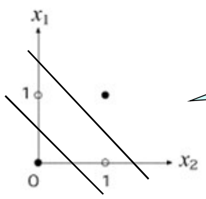


Logical AND

Limitation of One Neuron

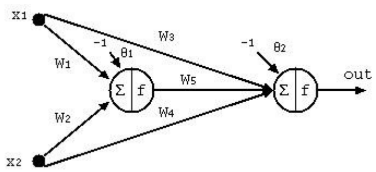
One neuron model is able to solve linear separable problems such as logical AND, OR, but it is not able to solve problems that are not linear separable such as logical XOR.

However, adding another neuron, the two-neuron-model is able to realize logical XOR.



Not linear separable.

Logical XOR



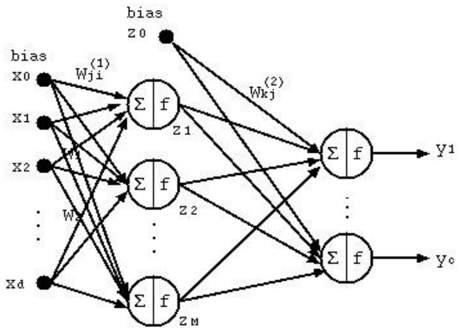
Two-neuron-model

$w_1 = w_2 = w_3 = w_4 = 1$

$w_5 = -2, \theta_1 = 1.5, \theta_2 = 0.5$

$f(x) = \text{step}(x)$

Multilayer Neural Networks



Cost function:

$$E = \frac{1}{2}\sum_{t=1}^{N}\sum_{k=1}^{c}(d_k(t) - y_k(t))^2$$

where

$$\begin{aligned} y_k(t) &= f_2(net_k(t)) \\ &= f_2(\sum_{j=0}^{M} w_{kj}^{(2)} z_j(t)) \\ &= f_2(\sum_{j=0}^{M} w_{kj}^{(2)} f_1(net_j(t))) \\ &= f_2(\sum_{j=0}^{M} w_{kj}^{(2)} f_1(\sum_{i=0}^{d} w_{ji}^{(1)} x_i(t)))) \end{aligned}$$

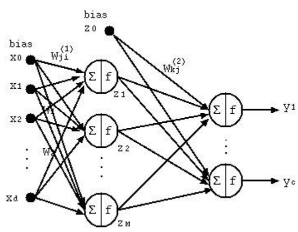
One neuron model has limited approximation ability. But a multilayer neural network has universal approximation ability.

Neural Network Training: BP Algorithm

Update weights using gradient method:

$$w_{ji}^{(1)} \leftarrow w_{ji}^{(1)} - \mu \frac{\partial E}{\partial w_{ji}^{(1)}}$$

$$w_{kj}^{(2)} \leftarrow w_{kj}^{(2)} - \mu \frac{\partial E}{\partial w_{kj}^{(2)}}$$



Calculate the gradient using BP algorithm:

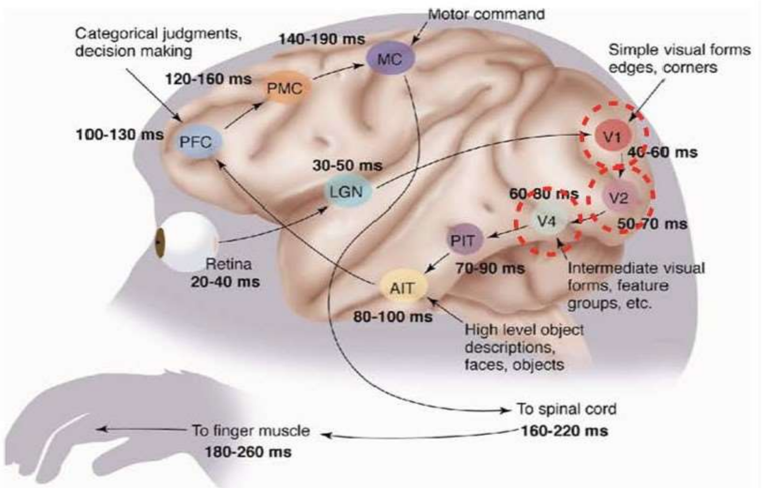
$$\frac{\partial E}{\partial w_{kj}^{(2)}} = -\sum_{t=1}^{N} \delta_{2k}(t) z_j(t) \ , \qquad \delta_{2k}(t) = (d_k(t) - y_k(t)) f_2^{'}(net_k(t))$$

$$\frac{\partial E}{\partial w_{ji}^{(1)}} = -\sum_{t=1}^{N} \delta_{1j}(t) x_i(t) \ , \qquad \delta_{1j}(t) = f_1^{'}(net_j(t)) \sum_{k=1}^{c} w_{kj}^{(2)} \delta_{2k}(t)$$

Note

Various techniques are needed to overcome local minimum problem and overtraining problem in the implementation of BP algorithm.

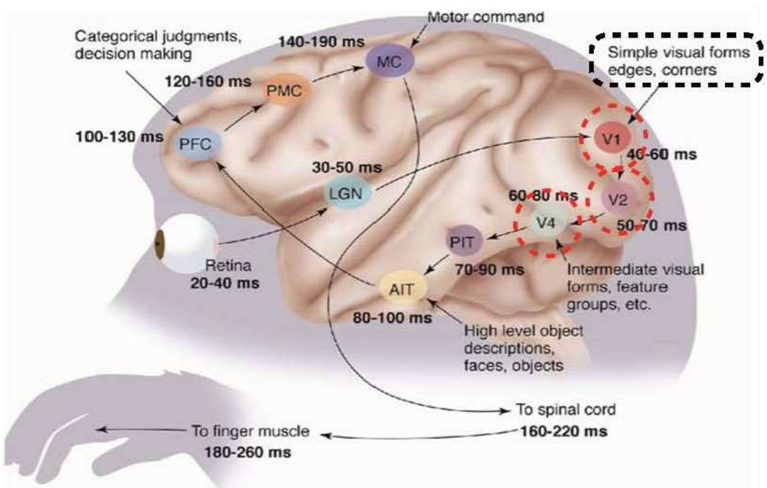
Deep Neural Network



[picture from Simon Thorpe]

18

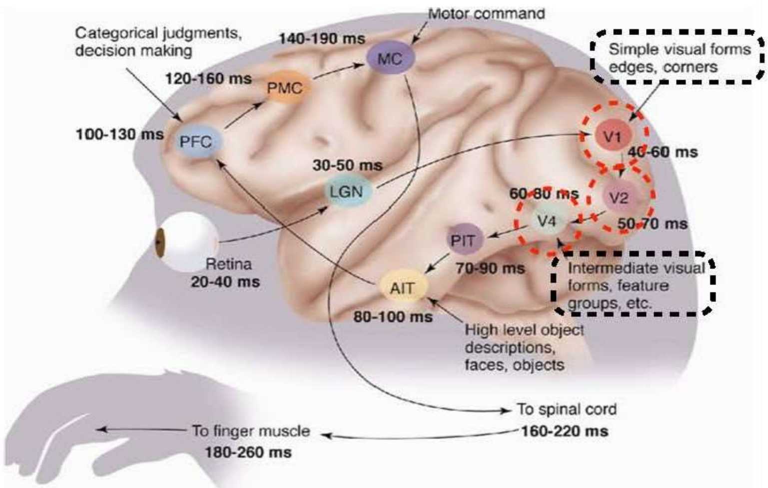
Deep Neural Network (cont'd)



[picture from Simon Thorpe]

18

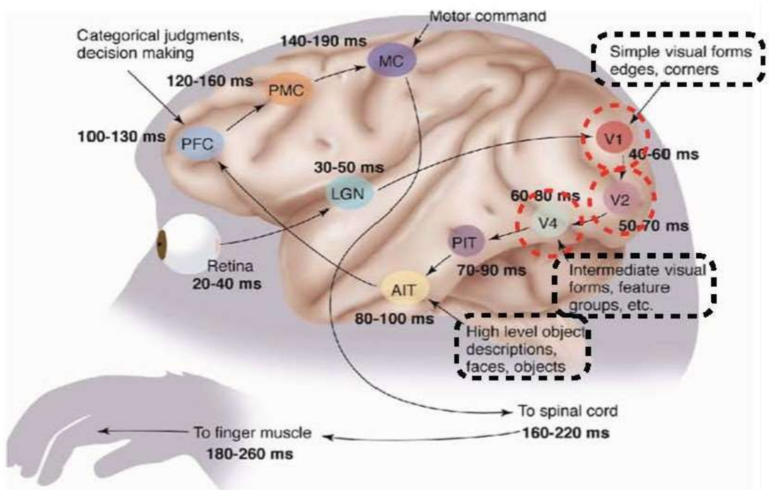
Deep Neural Network (cont'd)



[picture from Simon Thorpe]

18

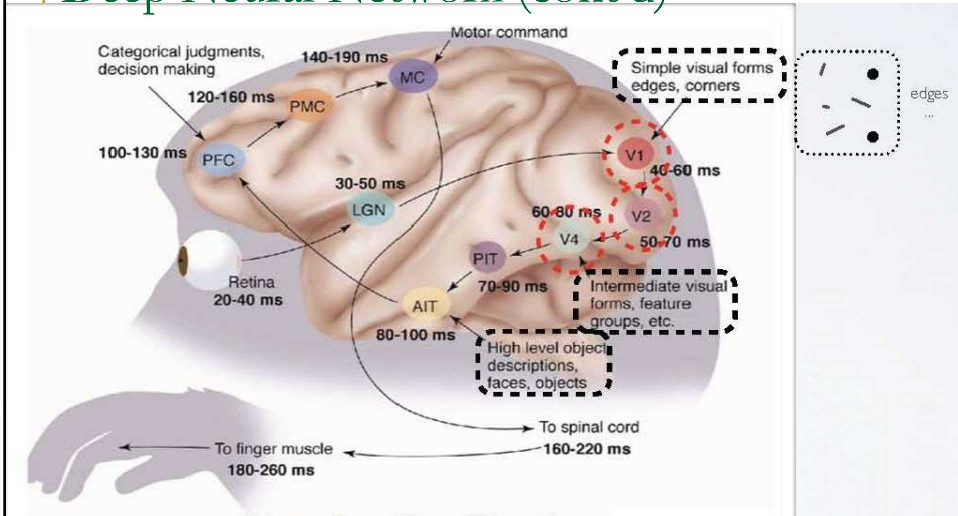
Deep Neural Network (cont'd)



[picture from Simon Thorpe]

18

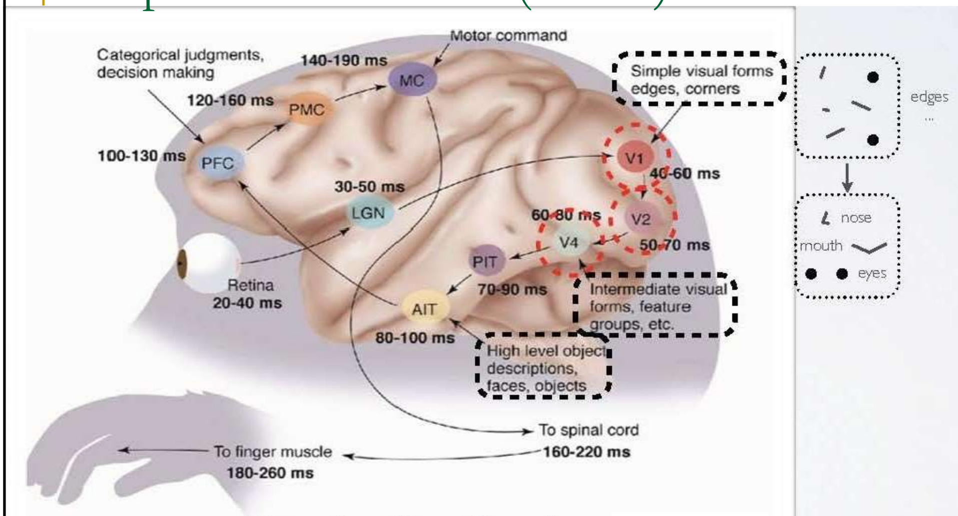
Deep Neural Network (cont'd)



[picture from Simon Thorpe]

18

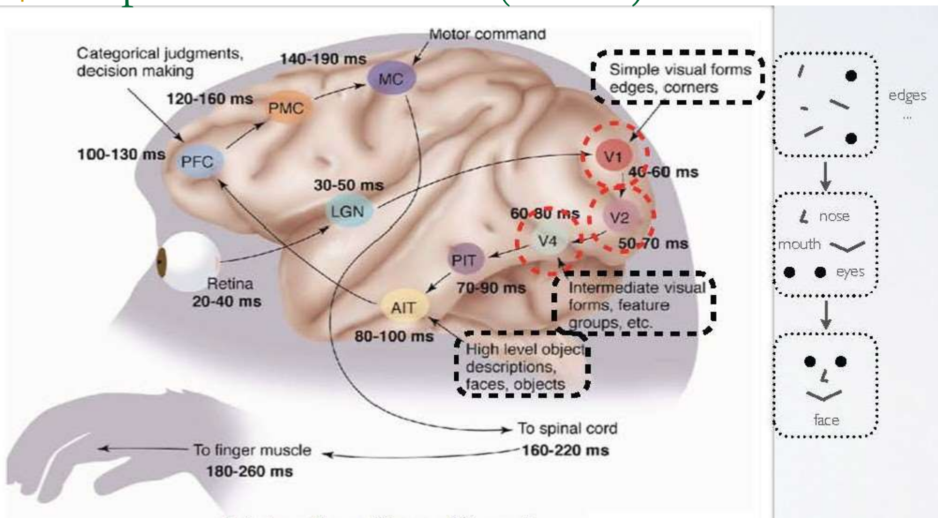
Deep Neural Network (cont'd)



[picture from Simon Thorpe]

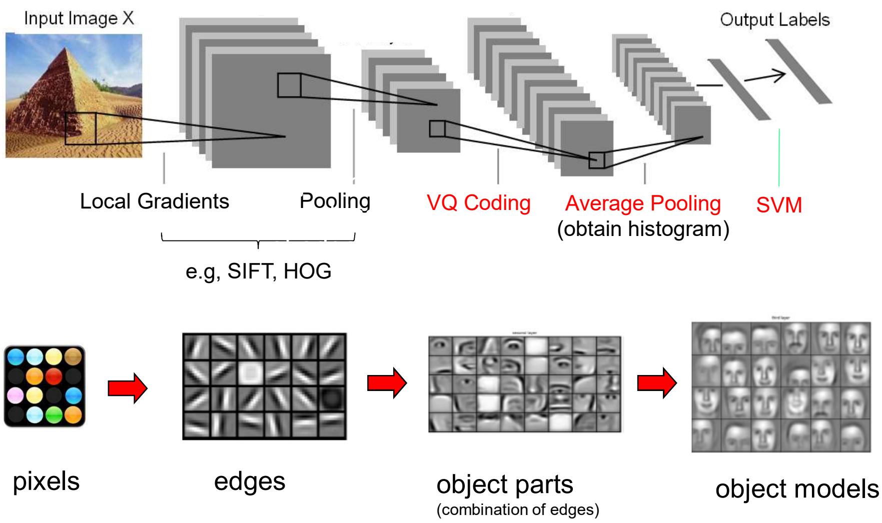
18

Deep Neural Network (cont'd)



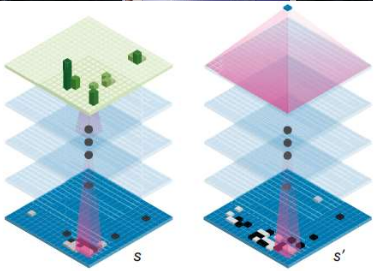
[picture from Simon Thorpe]

Deep Neural Network (cont'd)



A Successful Application of DNN: AlphaGo

1. **Deep Learning + Reinforcement Learning**



19