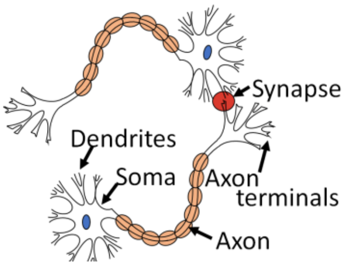
<Biological Neuron & Artificial Neural Network> JIEUN KIM

1.Biological Neuron

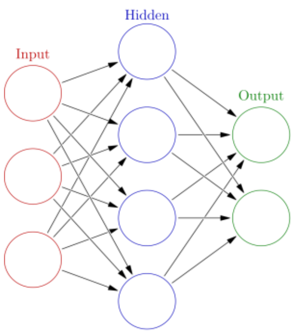


Neuron is a basic unit of nervous system. There are about 100 billion neurons in human brain. Neurons in brain and the nervous system functions such as information processing, memory, learning, sensory processing, and motor control. They are composed of dendrite(receive signals), soma, axon(translate signals to other cells). Synapse is connection between neurons.

Neural transmission - Neurons transmit signals using neurotransmitters and use both electrical & chemical signals. Electrical signals are used within neurons through ion channels such as sodium and potassium channels. Chemical signals are used at synapses. Synapse is a connection point between neurons, where electrical signals are converted into chemical signals at the axon terminal and transmitted to the dendrites of the next neuron. Electrical signals are converted into chemical signals at the axon terminal and transmitted to the dendrites of the next neuron.

All-or-none principle - Neurons transmit signals only when the stimulus reaches or exceeds a certain threshold.

2. Artificial Neural Network (ANN)

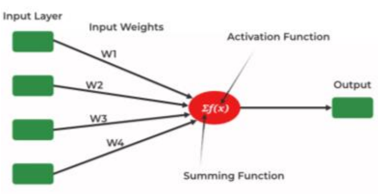


It's a mathematical model that imitates the function principles of neurons. ANN models are designed to help computers make intelligent decisions with limited human intervention. It's an interconnected group of circular nodes. It's composed of node(an artificial neuron) and edge(connection weight). One of the notable features of artificial neural networks is their exceptional parallelism. They can process and handle data in parallel, which makes them well-suited for various machine learning and AI tasks.

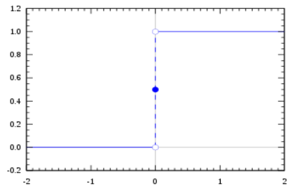
Working principle: It receives information from each neuron, calculates and outputs a value according to the activation function. When the output is wrong, learning is needed. Learning is to adjust the connection weight value to produce an accurate output. The learning of most artificial neural networks is to reduce errors by changing the weight value.

3. Applications. Widely utilized in various fields & easily find applications in our daily life(ex. Siri)

4. Perceptron

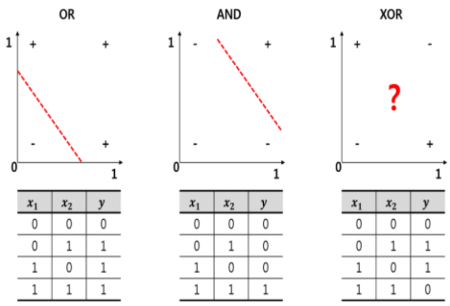


It takes multiple input values and applies certain weights to these inputs. Then sums them up. This sum is then passed through an activation function to produce an output.



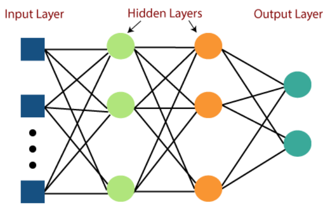
Heaviside step function is commonly used as activation function, and it produces binary output(0 or 1) based on whether the sum is above certain threshold or not. The result of activation function's computation is an output.

Learning process is to update weight. 1.Get outputs from the previous process 2. Initialize the weight: start with small values 3. Weighted_sum = sum of (input*weight) 4. Apply to activation function & get output 5. Calculate the error. error=target_output(actual output) – predicted_output 6. Changing the weight according to perceptron learning rule. New_weight(updated_weight)=old_weight(current weight) + learning rate * error * input_value 7. Iterate step 2-6

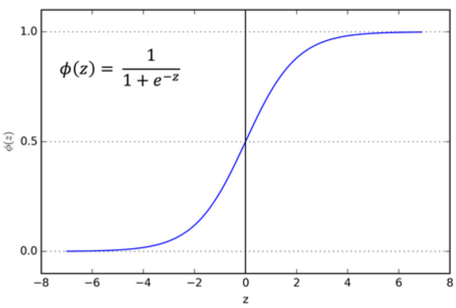


Limitation: A single layer perceptron is a linear classifier that only can classify linear separable data set. Nonlinear classification problem cannot be solved. For example, the XOR problem (XOR gate) cannot be handled. Since it can't solve the XOR gate which is the basic unit that makes up an electric circuit along with AND, OR, and NAND, there was dark age of ANN development.

5. Multilayer Perceptron



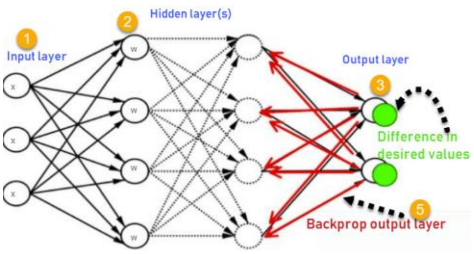
The limitation of single layer perceptron was overcome with the multilayer perceptron. It's a multilayer neural network that can non-linearly separate complex data sets. It's composed of 1 input layer, 1 or more hidden layers and 1 output layer. It's also a basis of deep learning.



It emerged by appearance of sigmoid activation function. Differentiable sigmoid function enables gradient descent and backpropagation which are used as the learning process algorithm of MLP. Also, Non-linear activation functions in hidden layers enable neurons in hidden layers combine input values to learn non-linear boundaries and solve non-linear classification problems.

Limitations: 1. Because the gradient of sigmoid function converges to 0 if output goes to both ends, sigmoid saturate&killing gradients and vanishing gradient problem are occurred. Overcome by using other functions like ReLU, Leaky ReLU, PReLU, ELU. 2. Non-zero centered problem: Output of sigmoid function is always positive value. Just like single layer perceptron learning process, change of weight is also proportional to input*error. So, quadrant 1,3 are allowed weight update directions and zigzag pattern are made. It's inefficient, and it can be overcome with hyperbolic tangent activation function.

Backpropagation: Learning algorithm of MLP.



Working process: 1. Feedforward. Provide an input, compute the weighted sum of inputs for each neuron, pass the result through an activation function, obtain the output of each neuron in each layer. 2. Calculate error using loss function 3. Backpropagation. Calculate the gradient of the loss function with respect to the network's parameters (weights and biases). Using the chain rule of calculus. Update the network's weights and biases in the direction that reduces the loss. The update rule typically uses the calculated gradients and a learning rate. Iterate those steps until loss converges to minimum