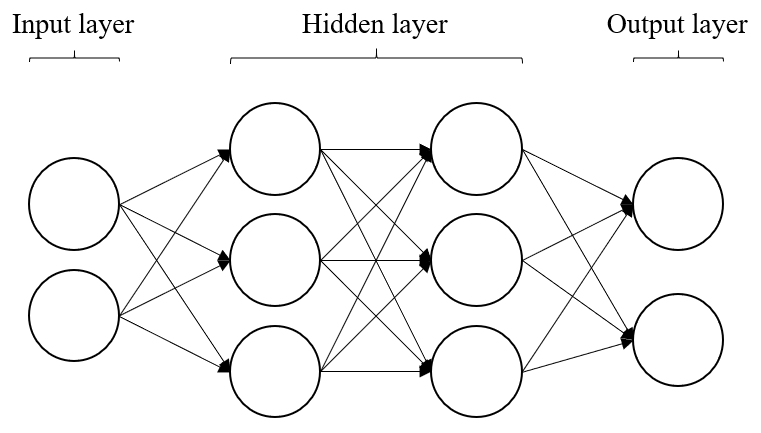
Deep Learning – Artificial Neural Networks

**Introduction**

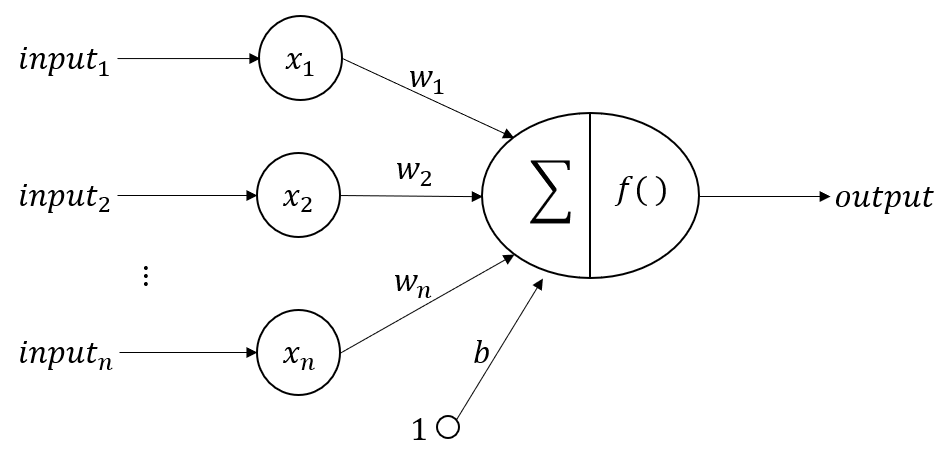
Deep Learning (DL), which is a branch of Machine Learning (ML) and Artificial Intelligence (AI). Using Artificial Neural Networks (ANN) for processing data, DL needs a massive data to train a model. ANN is a network that simulates neurons in human brain. As shown in Figure 1, an ANN which has 2 or more layers in the hidden layer can be called as a Deep Neural Network (DNN).

**Figure. 1 Deep Neural Network, DNN**

**Perceptron**

Perceptron is the basic component in a Neural Network (NN). As Shown in Figure 2, a perceptron also has an input layer and an output layer. There has an input tensor: for the input layer and every input has a corresponded weight in the weights tensor: , additionally adds a bias so that we can easier find a solution.

For the output layer we first multiply every input and weight then sum up with the bias , which comes out an equation: . After we get from the above equation we send it to the activation function and returns the output: ).



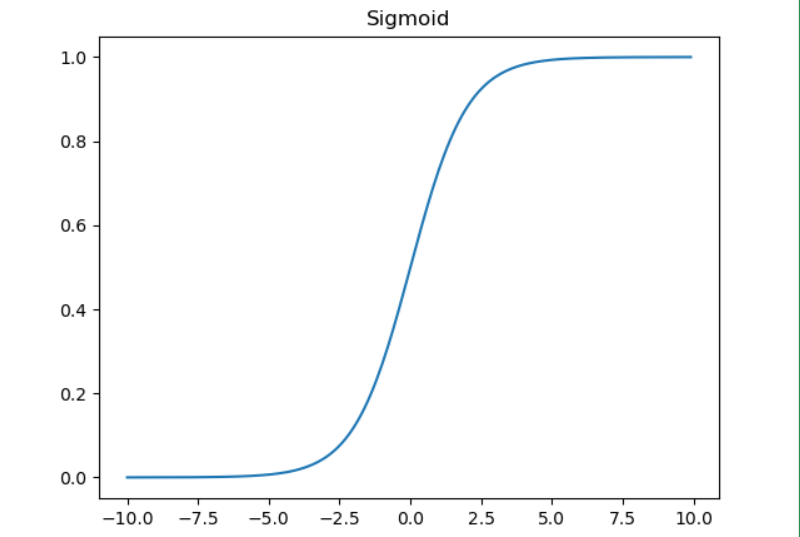
**Figure 2. Perceptron**

**Activation Function**

If there is no activation function in a NN no matter how many layers we pass through, it always constructs a linear function which can only deals with linear problems when fitting data. To solve a nonlinear problem, we use activation function to break the linearity which transfer the data into any range like 0 to 1 or -1 to 1 and that makes the NN fits more nonlinear problems. Here list some activation functions:

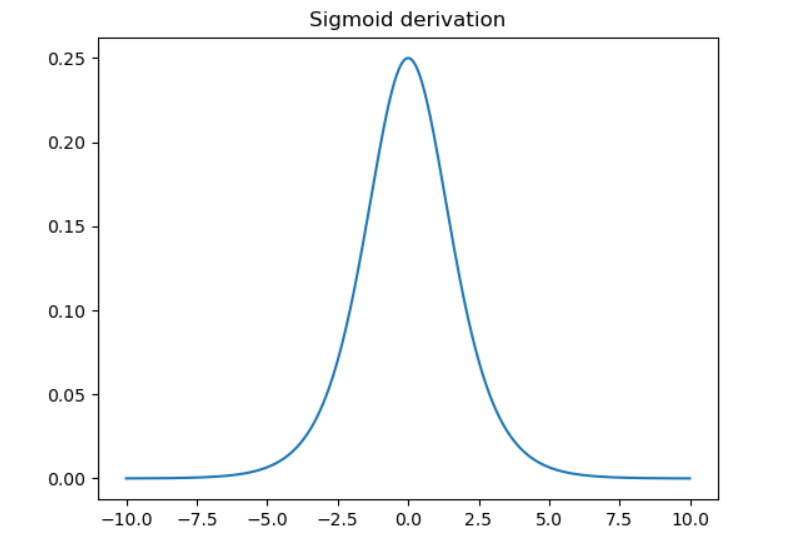
1. **Sigmoid**

As shown in Figure 3, a sigmoid function converts any data into a range of 0 to 1, and the equation:



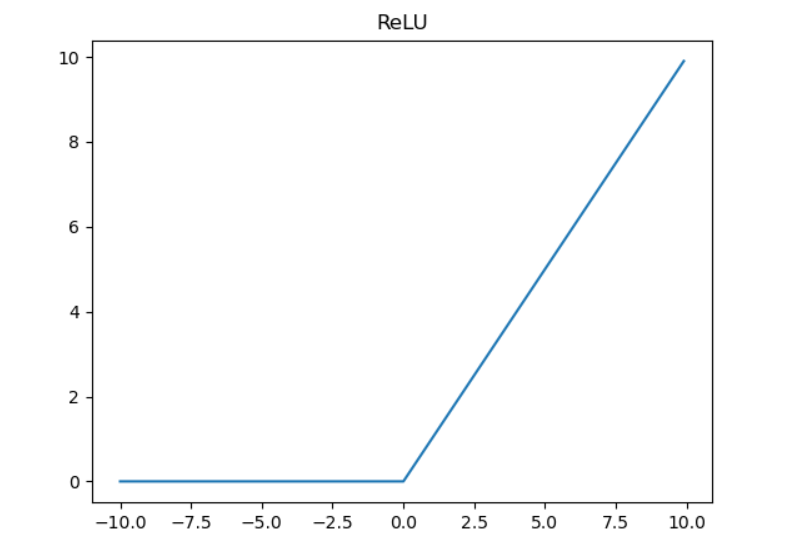
**Figure 3. Sigmoid function**

1. **Rectified Linear Unit (ReLU)**

 As shown in Figure 4, the maximum value is only 0.25 when derivates the sigmoid function, this will encounter a problem called Vanishing Gradient Problem which happens when we do backpropagation using chain rule to calculate the gradient.

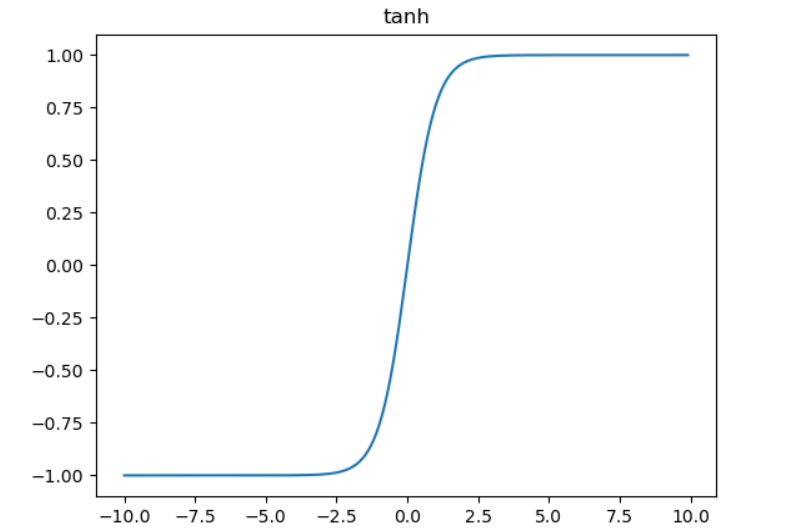
**Figure 4. Derivative of Sigmoid function**

ReLU function outputs 0 when the input is smaller than 0 else outputs a linear function which is the same as input. We can see that in Figure 5, the ReLU function has a derivation of 1 when it is greater than 0, which avoids the Vanishing Gradient Problem.

 The equation:

**Figure 4. ReLU function**

1. **Hyperbolic Tangent (Tanh)**

 Tanh is a trigonometric function, Figure 5 shows that the function has an output in a range of -1 to 1 while Sigmoid and ReLU do not have negative values, the equation:

**Figure 5. Tanh function**

1. **Softmax**

Softmax function converts the inputs into real numbers between 0 and 1 which presents in a form of probability, the formula is like:

for and

**Multilayer Perceptron (MLP)**

With single perceptron we can solve some linearly separable problems, but when we are about to do some problems that is not linearly separable, we need 2 or more layers of perceptron. As described at the beginning of the article, Figure 1 is a MLP, its every node connects to the every nodes in the next layer which is also called Full Connected and this kind of NN layer is called Dense Layer.

**Convolutional Neural Network (CNN)**