Machine Learning

Artificial Neural Network

Artificial Neural Networks

- Inspired by neural networks in human brain

- History of ANN

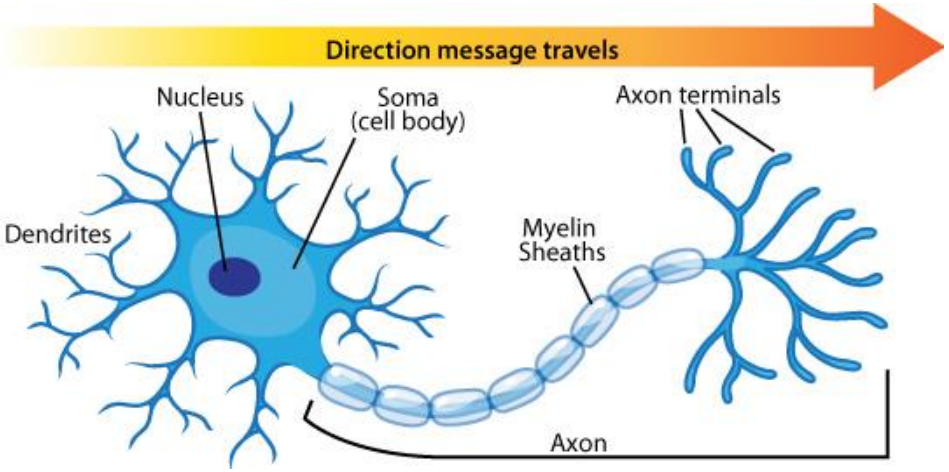
  - In 1950's, Rosenblatt's developed model of the perceptron. It was capable of learning certain classifications by adjusting connection weights.

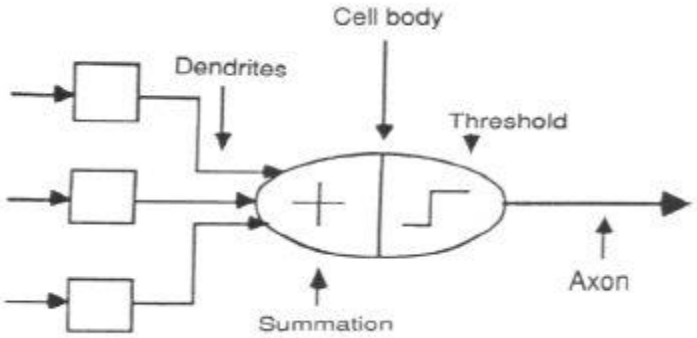
  - In early 1980's, there was renewed interest in neural networks.

  - However, lack of development in other areas such as Hardware, delayed popularity of ANNs for almost 30 years

Neural Network

- Neurons



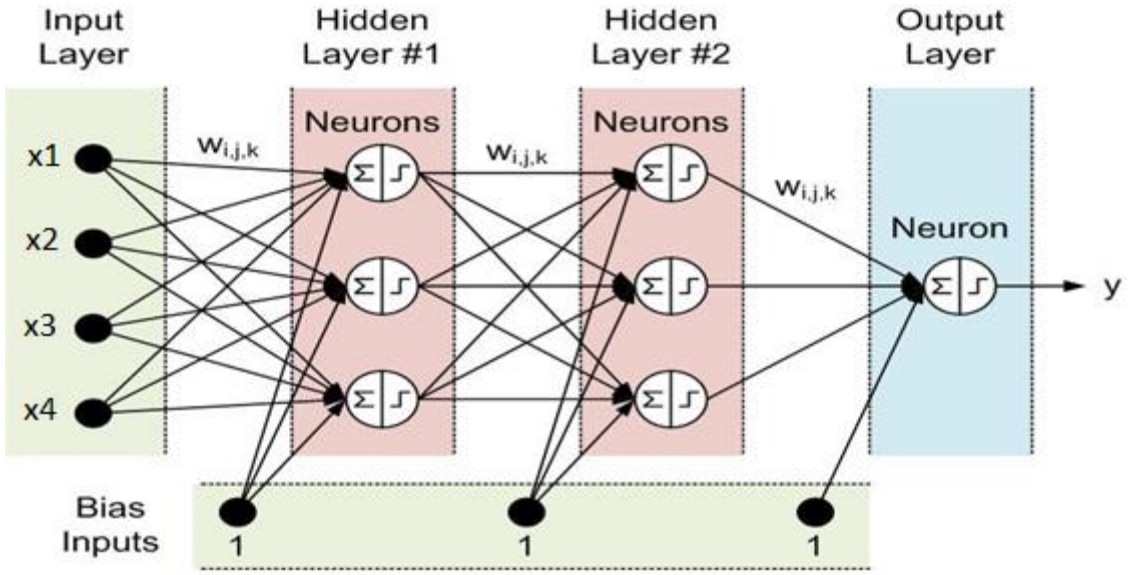


Neural Network

- The processing elements of a ANN is called a node. It represents the artificial neuron

- Each ANN consists of a collection of nodes grouped in layers.

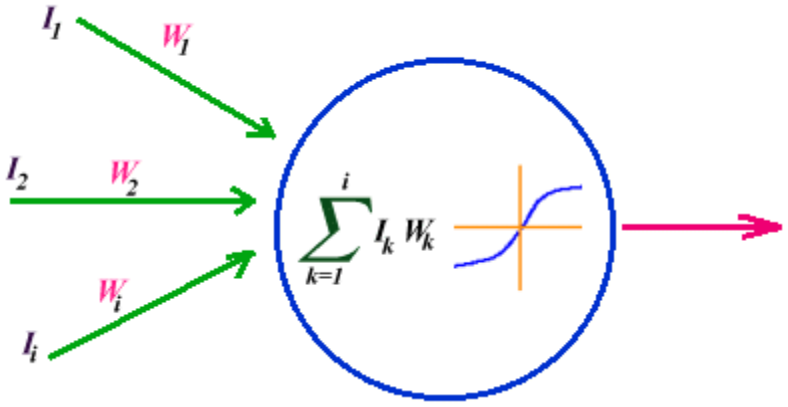
- The initial layer is the input layer and the last layer is the output layer. In between we have the hidden layers



Neural Network

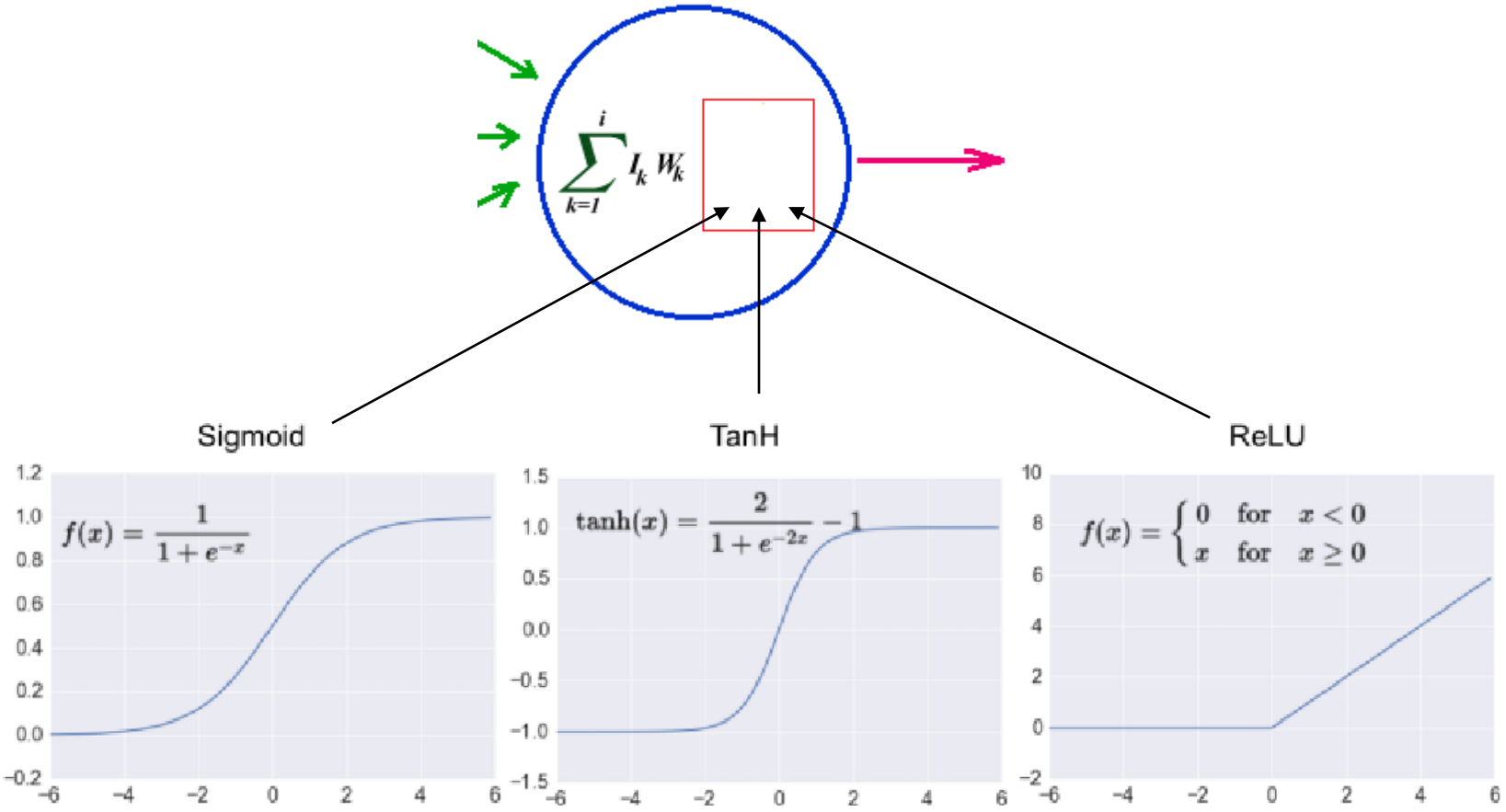
- Each neuron has input and a function that generates output

- Sigmoid function was commonly used in ANN. Recently it is replaced with ReLU function



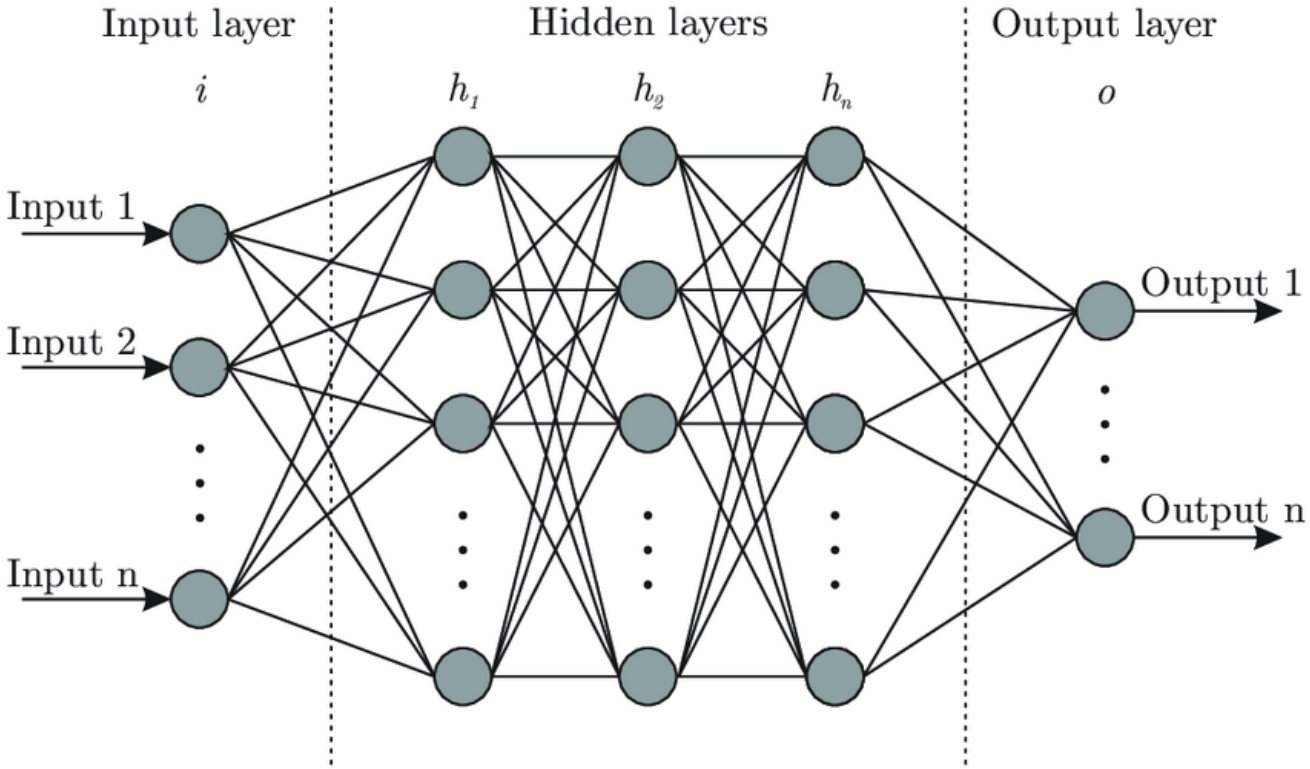
Neural Network

- Various options for the function used in neuron



Neural Network – Input Layer

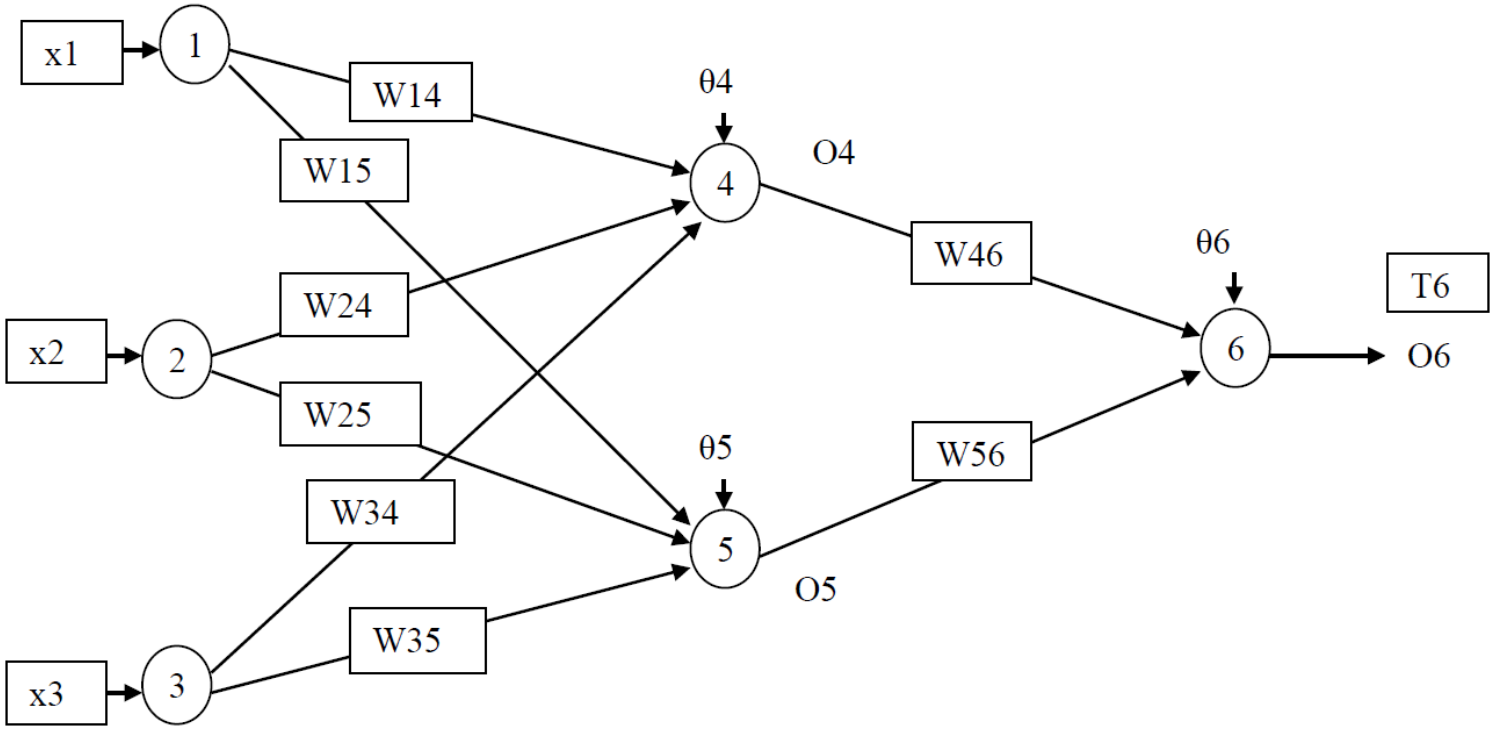
- Generic structure of ANN: The input layer does not perform any processing. It only holds the input data to supply it to the first hidden layer
- Following diagram shows a Neural Network with three hidden layers (note: bias inputs are not shown. This is for the purpose of understanding different layers)



Neural Network

- The weights (including bias) for hidden layers and output layers are randomly assigned.
- The output of each *input layer neuron* is fed to each neuron in the next hidden layer
- There can be one or more hidden layers each containing multiple neurons
- The output of each *hidden layer neuron* is fed to each neuron in the next layer (next layer can be hidden layer or output layer)
- Number of output neurons vary. For Regression, the output layer contains one neuron. For classification, the output layer contains one or multiple neurons depending on binary or multi-class classification

Neural Network



- ANN with 1 hidden layer and one output layer
  - One input layer of nodes (leftmost layer where external inputs are applied)
  - One output layer of nodes (rightmost layer where predicted values are available)
  - One or more hidden layers (one hidden layer in this example) between input layer and output layer

Neural Network - Learning

- Artificial neural networks work through the optimized weight values.
- The method by which the optimized weight values are attained is called learning
- During the learning process, neural network adjust weights, using backpropagation method, to produce the desired output
- When learning is complete, the trained neural network, with the updated optimal weights, can be used on unseen date to produce the predicted output value

Neural Network

- Key parameters
  - Hidden layers: Both the number of hidden layers and the number of neurons in each hidden layer can influence the quality of the results. For example, too few layers and/or neurons may not be adequate to sufficiently learn and too many may result in overfitting.
  - Number of cycles: A cycle is where a training example is presented and the weights are adjusted.
  - Learning rate: influences how fast the neural network learns.

Neural Network

- Input
  - i_1,i_2,i_3,i_4,i_5,i_6 : Net inputs to nodes 1,2,3,4,5,6
  - For input layer nodes j, $i_j = x_j$
  - For hidden and output layer node j, $i_j=\Sigma w_{ij}*O_i +\theta_j$

- Output
  - O_1,O_2,O_3,O_4,O_5,O_6: Outputs from nodes 1,2,3,4,5,6
  - For any input layer node j, $O_j= i_j$
  - For any hidden layer node j, $O_j=1/(1+e^{-ij})$ or ReLU etc.
  - For Output layer, $O_j=1/(1+e^{-ij})$ (for classification)

- $\mathbf{T_6}$=True value at output node 6. $\mathbf{e_j}$ = Error at node j, **L**= learning rate
- For an output node j, the error $e_j = O_j(1-O_j)(T_j-O_j)$
- For any node j at the hidden layer, the error $e_j = O_j(1-O_j)(\Sigma w_{jk}*e_k)$
- Weight updation: $w_{ij} = w_{ij} + L *O_i *e_j$, Bias updation: $\theta_j = \theta_j + L*(e_j)$

- *The above calculation are for a particular case and these vary. This is assuming – binary classification using sigmoid as activation function. These calculations will vary based on cost function, activation function, solver etc. These derivations involve partial derivatives, which are not covered as part of this class. Use of Softmax function in Output neurons is also not discussed here.*

Neural Network

- Advantages
  - Ability to implicitly detect complex nonlinear relationships between dependent and independent variables
  - Does not assume normal distribution of data

- Disadvantages
  - It is a black box model
  - Does not provide information about the relative significance of the various parameters

Neural Network

- Exercise: Artificial Neural Network