3. Deep Learning – Terminology

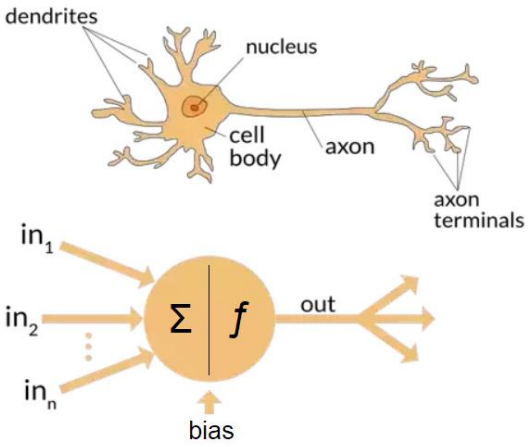
Contents

1. Neuron 2
2. MLP (Multi-Layer Perceptrons) 3
3. Neural network 4
4. Input, Hidden layers & Output 5
5. Weights 6
6. Bias 7
7. Activation Function 8
8. Types of activation functions 9
9. Forward Propagation 10
10. Back propagation 11
11. Cost Function 12
12. Gradient Descent 13
13. Learning Rate 13
14. Batches 14
15. Epochs 14

3. Deep Learning – Terminology

1. Neuron

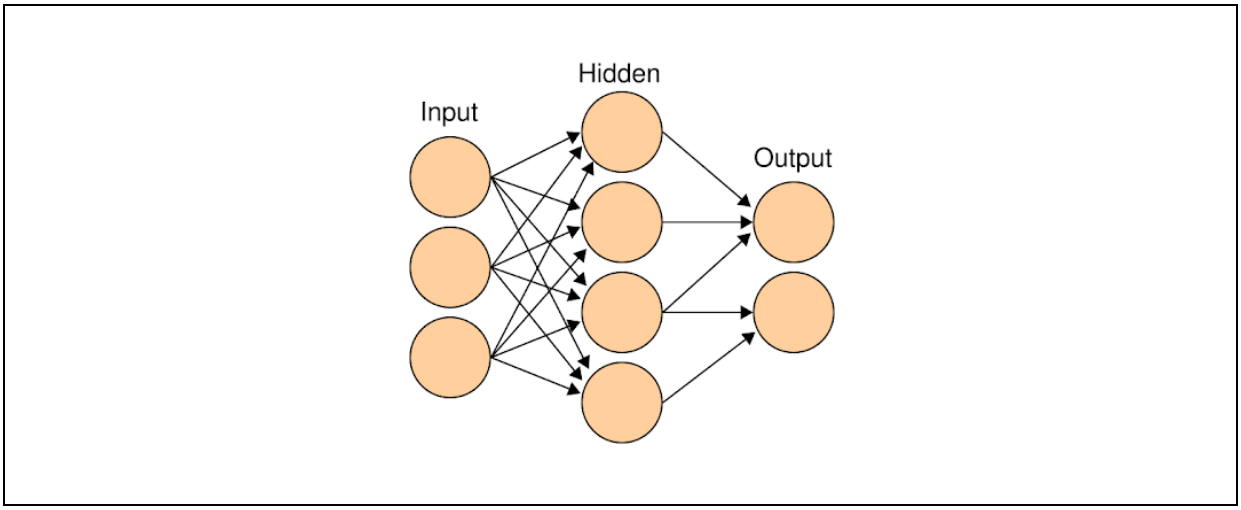
- ✓ Neuron forms the basic element of our brain.
- ✓ A group of neurons used to create neural network.
- ✓ A neuron receives an input, processes it and generates an output.



2. MLP (Multi-Layer Perceptrons)

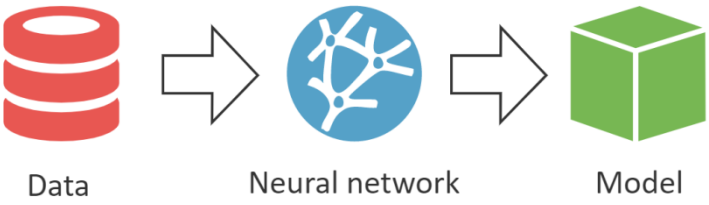
- ✓ A single neuron may not perform the complex tasks.
- ✓ So, it's required to use group of neurons to perform a complex task.
- ✓ In simple network we do have like,
  - Input layer.
  - Hidden layer.
  - Output layer.

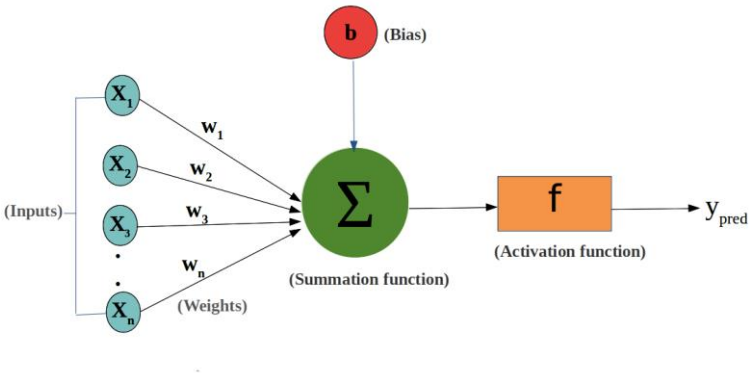
- ✓ Each layer has multiple neurons.
- ✓ All neurons in each layer are connected to all the neurons in the next layer.

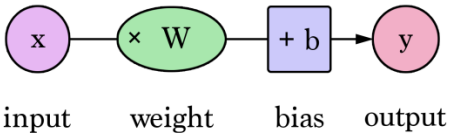


3. Neural network

- ✓ Neural Network is the backbone of deep learning.
- ✓ A Neural Network is combinations of basic Neurons also called as Perceptrons.
- ✓ The goal of a neural network is to find the mapping function.
  - Neurons having weights and bias which is updated during training.

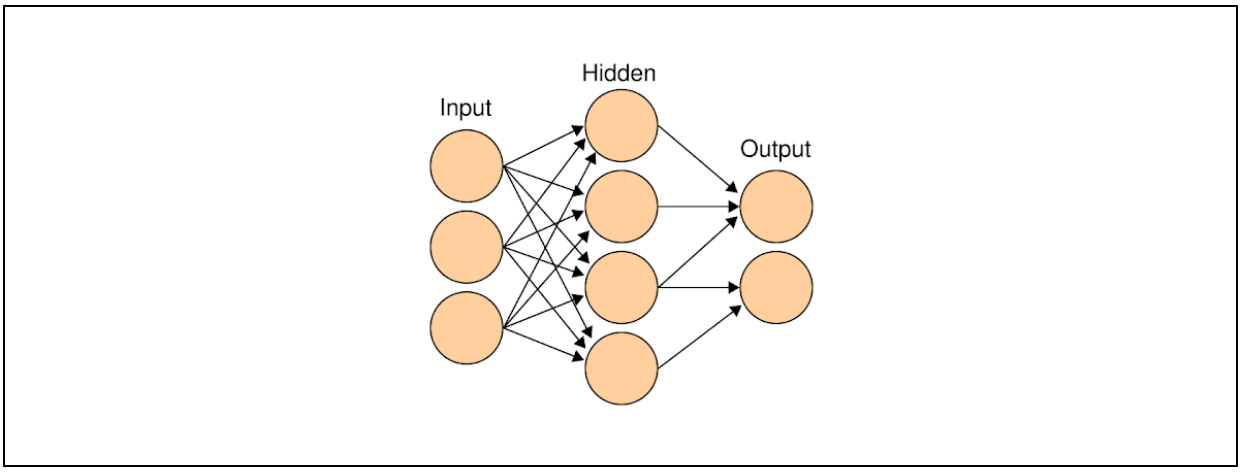






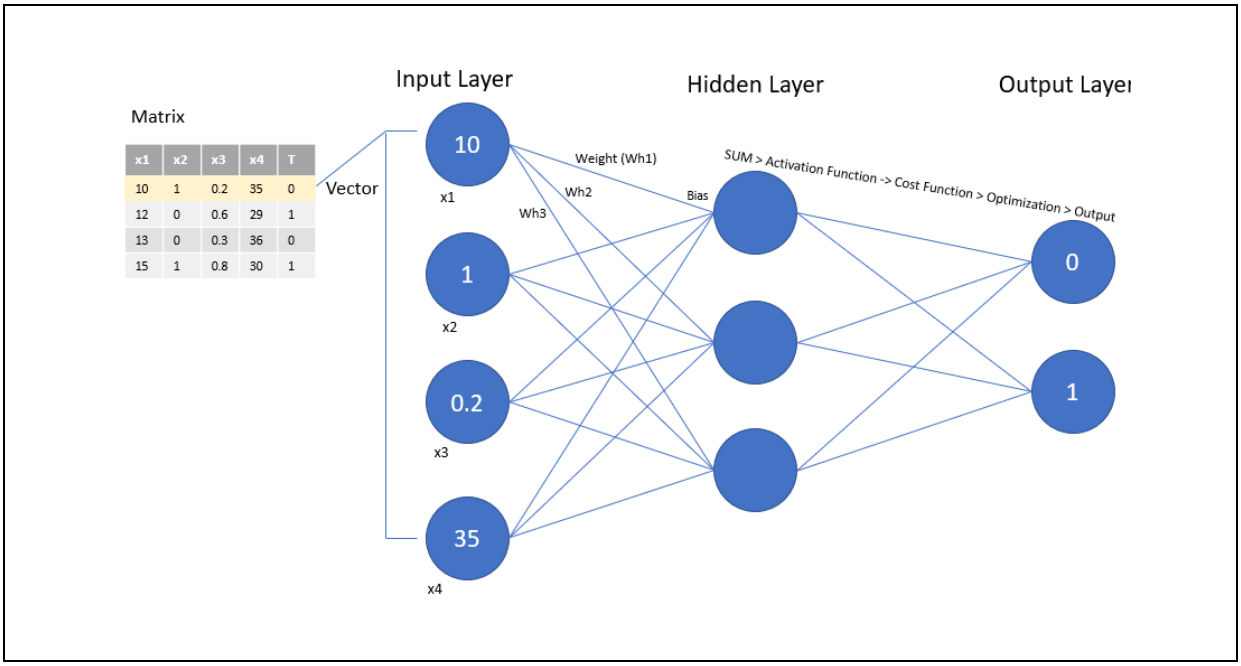
4. Input, Hidden layers & Output

- ✓ Input layer receives the input
- ✓ The processing layers are the hidden layers within the network.
  - These layers perform specific tasks on the incoming data.
  - These layers can pass result to the next layers
- ✓ Output layer generates the output
- ✓ Input and output layers are visible but hidden layers are hidden



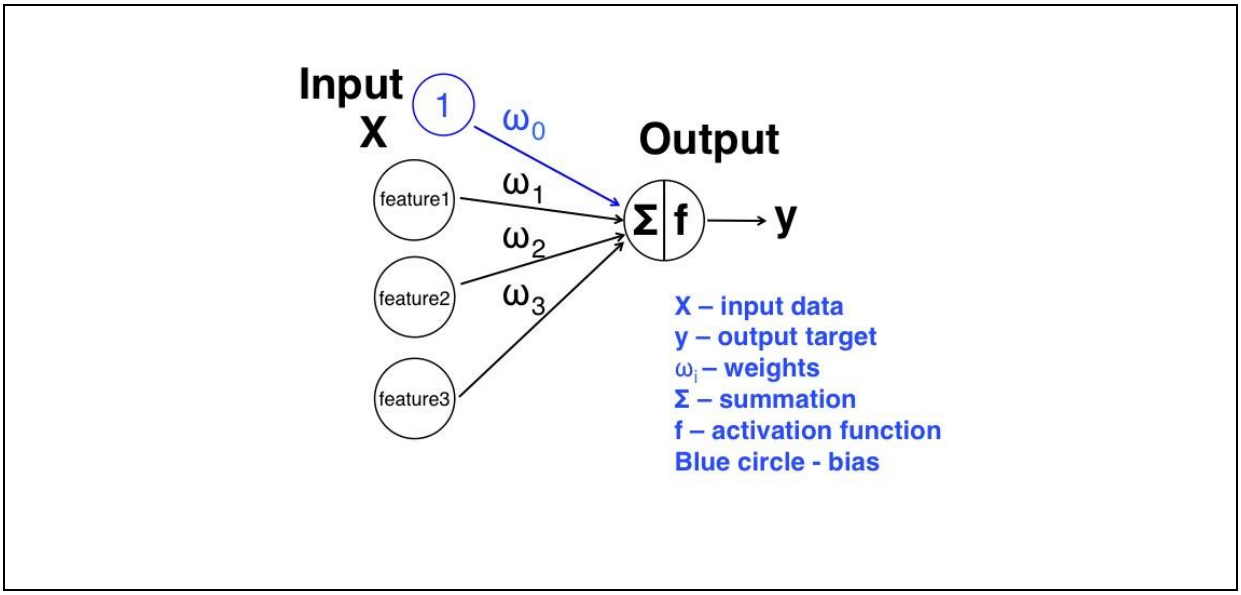
5. Weights

- ✓ When input enters into the neuron, it is multiplied by a weight.
- ✓ Assuming that, if a neuron has two inputs, then each input have separated weights.
- ✓ Here weights will be initialized randomly and these weights are updated during the model training.
  - Let's assume the input is **a** value and weight is **W1**.
  - Then after passing through the node the input becomes **a*W1**



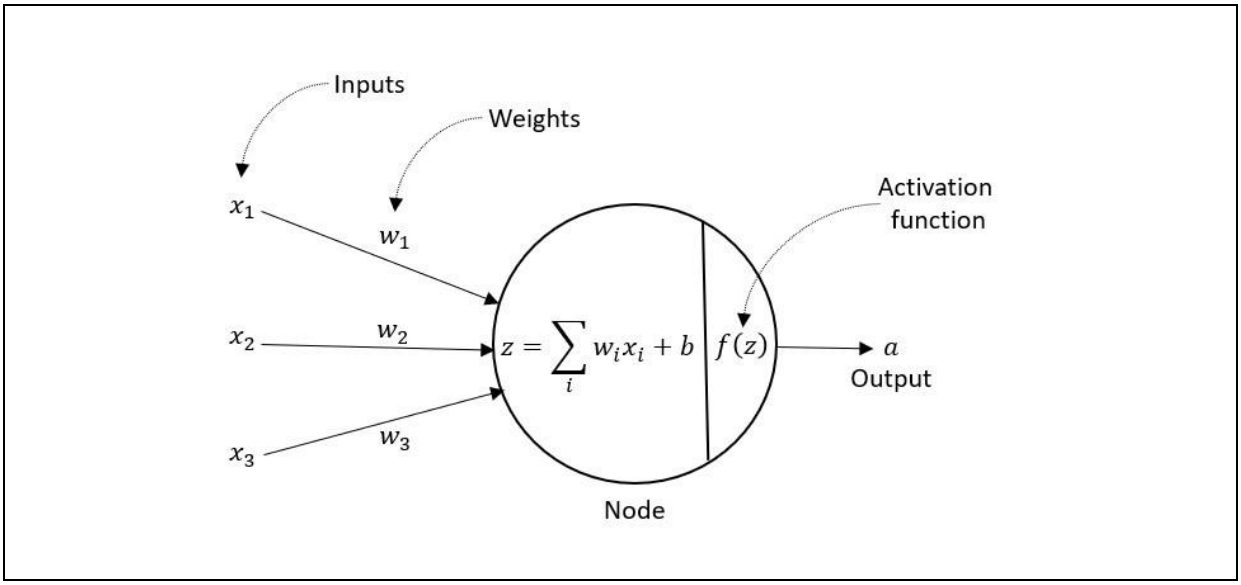
6. Bias

- ✓ In addition to the weights, bias also added to the input.
- ✓ After adding the bias, the result would look like, a * W1 + bias.



7. Activation Function

- ✓ The activation function translates the input signals to output signals.
- ✓ After applying activation function then the its looks like,
  - f(a*W1+b)
  - Here f() is the activation function.
- ✓ The activation function puts a nonlinear transformation to the linear combination



8. Types of activation functions

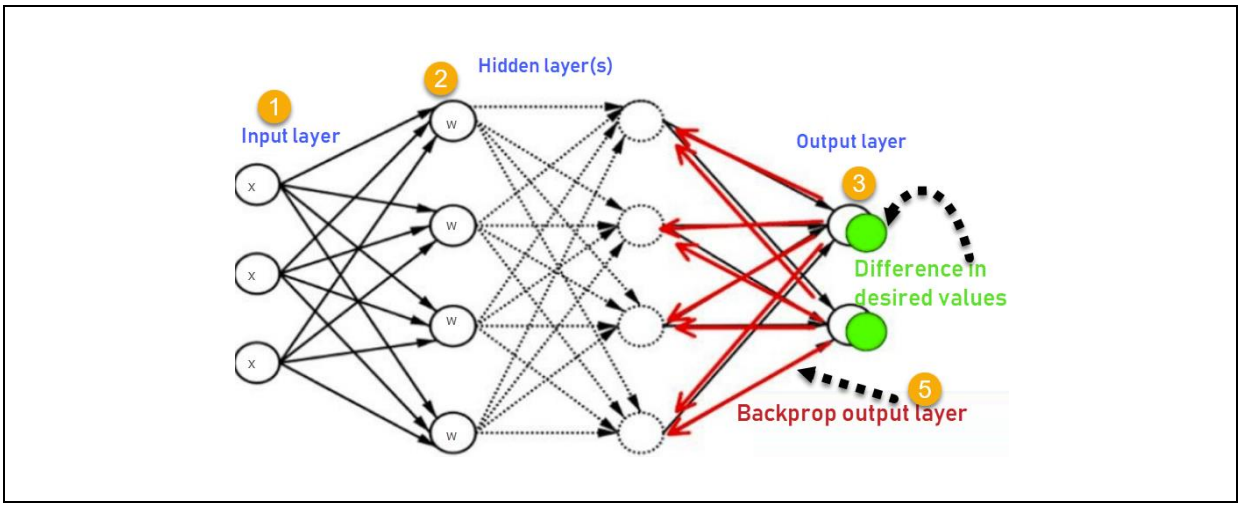
- ✓ Sigmoid
- ✓ Linear
- ✓ Tanh or hyperbolic tangent
- ✓ ReLU(Rectified Linear Units)
- ✓ Softmax

9. Forward Propagation

- ✓ In forward propagation, the information will be travelled into forward direction.
- ✓ The input layer provides input to the hidden layers and then the output is generated.
- ✓ In forward propagation input will not be travelled to backward direction.

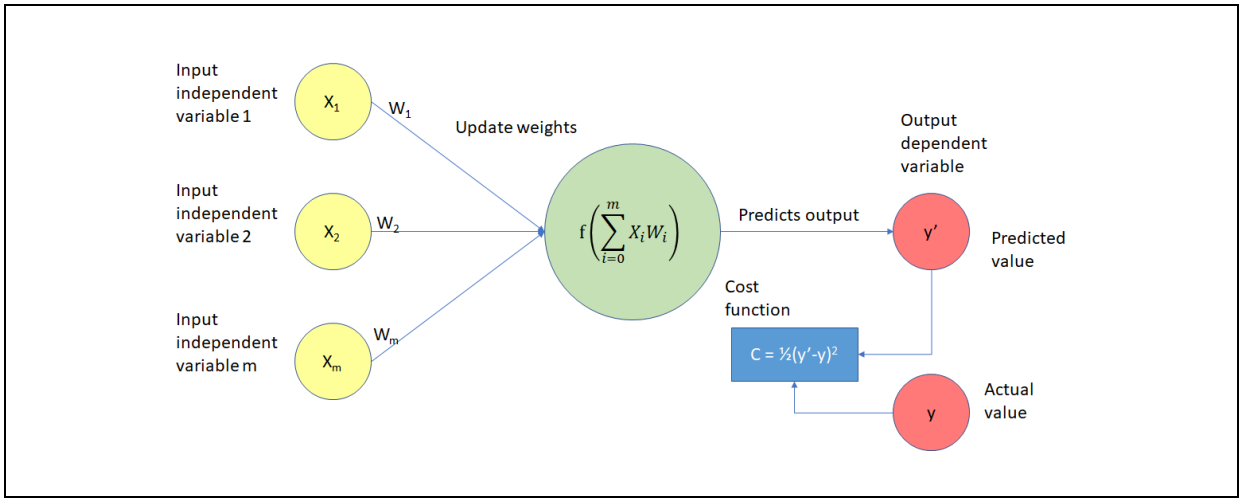
10. Back propagation

- ✓ During training, the network will get the results.
- ✓ These results will be compared with actual outputs by using loss/cost function.
- ✓ During comparing it will get error.
- ✓ To minimize this error internally weights supposed to be adjusted.
- ✓ So here back propagation helps to adjust the error.
- ✓ Back propagation means the,
    - The inputs results + error will travel in backward direction to adjust the weights



11. Cost Function

- ✓ When we create a network, the network tries to predict the output as close as possible to the actual value.
- ✓ We can measure this accuracy of the network by using the cost/loss function.
- ✓ The goal of running network is,
  - Increase our prediction accuracy
  - Reduce the error.
  - Minimizing the cost function.

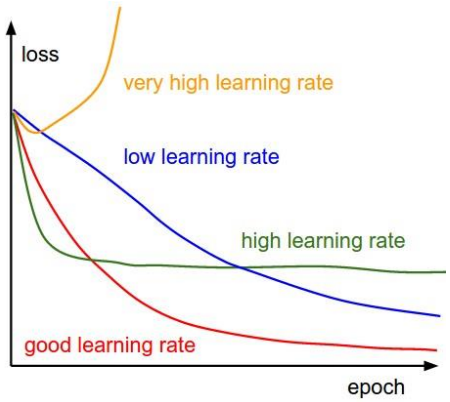


12. Gradient Descent

- ✓ Gradient descent is an optimization algorithm for minimizing the cost.

13. Learning Rate

- ✓ The learning rate is kind of hyper parameter to minimize the cost function in every iteration.
- ✓ We should choose the learning rate very carefully.
- ✓ If learning rate is large then it may miss minimum error point.
- ✓ If learning rate is very small then it takes long time to reach minimum error.
- ✓ So, optimize value is required.



14. Batches

- ✓ While training a neural network, instead of sending the entire input in onetime, generally it divides into several chunks of equal size randomly.
- ✓ It would be really good practice to train the model with batch of data instead of entire data.

15. Epochs

- ✓ The training of the neural network with all the training data for one cycle.