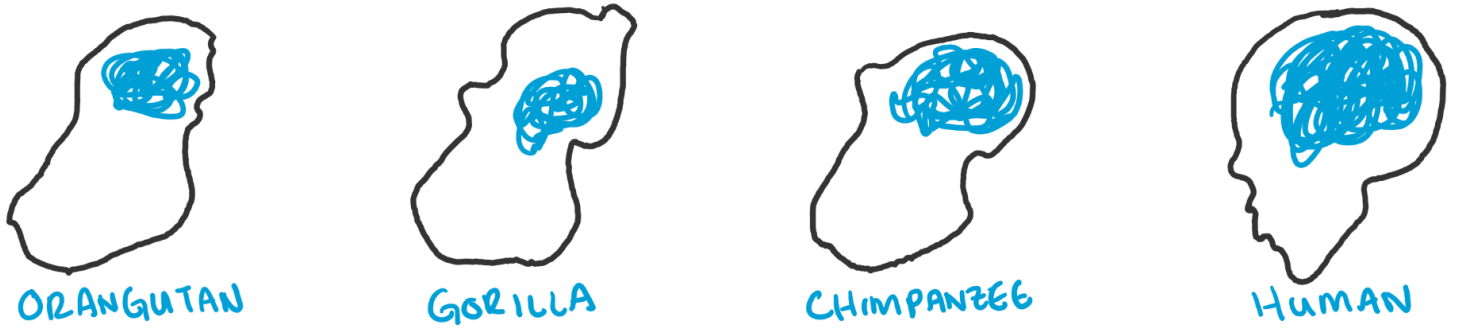
Deep Learning

"Deep learning is a particular kind of machine learning that is inspired by the functionality of our brain cells called neurons which led to the concept of artificial neural network"

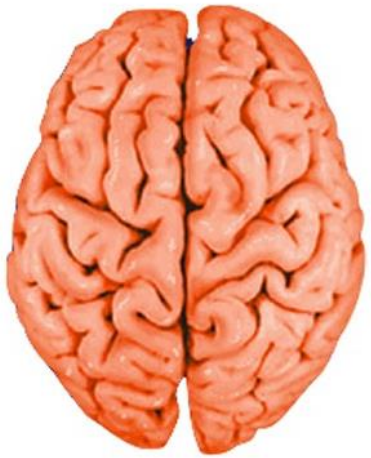
Anita Budhiraja
Scientist D

Our Evolutionary History

- Humans have the largest relative cerebral cortex size of all mammals. The blue area shows the brain in the skull.

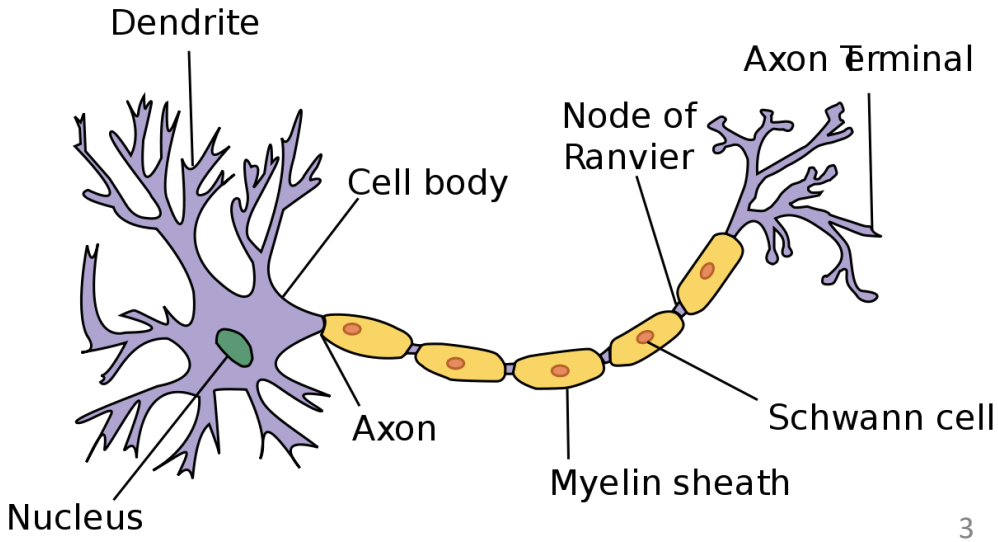


Human brain



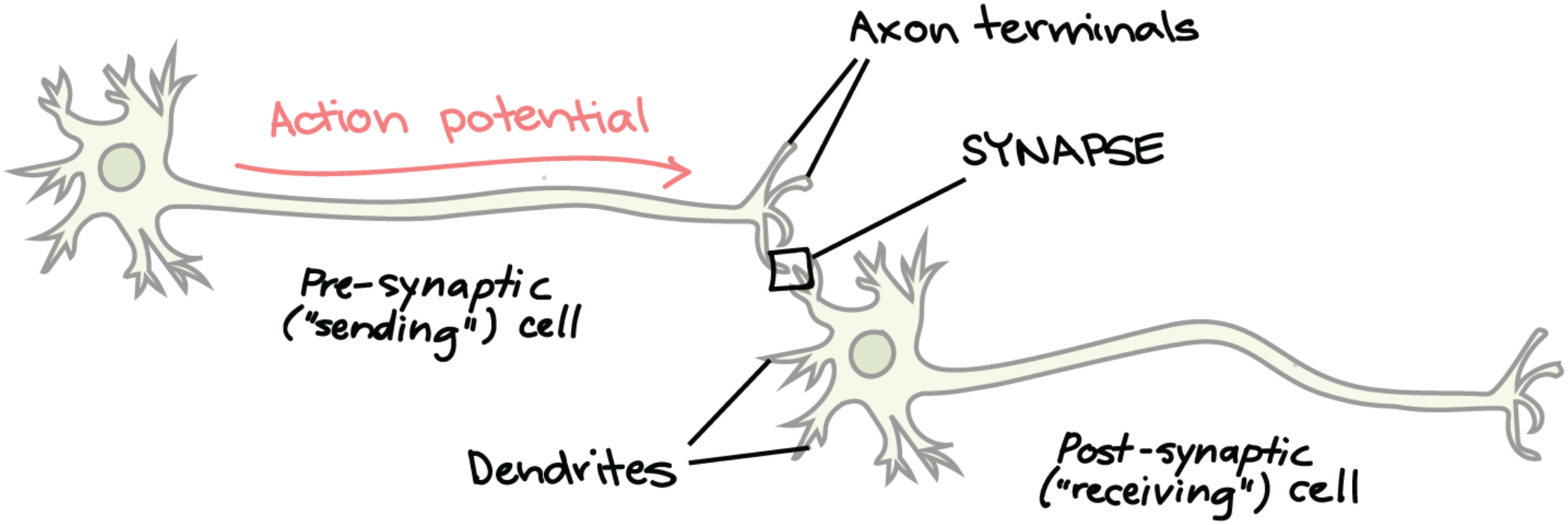
Neuron….*biological definition*

- A neuron is a nerve cell that carries electrical impulses.
- Neurons are the basic units of our nervous system.
- There are about 86 billion neurons in the human brain, which is about 10% of all brain cells.
- The human brain has about 16 billion neurons in the cerebral cortex.



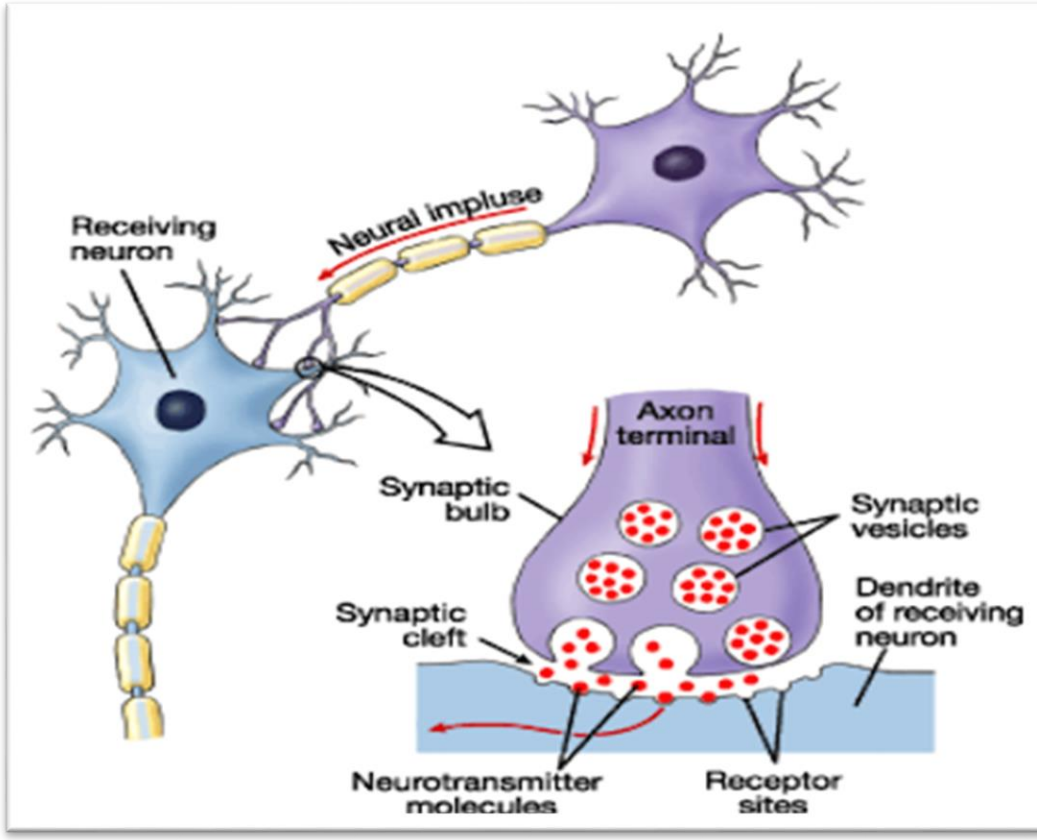
Neuron Synapse

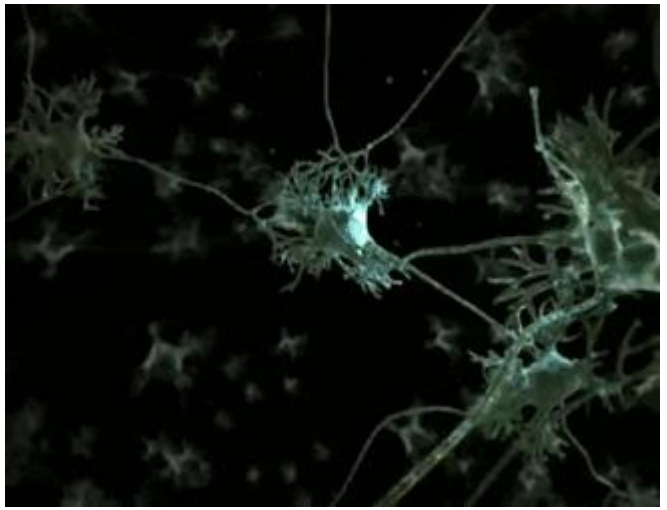
- Neurons are connected to one another in the brain, but they do not actually touch each other. Instead they have tiny gaps called synapses.
- Same way, synapses are like roads in a neural network. They connect inputs to neurons, neurons to neurons, and neurons to outputs.



Neuron Synapse

- In order to get from one neuron to another, you have to travel along the synapse paying the “toll” (weight) along the way.
- Each connection between two neurons has a unique synapse with a unique weight attached to it.
- When we talk about updating weights in a network, we’re really talking about adjusting the weights on these synapses.



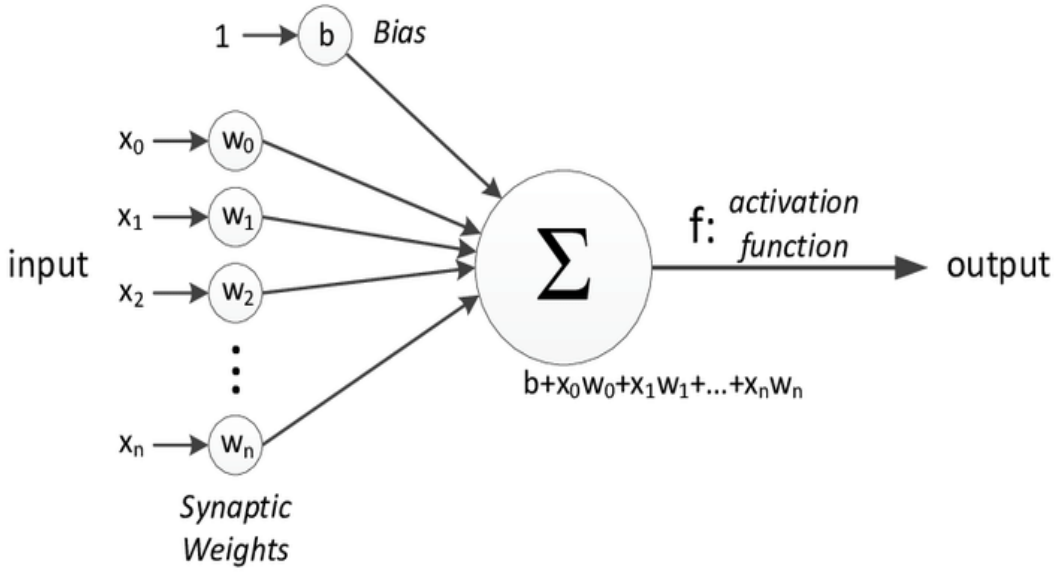


Artificial Neural Networks

- Artificial Neural Networks or ANN is an information processing paradigm that is inspired by the way the biological nervous system such as brain process information.
- It is composed of large number of highly interconnected processing elements (neurons) working in unison to solve a specific problem.
- *Perceptron is a general model of ANN which is inspired by a biological neuron.*
- A single layer neural network

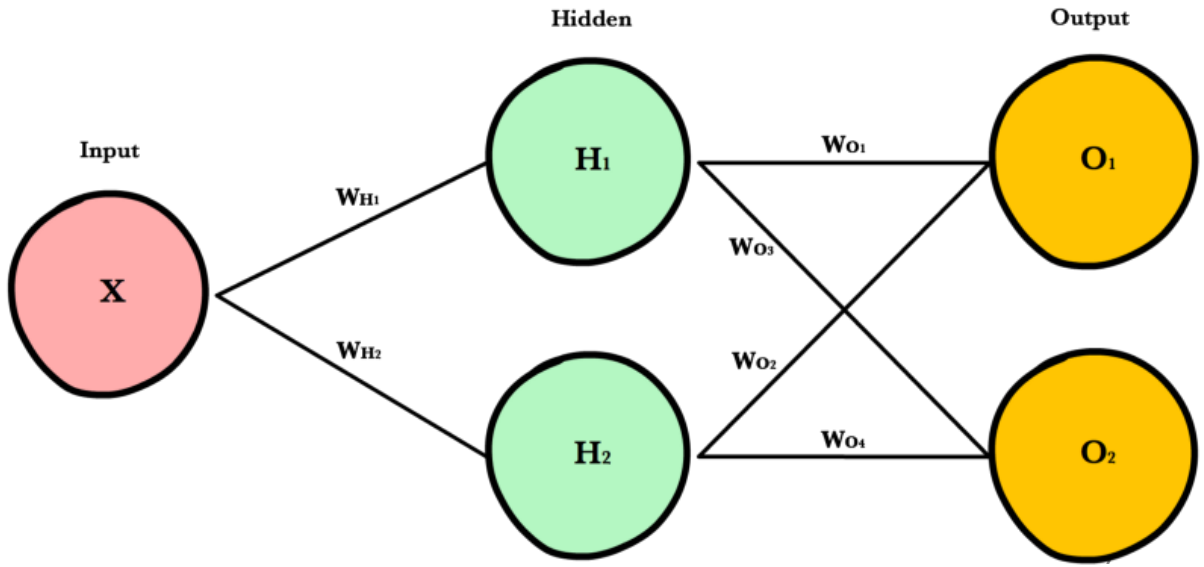
  is called a Perceptron.

  It gives a single output.



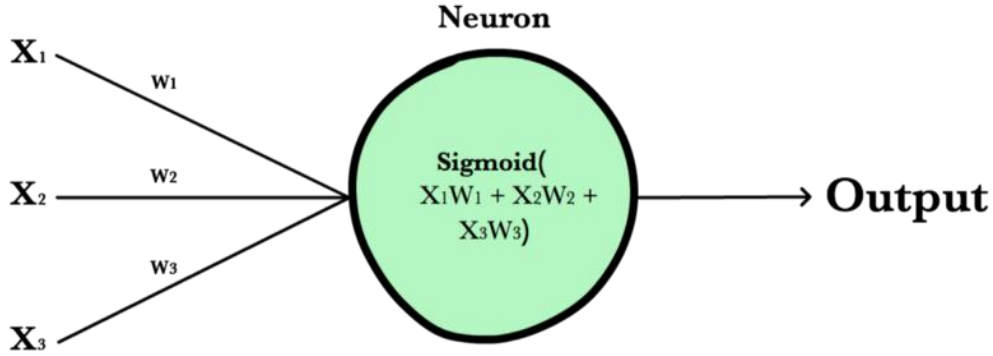
Neural Network

- Neural networks are a class of machine learning algorithms used to model complex patterns in datasets using multiple hidden layers and non-linear activation functions.
- A neural network takes an input, passes it through multiple layers of hidden neurons (mini-functions with unique coefficients that must be learned), and outputs a prediction representing the combined input of all the neurons.



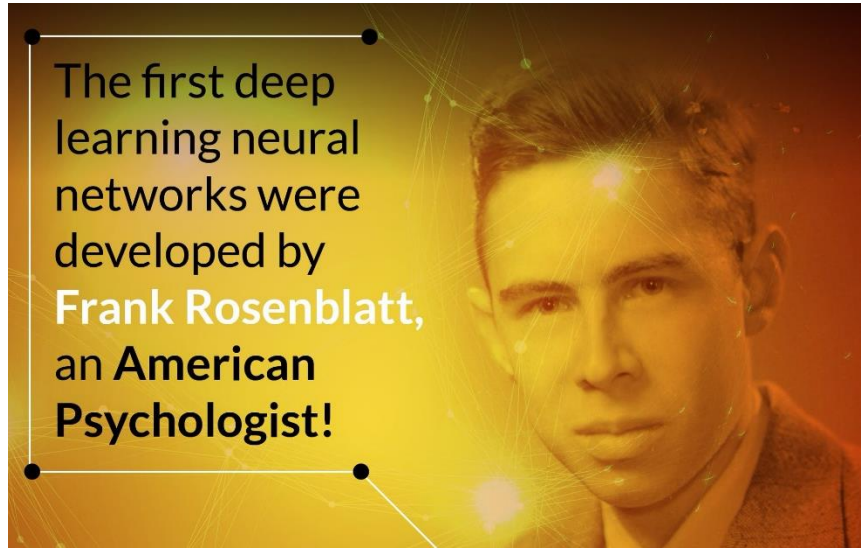
Neuron

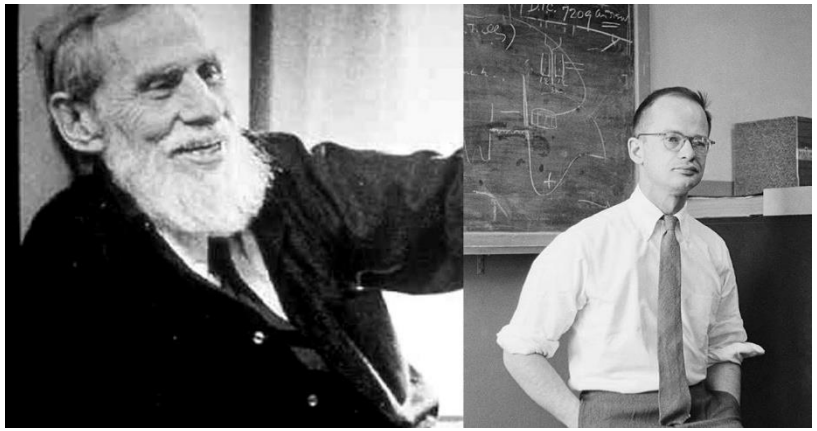
- A neuron takes a group of weighted inputs, applies an activation function, and returns an output.
- Inputs to a neuron can either be features from a training set or outputs from a previous layer's neurons.
- Weights are applied to the inputs as they travel along synapses to reach the neuron.
- The neuron then applies an activation function to the "sum of weighted inputs" from each incoming synapse and passes the result on to all the neurons in the next layer.

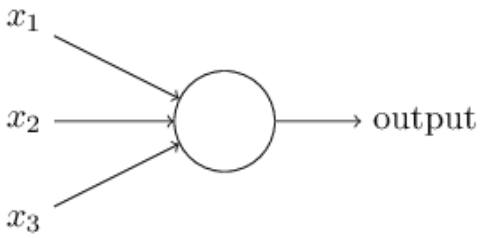


Perceptron … an artificial neuron

- Perceptrons were developed in the 1950s and 1960s by the scientist Frank Rosenblatt, inspired by earlier work by Warren McCulloch and Walter Pitts.
- The main neuron model used is one called the *sigmoid neuron.*
- A perceptron takes several binary inputs, $x_1, x_2, \ldots$ x1,x2,…, and produces a single binary output

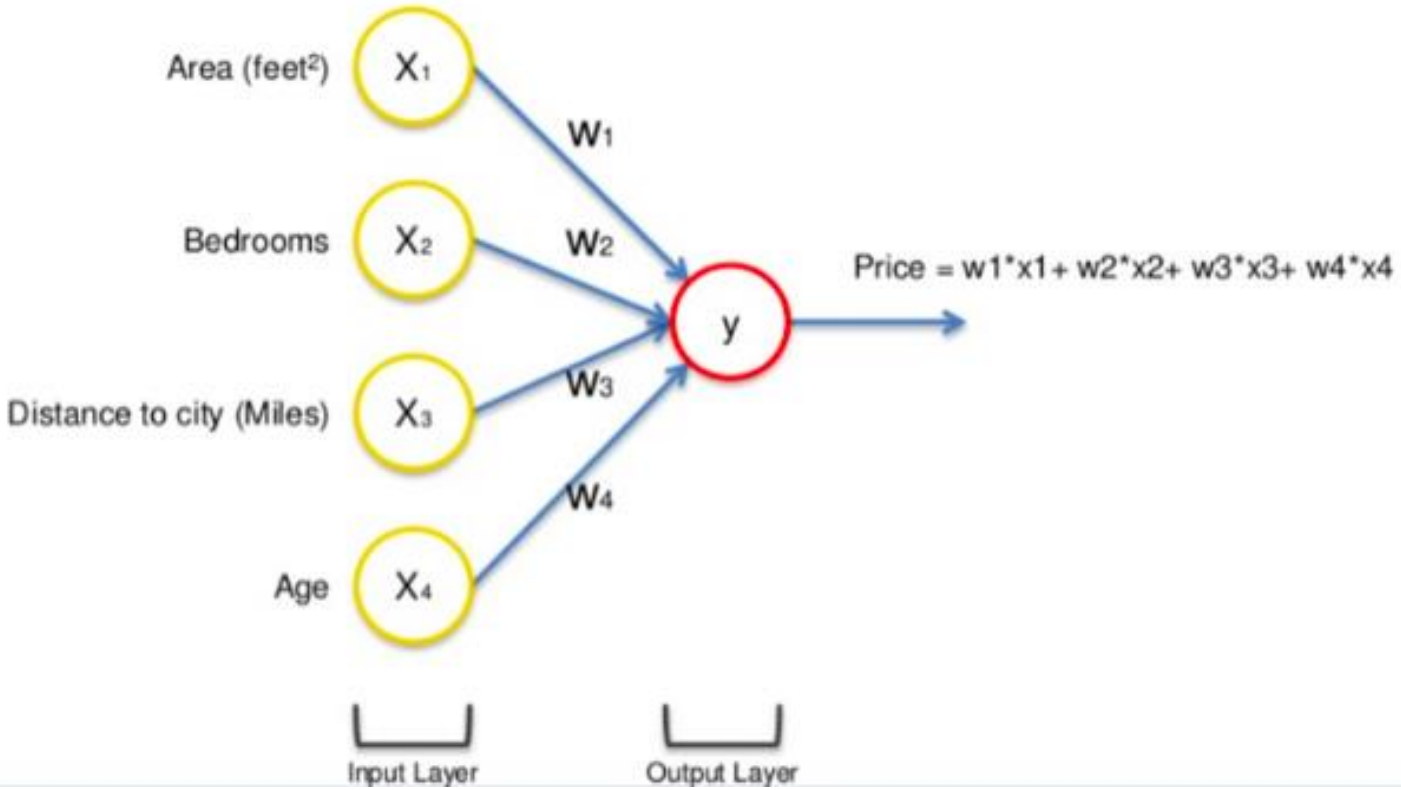




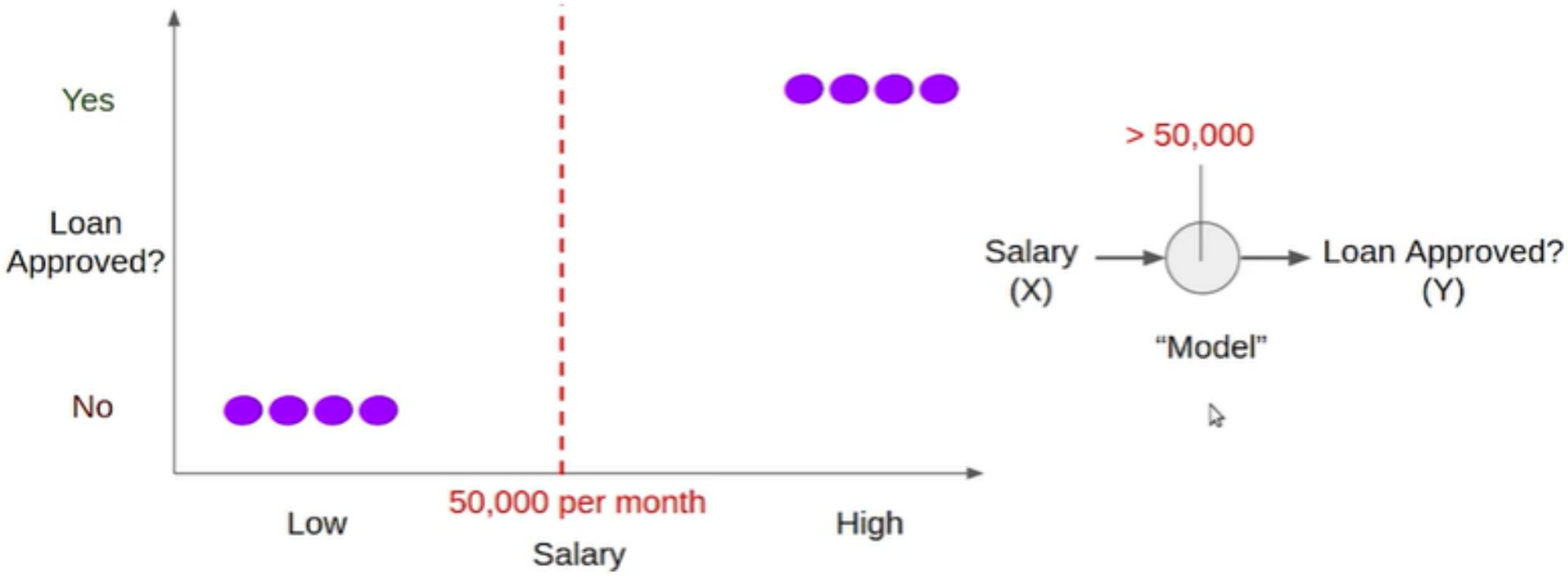


How Neural Network work

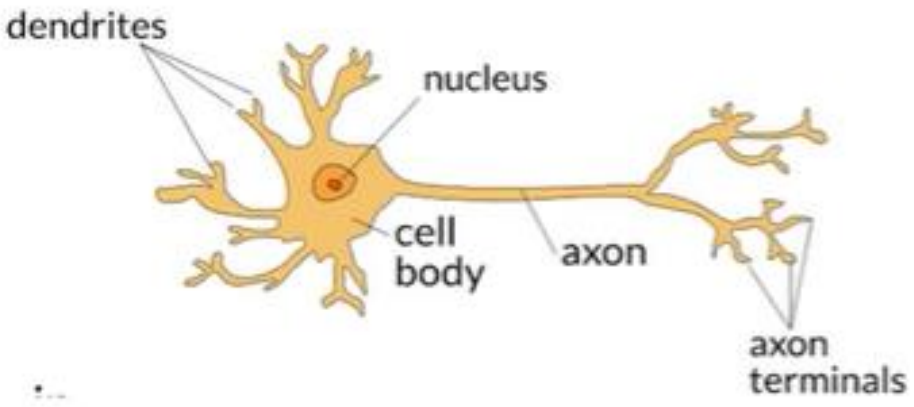
- example of the price of a property



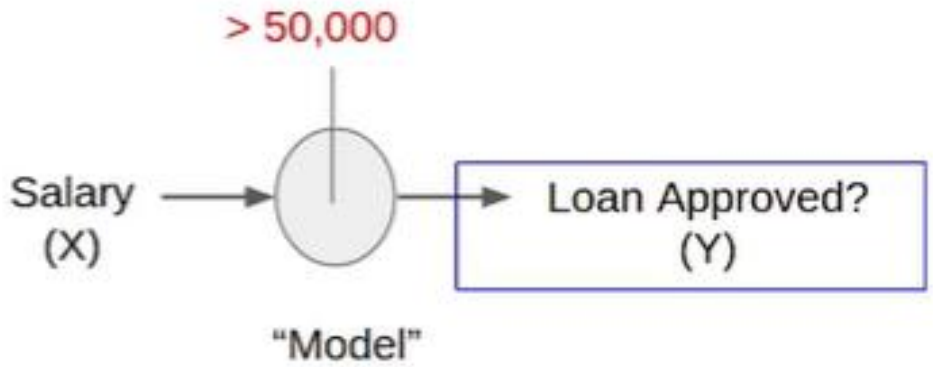
Understanding Neural Networks
Model to Approve Loan : Classification



Model to Approve Loan : Classification



Biological Neuron



Neuron

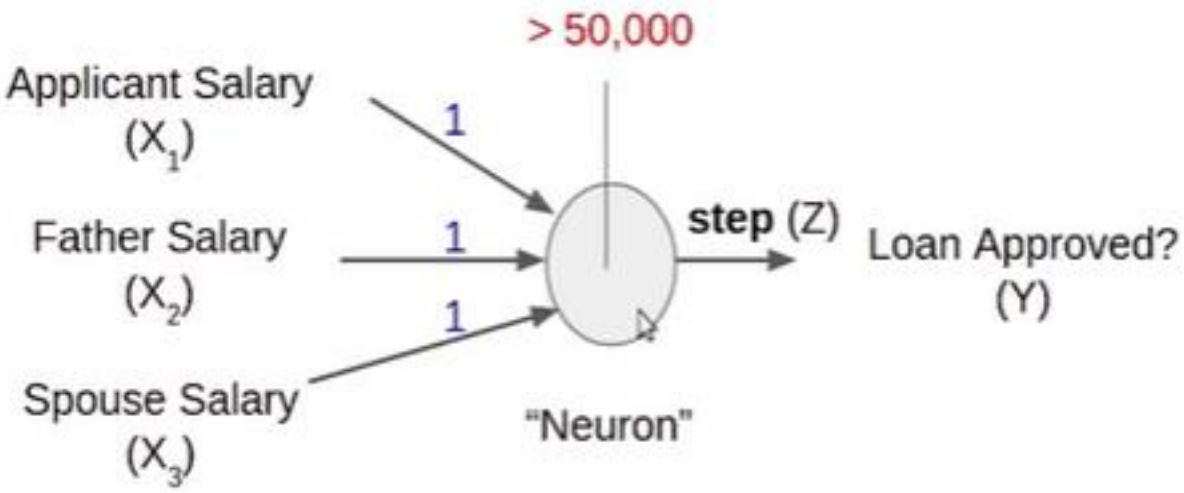
Weights…Bias

- Bias terms are additional constants attached to neurons and added to the weighted input before the activation function is applied.
- Bias terms help models represent patterns that do not necessarily pass through the origin.
- For example, if all your features were 0, would your output also be zero? Is it possible there is some base value upon which your features have an effect?
- Bias terms typically accompany weights and must also be learned by your model.

Model to Approve Loan : Classification

Multiple inputs like multiple dendrites.

Bias is something learnt by the neuron from the underlying data. If the input exceeds the magnitude of the bias, we want neuron should give yes. That is the loan can be given. This event is firing of a neuron.



Total Income > threshold

$X_1+X_2+X_3$ > threshold

$X_1+X_2+X_3$ - threshold > 0

"bias"

$X_1+X_2+X_3$ + bias > 0

Relationship in form of equation

if $X_1+X_2+X_3$ + bias > 0 then output should be 1

if $X_1+X_2+X_3$ + bias ≤ 0 then output should be 0

Activation Function : Graph of AF

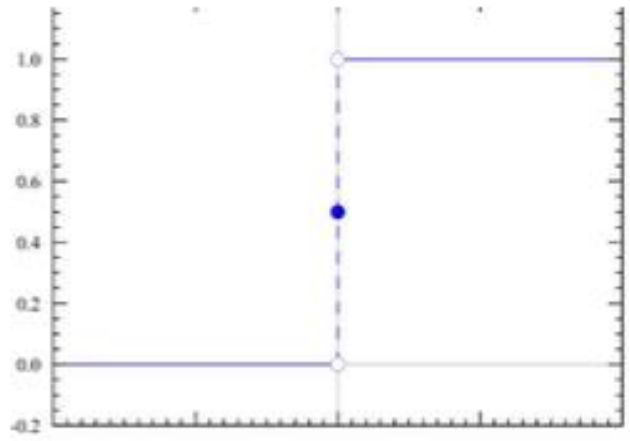
$Z = X_1+X_2+X_3 + bias$

Output = will be 1 if ($Z = X_1+X_2+X_3 + bias$) > 0 , it will be 0 otherwise.

Use Step function!

$$Output = \begin{cases} 1, Z > 0 \\ 0, Z \leq 0 \end{cases}$$

Output = step function (Z) or Output = **step** (Z)



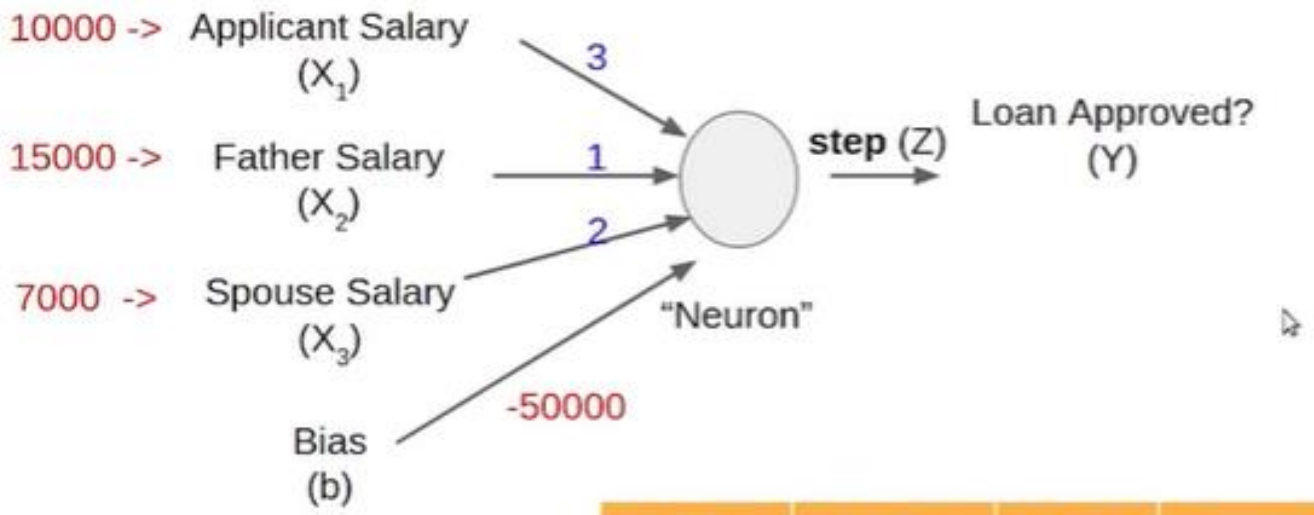
Step function is used to scale the output of the neuron. IN DL , we can choose these functions, these functions are called activation functions. Such step functions used as activation functions for a neuron, it is called perceptron.

Working of perceptron

- Rosenblatt proposed a simple rule to compute the output.
- He introduced *weights*, $w_1, w_2, \ldots$, real numbers expressing the importance of the respective inputs to the output.
- The neuron's output, 0 or 1, is determined by whether the weighted sum $\sum_j w_j x_j$ is less than or greater than some threshold value.

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

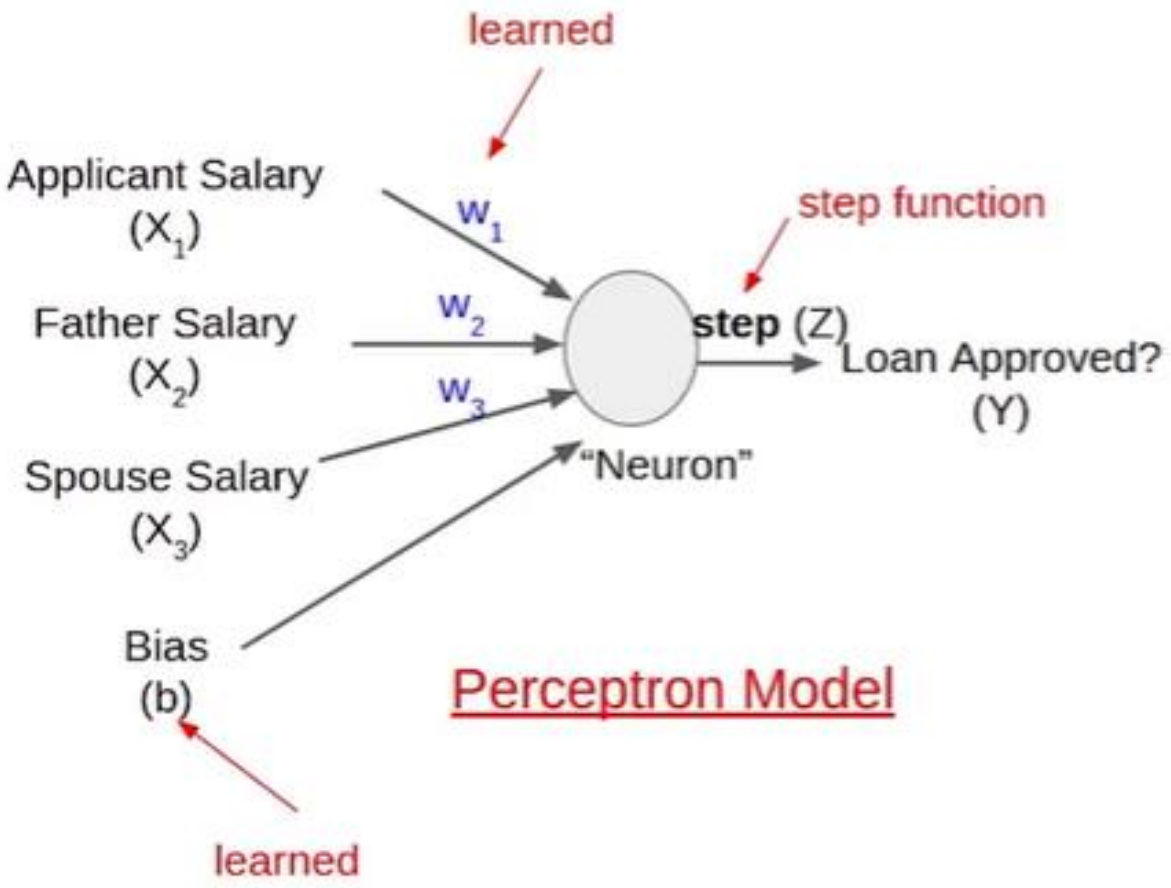
Weights in Perceptron



X1*w1	X2*w2	X3*w3	Sum of inputs	Z (Sum of inputs + bias)	step (Z)
10000*3	15000*1	7000*2	59000	9000	

Since 9000 is greater than 0, the function would give an output of 1 ie loan should be approved.
Y becomes the final output which is the output of the neuron. Neuron with a step function is called Perceptron model. Weights and bias are learnt during the training process.

Perceptron Model

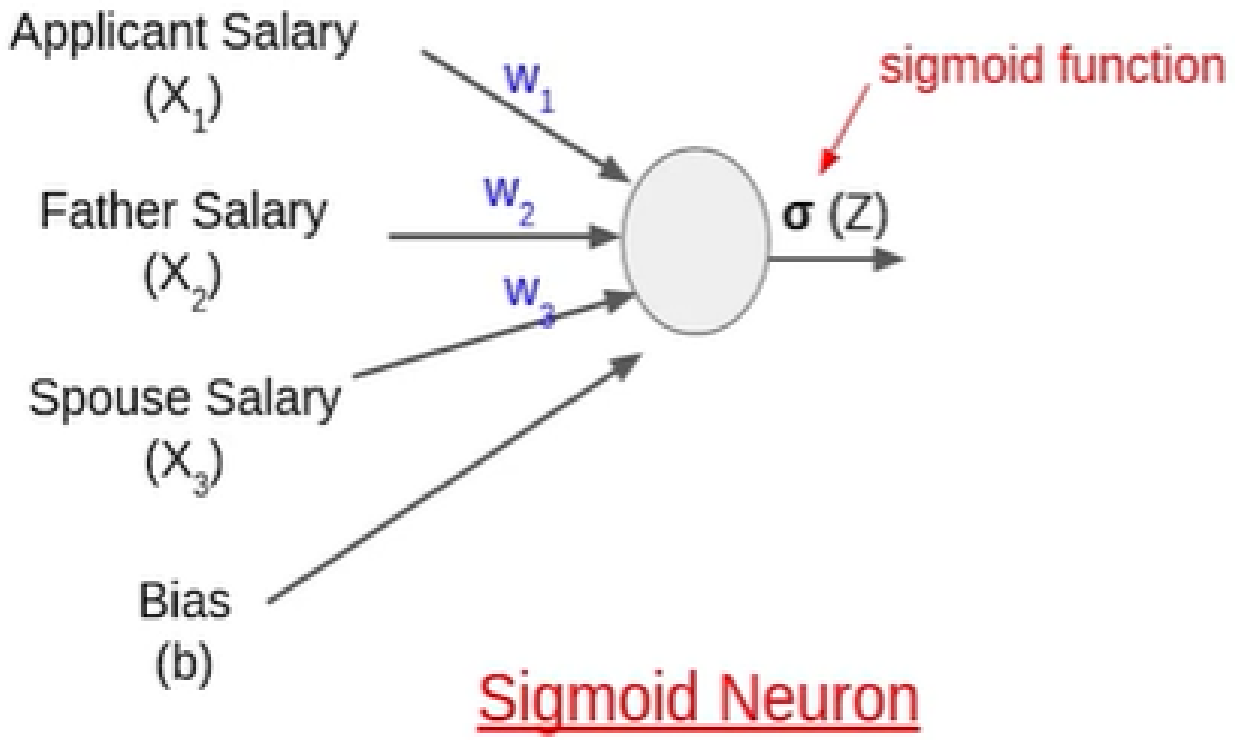


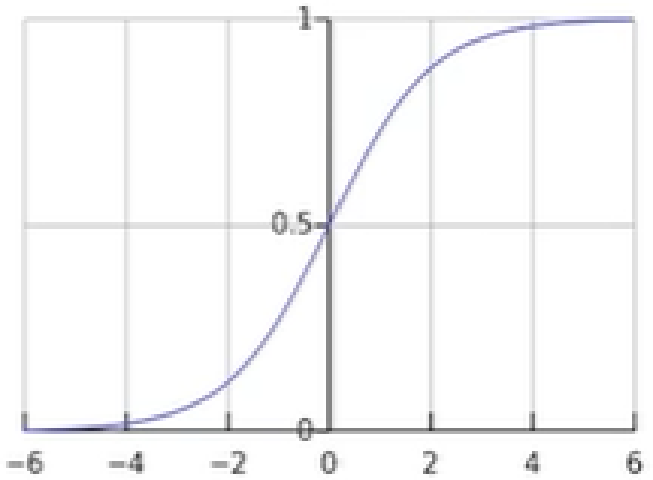
Sum of inputs = $X_1 * w_1 + X_2 * w_2 + X_3 * w_3$

$Z = X_1 * w_1 + X_2 * w_2 + X_3 * w_3 + b$ (bias)

$\hat{Y}$ (output) = **step** (Z)

Sigmoid Activation Function (Logistic Regression)





Sum of inputs = $X_1 * w_1 + X_2 * w_2 + X_3 * w_3$

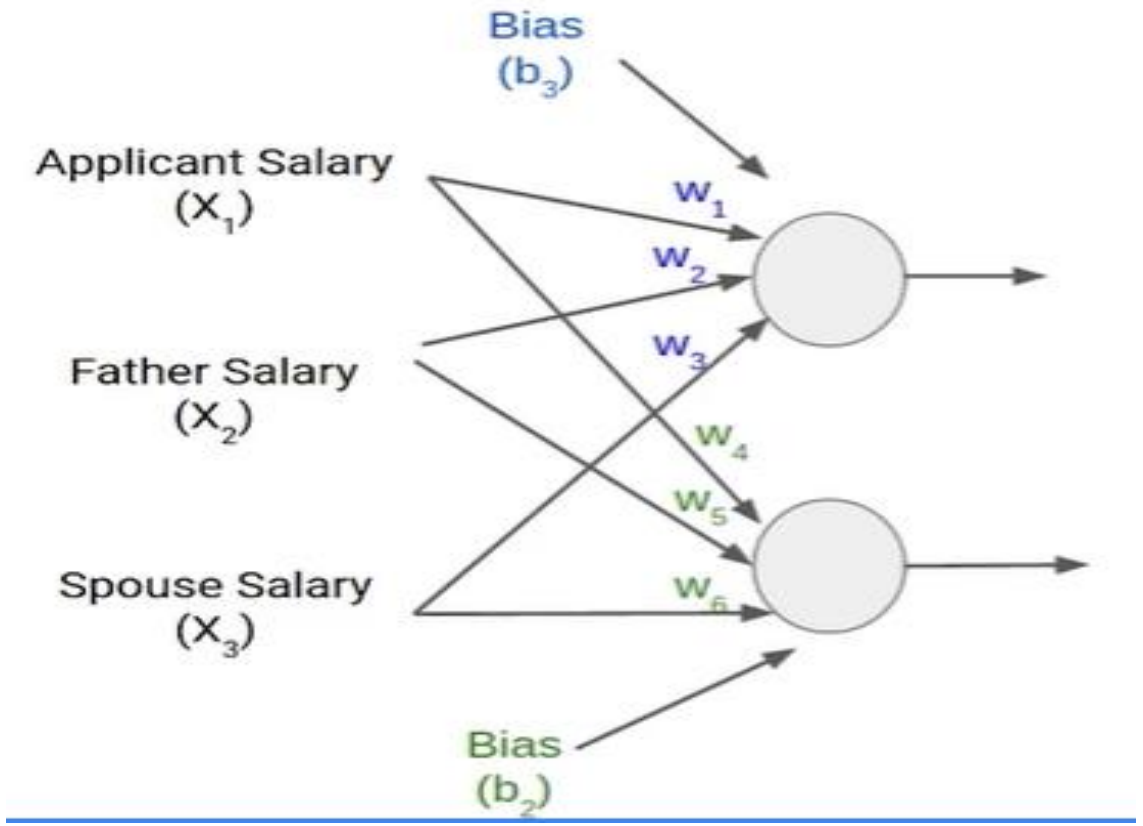
$Z = X_1 * w_1 + X_2 * w_2 + X_3 * w_3 + b$ (bias)

$\hat{Y}$ (output) = σ (Z)

Sigmoid function is used as an activation function, then equation of perceptron model will be that the range of input values can be anything between –infinity to infinity. But the output will be 0 or 1 only. The value returned by sigmoid can be anything between 0 and 1 so they can be considered as probabilities.

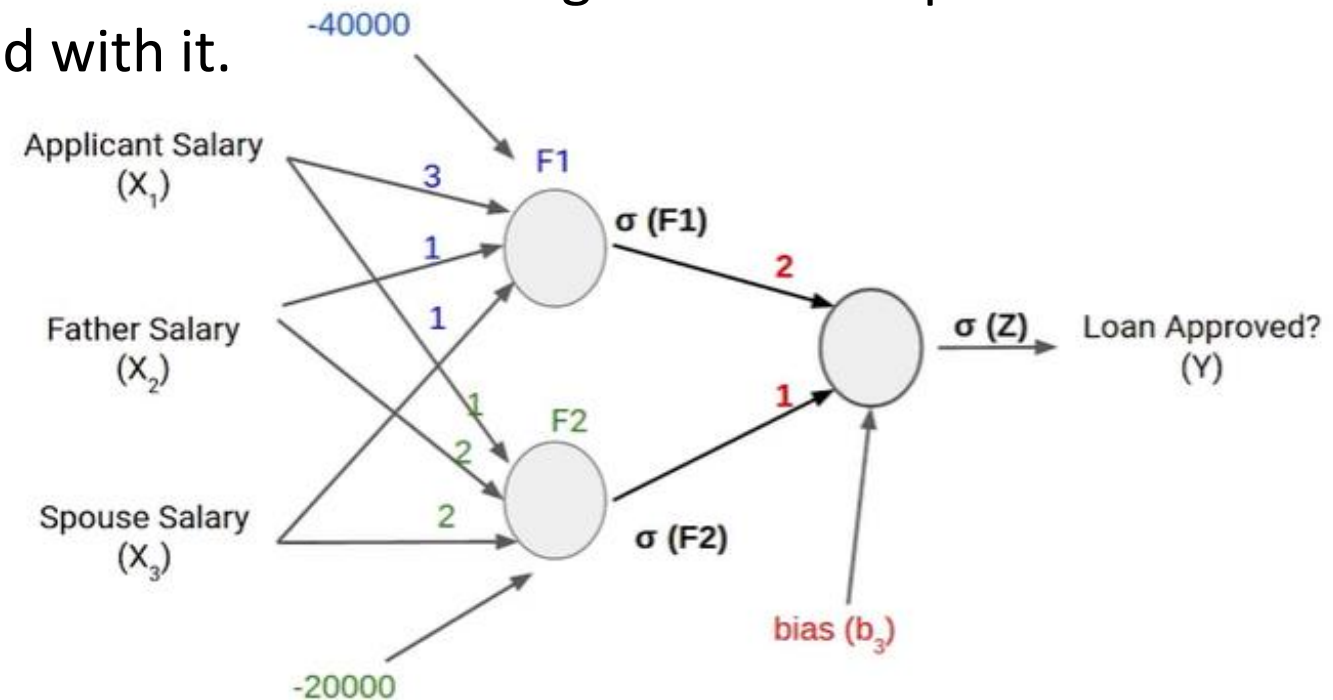
Single-layer Perceptron

- More than one neuron. Each neuron assigns weights to the input features.
- Some bias values will be associated with these neurons.

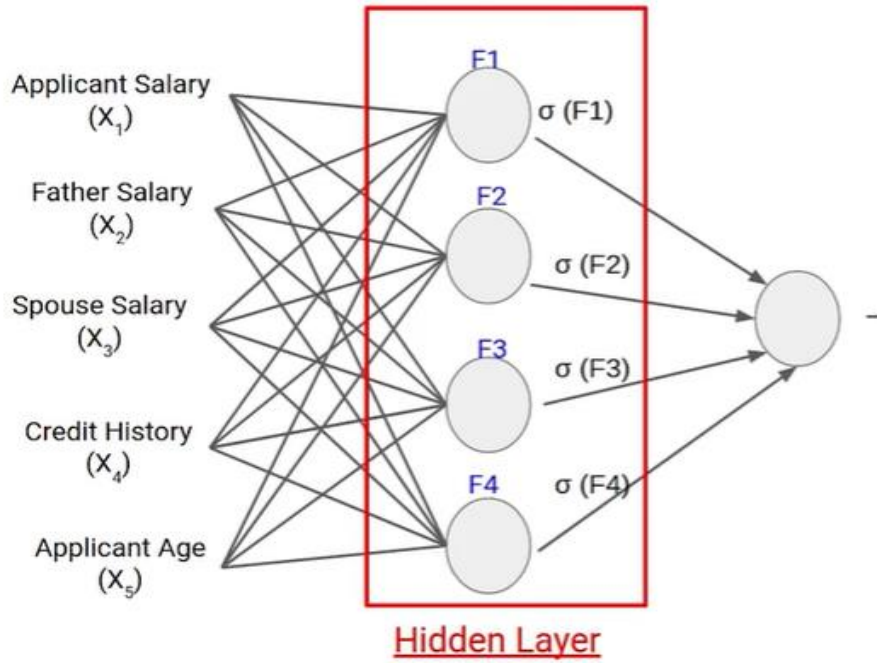


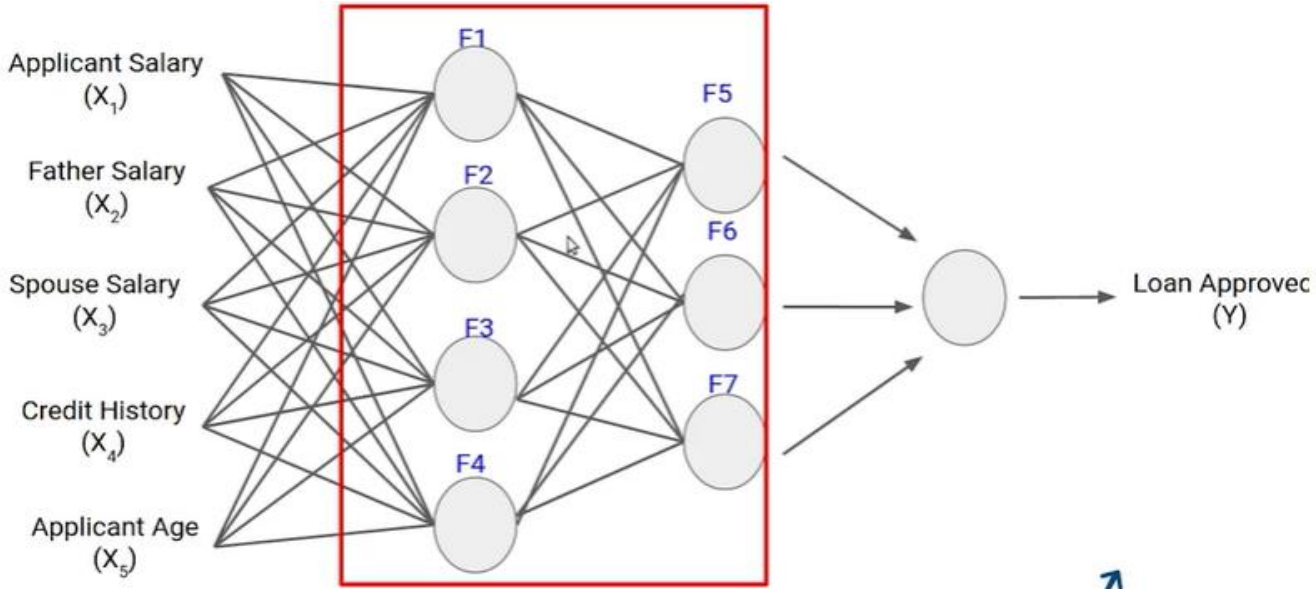
Multi-layered Perceptron

- Each neuron is automatically learning and creating a different feature from existing data on its own eg F1 and F2.
- The layer that performs the mathematical operations to generate the features are called the hidden layers.
- This happens during the entire training process.
- Then activation function is applied on the results.
- New neuron is created that works on calculating the final output which has some weight and bias associated with it.



Single-layered vs Multi-layered Perceptron





- No of output neurons depends on the number of classes in output.
- No of neurons in the hidden layer can be increased. Each of these will be extracting some features from the existing data like F1, F2, F3, F4.

- New Features like F1, F2, F3, F4 can further create new features like F5, F5, F7.
- Then there will be two hidden layers.

No of output neurons depends on the number of classes in output.

Weighted Input

- A neuron's input equals the sum of weighted outputs from all neurons in the previous layer. Each input is multiplied by the weight associated with the synapse connecting the input to the current neuron. If there are 3 inputs or neurons in the previous layer, each neuron in the current layer will have 3 distinct weights — one for each synapse.
- Single Input

  Z = Input ·Weight = XW
- Multiple Input

$$Z = \sum_{i=1}^{n} x_i w_i$$
$$= x_1 w_1 + x_2 w_2 + x_3 w_3$$

- it's exactly the same equation we use with linear regression!
- In fact, a neural network with a single neuron is the same as linear regression!
- The only difference is the neural network post-processes the weighted input with an activation function.

References

- Medium.com - Blockchain
- neuralnetworksanddeeplearning.com
- Fundamentals of Deep Learning - *Designing Next-Generation Machine Intelligence Algorithms* … Nikhil B
- Wikipedia
- Edureka
- AnalyticsVidya.com