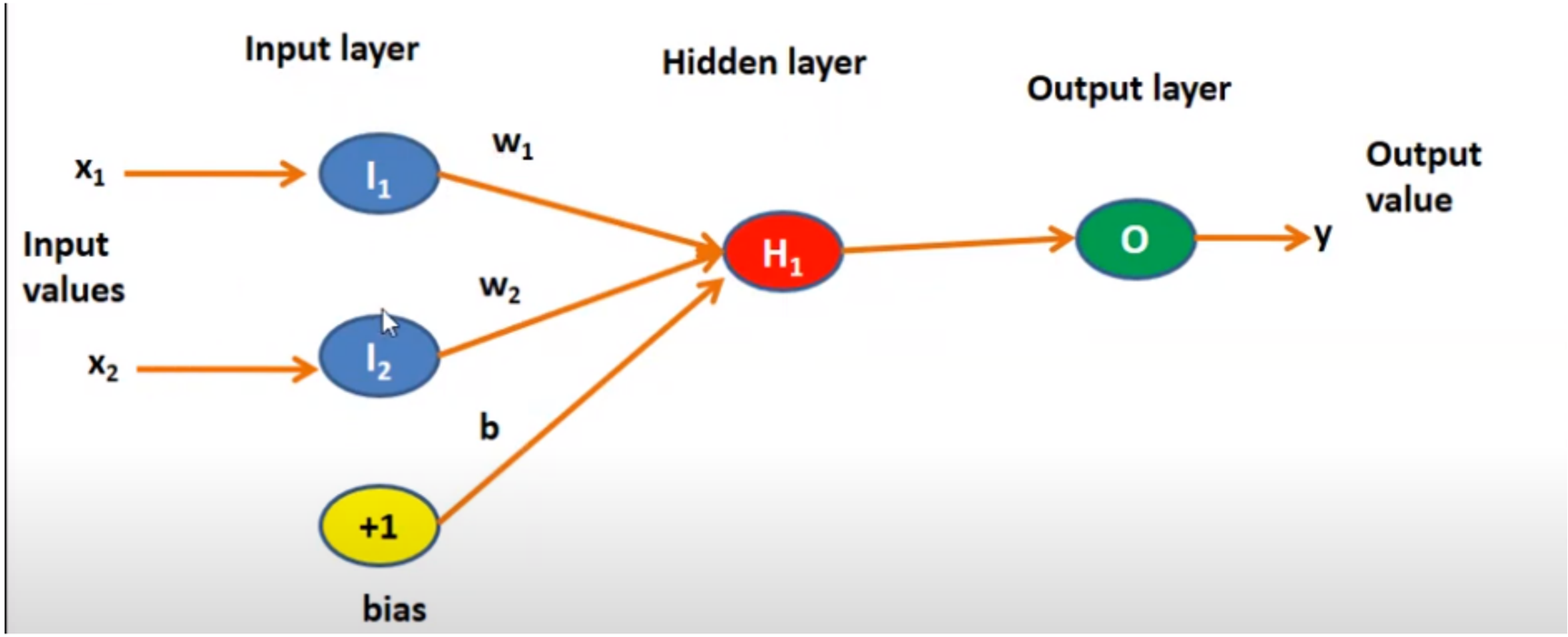
# *Artificial Neural Network*

# Artificial Neural Network

- Its a network which can solve Artificial intelligence problems.
- Decision Making Tools, which can be used to model complex relationships between input and output.
- ANN is a supervised learning system built of a large number of simple elements, called neurons or perceptrons.
- Each neuron can make simple decisions, and feeds those decisions to other neurons, organized in interconnected layers.
- Together, the neural network can emulate almost any function, and answer practically any question, given enough training samples and computing power.
- A simple neural network has only three layers of neurons:
- An input layer that accepts the independent variables or inputs of the model
- One hidden layer
- An output layer that generates predictionst

# Simple Artificial Neural Network



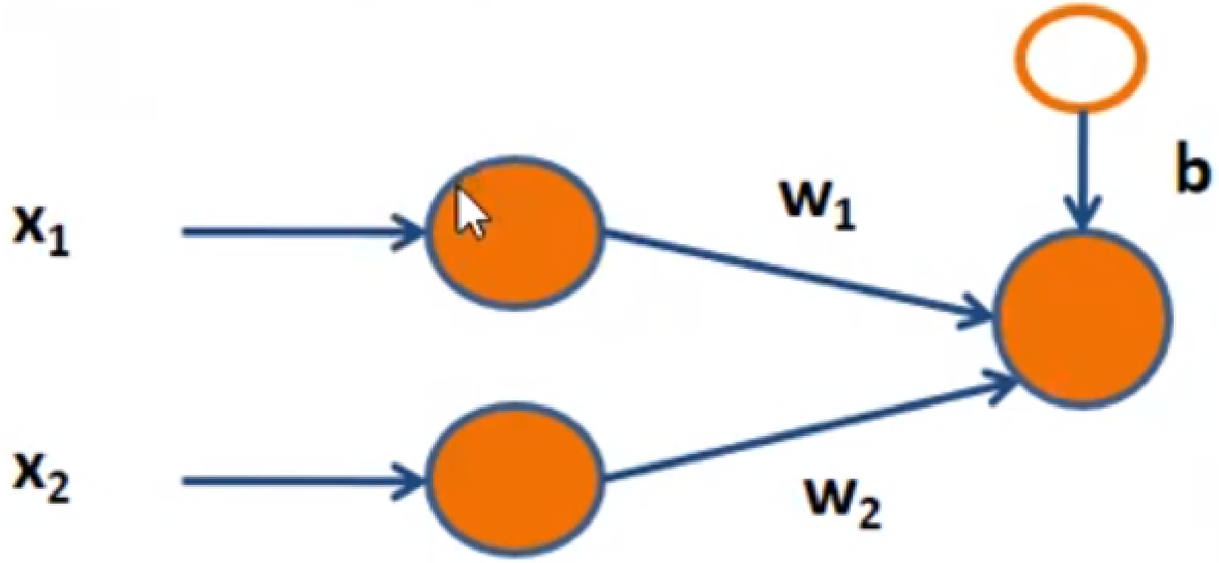

Step-1: each input is multiplied by a weight Input * weight= $X_1$* $W_1$+ X2* W2
Step-2: all the weighted inputs are added together with a bias b = (X1 * $W_1$) + (X2 * $w_2$) + b S=
Step-3: the sum is passed through an activation function y = f(s)

# Basic Terminology

- Neuron
- Hidden layer
- Connections
- Bias
- Inputs Layer
- Variance
- Output Layer
- Weights
- Error Function
- Activation function
- Hyperparameters

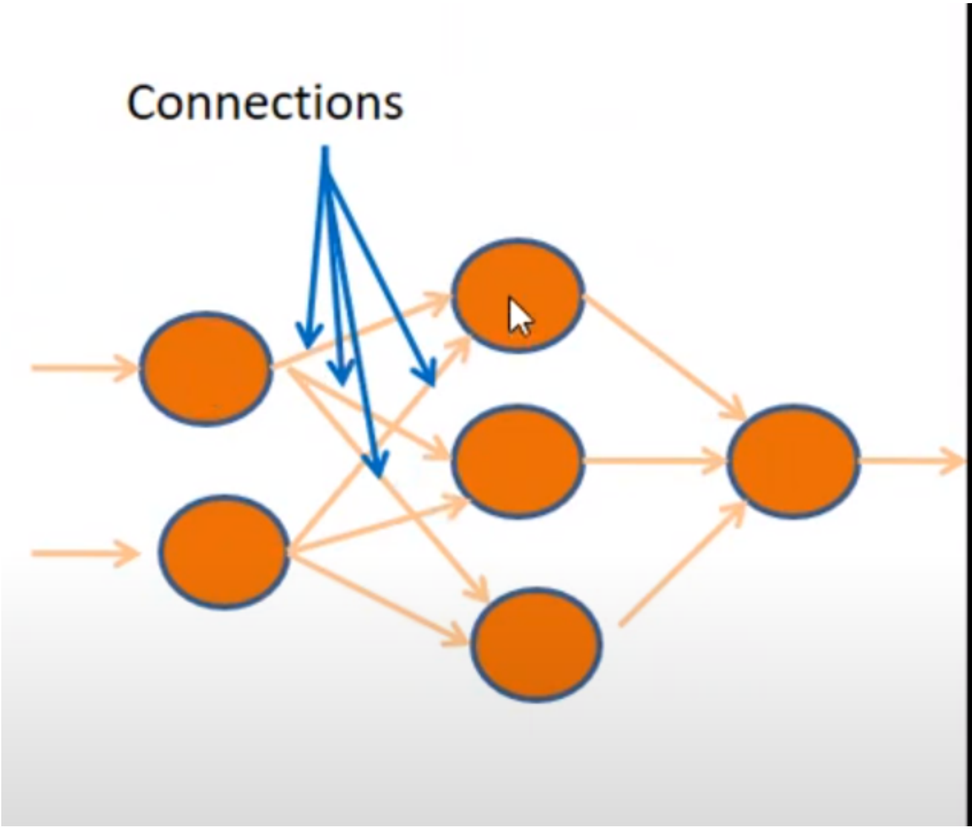
# Neuron(Node)

- It is the basic unit of a neural network. It gets certain number of inputs and a bias value. When a signal(value) arrives, it gets multiplied by a weight value.
- If a neuron has 2 inputs, it has 2 weight values which can be adjusted during training time.



# Connections

- It connects one neuron in one layer to another neuron in other layer or the same layer.
- A connection always has a weight value associated with it.
- Goal of the training is to update this weight value to decrease the loss(error).



# Input Layer

- The input layer is the most responsible layer for receiving the inputs and these inputs are loaded from some external sources like csv file, text file, web service etc.
- This is the first layer in the neural network. It takes input signals(values) and passes them on to the next layer. It doesn't apply any operations on the input signals(values) & has no weights and biases values associated.
- The number of neurons in an input layer depends on the shape of the training data.

  Input Shape - It is the shape of the input matrix we pass to the input layer. Our network's input layer has 2 neurons and it expects 2 values of 1 sample. Desired input shape for our network is (1, 2, 1) if we feed it one sample at a time. If we feed 100 samples input shape will be (100, 2, 1).

# Output Layer

- The output layer is mostly responsible for producing the final output results.
- Neural networks generate their predictions in the form of a set of real values or Boolean decisions.
- Each output value is generated by one of the neurons in the output layer.
- The output layer takes the inputs which are passed in from the layers before it, and performs the calculations through its neurons and then the output is computed.
- In complex neural networks the output layer receives inputs from the previous hidden layers.

# Weights

- Each neuron is given a numeric weight.
- A weight represent the strength of the connection between units. If the weight from node 1 to node 2 has greater magnitude, it means that neuron 1 has greater influence over neuron 2.
- Weights near zero means changing this input will not change the output. Negative weights mean increasing this input will decrease the output. A weight decides how much influence the input will have on the output.
- The weights, together with the activation function, define each neuron's output.
- Neural networks are trained by fine-tuning weights, to discover the optimal set of weights that generates the most accurate prediction.

# Activation Function

- Basically the activation functions decides weather the neuron activated or not.

$$G = \text{Activation}(\Sigma(\text{weight} * \text{input}) + \text{bias})$$

- Common activation functions are sigmoid, TanH (hyperbolic tangent), ReLu (Rectified Linear Unit) etc.
- It squashes the values in a smaller range viz. a Sigmoid activation function squashes values between a range 0 to 1.
- Activation functions help generate output values within an acceptable range, and their non-linear form is crucial for training the network.

# Hidden Layer

- The hidden layers are placed in between the input and output layers.
- These are not visible to the external systems and these are private to the neural networks.
- Hidden layers have neurons(nodes) which apply different transformations to the input data. One hidden layer is a collection of neurons stacked vertically
- For large majority of problems one hidden layer is sufficient.
- Training Set
- A set of inputs for which the correct outputs are

# Bias

- It is an extra input to neurons and it is always 1, and has it's own connection weight. This makes sure that even when all the inputs are none (all O's) there's going to be an activation in the neuron.
- Bias neuron is used in every layer of the neural network.
- These are extra neurons added to each layer, which store the value of 1
- When training neural networks, we try to balance between bias and variance.
- Bias measures how well the model fits the training set

# Variance

- It measures how well the model works with unknown inputs that were not available during training.
- able to make accurate predictions on the validation set.

# Error/ Loss Function

- Defines how far the actual output of the current model is from the correct output.
- When training the model, the objective is to minimize the error function and bring output as close as possible to the correct value.
- Eg. Mean Squared Error
  MSE function is used for regression tasks. Here loss is calculated by taking the mean of squared differences between actual(target) and predicted values.
- Loss is nothing but a prediction error of Neural Network. And the method to calculate the loss is called Loss Function.
- Loss is used to calculate the gradients. And gradients are used to update the weights of the Neural Network

# Error/ Loss Function

- A hyperparameter is a setting that affects the structure or operation of the neural network.
- These include learning rate, number of hidden layers, number of neurons in a given layer etc.
- Tuning the hyperparameters refers to the process of choosing the best values of the hyperparameters.
- Tuning is the primary way to build a network that provides accurate predictions for a certain problem.
- Must be set manually and tuned.

# *THANK-YOU*