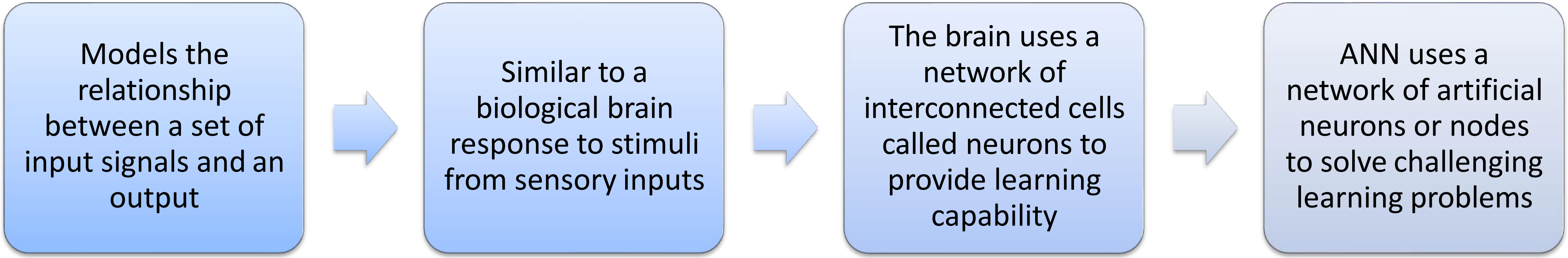
Artificial Neural Network

Artificial neural network

- A Machine Learning algorithm that is roughly modelled around what is currently known about how the human brain functions.

Understanding neural network



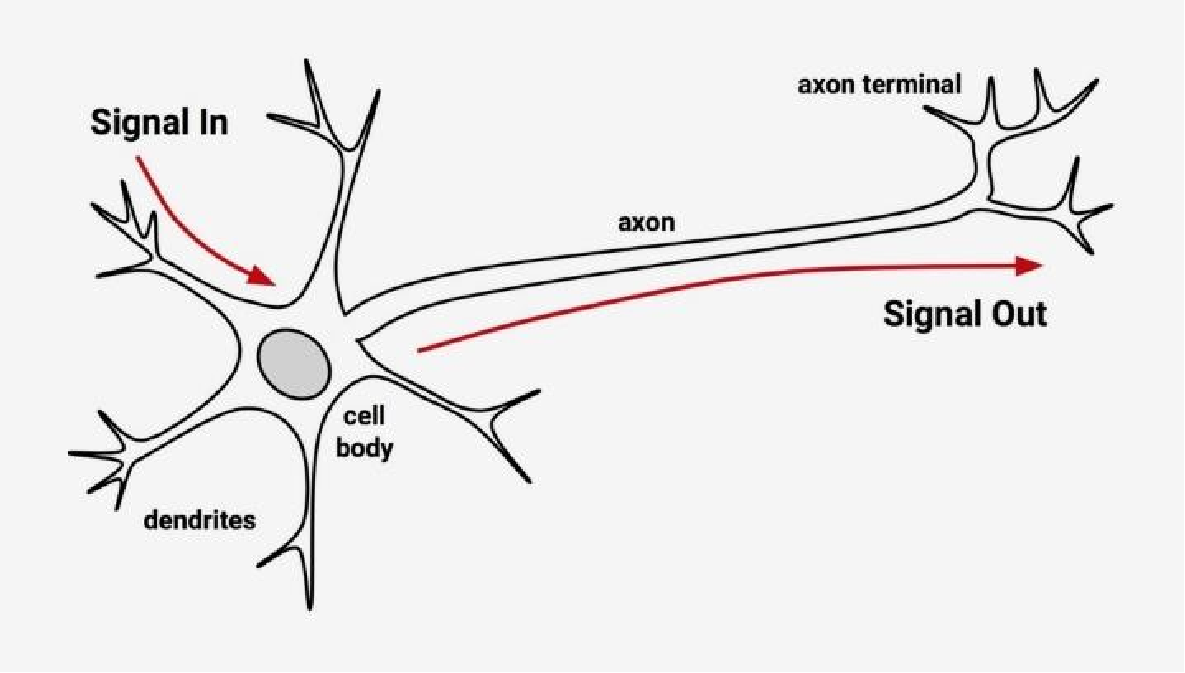
Models the relationship between a set of input signals and an output

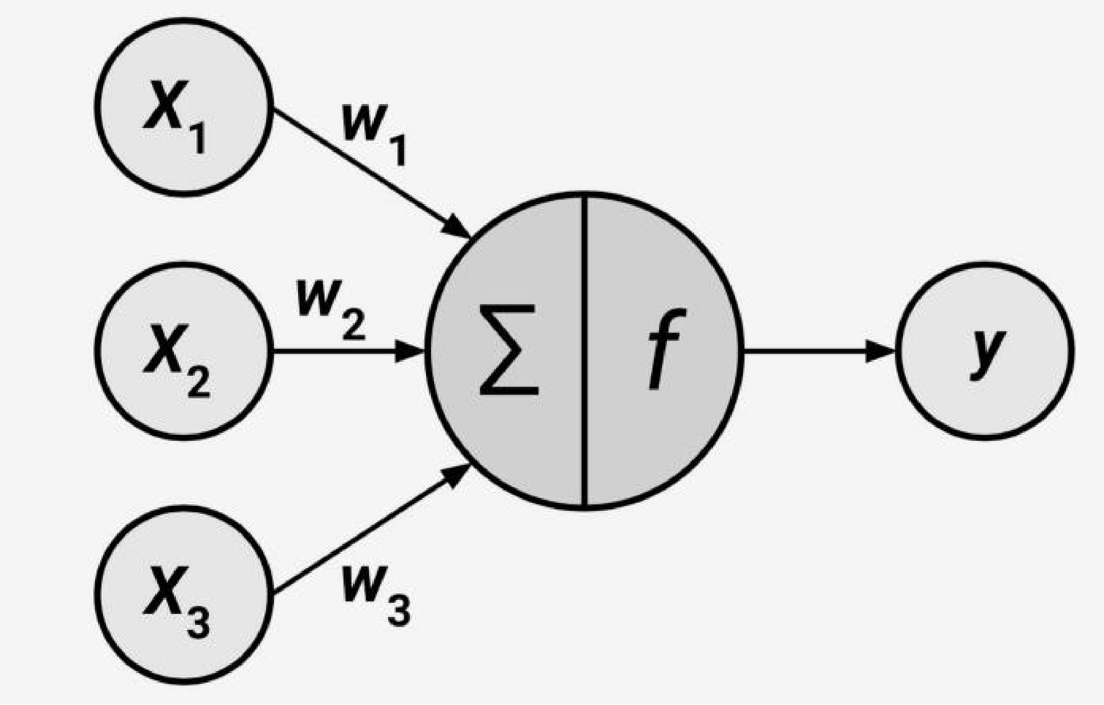
Similar to a biological brain response to stimuli from sensory inputs

The brain uses a network of interconnected cells called neurons to provide learning capability

ANN uses a network of artificial neurons or nodes to solve challenging learning problems

Brain vs neural network





Why learn neural networks?

Ability to learn

- Neural Networks figure out how to perform their function on their own
- Determine their function based only upon sample inputs

Ability to generalize

- Produce outputs for inputs it has not been taught how to deal with

Adaptivity – can be easily retrained to changing environmental conditions

Representation of a neuron

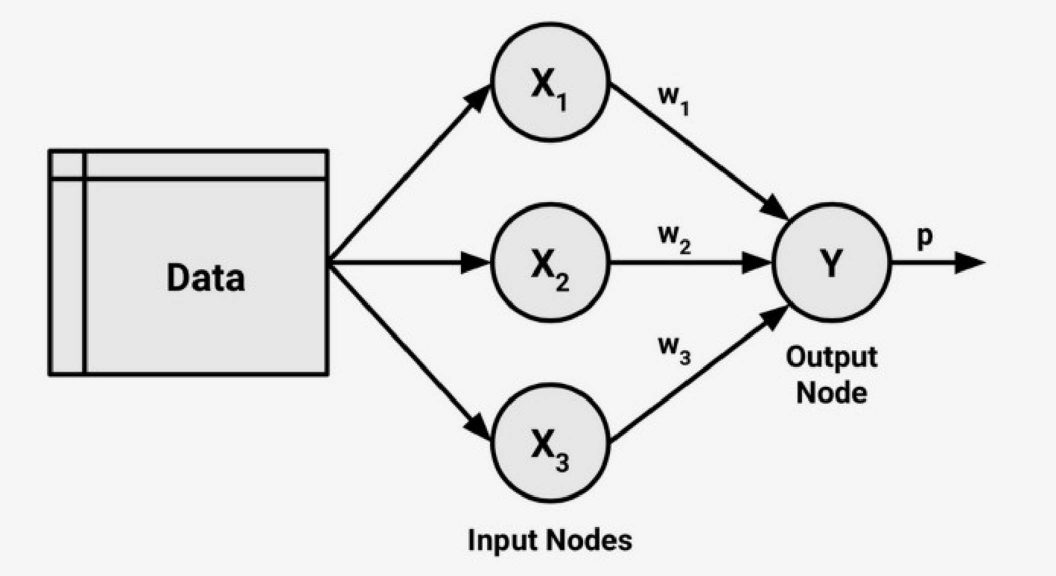
$$y(x) = f\left(\sum_{i=1}^{n} w_i x_i\right)$$

- w - weights
- n – number of inputs
- xi – input
- f(x) – activation function
- y(x) – output axon

Neural network architechture

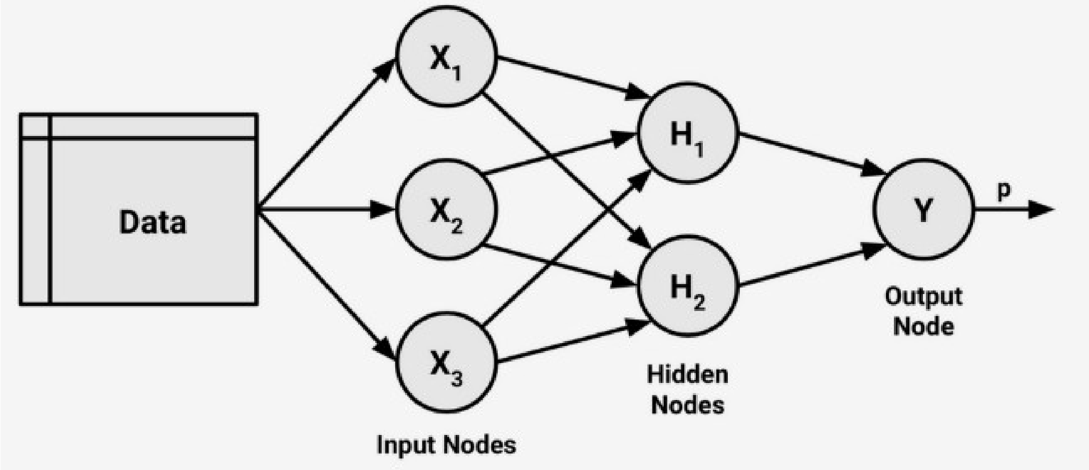
- Made of layers with many interconnected nodes (neurons)
- There are three main layers, specifically
  - Input Layer
  - Hidden Layer
  - Output Layer
- Hidden Layer can be one or more

Single Layered Network



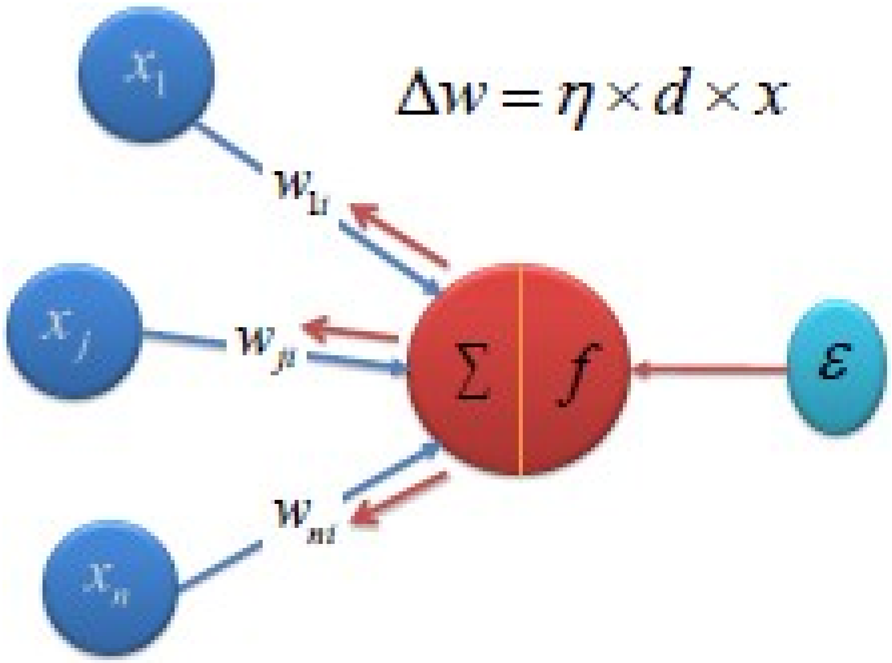
- Input nodes process the incoming data exactly as received
- Network has only one set of connection weights (w1, w2, and w3)
- It is therefore termed a single-layer network

Multilayer network



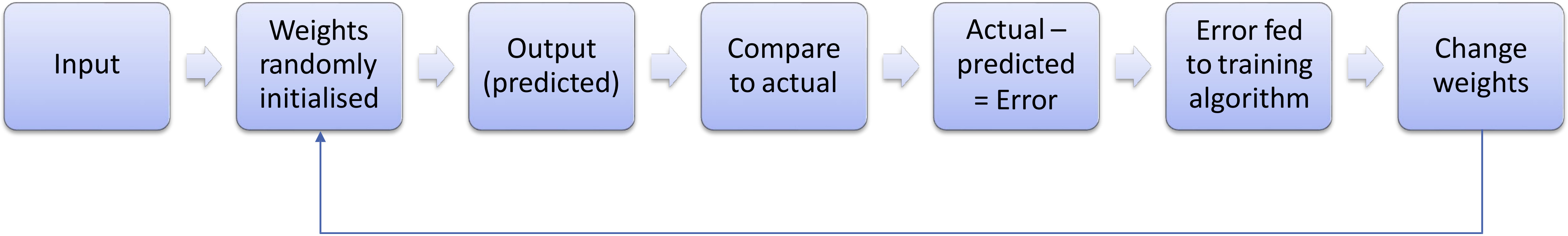
Adds one or more hidden layers that process the signals from the input nodes prior to reaching the output node

Backward error propagation or back propagation

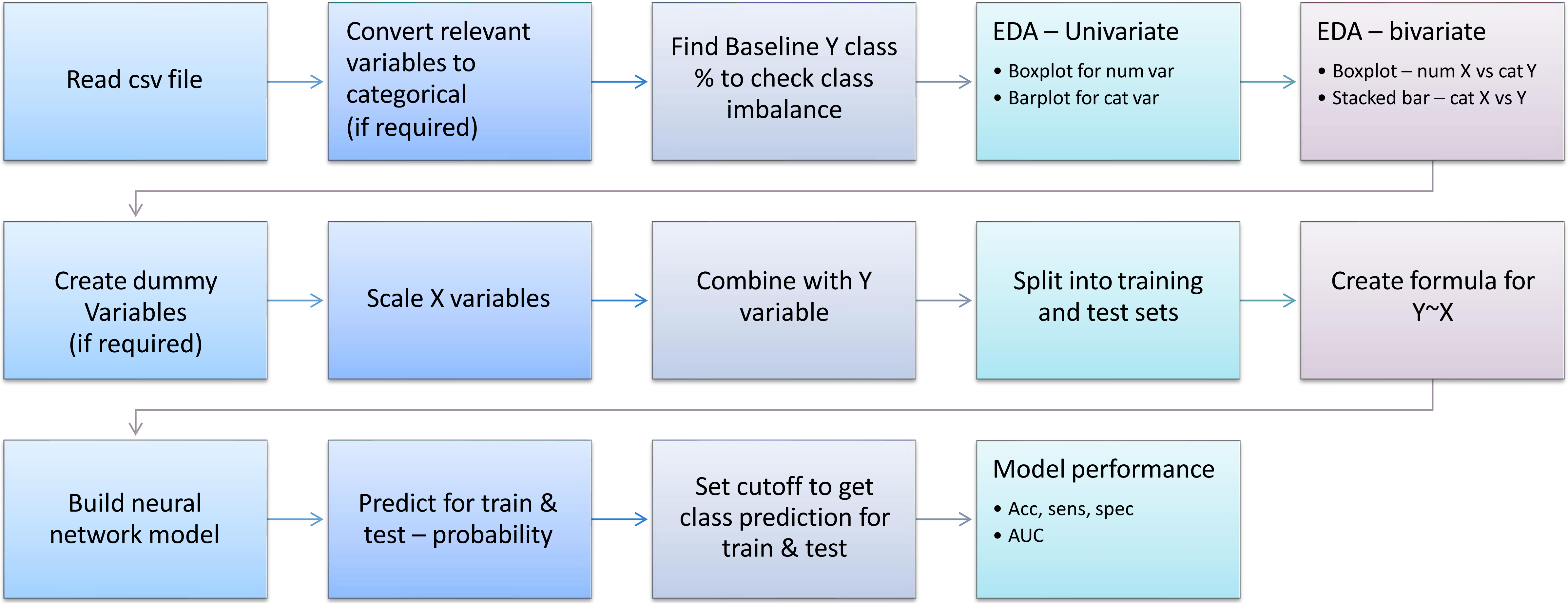


- The output node gives a predicted value
- The difference between predicted value and actual value is the error
- Error propagated backward by apportioning them to each node's weights
- In proportion to the amount of this error the node is responsible for

How is the synaptic weights of neurons determined?


Input
Weights randomly initialised
Output (predicted)
Compare to actual
Actual – predicted = Error
Error fed to training algorithm
Change weights

Overall flow of the NN classification process



Read csv file

Convert relevant variables to categorical (if required)

Find Baseline Y class % to check class imbalance

EDA – Univariate

- Boxplot for num var
- Barplot for cat var

EDA – bivariate

- Boxplot – num X vs cat Y
- Stacked bar – cat X vs Y

Create dummy Variables (if required)

Scale X variables

Combine with Y variable

Split into training and test sets

Create formula for Y~X

Build neural network model

Predict for train & test – probability

Set cutoff to get class prediction for train & test

Model performance

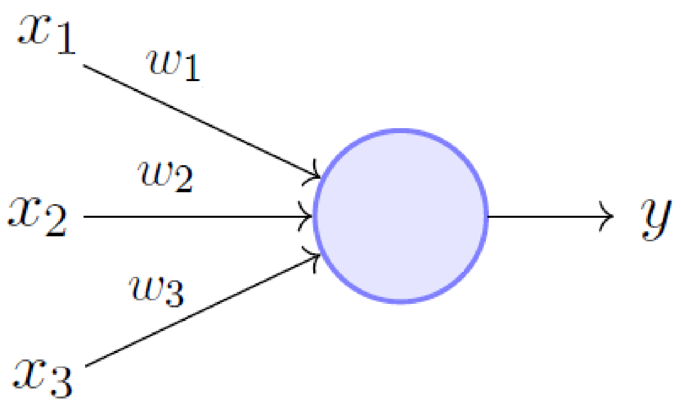
- Acc, sens, spec
- AUC

HappyLearning

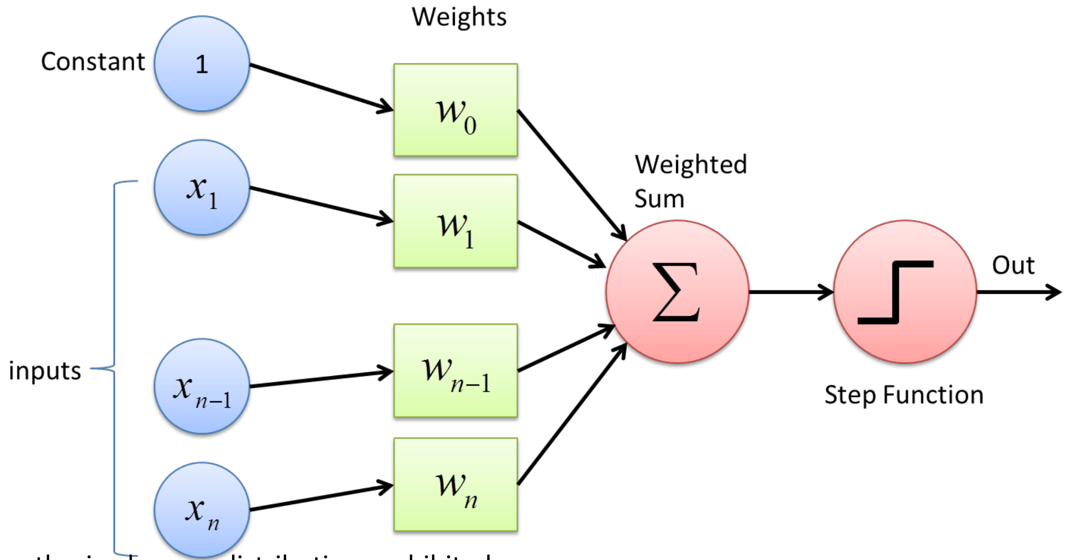
APPENDIX

Perceptron

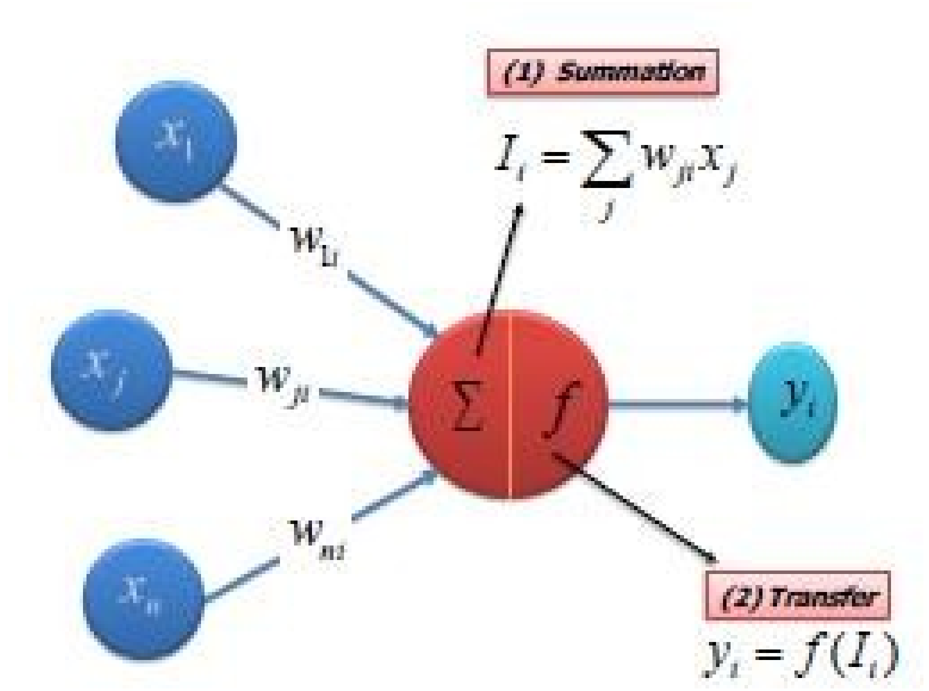
- Perceptrons are a type of artificial neurons developed in 1950’s – 60’s
- A perceptron takes several inputs and produces a single binary output
- The perceptron output determined by whether the weighted sum is greater than or less than a threshold value



Perceptron Model (Minsky-Papert in 1969)

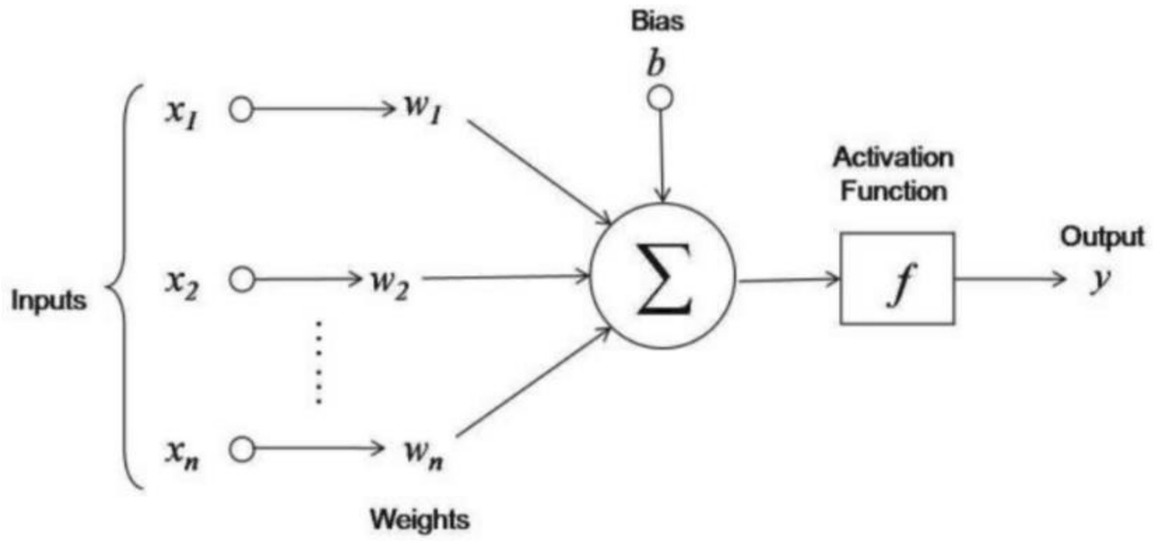


Feed forward network



Networks in which the input signal is fed continuously in one direction from the input layer to the output layer

ANN - neurons



- A neuron is an information-processing unit that is fundamental to the operation of a neural network.
- Three basic elements of the neuron model:
- –Synaptic weights
- –Combination (Addition) function
- –Activation function
- External input bias to increase or lower the net input to the Activation function

How does a Neuron Work

- Neuron Output

= fn ($wi *xi_{ni=1}+bias$)

- fn is some Activation Function
- Activation Function is sometime also called Transfer Function
- The output of the Activation Function is passed on to the neurons in the next layer and so on till the final output layer

Activation function

Mechanism by which the artificial neuron processes incoming information and passes it throughout the network.

threshold activation function - as it results in an output signal only once a specified input threshold has been attained.

Types of activation functions

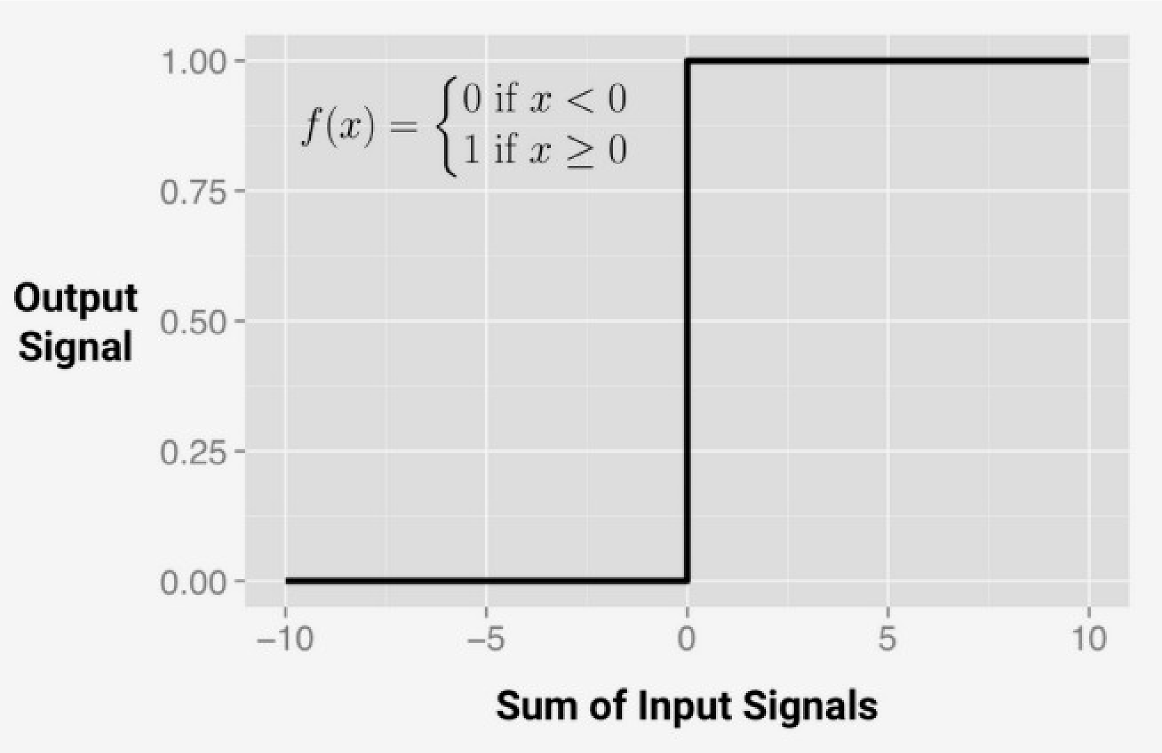
Unit step activation function

Sigmoid activation function

Hyperbolic tangent activation function

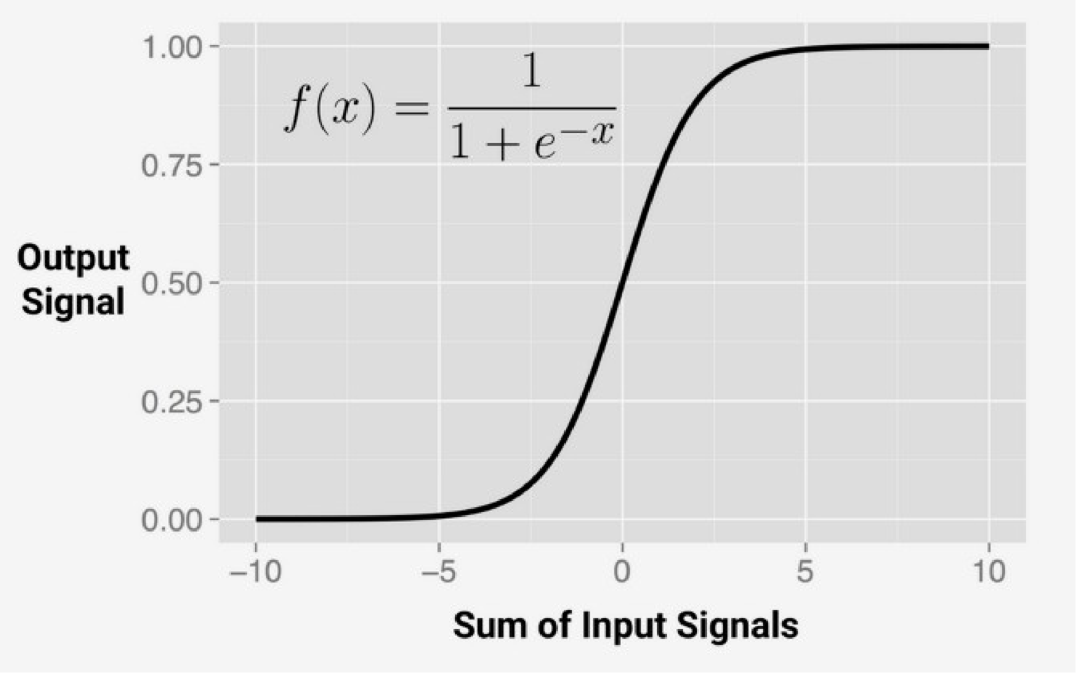
Rectified Linear Unit Activation Function

Unit step activation function



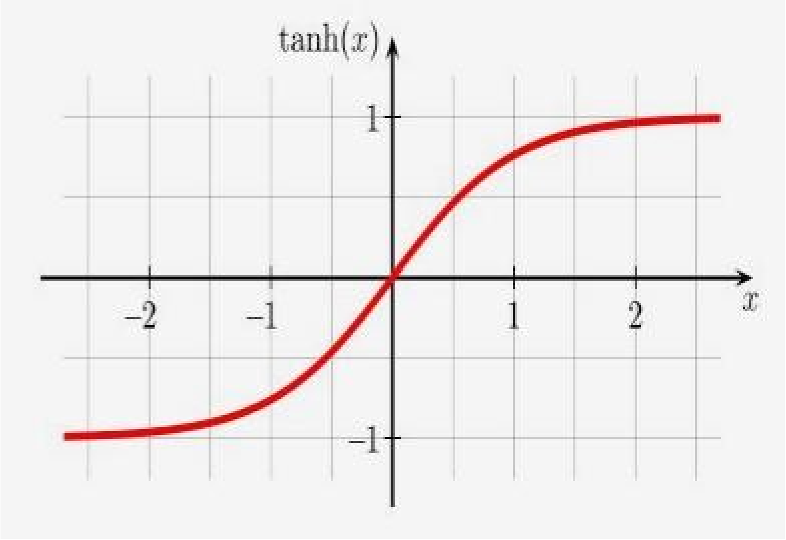
- Neuron fires when the sum of input signals is at least zero.
- Because its shape resembles a stair, it is sometimes called a unit step activation function.

Sigmoid activation function



- Most commonly used activation function
- Unlike the step activation function, the output signal is no longer binary
- Output values can fall anywhere in the range from zero to one.
- Sigmoid is differentiable $\Rightarrow$ possible to calculate the derivative across the entire range of inputs
- This feature is crucial for creating efficient ANN optimization algorithms.

Hyperbolic Tangent Activation Function



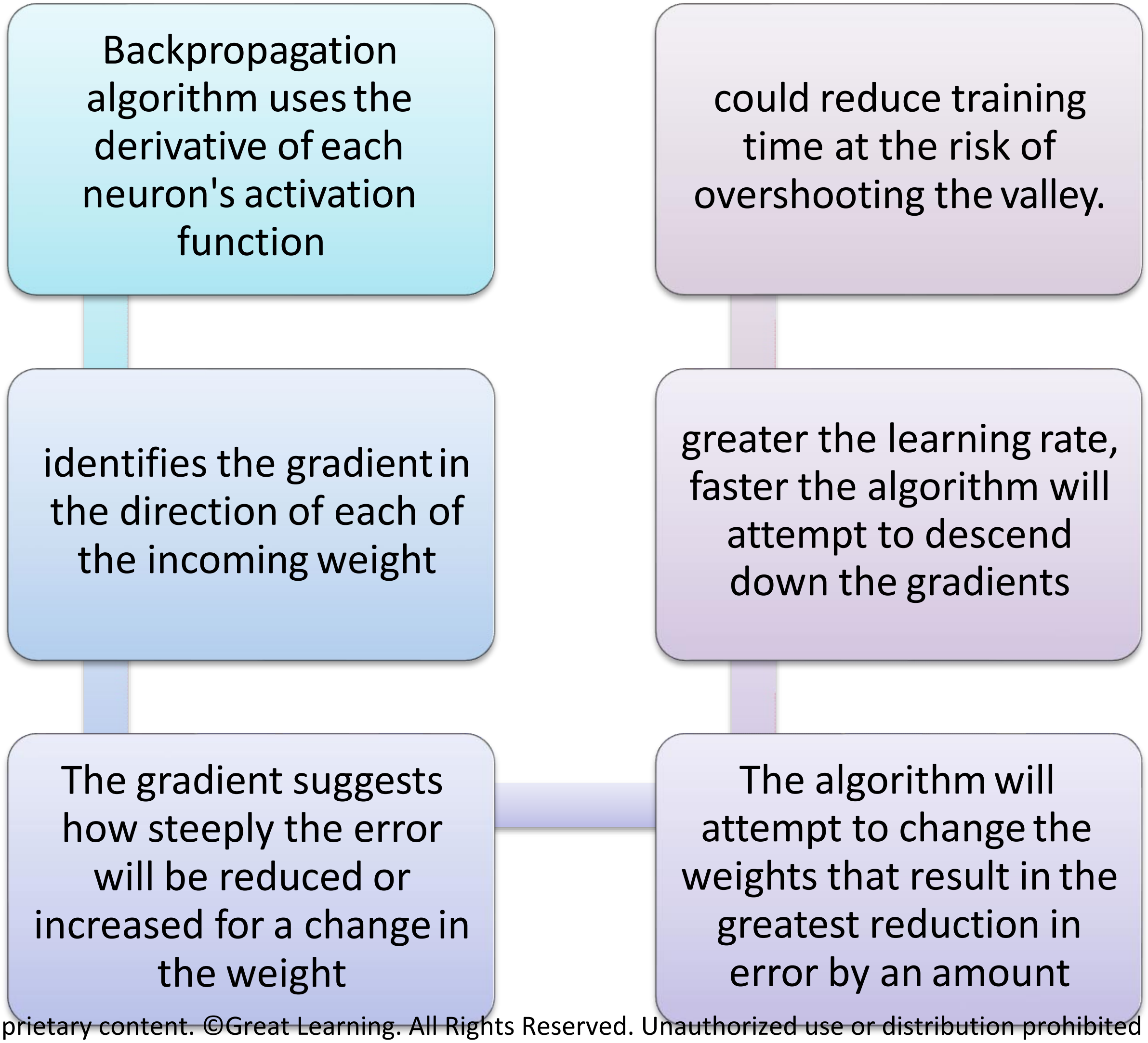
- Also known as tanh is a transfer function
- Widely used in deep neural networks
- Particularly as the activation function in Recurrent Neural Networks
- Shifted version of sigmoid activation
- Outputs a wider range of values

How is the synaptic weights of neurons determined?

- The synaptic weights of neurons are determined based on the neural net learning process (Learning Algorithm)
- Most common measure of the Error (cost function) is mean square error E = (y – d)^2
- Iterations of the above process of providing the network with an input and updating the network's weight to reduce error is used to train the network

How gradient descent works

hence the importance of having a differentiable activation function



Backpropagation algorithm uses the derivative of each neuron's activation function

identifies the gradient in the direction of each of the incoming weight

The gradient suggests how steeply the error will be reduced or increased for a change in the weight

The algorithm will attempt to change the weights that result in the greatest reduction in error by an amount

greater the learning rate, faster the algorithm will attempt to descend down the gradients

could reduce training time at the risk of overshooting the valley.

amount - learning rate

Cost function

- Cost Function is a loss function, a function to be minimized
- In ANN, the cost function is used to return a number to indicate how well the neural network performed to map training examples to correct output
- Desirable properties of Cost Function for ANN
  - Non-negativity
  - Cost function should tend to zero as actual output is close to the desired output
  - Globally continuous and differentiable
- Cost Functions e.g.
  - Quadratic Cost Function
  - Cross-Entropy Cost Function

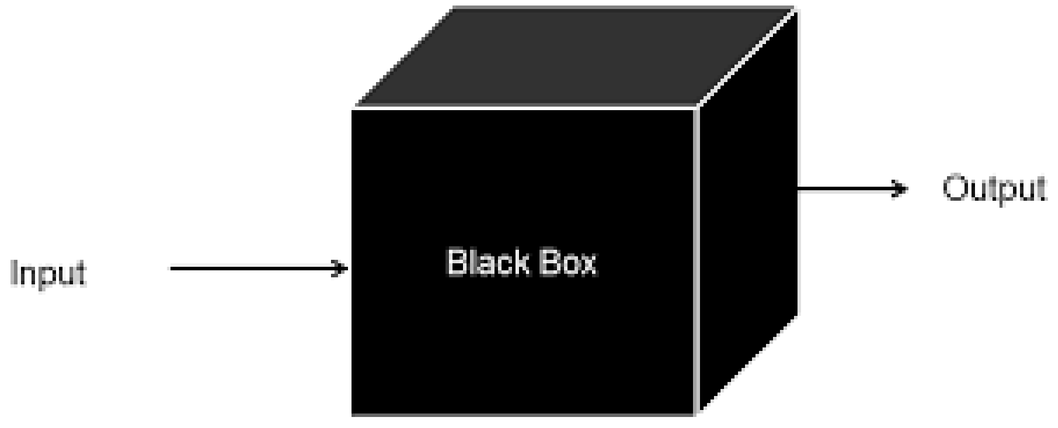
Learning Rate

- Hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated
- Choosing the learning rate is challenging
- a value too small may result in a long training process that could get stuck
- a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process.

Epoch

- one **epoch** = one forward pass and one backward pass of *all* the training examples
- **batch size** = the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- number of **iterations** = number of passes, each pass using [batch size] number of examples. To be clear, one pass = one forward pass + one backward pass (we do not count the forward pass and backward pass as two different passes).
- Example: if you have 1000 training examples, and your batch size is 500, then it will take 2 iterations to complete 1 epoch.

Drawback (Strength) of Neural Network Model



Internal behavior of the code is unknown

- It does not provide you with Business Insights