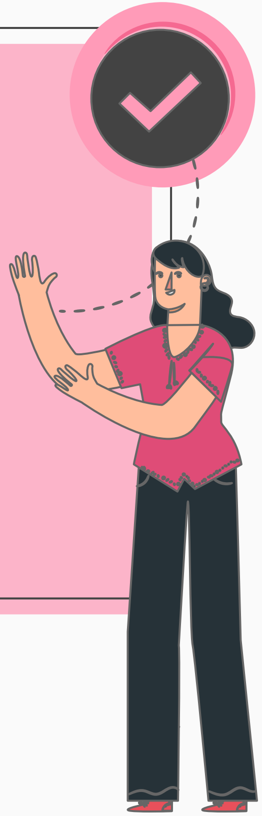
ACTIVATION FUNCTIONS



SureStart

ACTIVATION FUNCTION

An activation function is a **nonlinear function** applied by a neuron to introduce non-linear properties in the network.

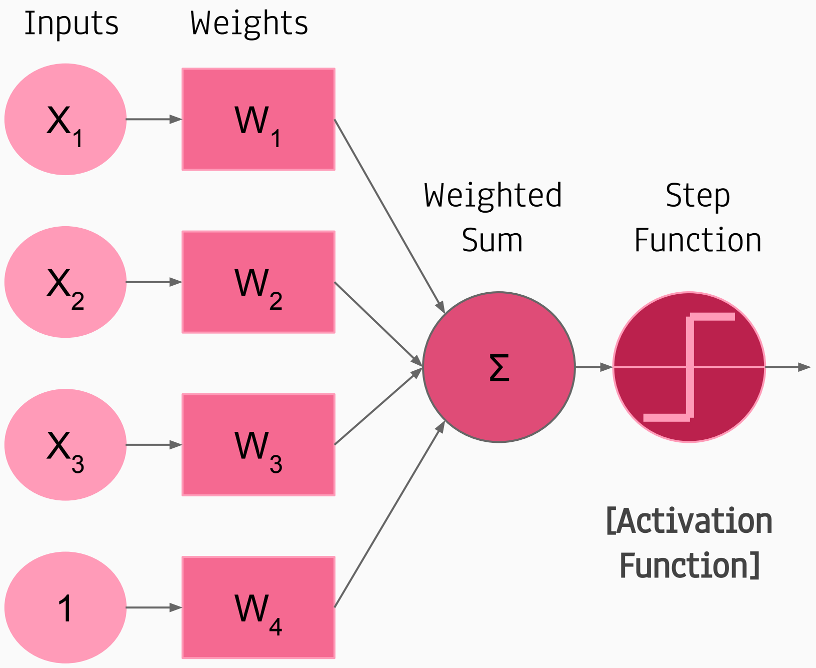
The activation functions basically decide whether the certain neuron should be activated or not.

The choice of activation function in the hidden layer controls *how well the network model learns the training dataset*.

The choice of activation function in the output layer will define the *type of predictions* the model can make.

All hidden layers typically use the same activation function.

The output layer will typically use a different activation function from the hidden layers and is dependent upon the type of prediction required by the model.



LINEAR VS NONLINEAR RELATIONSHIP

LINEAR RELATIONSHIP

In a linear relationship, a change in the first variable corresponds to a constant change in the second variable.

- A straight-line function
- Values can get very large
- The linear function alone doesn't capture complex patterns

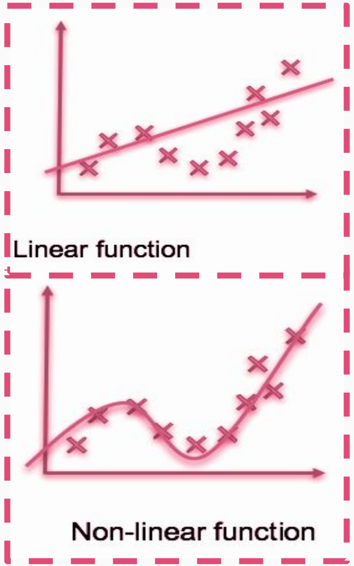


Figure: source

NONLINEAR RELATIONSHIP

In a non-linear relationship, the first variable doesn't necessarily correspond with a constant change in the second. However, they may impact each other but it appears to be unpredictable.

- A nonlinear function can capture more complex patterns.
- Output values are bounded, so that they don't get too large.
- Can suffer from "vanishing gradient".

Vanishing gradient? – For very low values of input X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further or being too slow to reach an accurate prediction.

SureStart

ACTIVATION LAYERS

FOR HIDDEN LAYERS

The most commonly used activation layer for hidden layers are -

Rectified Linear Activation (ReLU)
Logistic (Sigmoid)
Hyperbolic Tangent (Tanh)

Multilayer Perceptron: ReLU
Convolutional Neural Network: ReLU
Recurrent Neural Network: Tanh and/or Sigmoid

FOR OUTPUT LAYERS

The most commonly used activation layer for output layers are -

Linear
Logistic (Sigmoid)
Softmax

REGRESSION

One node, linear activation

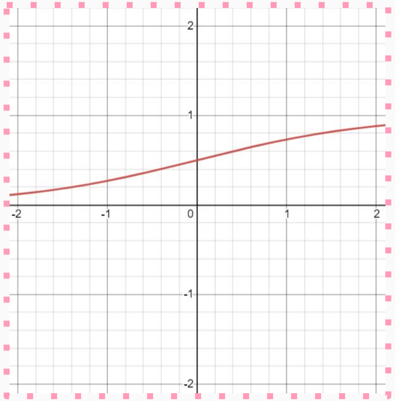
CLASSIFICATION

Binary Classification: One node, sigmoid activation.
Multiclass Classification: One node per class, softmax activation.
Multilabel Classification: One node per class, sigmoid activation.

ACTIVATION FOR HIDDEN LAYERS

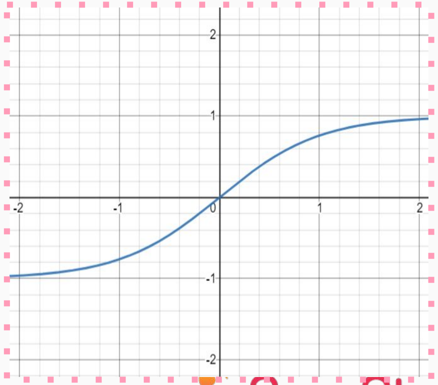
LOGISTIC (SIGMOID)

- Uses: Can be used for hidden layers but mostly used for output layers of a binary classification, where result is either 0 or 1. Value for sigmoid function lies between 0 and 1, result can be predicted to be 1 if value is greater than 0.5 and 0 otherwise.
- Value Range: 0 to 1
- Nature: Non-linear. Notice that X values lies between -2 to 2, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.
- Smooth gradient, preventing “jumps” in output values.



HYPERBOLIC TANGENT (TANH)

- Uses: In hidden layers, as an activation function.
- Value Range: Output values bound between -1 and 1.
- Nature: A nonlinear function.
- Zero centered– Usually used in hidden layers of a neural network as its values lie between -1 to 1 hence the mean for the hidden layer comes out be 0 (or very close to 0) and helps in centering the data by bringing mean close to 0. *This makes learning easier to the model inputs that have strongly negative, neutral, and strongly positive values.*



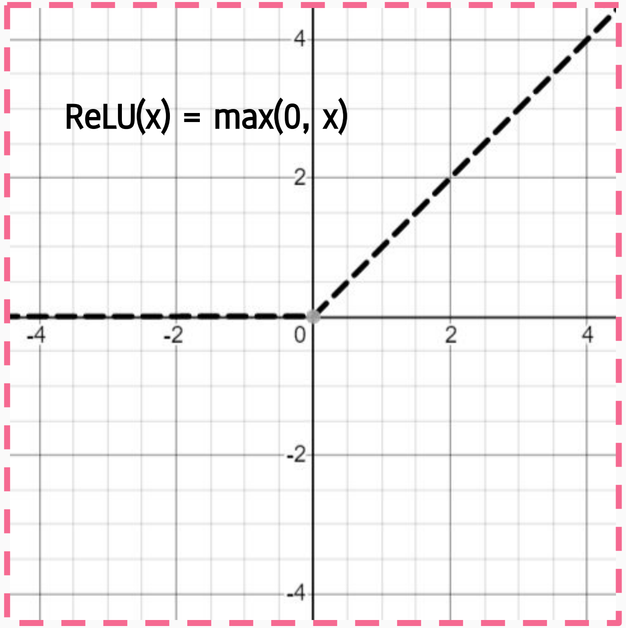
SureStart

ACTIVATION FOR HIDDEN LAYERS - RELU

RECTIFIED LINEAR ACTIVATION (RELU)

It is the most widely used activation function. Mainly implemented in hidden layers of neural network

- Value Range: $[0, \infty)$
- Nature: non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.
- Uses:
    - ReLU is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations.
    - At a time only a few neurons are activated because of the characteristic of output 0 for negative values of input. This makes the network sparse (the strong activation of a relatively small set of neurons) making it efficient and easy for computation.
    - ReLU learns much faster than sigmoid and Tanh function.

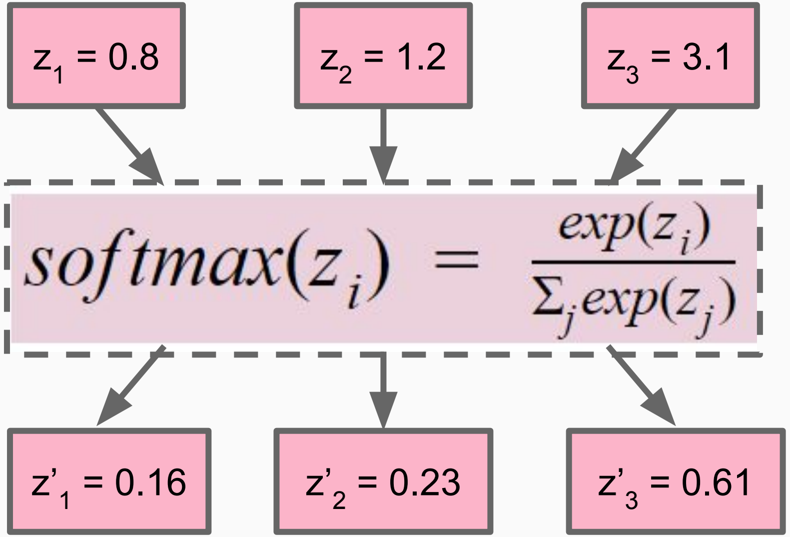


ACTIVATION FOR OUTPUT LAYERS: SOFTMAX

SOFTMAX

The softmax function is also a type of sigmoid function and is handy when we are trying to handle classification problems.

- Value Range: [0, 1]
- Nature: non-linear
- Uses:
  - The softmax function is ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.
  - The softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.
  - Usually used when trying to handle multiple classes.



SureStart

RELATION BETWEEN ACTIVATION FUNCTION WITH LOSS FUNCTION

PROBLEM TYPE	OUTPUT TYPE	FINAL ACTIVATION FUNCTION	LOSS FUNCTION
Regression	Numerical value	Linear	Mean Squared Error [MSE]
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple label, multiple classes	Sigmoid	Binary Cross Entropy

SureStart