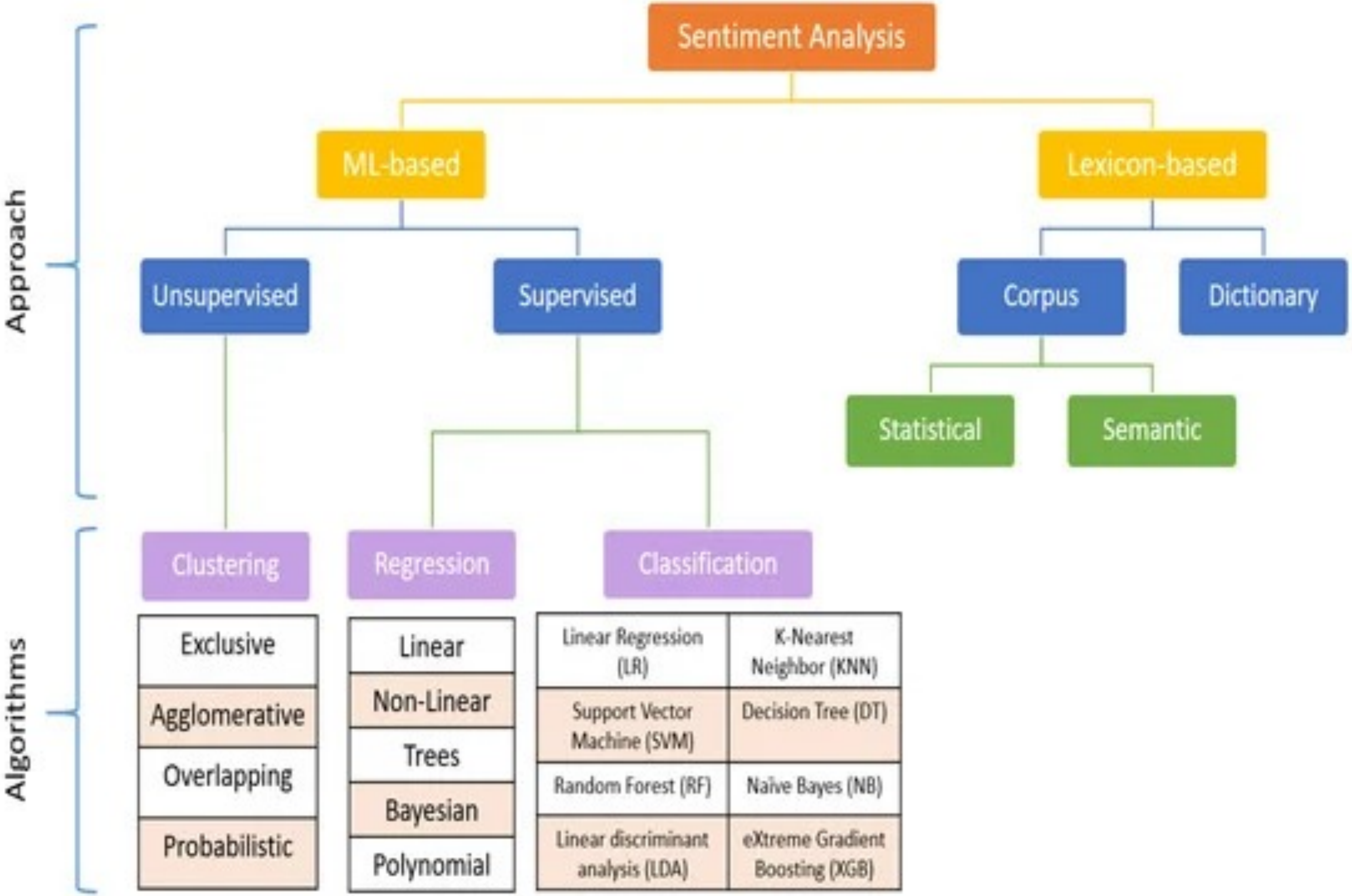
Artificial Neural Networks

Mustafa Coşkun


Approach
Algorithms
Sentiment Analysis
ML-based
Lexicon-based
Unsupervised
Supervised
Corpus
Dictionary
Statistical
Semantic
Clustering
Regression
Classification
Exclusive
Agglomerative
Overlapping
Probabilistic
Linear
Non-Linear
Trees
Bayesian
Polynomial
Linear Regression (LR)
K-Nearest Neighbor (KNN)
Support Vector Machine (SVM)
Decision Tree (DT)
Random Forest (RF)
Naïve Bayes (NB)
Linear discriminant analysis (LDA)
eXtreme Gradient Boosting (XGB)

What is ANN?

- ANN(Artificial Neural Network)
- The Artificial Neural Network consists of an input layer, a hidden layer, and an output layer.

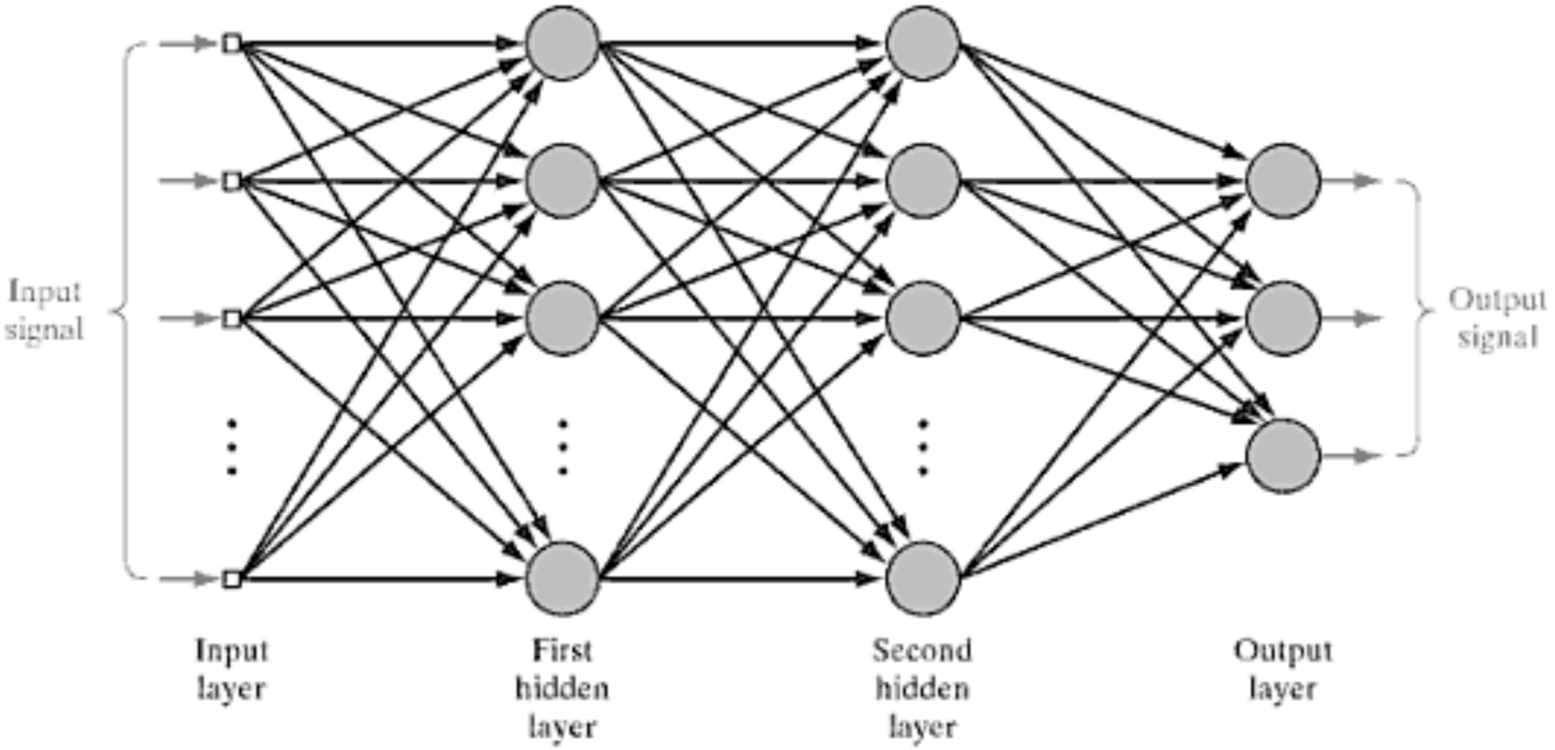


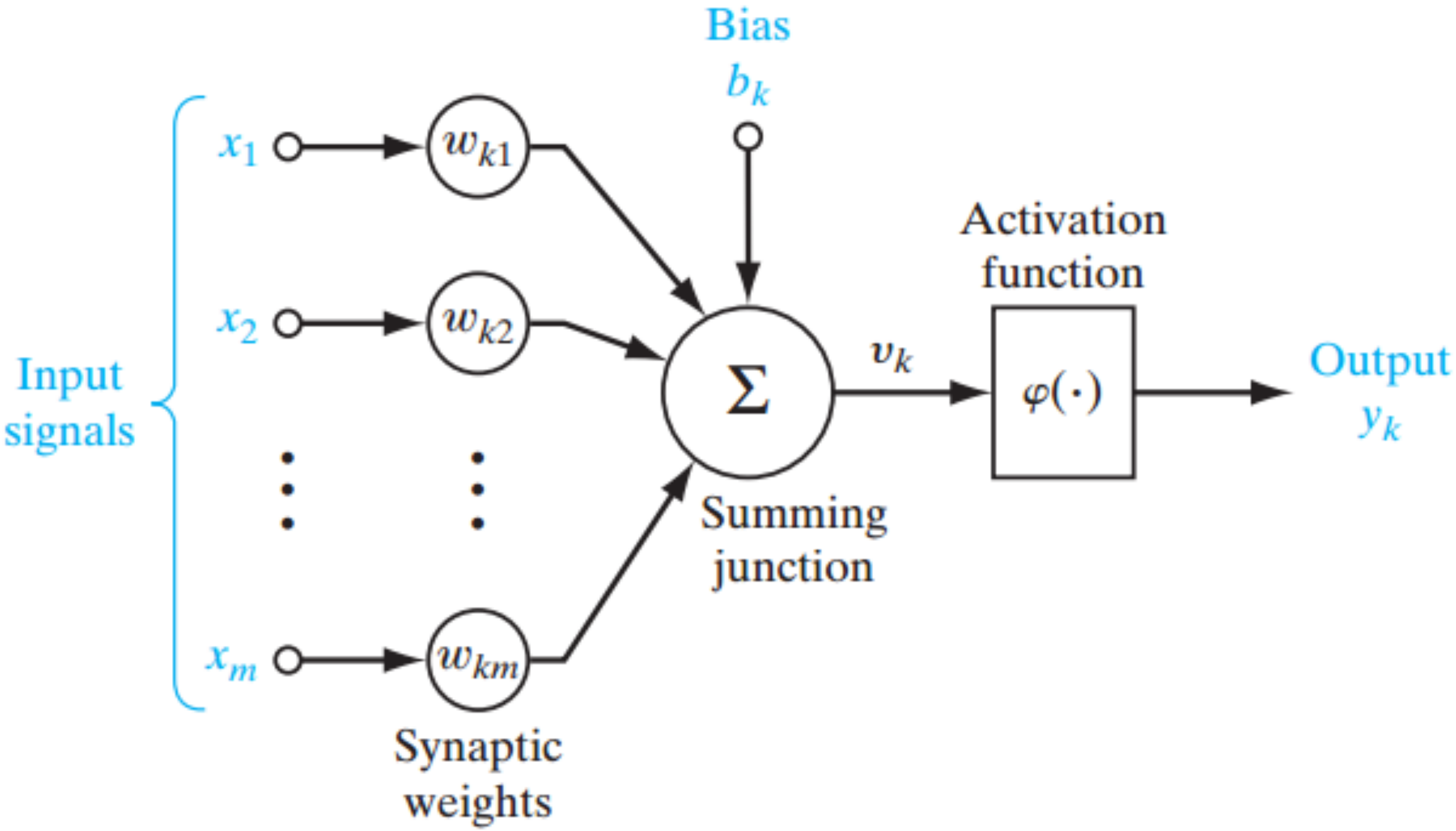
FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers.

Is there a difference between NN and ANN?

- Neural Network is a broad term that encompases various types of networks.
- Is ANN one of the types?
- neural network alone is not an algorithm but a framework which assists the algorithms to work. ANN is the most basic type of implementation of neurals. ANN was the term coined much earlier and nowadays the two terms are interchangeably used.

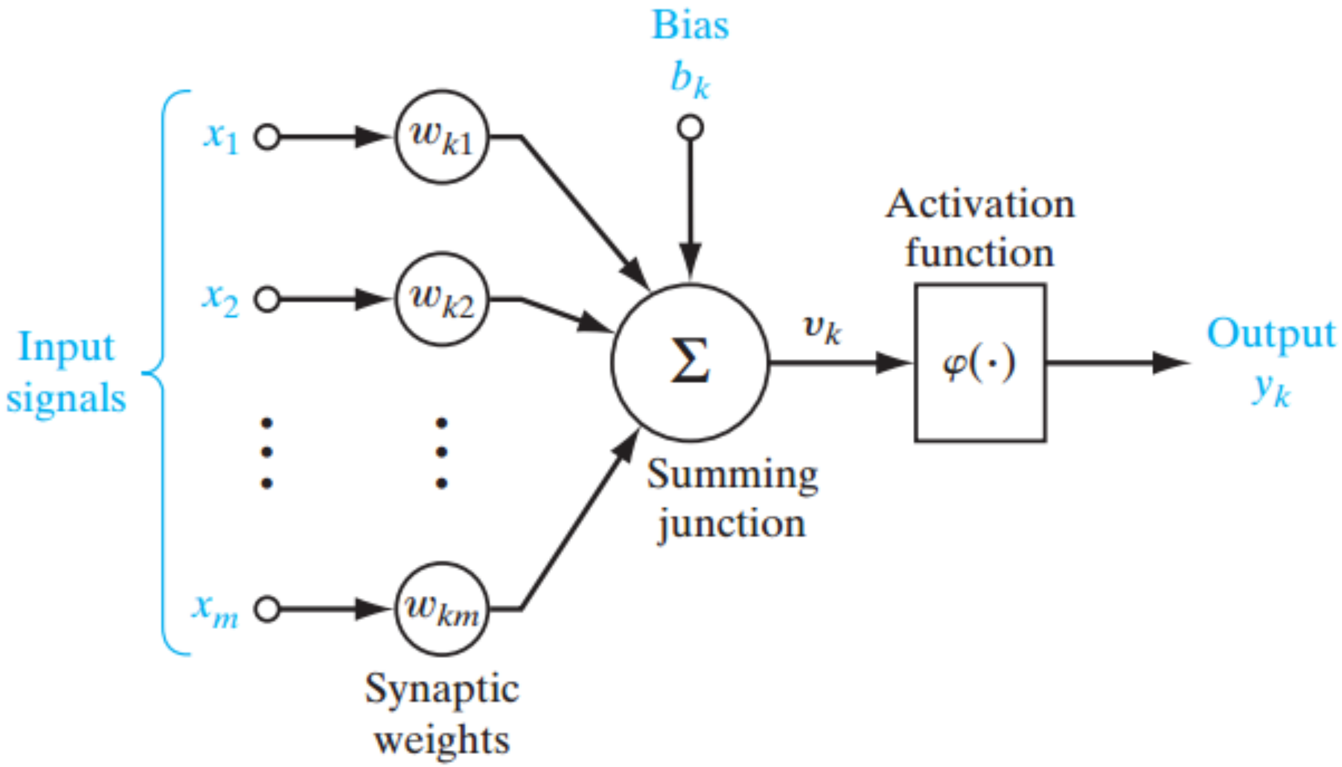
SLP(Single Layer Perceptron)

- If ANN model has no hidden layer, it is called single layer perceptron.



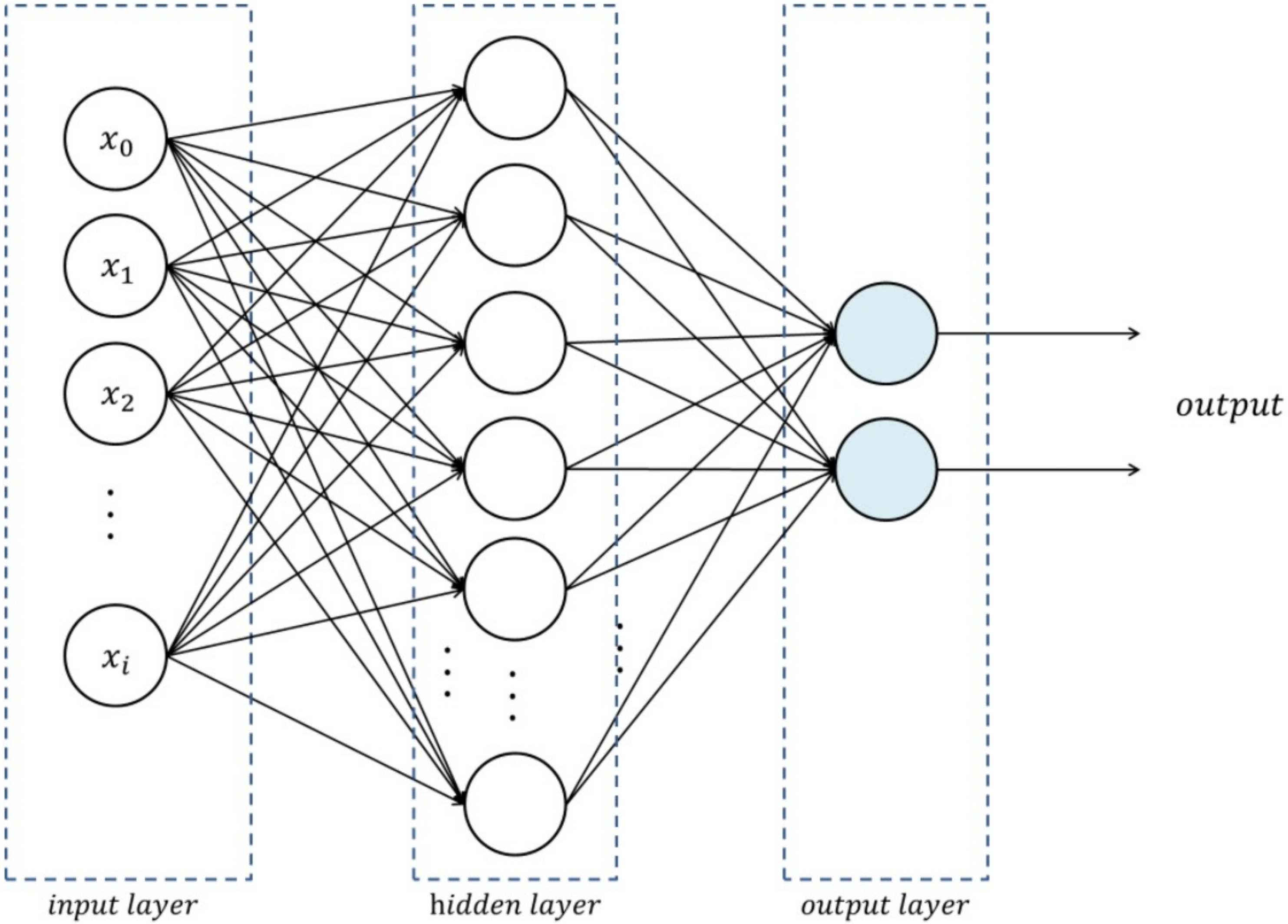
Basic equation of ANN-SLP

- **Output = Weight * Input + Bias**
- Weight : a value that can give different weights depending on features and output
- bias : a value that can give different weights depending on features



MLP(Multi Layer Perceptron)

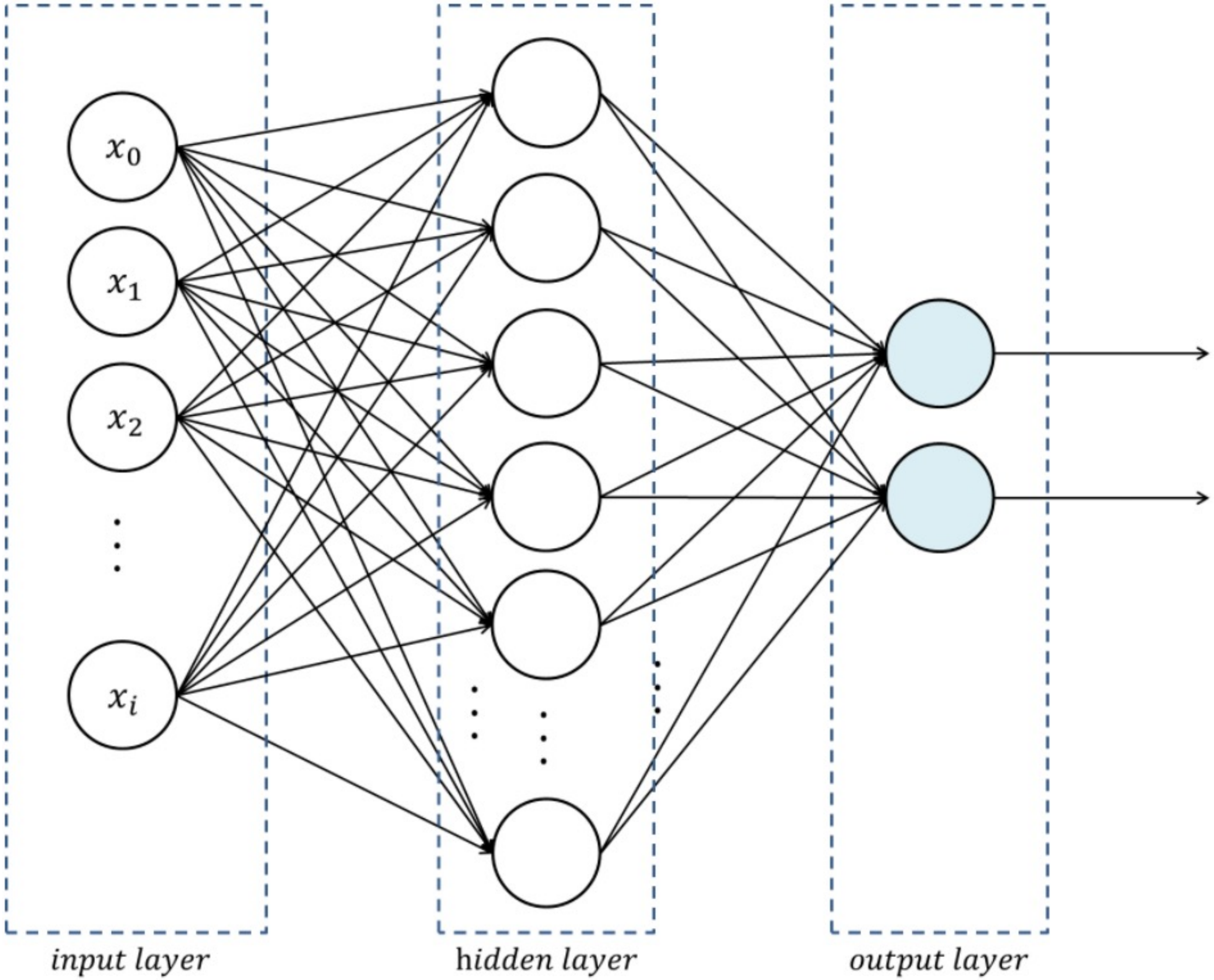
- MLP(Multiple Layer Perceptron) model is ANN which has multiple hidden layers (more than 1)



Basic equation of ANN-SLP

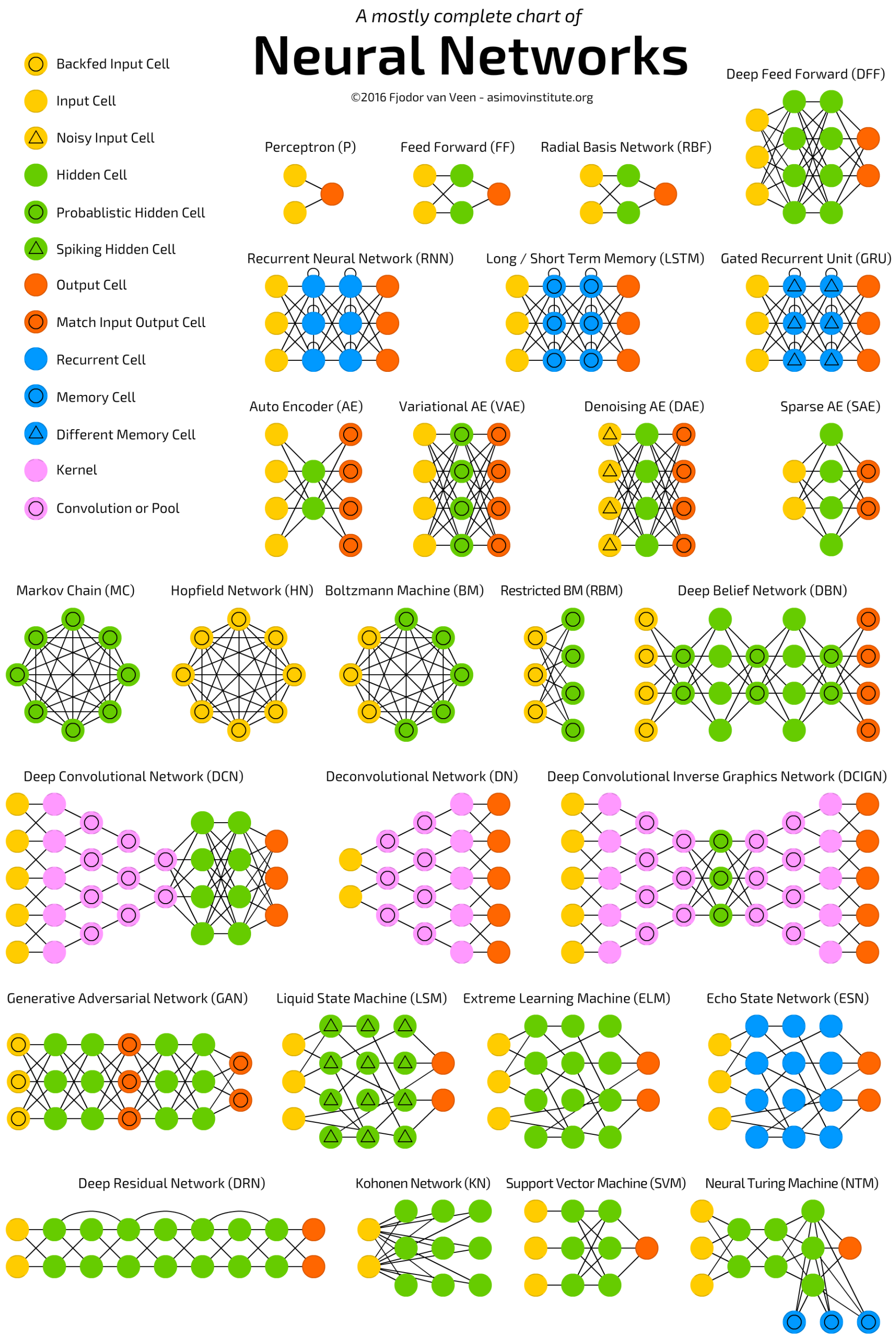
Output = (Weight1 x Input1) + (Weight2 x Input2) + ... + (WeightN x InputN) + Bias

- Weight : a value that can give different weights depending on features and output
- bias : a value that can give different weights depending on features



Types of Neural Network

https://i.stack.imgur.com/LgmYv.png


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Flowchart of ANN

1
- Assign Random weights to all the linkages to start the algorithm

2
- Using the inputs and the (Input ->Hidden node) linkages find the activation rate of Hidden Nodes

3
- Using the activation rate of Hidden nodes and linkages to Output, find the activation rate of Output Nodes

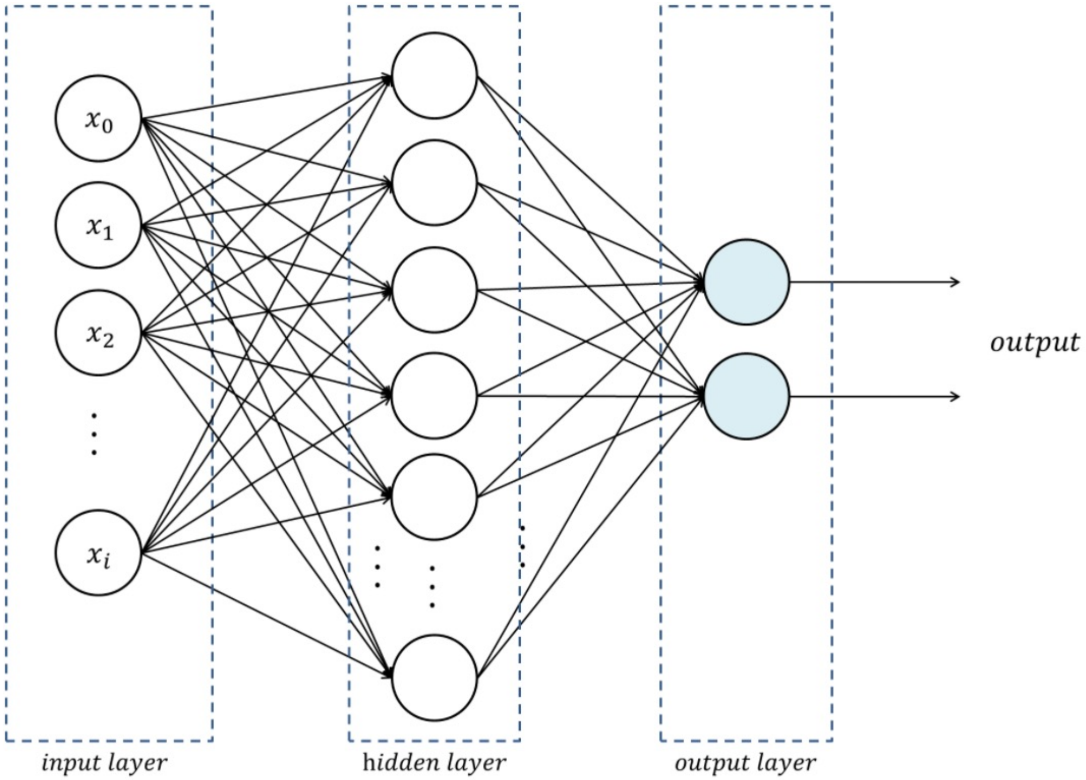
4
- Find the error rate at the output node and recalibrate all the linkages between Hidden Nodes and Output Nodes

5
- Using the Weights and error found at Output node, cascade down the error to Hidden Nodes

6
- Recalibrate the weights between hidden node and the input nodes

7
- Repeat the process till the convergence criterion is met

8
- Using the final linkage weights score the activation rate of the output nodes

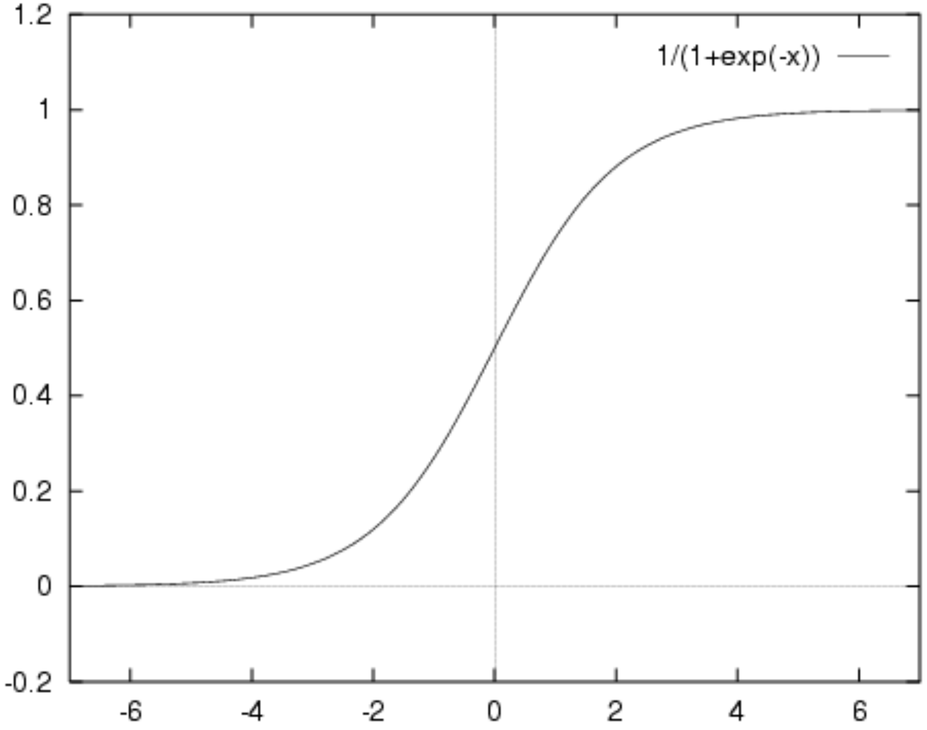


Activation Function

- Activation functions are really important for a Artificial Neural Network to learn and make sense of something really complicated and Non-linear complex functional mappings between the inputs and response variable. They introduce non-linear properties to our Network. Their main purpose is to convert a input signal of a node in a A-NN to an output signal. That output signal now is used as a input in the next layer in the stack.
-

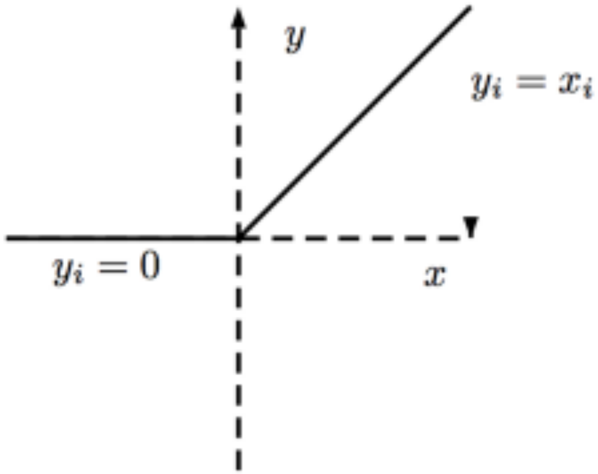
Most popular types of Activation functions

- Sigmoid or Logistic
- **Sigmoid Activation function**: It is a activation function of form f(x) = 1 / 1 + exp(-x) . Its Range is between 0 and 1. It is a S — shaped curve. It is easy to understand and apply but it has major reasons which have made it fall out of popularity:
  - Vanishing gradient problem
  - Secondly , its output isn’t zero centered. It makes the gradient updates go too far in different directions. 0 < output < 1, and it makes optimization harder.
  - Sigmoids saturate and kill gradients.
  - Sigmoids have slow convergence.

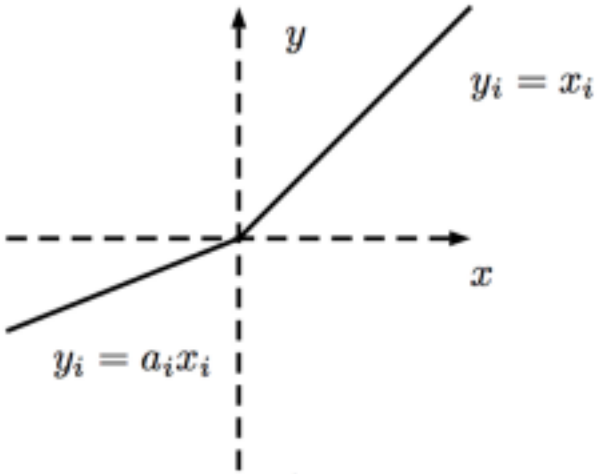


Most popular types of Activation functions

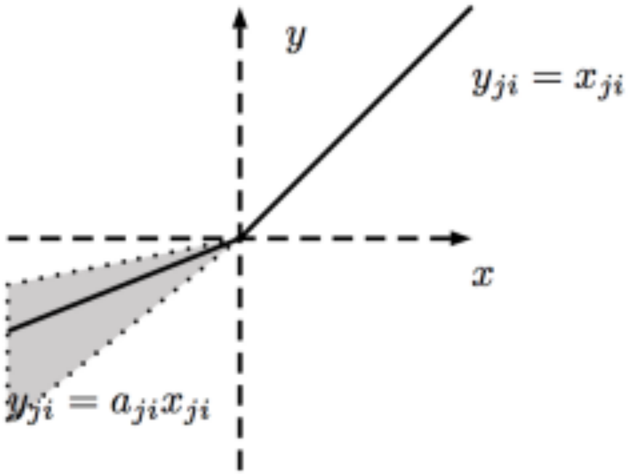
- **ReLu- Rectified Linear units** : It has become very popular in the past couple of years. It was recently proved that it had 6 times improvement in convergence from Tanh function. It's just R(x) = max(0,x) i.e if x < 0 , R(x) = 0 and if x >= 0 , R(x) = x. Hence as seeing the mathamatical form of this function we can see that it is very simple and efficinent . A lot of times in Machine learning and computer science we notice that most simple and consistent techniques and methods are only preferred and are best. Hence it avoids and rectifies vanishing gradient problem . Almost all deep learning Models use ReLu nowadays.
- But its limitation is that it should only be used within Hidden layers of a Neural Network Model.
- Hence for output layers we should use a Softmax function for a Classification problem to compute the probabilites for the classes , and for a regression problem it should simply use a linear function.



ReLU



Leaky ReLU/PReLU



Randomized Leaky ReLU

What happens without activation function?

- If we do not apply a Activation function then the output signal would simply be a simple linear function.A linear function is just a polynomial of one degree. Now, a linear equation is easy to solve but they are limited in their complexity and have less power to learn complex functional mappings from data. A Neural Network without Activation function would simply be a Linear regression Model, which has limited power and does not performs good most of the times. We want our Neural Network to not just learn and compute a linear function but something more complicated than that. Also without activation function our Neural network would not be able to learn and model other complicated kinds of data such as images, videos , audio , speech etc. That is why we use Artificial Neural network techniques such as Deep learning to make sense of something complicated ,high dimensional,non-linear -big datasets, where the model has lots and lots of hidden layers in between and has a very complicated architecture which helps us to make sense and extract knowledge form such complicated big datasets.