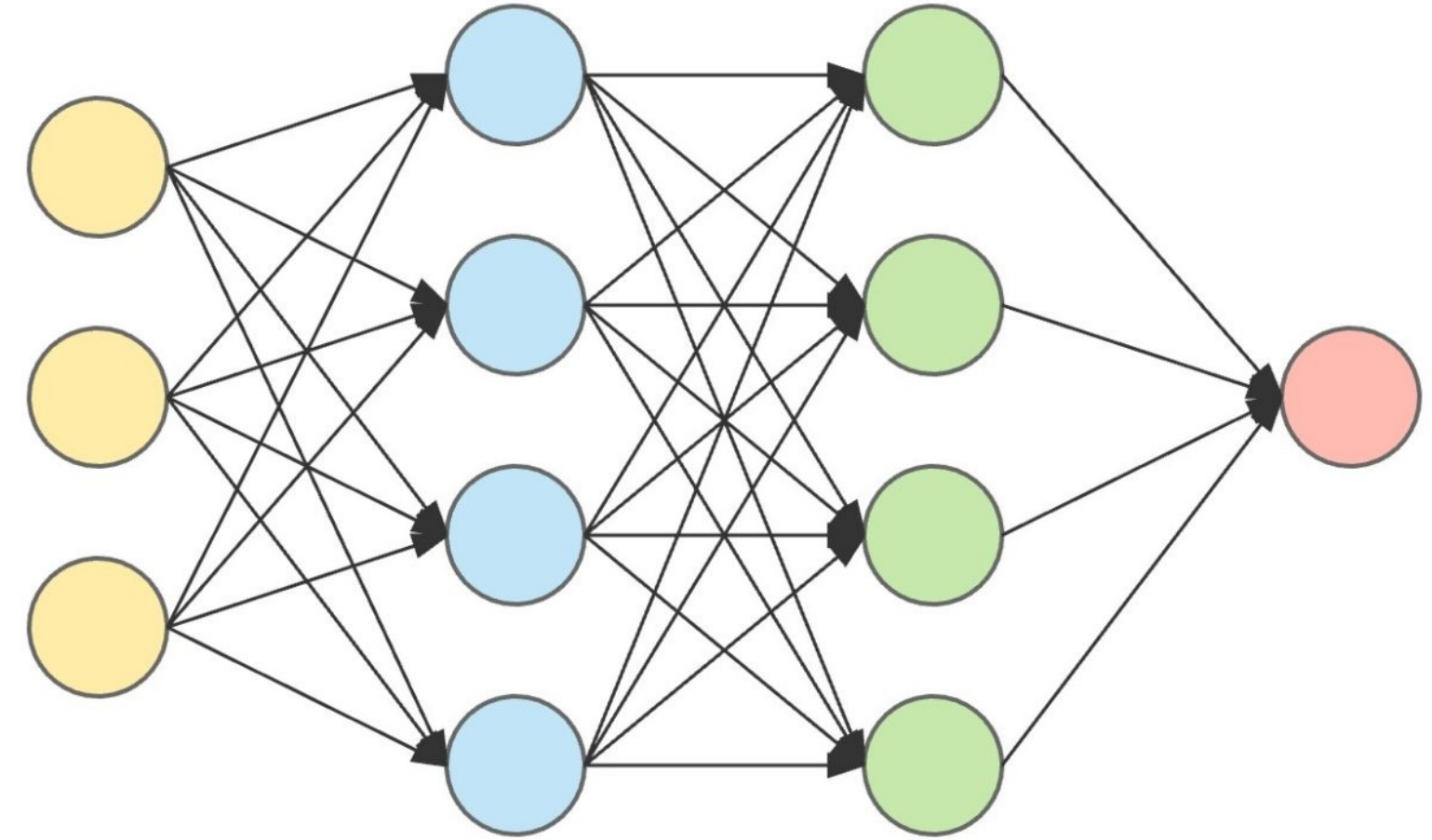
Fundamentals of Deep Learning

Mohammed Almelabi

End-to-end process

- Data ingestion
- Data cleaning
- Data preparation
- Exploratory data analysis
- Network design and training the model
- Check the accuracy and iterate
- Initial Model is created
- Model is deployment

Motivation behind Neural Network



Layers in a Neural Network

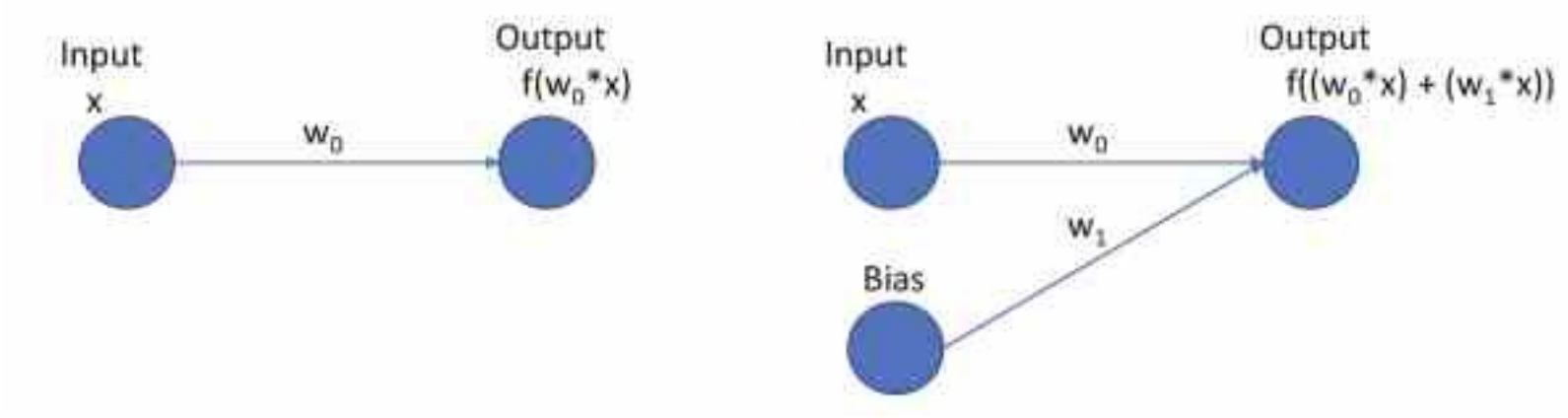
- A basic Neural Network architecture consists of predominantly three layers:
- Input layer
- Hidden layers
- Output layer

Hyperparameters

- The learning rate
- Number of hidden layers in the network
- Number of neurons in each layer
- Activation function
- Number of epoch
- Batch size
- Dropout
- Network weight

Bias term

- Bias is just like adding an intercept value to a linear equation
- It is an extra or additional parameter in a network



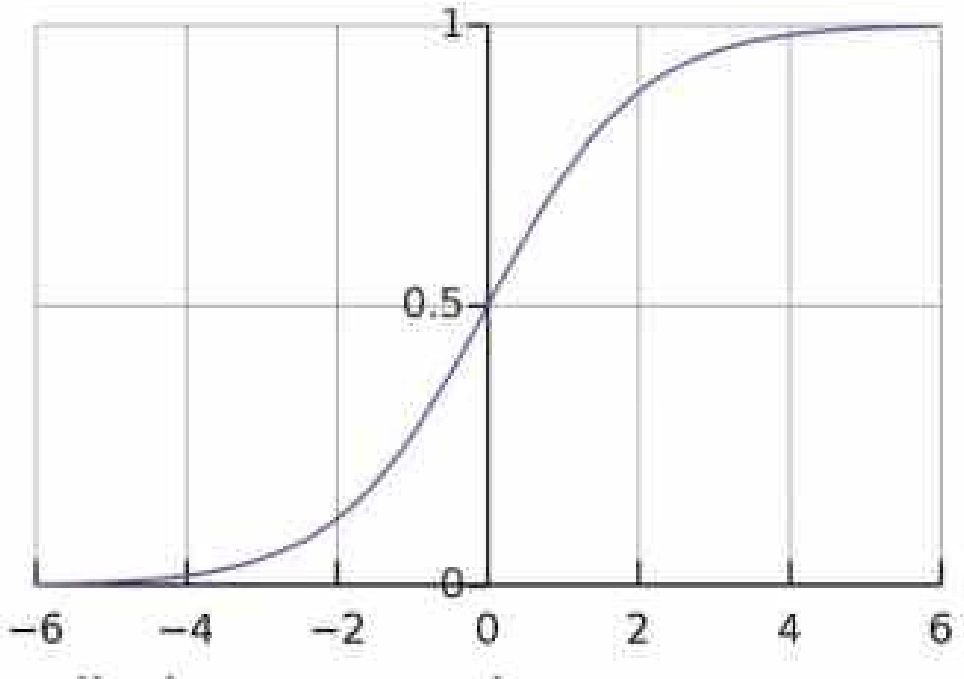
Activation functions

- Sigmoid function
- Tanh function
- Rectified Linear Unit or ReLU
- Softmax function

Sigmoid function

- The Sigmoid function is used if the output value of a neuron is between 0 and 1.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1}$$

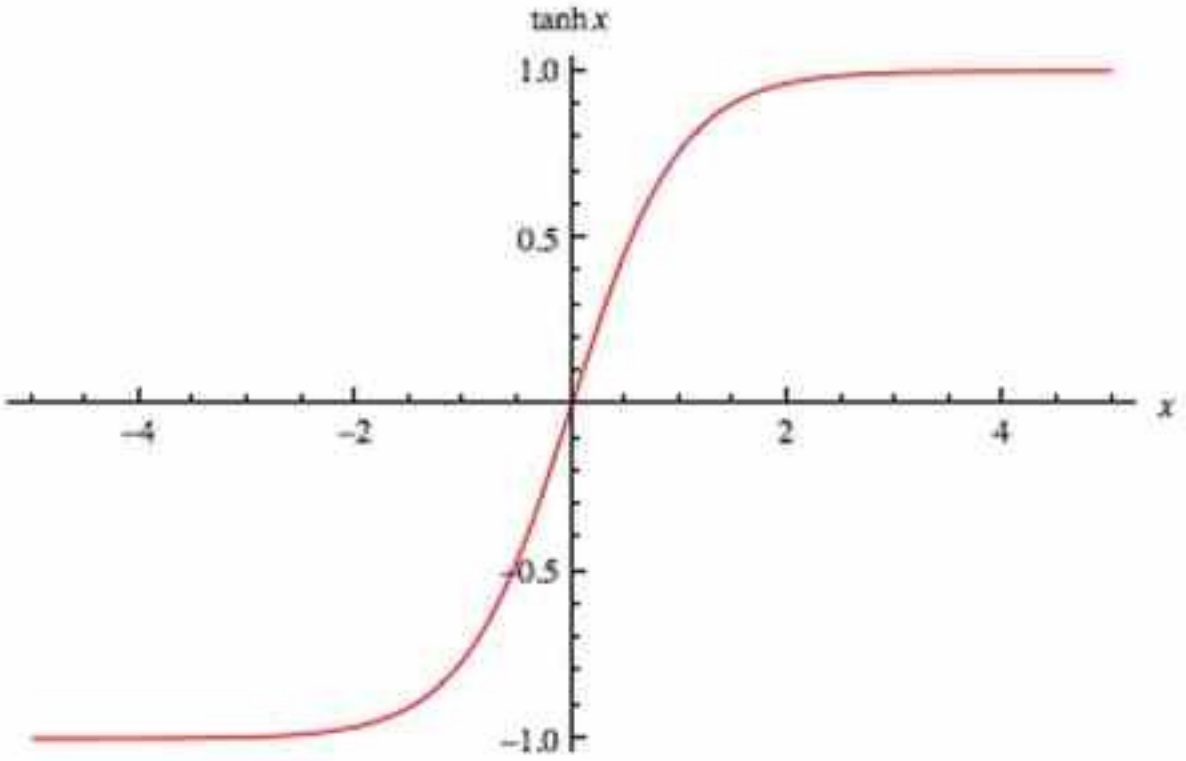


Tanh function

- Tangent hyperbolic function or tanh is a differentiable hyperbolic function.
- It is a scaled version of the Sigmoid function. It is a smooth function, and its input values are in the range of –1 to +1.
- A tanh function is generally used in the hidden layers. It makes the mean closer to zero which makes the training easier for the next layer in the network.

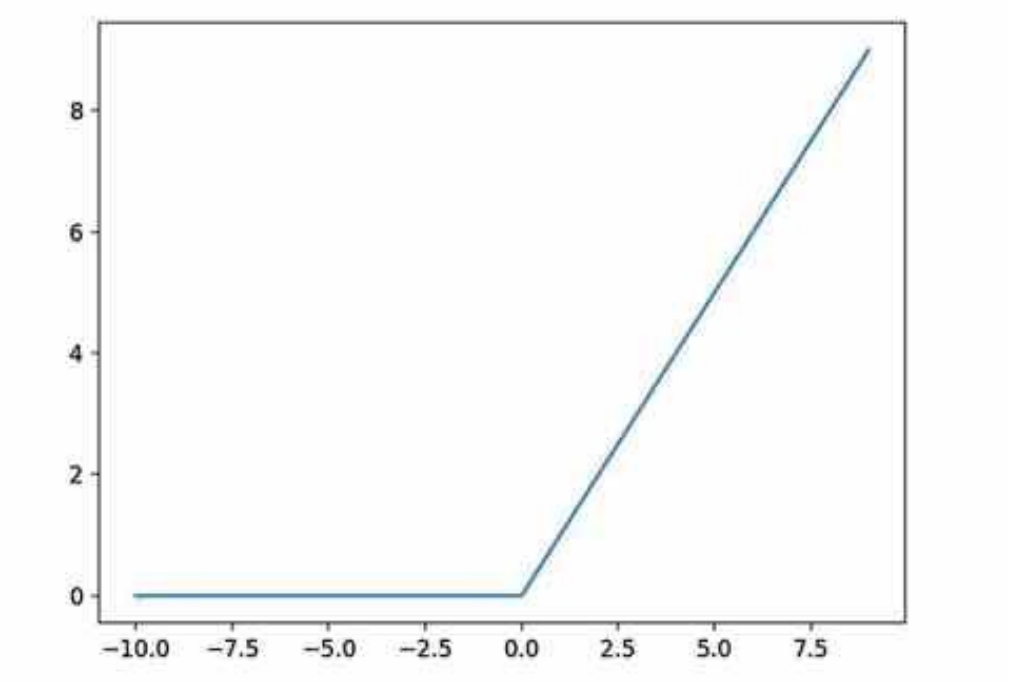
Tanh function

$$Tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



Rectified Linear Unit or ReLU

- The most popular activation function
- F(x) = max (0, x)
- It will give output as x if positive otherwise it will produce 0



Softmax function

- The softmax function is used in the final layer to generate an output.
- The output can be a final classification of an image for distinct categories.
- It calculates the probabilities for each of the target classes over all the possibilities.
- It is good for multiclass classification problems.

Activation functions

- There are other activation functions like Leaky ReLU, ELU.

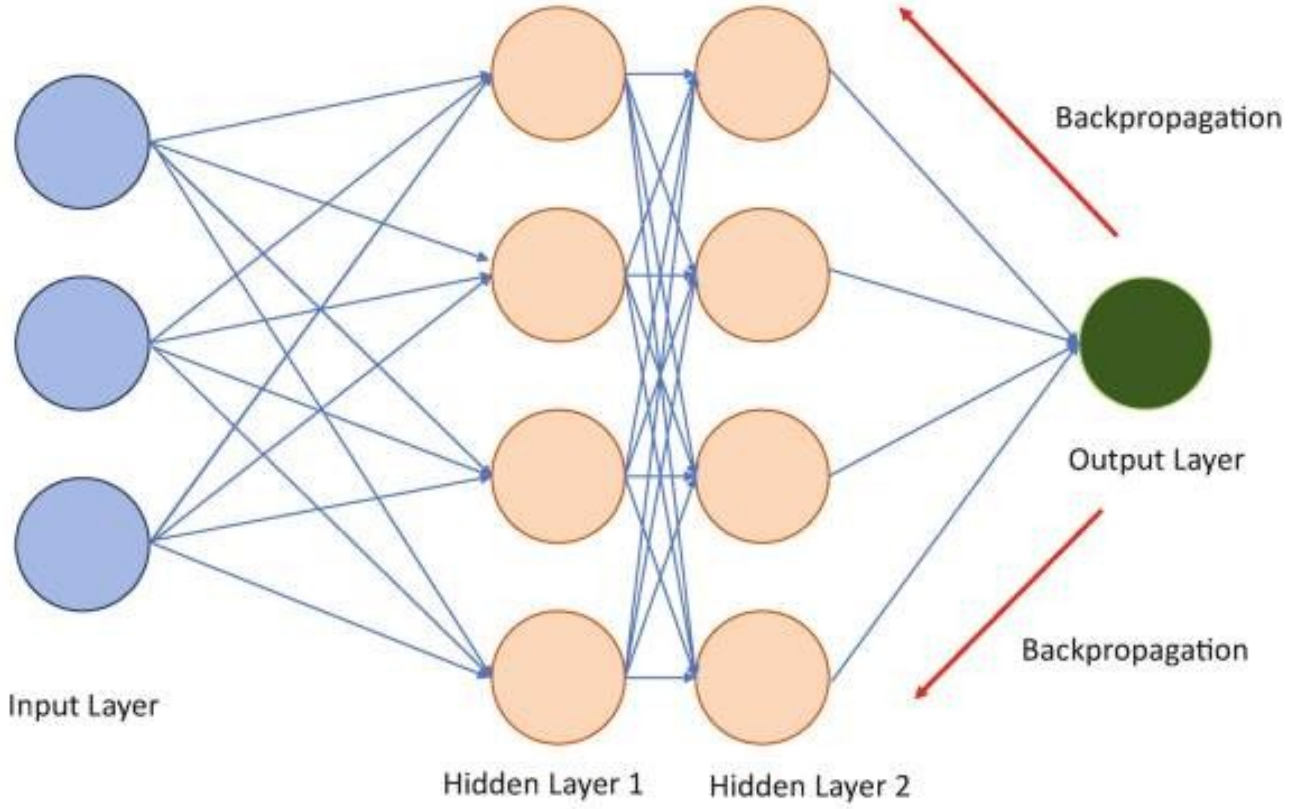
AF	Value	Positive	Challenges
Sigmoid	[0,1]	1-Nonlinear 2-Continuous differentiable	Output non-zero centerd
Tanh	[-1,1]	Gradient is stronger	Vanishing gradient
ReLU	[0,inf]	Easy to compute	Used only in hidden layers
Leaky ReLU	Max(0,x)	A variant of ReLU	Cannot be use for complex classification
ELU	[0,inf]	Alternative to ReLU	Can blow up the activations
Softmax	Calculates probabilites	Used in output layer	

Learning rate

- Learning rate defines the corrective steps which a model takes to reduce the errors.
- Learning rate governs the adjustments to be made to the weights during training of the network.
- In most of the cases, having a learning rate of 0.01 is acceptable.

Backpropagation

- Learning rate defines the size of the corrective steps to reduce the error, backpropagation is used to adjust the connection weights.

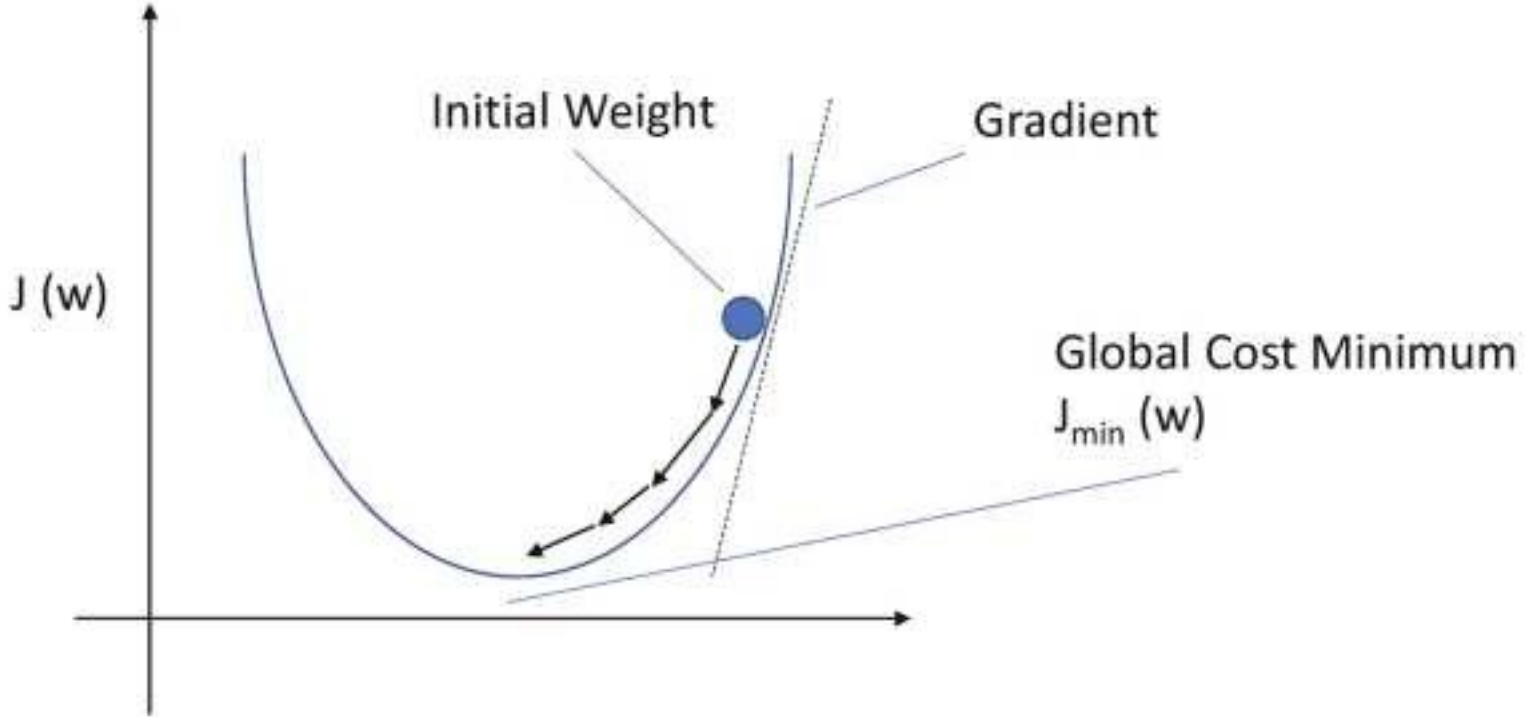


Overfitting

- if the network is working well on the training dataset but not so great on unseen dataset, it is called overfitting.
- To tackle overfitting, you can train your network with more training data. Or reduce the networks' complexity.
- Batch normalization and Dropout are two other techniques to mitigate the problem of overfitting

Gradient descent

- Gradient descent is used to find the global minimum or global maximum of a function. It is a highly used optimization technique.



Loss functions

- Loss is the measure of our model's accuracy.
- It is the difference of actual and predicted values.
- Different loss functions for regression and classification problems.

Loss Function	Equation for the Loss	Used for
Cross-entropy	$-y(\log(p) + (1-y)\log(1-p))$	Classification
Hinge loss	$\max(0, 1 - y * f(x))$	Classification
Absolute error	$\lvert y - f(x)\rvert$	Regression
Squared error	$(y - f(x))^2$	Regression
Huber loss	$L_\delta = \frac{1}{2}(y - f(x))^2$, if $\lvert y-f(x)\rvert <= \delta$ else $\delta\lvert y-f(x)\rvert - \frac{1}{2}\delta^2$	Regression

Deep Learning libraries

- TensorFlow developed by Google is arguably one of the most popular and widely used.
- Keras is one of the easiest Deep Learning frameworks for starters.
- PyTorch developed by Facebook, one of the popular Deep Learning libraries. It allows data parallelism and distributed learning models.
- Sonnet is developed using TF. Sonnet is designed for complex Neural Network applications and architectures.
- MXNet highly scalable Deep Learning tool

How to learn it?

- Deep Learning is a continuous learning experience that requires discipline, and commitment.
- Stay focused and disciplined.