Deep learning frameworks :-

① TensorFlow - Google's open source platform, TensorFlow is the most popular tool for machine learning and deep learning.

② PyTorch - it is OpenSource Deep learning framework developed by Facebook.

keras - Tool can run on top of tensorflow, Theano, microsoft Cognitive toolkit and plaid ML.

Tensor Flow Architecture :- Mainly Categorise 3 parts -

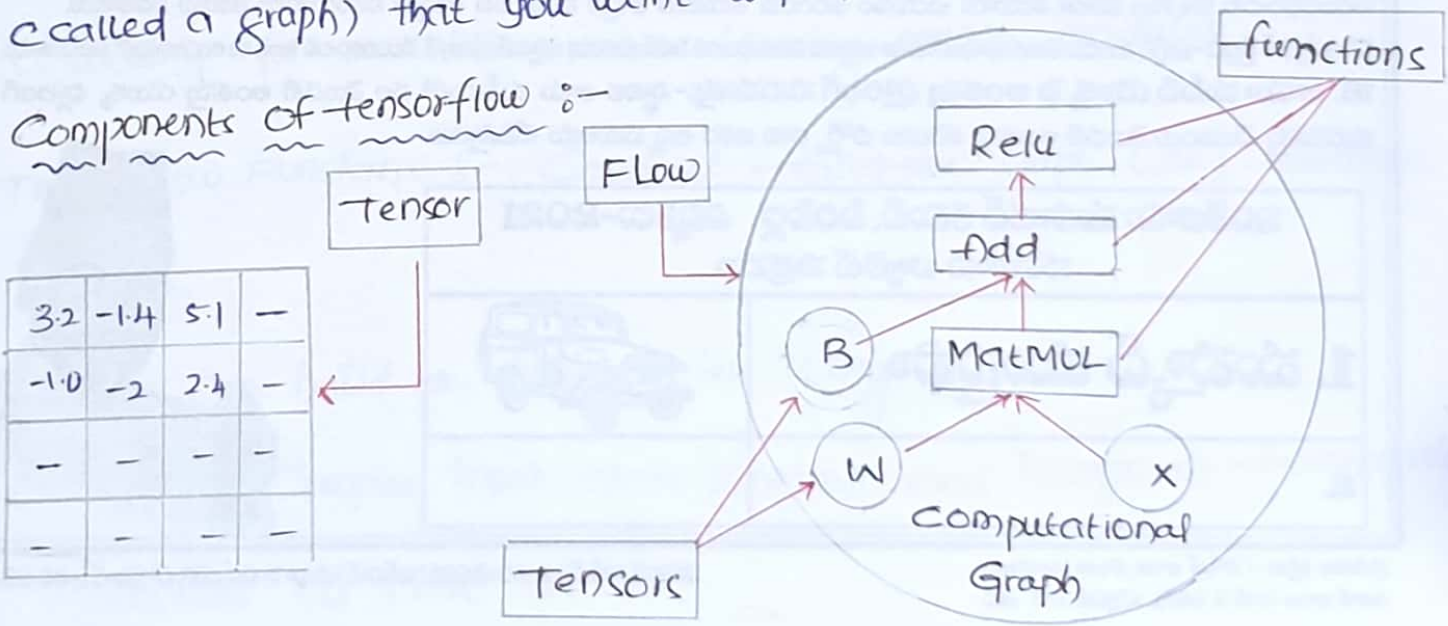
① preprocessing the Data.

② Build the model

③ Train and estimate the model

* it takes input as a multi-dimentional array, also known as Tensors., construct a shoot of flow chart of operations (called a graph) that you want to perform on the input.

Components of tensorflow :-



Tensor :- Input data or Result of the Computation.

Graphs :- The graph gathers and describes all the series computations done during the training.

Core modules and API of TensorFlow 2.0 :-

Keras :- Easy to use tensorflow API for quick photo typing of a deep learning model

With latest intigration and modification can be used for ~~d~~ production as well.

Data :- Data API helps users to build data pipeline for training a deep learning model.

Accelarators :- Build a distribution Strategy for model training using tensorflow accelarators.

Train on multiple GPU's.

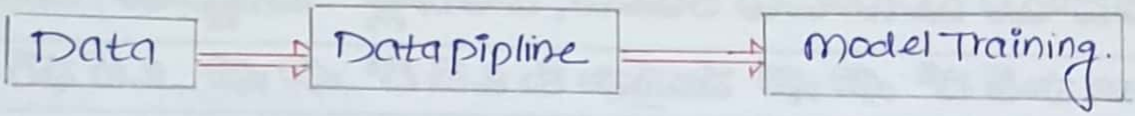
TensorFlow Hub :- Leaverage transfer learning using TF hub. a library for reusable machine learning ~~modules~~. modules.

TensorFlow Functions :- Construct tencorflow graph. Using functions module.

* We can build a data pipeline using tf.data.

* Design Complex input data pipeline and Incorporate transformation functions as part of it.

Data pipe-lines using TensorFlow 2.0 :-



TensorFlow keras :-

model :- i) Construct overall model structure

ii) A frame work for the model

Layers :- i) Configure Layers based on the deep learning model

ii) Convolution layers, Recurrent layers, Dense layers etc.

Compile :- i) Assign a optimizer and loss functions.

ii) Compile the model.

Training :- i) Train the model based on the no of Epochs provided.

CPU vs GPU vs TPU :-

Steps involved in making a model :-

1. Feed Forward
2. Back propagation
3. Fully Connected Layer - Forward pass
4. Fully Connected Layer - Backward pass.
5. Activation functions
6. Activation functions in practice.
7. Soft Max
8. Cross entropy.
9. Hands on MNIST - with all building blocks.

FCN (Fully Connected Network) - Forward :-

Neurons have Connections to all activations of the previous layer.

Number of Connections add up very quickly due to all Combinations.

forward pass - One matrix multiplication followed by by offset and activation function.

FCL - Backward pass :-

We have gradients Comming from next layer called Grad input

Size is always equal to the Size of Output.

We will Calculate the gradients which we need to pass to previous layers for Continuing the chain rule.

We will Calculate gradients also. (updating weights)

Activation functions :-

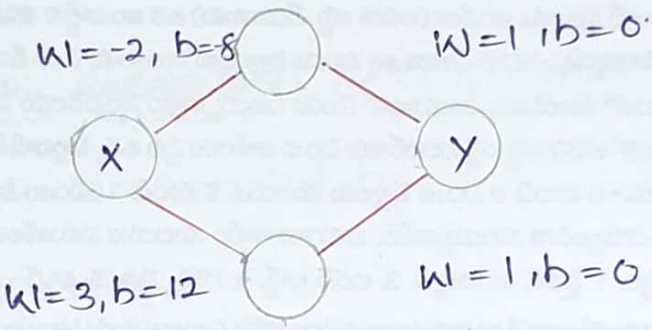
An AF takes a single input value and applies a function to it to add linearity.

Converts linear into non-linear for having a non-linear decision boundary.

AF's can decide which neuron is Switch ON, it acts as gate.

In practise - we use ReLU a lot, monitor the dead neurons, Never use Sigmoid, try leaky ReLU, Tanh

Vanishing Gradient :-



→ If we looking for updating the weights of w_3 or w_1, it's a huge multiplications of Gradient with Jacobians.

→ If all values are lessthan 1 we have vanishing gradient Problems.

→ If all values are grater than 1 wee have exploding gradient problems.

Soft Max :- Softmax is multinomial logistic classifier, it can handle multiple classes.

Soft max is typically the last layer of NN.

Softmax it self is an activation-function, hence it need not be combined with any activation-function.

Cross-entropy loss :-

① CE loss also called logloss quentifies, the deviation between predicted output and grand truth.

② perfect prediction whould have zero loss

③ with gradient descent, we will try to reduce the error.

Neural Networks ~~Connection~~ & Construction :-

Neural Networks Constructed by -

① Data preprocessing.

② Data ~~Agumentation~~

③ weight intiallization

④ Regularization.

Impliment all these ~~he~~ build networks.

① Data preprocessing :-

In data preprocessing we do data Normalization. by using label encoding or one hot encoding. wich are predefined methods.

Data Augumentation :- In this we do -

① Rotate

② Gray scale

③ Adding Random noise

④ Horizantal flip

⑤ Color jitter

Doing all these transformations, Since for each epoch do some random transformation for the Image.

if we Slightly move the Image, the pixels are changed and it altogether Sees a diffrent image.

Weight initialization :-

For Suppose if we have 3 Inputs, 4 hidden layers and 1 O/P neurone ie (3x12 etc. and 4x4)

if we go with initilization as zero, its not a good idea. because output will always move in Same direction.

Take the random numbers and Scale it based on the input neurons.

Dropout of a neuron :-

→ Dropout of a neuron is given as per probability, if P=0.5, if we drop 2 neurons in one layer and 3 in the next layer that dropsout total 5 neuron which will be randomly created paths.

Regularization (batch normalization) :-

In weight, we want our activations should be unit gaussians hence we initialized the weights similarity.

Usually inserted after all fully connected layers before the non-linearity.

Affine transformation :- (B.N)

Improve the gradient flow.

Allows higher learning rates.

Reduce strong dependency on Initialization.

Dropout :-

It forces the network to have redundant representation.

Predicting the same image.

Dropout is training a large ensemble of models.

Each binary mask is one model, gets trained on only one batch

At test time all neurons are ON.

We must scale the activations for each neuron so that output at test-time == expected output at train time.

Ex :- [for Cross Entropy]

Cross entropy is the mesure from the yield of Information-theory building upon entropy and generally calculating the diff b/w probability distributions.

Cross entropy can be used as a loss-function when optimizing classification models like logistic regression and artificial neural networks.

Ex :-

Day	Weather	Sunny	Raining
1	Sunny	1	0
2	Sunny	1	0
3	Rainy	0	1
4	Rainy	0	1
5	Sunny	1	0

Probability of Sunny = 0.57

probability of Rainy = 0.43

$$\text{Cross-Entropy} = -(1 * \log(0.57) + 0 * (\log(0.43))$$

$$= 0.23$$