Assignment - 1

Study of Deeplearning packages : Tensorflow, keras Theano and PyTorch. Document the distinct features and functionality of the packages.

Aim:

Study of Deep learning packages.

Objectives:

1. Tensorflow
2. keras
3. Theano
4. PyTorch

Theory:

1. Tensorflow

Tensor is a generalization of vector and matrices of potentially higher dimension array of data with varying dimension and ranks that are fed as input to the neural network are called tensor

They appear in multidimensional arrays.

When the data stored in tensor and fed into the neural network, the output we get is as shown below.

There are some terms associated with tensors that we need to familiarize ourselves with-

⇛ → Dimension

Dimension is the size of the array

elements.

3	5	4
2	6	7
8	1	3
2	5	9
1	4	2

Data in the form of array is fed as i/p to the network.

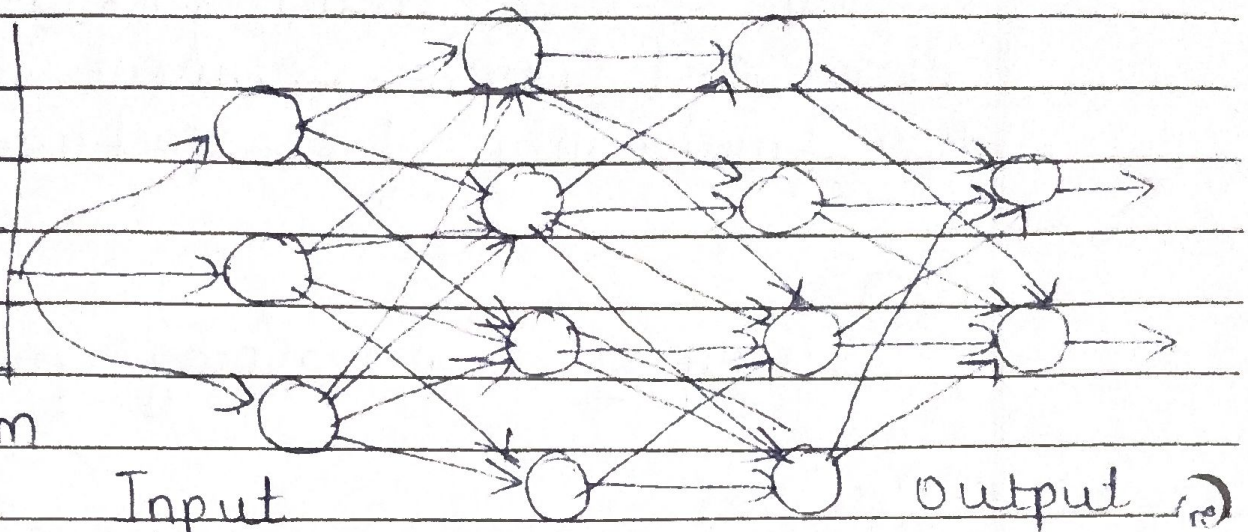

Fig. Neural Network Tensor

Algorithm supported by Tensorflow -

↳ Classification - tf.estimater.linearClassifier
↳ linear regression - tf.estimater.linearRegression
↳ Boosted tree classification - tf.estimater.BoostedTreeClassifier.
↳ Boosted tree regression - tf.estimater.DNNLinearCombinedClassifier

Tensorflow however data can be stored and manipulated using three different programming elements -

1. Constants
2. Variables
3. Placeholders.

Example -

"Hello World" program in tensorflow

```
hello = tf.constant('Hello')
type(hello)
tensorflow.python.framework.ops.Tensor
World = tf.constant('World')
result = hello + world
result
<tf.Tensor 'add=0' shape=() dtype=stringtype(resul
tensorflow.python.framework.ops.Tensor
with tf.session() as sess: result = sess.run(hello+world)
result
b'HelloWorld'
```

2. Keras

Keras is a deep learning API written in Python running on top of the machine learning platform tensorflow.

Keras is
- ↳ Simple
- ↳ flexible
- ↳ powerful

Sequential model

```
from .tensorflow.keras.models import sequential model =
    sequential()
```

Evaluate your test loss and metrics in one line

```
loss_and_metrics = model.evaluate(x-text, batch-size=128)
```

3. Theano

Theano is a python library for fast numerical computation that can be run on CPU or GPU
It is a key foundation library for deep learning in python that you can use directly to create deep learning models or wrapper libraries that greatly simplify the process.

Examples

```
import theano
from theano import tensor
a = tensor.dscalar()
b = tensor.dscalar()
c = a+b
f = theano.function([a,b],c).
```

4. PyTorch

PyTorch is the premiere open source deep learning framework developed and maintained by facebook.

PyTorch deep learning model lifecycle

The five steps are —

1. prepare the data

load and prepare data. Neural network models require numerical input data and numerical output data.

eg: construction of your dataset object can load your data file.

2. Define the model

Activation functions can also be define as layer such as ReLU, Softmax and sigmoid. Common examples include xavier and He weight ~~initiati~~ initialization scheme

1. . . .

2 xavier_uniform_(self.layer.weight).

3. Train and evaluate the model

once the model is fit, it can be evaluated on the test data set. This can be achieved by using the dataloader for the test dataset and collection collecting the predictions for the test set then comparing the prediction expected

1. . . .

2. For i, (inputs, targets) in enumerate

3. Evaluate the model on test set

4. ~~yhat~~ $\hat{y}$ = model(inputs)

5. . . . .

4. Make predictions.

Assignment - 2

Implementing feedforward neural networks with Keras and Tensorflow

Aim:

To implement feedforward neural networks.

Objectives:

a. import the necessary packages.
b. Load the training and testing data
c. Define the network architecture using keras
d. Train the model using SGD.
e. Evaluate the network.
f. plot the training loss and accuracy.

Theory:

To demonstrate, obtain baseline standard Neural Network which we will later compare to convolution neural network.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
get_ipython().run_line_magic('matplotlib.inline').
```

MNIST:

MNIST Stands for "Modified National Istitute of Standard and Technology"

It is a dataset of 70,000 handwritten image each image is of 20x20 pixels i.e about 784 features.

import dataset and split into train and test data

```
mnist = tf.keras.dataset.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data().
```

shape of training dataset 60,000 images having 28x28 size x_train.shape.

Network Architecture using Keras

Tensorflow is an open source library for creating and working with neural networks.

Creating the model

ReLU is one of the most popular activation functions

Mathematically,

$$y = \max(0, x)$$

returns 0 if the input is -ve and is linear if input is +ve.

Softmax is another activation function.

Training the model

```
history = model.fit(x_train, y_train, validation_data
          (x_test, y_test), epochs = 10).
```

Evaluate the model

```
test_loss , test_acc = model.evaluate (x_test , y_test)
print ("loss = %0.3f" % test_loss)
print ("Accuracy = %0.3f"% test_acc)
```

Prediction of data

Plot graph for accuracy and loss

```
get_ipython().run_line_magic('pinfo', 'history.history')
history.history.keys()
```

Assignment - 3

Build image classification model by dividing the model into 4 stages

Aim:

Build the image classification model by dividing the model into following 4 stages -

a. loading and preprocessing the image data.
b. Training the model
c. define the model
d. Estimating the model performance.

Objectives:

a. loading and processing the image data
b. training the model
c. Define the model architecture
d. Estimating the model performance.

Theory:

a. Loading and processing the image data
→ use high level keras preprocessing utilities and layers such as tf.keras.layers.Rescaling to read a directory of image on disk
→ write input pipeline from scratch
→ download a dataset from the large catalog available

```
import numpy as np
import os
import PIL.
```

```
import PIL.Image
import tensorflow as tf
import tensorflow-dataset as tfds.

print (tf--version)
```

Output

2.9.1

b. Training the model.

Training means learning good values for all the weights and bias from labelled example.

There are 6 steps

1. Setup a google cloud account
2. Create a project
3. Deploy deep learning virtual machine
4. Access jupyter notebook GUI
5. Add GPUs to virtual machine
6. Change virtual machine configuration.

c. Define the model architecture

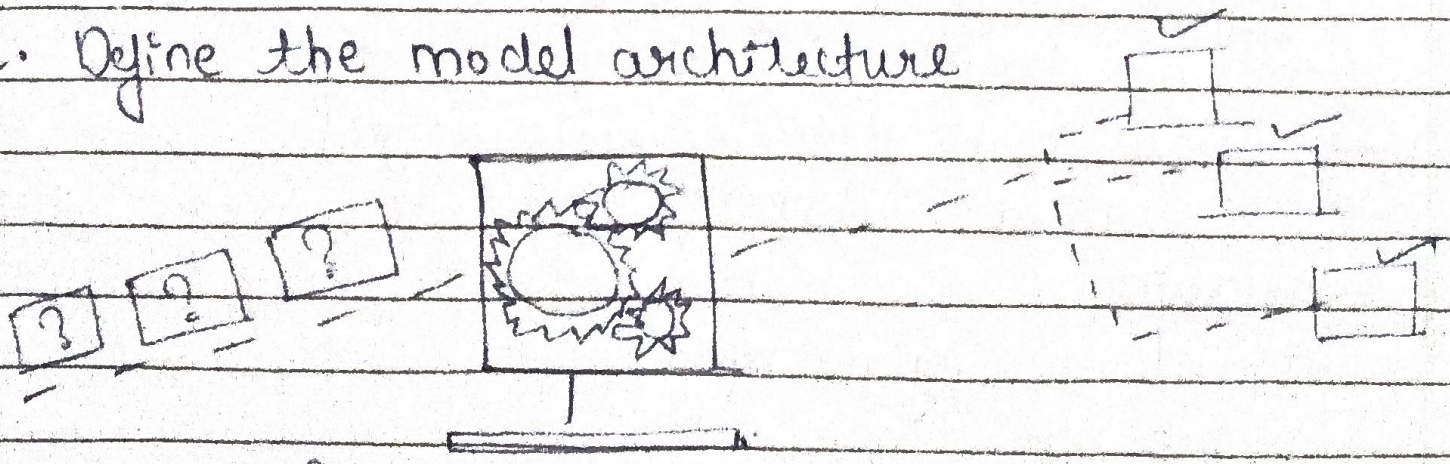

Fig. Image processing

Architectures
↳ AlexNet (2012)
↳ ZfNet (2013)
↳ GoogleNet (2014)
↳ ResNet (2015)
↳ resNext (2016)
↳ DenseNet (2016)
↳ SENet (2018)

d. Estimating the model performance
most frequent classification evaluation metric should be accuracy.
1. Confusion matrix for a 2 class classification problem
2. The key classification metrics: Accuracy, Precision Recall.

Confusion Matrix.

		Actual Value	
		Positive	negative
Predicted Value	Positive	TP	FP
	Negative	FN	TN

3. The difference betn precision and recall in specific cases.
4. Decision threshold and receiver operating characteristic curve (ROC).

Equations

1. TPR

$$\text{True positive rate} = \frac{\text{true Positives}}{\text{true positives + false negative}}$$

2. FPR

$$\text{false positive rate} = \frac{\text{false Positive}}{\text{false positives + true negatives}}$$