TENSORFLOW CHEAT SHEET

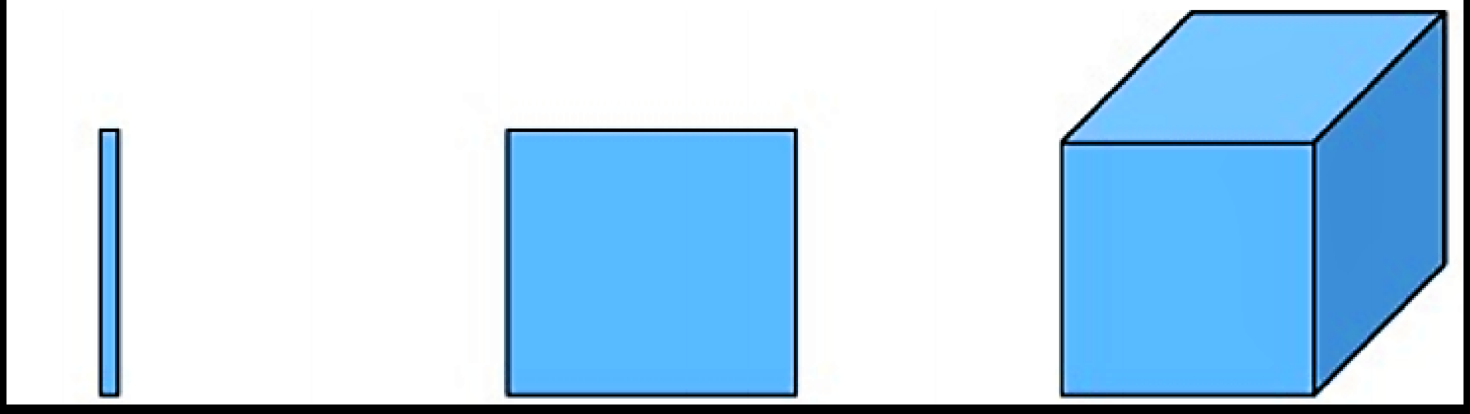
Tensorflow:

TensorFlow is an open-source library of software for dataflow and differential programming for various tasks. Similarly, TensorFlow is used in machine learning by neural networks. Developed by Google in 2011 under the name DistBelief, TensorFlow was officially released in 2017 for free. The library is able to run on multiple CPUs and GPUs and is available across multiple platforms, including mobile. The name comes from multidimensional arrays known as tensors, which are commonly used in neural networks.

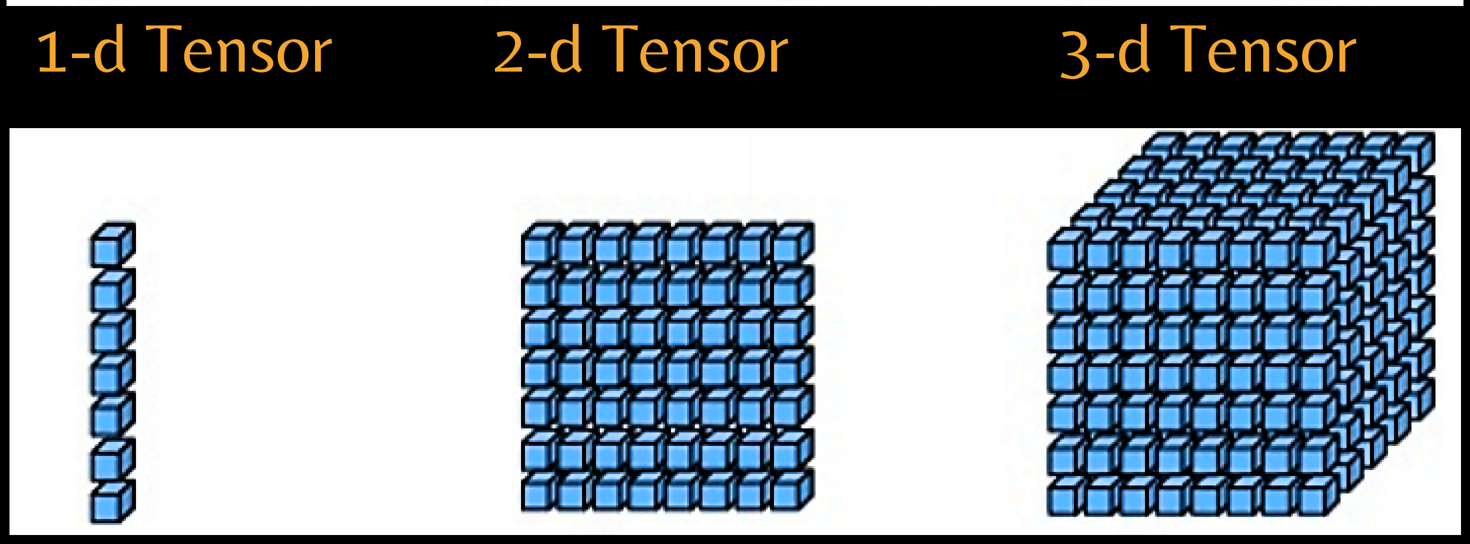
Tensor:

Like vectors and matrices, tensors can be represented in Python using the N-dimensional array (ndarray).

A tensor can be defined in-line to the constructor of array() as a list of lists.



1-d Tensor 2-d Tensor 3-d Tensor



4-d Tensor 5-d Tensor 6-d Tensor

Installation:

HOW TO INSTALL TENSORFLOW?

```
device = cpu/gpu
python_version = cp27/cp34
sudo pip install
https://storage.googleapis.com/
tensorflow/linux/$device/tensorflow-
0.8.0-$python_version-none-linux_x86
_64.whl
```

INSTALLING TF IN PYTHON NOTEBOOK:

```
pip install tensorflow
pip install tensorflow-gpu
```

Importing TensorFlow:

```
%tensorflow_version 2.x
import tensorflow as tf
```

TensorFlow operations:

CREATING TENSORS:

```
con=tf.constant(1)      # creating a constant
var=tf.variable()       # creating a variable
```

TENSORS WITH VALUE AND DATATYPE:

```
string = tf.Variable("Sarthak Saraiya", tf.string)
number = tf.Variable(129, tf.int16)
floating = tf.Variable(3.567, tf.float64)
a = tf.constant([3., 3., 3.])
b = tf.constant([2., 2., 2.])
```

ADDITION OF TWO TENSORS:

```
sum = tf.add(a, b)              # [ 5. 5. 5. ]
```

SUBTRACTION OF TWO TENSORS:

```
diff = tf.subtract(a, b)        # [ 1. 1. 1. ]
```

PRODUCT OF TWO TENSORS:

```
prod = tf.multiply(a, b)        # [ 6. 6. 6. ]
```

DIVISION OF TWO TENSORS:

```
quot = tf.divide(a, b)          # [ 1.5 1.5 1.5 ]
```

RANK OF A TENSOR:

```
tf.rank(b)                      # 1
```

SHAPE OF A TENSOR:

```
tf.shape(b)                     # shape=(1,)
```

SIZE OF A TENSOR:

```
tf.size(b).numpy()              # 3
```

MATRIX MULTIPLICATION OF TWO TENSORS:

```
matrix1 = tf.constant([[3., 3.]])
matrix2 = tf.constant([[2.],[2.]])
product = tf.matmul(matrix1, matrix2)
```

SOME USEFUL FUNCTIONS:

```
tf.get_default_session()
tf.get_default_graph()
tf.reset_default_graph()
ops.reset_default_graph()
tf.device(“/cpu:0”)
tf.name_scope(value)
tf.convert_to_tensor(value
```

MAIN CLASSES:

```
tf.Graph()
tf.Operation()
tf.Tensor()
tf.Session()
```

TF Optimizers:

GradientDescentOptimizer
AdadeltaOptimizer
AdagradOptimizer
MomentumOptimizer
AdamOptimizer
FtrlOptimizer

TENSORFLOW CHEAT SHEET

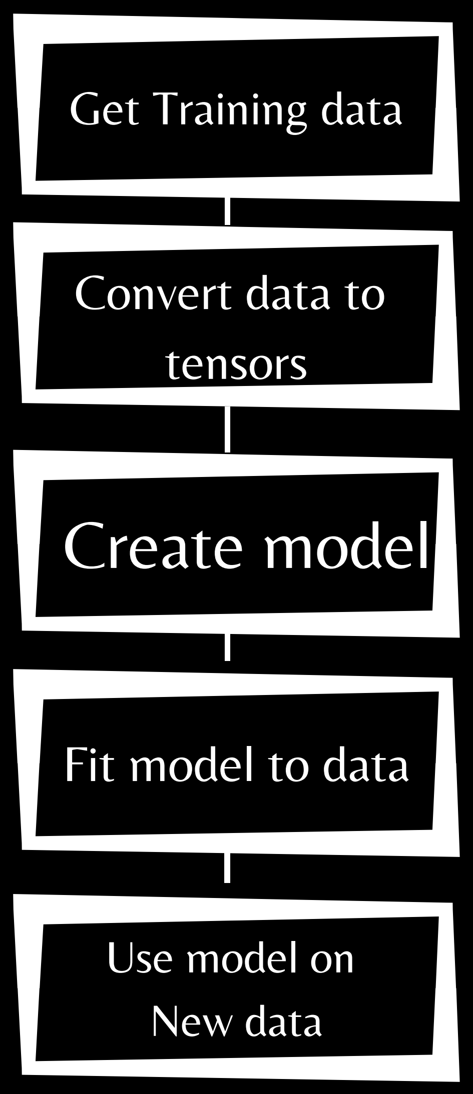
TF Core Learning Algorithms:

1) LINEAR REGRESSION:

Linear Regression is a very common statistical method that allows us to learn a function or relationship from a given set of continuous data.

EPOCH:

The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset.



Get Training data

Convert data to tensors

Create model

Fit model to data

Use model on New data

LR with TF:

✓ Importing the Necessary Modules (numpy, tensorflow)

✓ Creating a Random Data Set

✓ Setting up the Hyperparameters (learning_rate,epoch,display_step)

✓ Initializing Weights and Biases

✓ Defining Linear Reg and Cost Func

✓ Building Optimizers and Gradients

✓ Constructing the Training Loop

✓ Visualizing Linear Regression

2) CLASSIFICATION:

Classification aims at predicting the probability of each class given a set of inputs. The label (i.e., the dependent variable) is a discrete value, called a class.

3) CLUSTERING:

Clustering is a Machine Learning technique that involves the grouping of data points. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features.

Basic Algorithm for K-Means.

Step 1: Randomly pick K points to place K centroids

Step 2: Assign all the data points to the centroids by distance. The closest centroid to a point is the one it is assigned to.

Step 3: Average all the points belonging to each centroid to find the middle of those clusters (center of mass). Place the corresponding centroids into that position.

Step 4: Reassign every point once again to the closest centroid.

Step 5: Repeat steps 3-4 until no point changes which centroid it belongs to.

https://colab.research.google.com/drive/15Cyy2H7nT40sGR7TBN5wBvgTd57mVKay#forceEdit=true&sandboxMode=true&scrollTo=ujwvc6ASsHID (Click on the google colab link to see the implementation)

Neural Network:

A typical implementation of Neural Network would be as follows:

- Define Neural Network architecture to be compiled
- Transfer data to your model
- Under the hood, the data is first divided into batches, so that it can be ingested. The batches are first preprocessed, augmented and then fed into Neural Network for training
- The model then gets trained incrementally
- Display the accuracy for a specific number of timesteps
- After training save the model for future use
- Test the model on a new data and check how it performs

https://colab.research.google.com/drive/1m2cg3D1x3j5vrFc-Cu0gMvc48gWyCOuG#forceEdit=true&sandboxMode=true&scrollTo=Hivk879ZQhxU (Click on the google colab link to see the implementation)

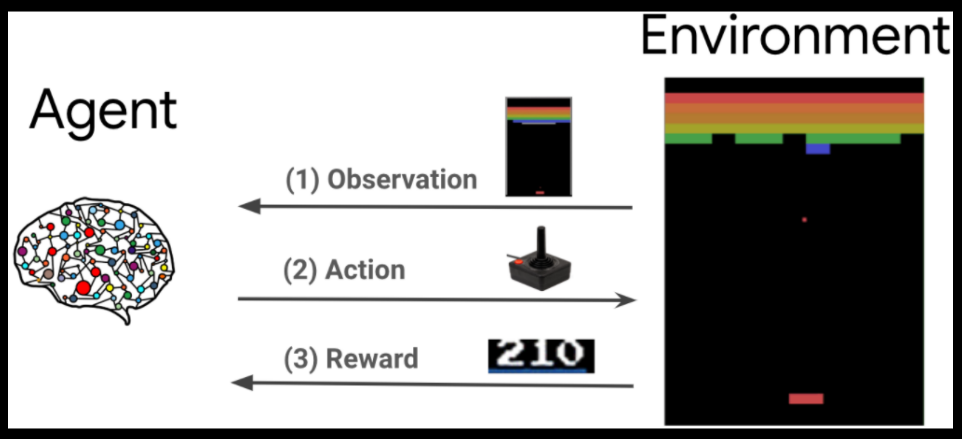
Natural Language Processing:

Natural Language Processing (or NLP for short) is a discipline in computing that deals with the communication between natural (human) languages and computer languages.

https://colab.research.google.com/drive/1ysEKrw_LE2jMndo1snrZUh5w87LQsCxk#forceEdit=true&sandboxMode=true (Click on the google colab link to see the implementation)

Reinforcement Learning:

Reinforcement learning (RL) is a general framework where agents learn to perform actions in an environment so as to maximize a reward. The two main components are the environment, which represents the problem to be solved, and the agent, which represents the learning algorithm.



Important links:

✓ https://www.tensorflow.org/tutorials

✓ https://www.geeksforgeeks.org/introduction-to-tensorflow/

✓ https://www.tutorialspoint.com/tensorflow

✓ https://www.datacamp.com/community/tutorial

✓ https://www.youtube.com/watch?v=tPYj3fFJGjk