

Tensorflow =>

cmd=>pip install tensorflow

The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining.

```
In [2]: import tensorflow as tf
```

```
In [3]: print(tf.__version__)
```

2.14.0

```
In [5]: print(tf.reduce_sum(tf.random.normal([100, 1000])))
```

tf.Tensor(-96.71722, shape=(), dtype=float32)

Keras

```
In [6]: from keras import datasets
```

```
In [8]: print(kf.__version__)
```

2.14.0

Load MNIST

```
In [9]: (train_images, train_labels) , (test_images , test_labels) = datasets.mnist.load_data()
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11490434/11490434 [=====] - 3s 0us/step

```
In [10]: train_images.shape , test_images.shape
```

```
Out[10]: ((60000, 28, 28), (10000, 28, 28))
```

Theano

```
In [17]: import numpy
```

```
In [ ]: import theano.tensor as T
```

```
In [ ]: from theano import function
```

Adding Two variables

```
In [ ]: x = T.dscalar('x')
```

```
In [ ]: y = T.dscalar('y')
```

```
In [ ]: x = x+y
```

```
In [ ]: f= function([x,y],z)
```

```
In [ ]: f(5,7)
```

PyTorch

cmd=>pip3 install torch torchvision torchaudio

```
In [25]: import torch
```

```
In [27]: import torch.nn as nn
```

```
In [28]: print(torch.__version__)
```

2.1.0+cpu

Neural Network

A neural network is an artificial intelligence model inspired by how the human brain functions. Neural networks comprise of numerous neurons that take inputs and produce outputs using a set of trainable parameters, such as weights and biases.

CNN

A complete convolutional neural network can be broken down into two parts:

CNN: The convolutional neural network that comprises the convolutional layers.

ANN: The artificial neural network that comprises dense layers.

Flatten Layer

The flatten layer is a component of the convolutional neural networks (CNN's).

Intuition behind flattening layer is to convert data into 1-dimensional array for feeding next layer. we flattened output of convolutional layer into single long feature vector.

Convolutional layer

Image filtering (kernel) is a process of modifying an image by changing its shades or colour of pixels. It is also used for brightness and contrast.

Pooling layer

Pooling layer is used to reduce feature map dimensions. Thus it reduces the number of parameters to learn and the amount of computation performed in the network. Pooling layer summarises

features present in a region of feature map generated by convolutional layer.

In []: