**DESCRIPTION OF PYTORCH AS A FRAMEWORK**.

PyTorch: It is the framework of machine learning produced by Facebook in 2016. PyTorch is an open source liberty use for machine learning training programs based on the popular Torch library. PyTorch was is designed to provide a good flexibility with a high speeds for deep neural network implementation. Description

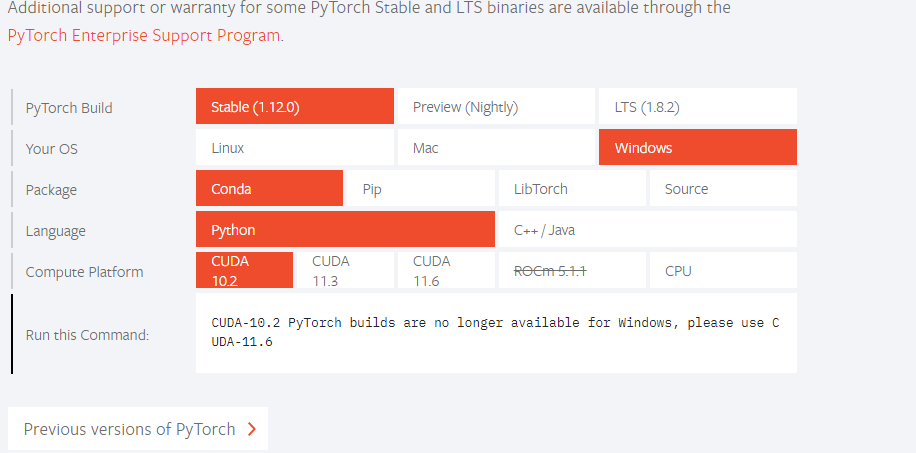
PyTorch differ from other deep learning frameworks like TensorFlows in that it uses **dynamic computation graphs**. TensorFlows are defined prior to runtime in a forward computation to achieve a dynamic graphs

**How to Install PyTorch**

Initially install python and all basic libraries to work with PyTorch

1. Install [Anaconda](https://pytorch.org/get-started/locally/" \l "windows-anaconda)
2. Install [pip](https://pytorch.org/get-started/locally/" \l "windows-pip)
3. Then [Verification](https://pytorch.org/get-started/locally/" \l "windows-verification)

Go on <https://pytorch.org/> to get the installation command of PyTorch.



## PyTorch supports the various tensor function with different operations like NumPy.

## 

## PyTorch-tensors are similar to NumP arrays with additional feature such that they can be used on Graphical Processing Unit (GPU) to accelerate computing.

## PyTorch-Tensor can run on Central Processing Unit as well as Graphical Processing Unit. If a PyTorch-Tensor require o be executed on Graphical Processing Unit, the user just need to cast the Tensor to a CUDA datatype.

**PyTorch vs TensorFlow**

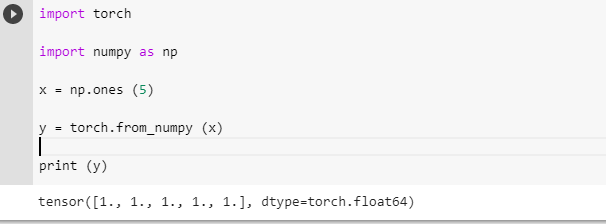
**Table 1 PyTorch vs TensorFlow by owner**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Classification | **PyTorch** | **TensorFlow** |
| 1 | **Dynamic vs Static** | PyTorch uses dynamic computation graphs | TensorFlow uses static computation graphs |
| 2 | **Data Parallelism** | PyTorch uses asynchronous execution of Python to implement data parallelism | TensorFlow you need to manually configure every operation for data parallelism |
| 3 | **Visualization Support** | PyTorch users can utilize TensorBoard to log PyTorch models and metrics within the TensorBoard UI. Scalars, images, histograms, graphs, and embedding visualizations. PyTorch initially had a visualization library called Visdom. | TensorFlow has a very good visualization library called TensorBoard. The visualization support helps developers to track the model training process nicely |
| 4 | **Model Deployment** | PyTorch does not provide a framework like serving to deploy models onto the web using REST Client. | TensorFlow has great support for deploying models using a framework called TensorFlow serving. TensorFlows uses REST Client API for using the model for prediction once deployed. |

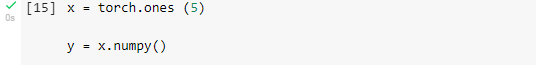
# **Prerequisites to Build the Model**

1. Import all necessary libraries for loading our data
2. Access the data in the dataset
3. Loading the data
4. Iterate over the data
5. [Optional] Visualize the data

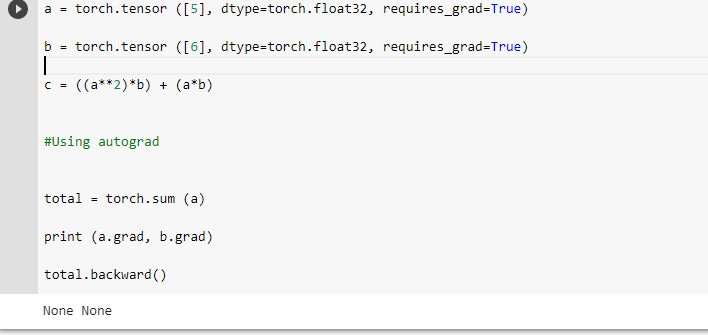
Converting Numpy array to tensor



**Tensor to Numpy**:



## Autograd Module.



PyTorch can provides an **autograd** package that provides **automatic differentiation** for all operations in Tensors which can be define by a run framework. These means that a backdrop is defined by how a code is run and that every single iteration can be different in a works by recording for all the operations perform and replay in the compute backward gradients

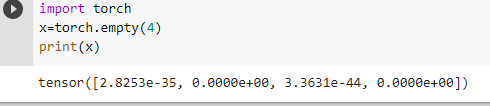
Optima Model

## 

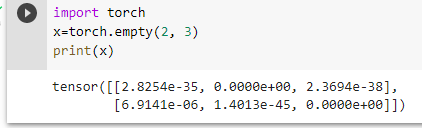
Users can instead of manually updating the weights of the model as have seen earlier, but use the **optim** package to define an **Optimizer** that will update the weights for him or her.

**Basic Coding with Pytorch different dimensions and ransoms values**

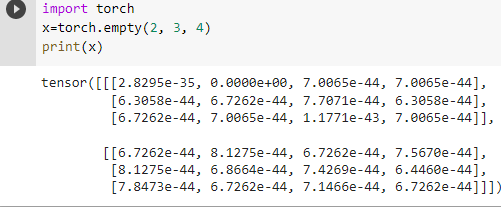
**1D dimension metrics**



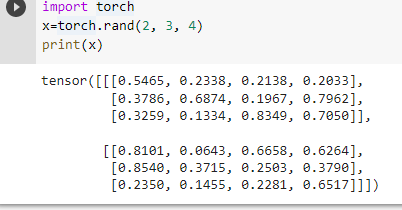
**2D dimension Metrics**



**3D dimension Metrics**

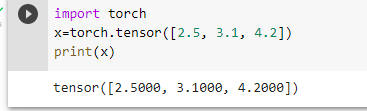


**Pytorch With random Values**

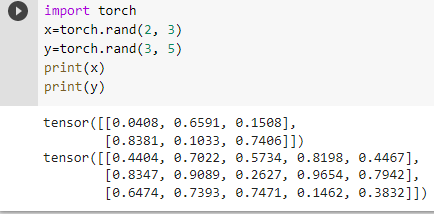


**Pytorch versus tensor**

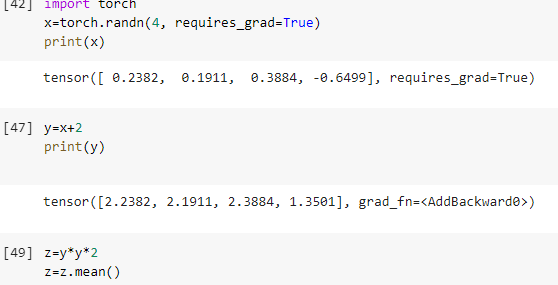
1 Single operation



2 Two operations



**Autograd Basic Coding with Pytorch**



**To conclude** **Deep learning** is a subset of machine learning where artificial **neural networks**, algorithms inspired by the human brain, learn from large amounts of data. Every once in a while, there comes a library or framework that provides us new insights into the field of Deep Learning that allows attaining remarkable progress. At the moment Pytorch allow us attain a remarkable progress.