Pytorch Series (Lecture # 1)

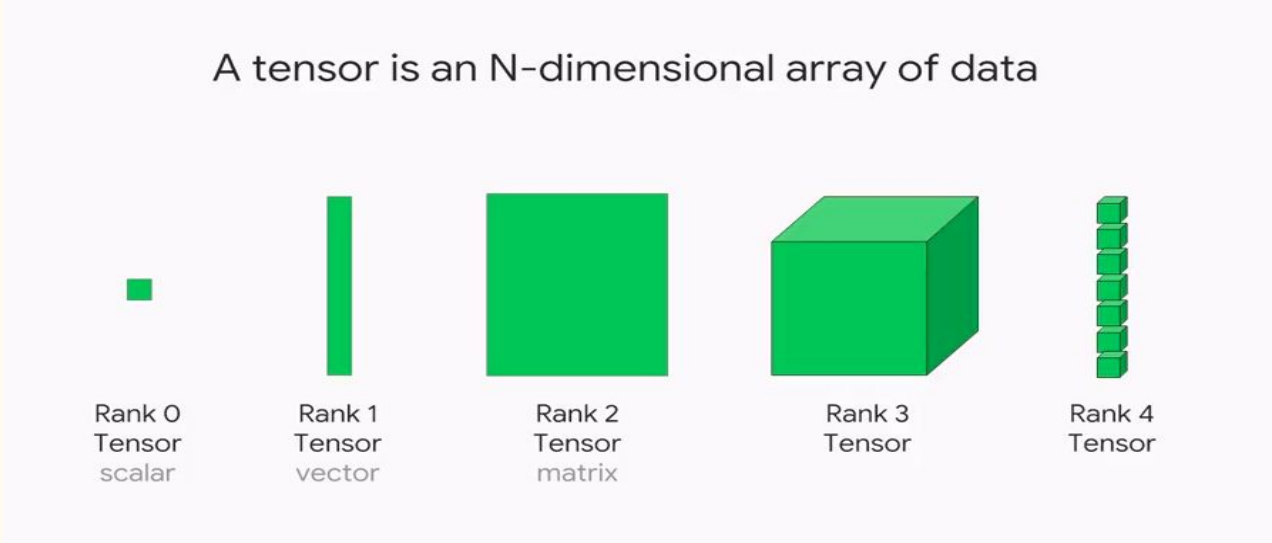
Rizwan Ali Shah

PyTorch: A Brief History

- The initial release of PyTorch was in October of 2016, and before PyTorch was created, there was and still is, another framework called *Torch*.
- Torch is a machine learning framework is based on the Lua programming language.
- Soumith Chintala is credited with bootstrapping the PyTorch project.
- PyTorch was created because Lua version of Torch was aging, so a newer version written in Python was created named “PyTorch”.

Introduction

- *PyTorch is a deep learning framework and a scientific computing package.*
- *Scientific computing aspect of PyTorch is primarily a result PyTorch's tensor library and associated tensor operation*



Introduction

- Transition between Numpy and Pytorch is very easy.
  - a.numpy() (from torch to numpy)
  - torch.from_numpy(a) (from numpy to torch)

- PyTorch tensors, GPU support is built-in and very easy with PyTorch to move tensors to and from a GPU.
  - PyTorch tensor operations can be performed on a GPU.

- PyTorch tensors and their associated operations are very similar to NumPy n-dimensional arrays.

Introduction

"Tensors are very important for deep learning and neural networks because they are the data structure that we ultimately use for building and training our neural networks."

Deep Learning With PyTorch

The table gives us a list of PyTorch packages and their corresponding descriptions.

Package	Description
torch	The top-level PyTorch package and tensor library.
torch.nn	A subpackage that contains modules and extensible classes for building neural networks.
torch.autograd	A subpackage that supports all the differentiable Tensor operations in PyTorch.
torch.nn.functional	A functional interface that contains typical operations used for building neural networks like loss functions, activation functions, and convolution operations.
torch.optim	A subpackage that contains standard optimization operations like SGD and Adam.
torch.utils	A subpackage that contains utility classes like data sets and data loaders that make data preprocessing easier.
torchvision	A package that provides access to popular datasets, model architectures, and image transformations for computer vision.

Why Use PyTorch For Deep Learning?

- PyTorch is a shallow framework that stays out of the way.
- PyTorch uses a computational graph that is called a dynamic computational graph.
- Neural Networks with PyTorch are super close to programming neural networks from scratch.
- Focus on neural networks and less on the actual framework.
- Its as fast as the competitor deep learning libraries
- Close to Python ecosystem
- PyTorch will be capable of adapting to the rapidly evolving deep learning environment as things change over time.

Installing PyTorch With Anaconda

Steps to follow:

- Download and install Anaconda (choose the latest Python version).
  - https://www.anaconda.com/products/individual
- Go to PyTorch's site and find the *get started locally* section.
  - conda install pytorch torchvision cpuonly -c pytorch (CPU only)
  - conda install pytorch torchvision cudatoolkit=10.2 -c pytorch (GPU)
- Specify the appropriate configuration options for your particular environment.
- Run the presented command in the terminal to install PyTorch.
- Verify in Command Prompt
  - conda list torch

Verify The PyTorch Install

Steps to verify the install:

1. To use PyTorch we import torch.
2. To check the version, we use torch.__version__

Now, to verify our GPU capabilities:

1. torch.cuda.is_available()
2. torch.version.cuda

Note : “If your torch.cuda.is_available() call returns false, it may be because you don’t have a supported Nvidia GPU installed on your system”.

If you’re interested in checking whether your Nvidia GPU supports CUDA, you can check for it here.