**Introduction to pytorch**

Pytorch is a framework developed by Facebook. This is a giant in technology that invests a lot of resources in the development of Artificial Intelligence. Pytorch is developed with an open source license so it makes for a very large community. A large community means a lot of resources to learn and your problems may have already been solved by someone else. share with the community. Pytorch along with Tensorflow and Keras is one of the popular frameworks used in Deep Learning problems today. In particular, in research fields, almost all authors use pytorch to develop their problems. Pytorch shows its advantage in the research field because it is very easy for you to debug and visualize, in addition it follows the Dynamic Graphs mechanism that allows to reduce the time to train the model.

**Tensor in pytorch**

Tensor is like Numpy array but converted to tensor for use on GPU. Numpy's math, transformation, and basic operations are all possible on Tensor. Here, I will give some basic syntax on Tensor so that you can understand the basics

#Tạo một Tensor từ list cho trước sử dụng torch.Tensor

t = torch.Tensor([[1,2,3],[3,4,5]])

#Tạo một Tensor với kích thước (2, 3) cho trước và có giá trị ngẫu nhiên tuân theo phân phối chuẩn với trung vị bằng 0 và phương sai bằng 1

t = torch.randn(2, 3)

# Tạo một Tensor với kích thước (2, 3) cho trước và tất cả phần tử có giá trị đều bằng 1

t = torch.ones(2, 3)

# Tạo một Tensor với kích thước (2, 3) cho trước và tất cả phần tử có giá trị đều bằng 0

t = torch.zeros(2, 3)

#Tạo một tensor có kích thước (2,3) với giá trị nằm trong khoảng từ 0->10

t = torch.randint(low = 0,high = 10,size = (2,3))

#Sử dụng torch.from\_numpy để chuyển đổi từ Numpy array sang Tensor

a = np.array([[1,2,3],[3,4,5]])

t = torch.from\_numpy(a)

# Sử dụng .numpy() để chuyển đổi từ Tensor sang Numpy array

t = t.numpy()

Numpy arrays vs PyTorch tensors.

Numpy is a Python library for storing and processing calculations with real numeric data. However, Numpy is written in C / C ++, so processing and computation speed is very fast.

PyTorch tensors are similar in function and purpose to Numpy arrays, but with a few more advantages:

* Perform fast computation on GPU, because DL models that want to be accelerated need to be processed through GPUs, so tensors that support fast computation on GPUs are essential.
* Pytorch tensors can save computational graphs, so they can calculate derivatives quickly, serving the backpropagation algorithm in Deep Learning, I will talk about it in the next lesson about autograd.

Torch Properties.

Torch tensors contain only numeric and bool (True/False) data. Each torch tensor belongs to a data type, in the dtype attribute. Here is a list of the data types torch tensors can contain:

torch.float32 or torch.float: 32-bit floating-point

torch.float64 or torch.double: 64-bit, double-precision floating-point

torch.float16 or torch.half: 16-bit, half-precision floating-point

torch.int8: signed 8-bit integers

torch.uint8: unsigned 8-bit integers

torch.int16 or torch.short: signed 16-bit integers

torch.int32 or torch.int: signed 32-bit integers

torch.int64 or torch.long: signed 64-bit integers

torch.bool: Boolean