



Recitation 0.4

PyTorch

Ahmed Safwat Abouhashem
Nayasha Gandotra

PyTorch

Open-source deep learning framework — developed by Facebook AI Research

Pythonic and User-Friendly — easy to learn

Flexible and intuitive — easy to build, modify, and debug custom models

Dynamic computation graphs

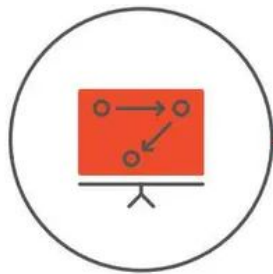
GPU-accelerated tensor computations

- Tensors are multidimensional arrays
 - Vector \rightarrow 1D tensor
 - Matrix \rightarrow 2D tensor
 - Extendable to n dimensions

Popular — large community & adopted by top AI labs and researchers



RESEARCH TO PRODUCTION



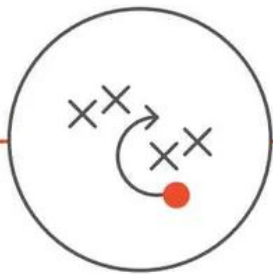
1

DETERMINE
APPROACH



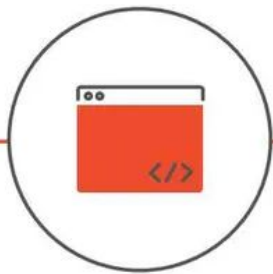
2

PREPARE
DATA



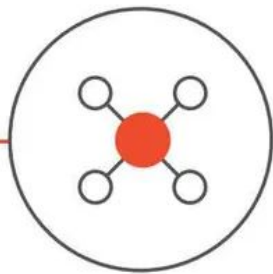
3

BUILD AND
TRAIN
MODEL



4

TRANSFER
MODEL TO
PRODUCTION



5

DEPLOY
AND SCALE

Python API

control where torch.Tensor is allocated

```
>>> torch.device('cuda', 0)
device(type='cuda', index=0)
```

```
>>> torch.device('mps', 0)
device(type='mps', index=0)
```

```
>>> torch.device('cpu', 0)
device(type='cpu', index=0)
```

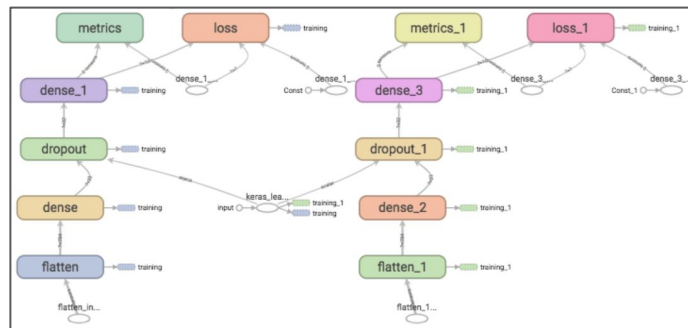
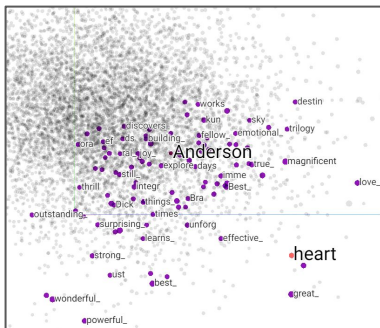
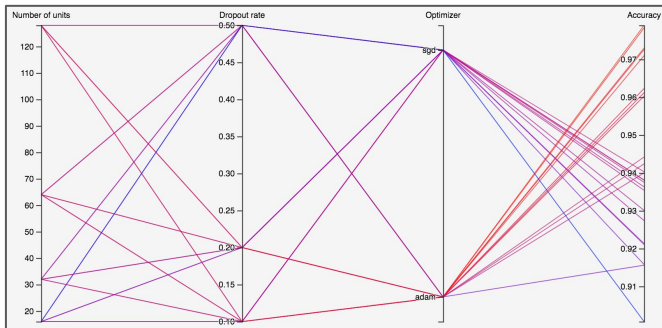
automatic differentiation (**torch.autograd**)

```
requires_grad=True
```

various optimization algorithms (**torch.optim**)

```
optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.9)
optimizer = optim.Adam([var1, var2], lr=0.0001)
```

visualize hyperparameter tuning, model graphs, ...etc (**tensorboard**)



Convolution layers

<code>nn.Conv1d</code>	Applies a 1D convolution over an input signal composed of several input planes.
<code>nn.Conv2d</code>	Applies a 2D convolution over an input signal composed of several input planes.
<code>nn.Conv3d</code>	Applies a 3D convolution over an input signal composed of several input planes.
<code>nn.ConvTranspose1d</code>	Applies a 1D transposed convolution operator over an input image composed of several input planes.
<code>nn.ConvTranspose2d</code>	Applies a 2D transposed convolution operator over an input image composed of several input planes.
<code>nn.ConvTranspose3d</code>	Applies a 3D transposed convolution operator over an input image composed of several input planes.

Non-linear activation

<code>nn.Sigmoid</code>	Applies the Sigmoid function element-wise.
<code>nn.Tanh</code>	Applies the Hyperbolic Tangent (Tanh) function element-wise.
<code>nn.ReLU</code>	Applies the rectified linear unit function element-wise.
<code>nn.LeakyReLU</code>	Applies the LeakyReLU function element-wise.
<code>nn.Softmax</code>	Applies the Softmax function to an n-dimensional input Tensor.
<code>nn.Softmax2d</code>	Applies SoftMax over features to each spatial location.

<https://pytorch.org/>

Loss functions

<code>nn.L1Loss</code>
<code>nn.MSELoss</code>
<code>nn.CrossEntropyLoss</code>
<code>nn.CTCLoss</code>
<code>nn.NLLLoss</code>
<code>nn.PoissonNLLLoss</code>

Modeling

- Computer vision
- Language
- Specialized
- Medical & Biology
- Multimodal
- Adversarial & Robustness
- 3D



<https://github.com/microsoft/torchvision>

popular datasets
pretrained models

Rich ecosystem!



Detectron2

object detection &
segmentation



<https://github.com/microsoft/detectron2>

Albumentations



image augmentation



<https://github.com/albumentations-team/albumentations>

Training

- General
- Privacy
- Reinforcement Learning



TorchText

<https://github.com/pytorch/text>

natural language



TorchAudio

<https://github.com/pytorch/audio>

audio and signal
processing



TorchGeo

<https://github.com/microsoft/torchgeo>

geospatial data



TorchRec

<https://github.com/pytorch/torchrec>

recommendation
systems



PyG

https://github.com/pyg-team/pytorch_geometric

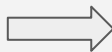
Graph Neural
Networks

Optimizations

- Compilers & Runtimes
- General
- MLOps

<https://landscape.pytorch.org/>

popular datasets
pretrained models



easy to use / benchmarking / transfer learning

Pytorch Practice Notebook

1. Pre-reqs:

- a. Python - see recitation 0.1
- b. Notebooks - see recitation 0.2
- c. Numpy - see recitation 0.3
- d. Some math/ matrix algebra



Recitation 0.4

PyTorch

Focus on solving the problem at hand!

<https://pytorch.org/>

(Tutorials, Community hub, Docs, others!)