DL Dev Course: Week 06
PyTorch

Why PyTorch

- Python version of Torch (orig: LUA)
- Research papers often use Torch -> PyTorch
- Native Python/Numpy feel -Pythonic
- Debugging
- Can look into tensors much easier etc.
- Papers often release with PyTorch code

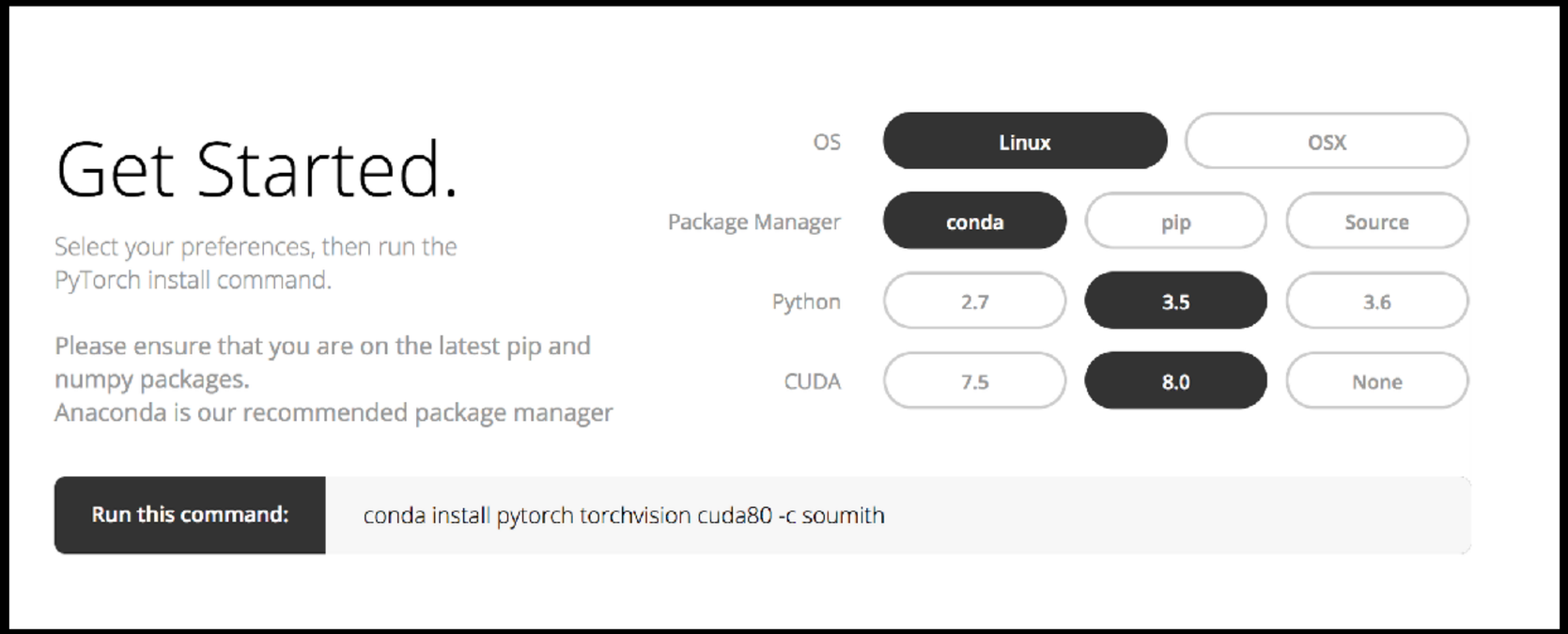
Dynamic Compute Graphs

- Define by run
- More freedom
- Changeable graphs dependent on the inputs
- Debugging is much much easier
- Dynamic vs Static Graph Definition
- Can change on the fly

Why TensorFlow

- Industry Standard for Production
- TensorFlow Serving
- Scalable + Distributed Training
- Optimized for TPUs etc etc
- TensorBoard vs Matplotlib

Installing


Get Started.
Select your preferences, then run the PyTorch install command.
Please ensure that you are on the latest pip and numpy packages.
Anaconda is our recommended package manager
OS
Linux
OSX
Package Manager
conda
pip
Source
Python
2.7
3.5
3.6
CUDA
7.5
8.0
None
Run this command:
conda install pytorch torchvision cuda80 -c soumith

Key concepts

- **`nn.Module`** is a building block
- **`torch.nn`** = ready made modules
- **`torch.nn.Sequential`** = similar to Keras
- **`torch.autograd`** = Auto gradient on variables
- **`Variable`** = Auto gradient on variables
- **`Model.zero_grad`** =zero the gradients before pass

Key concepts 2

- `forward` = run the network forward
- `loss.backward()` = computes the gradients of loss in relation to the learnable parameters
- `.cuda()` = runs on the GPU

Optimizers

- `torch.optim` = Optimizers
- `optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)` = Model.parameters is weights etc.
- `optimizer.step()` = makes 1 step update of trainable parameters

Basics Notebooks

TorchVision