# 7. Classification for MNIST using deep neural networks

In this section, we are going to use deep neural networks to perform the same classification task as in previous sections. We will use PyTorch, a python deep learning framework. Using a framework like PyTorch means you don't have to implement all of the details (like in the earlier problem) and can spend more time thinking through your high level architecture.

**Setup Overview** To setup PyTorch, navigate to their website in your browser, select your preferences and begin downloading. Your selection for **OS** and **Package Manager** will depend on your local setup. For example, if you are on a Mac and use `pip` as your Python package manager, select "OSX" and "Pip". We recommend you select Python version 3 for use with PyTorch. Finally, you are not required to train large models for this course, so you can safely select "None" for CUDA. If you have access to a NVIDIA GPU enabled device with the CUDA library installed, and want to try training your neural models on GPUs, feel free to install PyTorch with CUDA selected but you will have to troubleshoot on your own.

**Test your installation** Once you have successfully installed PyTorch using the instructions on their website, you should test your installation to ensure it is running properly before trying to complete the project. For basic functionality, you can start a python REPL environment with the `python` command in your terminal. Then try importing PyTorch with `import pytorch`.

## Discussion

Hide Discussion

**Topic:** Unit 3 Neural networks (2.5 weeks):Project 3: Digit recognition (Part 2) / 7. Classification for MNIST using deep neural networks

Add a Post

Show all posts ▼ by recent activity ▼

import torch
In my environment (and I suspect this is general), I imported pytorch with `import torch`, not `import pytorch`. 3
Community TA