MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

Course Progress Dates Discussion Resources

Course / Unit 3. Neural networks (2.5 weeks) / Project 3: Digit recognition (Pa

Previous ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

1. Introduction

Bookmark this page

Your friends now want you to try implementing a neural network to classify MNIST digit

Setup:

As with the last project, please use Python's **NumPy** numerical library for handling arra use **matplotlib** for producing figures and plots.

1. *Note on software: For all the projects, we will use python 3.6 augmented with the **Nu** the **matplotlib** plotting toolbox. For THIS project, you will also be using **PyTorch** for i Nets* and **scipy** to handle sparse matrices.

2. Download mnist.tar.gz and untar it in to a working directory. The archive contains the dataset directory, along with the following python files:

   - part2-nn/neural_nets.py in which you'll implement your first neural net from sc
   - part2-mnist/nnet_fc.py where you'll start using PyTorch to classify MNIST digi
   - part2-mnist/nnet_conv.py where you will use convolutional layers to boost per
   - part2-twodigit/mlp.py and part2-twodigit/conv.py which are for a new, m the MNIST dataset

Tip: Throughout the whole online grading system, you can assume the NumPy python l imported as np. In some problems you will also have access to python's random library you've already implemented. Look out for the "Available Functions" Tip before the code last project.

This project will unfold both on MITx and on your local machine. However, we encourag the functions locally. For this project, there will not be a test.py script. You are encou own test cases to make sure your code works as you expected before submitting it to t

Tip: You may work through the Pytorch tutorial in Introduction to ML Packages (Part 2) section).