

EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the [Privacy Policy](#).



MITx: 6.86x

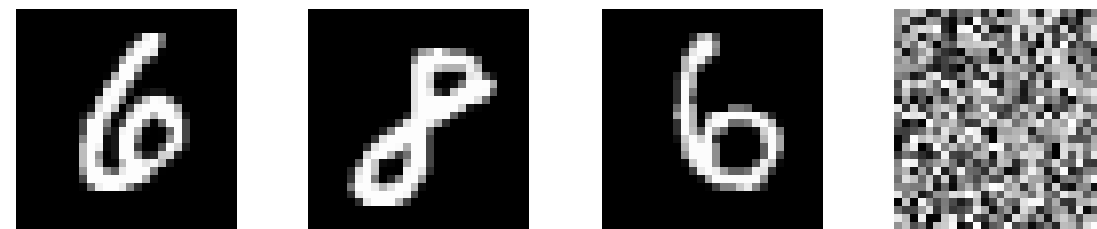
Machine Learning with Python-From Linear Models to Deep Learning

[Help](#)[sandipan\\_dey](#) ▼

[Course](#) > [Unit 3 Neural networks \(2.5 weeks\)](#) > [Project 3: Digit recognition \(Part 2\)](#) > 1. Introduction

## 1. Introduction

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



### Setup:

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

1. *Note on software: For all the projects, we will use python 3.6 augmented with the **NumPy** numerical toolbox, the **matplotlib** plotting toolbox. For THIS project, you will also be using **PyTorch** for implementing the Neural Nets and **scipy** to handle sparse matrices.*
2. Download [mnist.tar.gz](#) and untar it in to a working directory. The archive contains the various data files in the Dataset directory, along with the following python files:
  - `part2-nn/neural_nets.py` in which you'll implement your first neural net from scratch
  - `part2-mnist/nnet_fc.py` where you'll start using PyTorch to classify MNIST digits
  - `part2-mnist/nnet_cnn.py` where you will use convolutional layers to boost performance
  - `part2-twodigit/mlp.py` and `part2-twodigit/cnn.py` which are for a new, more difficult version of the MNIST dataset

**Tip:** Throughout the whole online grading system, you can assume the NumPy python library is already imported as `np`. In some problems you will also have access to python's `random` library, and other functions you've already implemented. Look out for the "Available Functions" Tip before the codebox, as you did in the last project.

This project will unfold both on MITx and on your local machine. However, we encourage you to first implement the functions locally. For this project, there will not be a `test.py` script. You are encouraged to think of your own test cases to make sure your code works as you expected before submitting it to the online grader.

## Discussion

Hide Discussion

**Topic:** Unit 3 Neural networks (2.5 weeks):Project 3: Digit recognition (Part 2) / 1. Introduction

Add a Post

Show all posts ▼by recent activity ▼

There are no posts in this topic yet.

✕

Learn About Verified Certificates