



MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

[Help](#)

smitha_kannur ▾

[Course](#)

[Progress](#)

[Dates](#)

[Discussion](#)

[Resources](#)

[Home](#) / [Course](#) / [Unit 3 Neural networks \(2.5 weeks\)](#) / [Project 3: Digit recognition \(Part 2\)](#)

[< Previous](#)

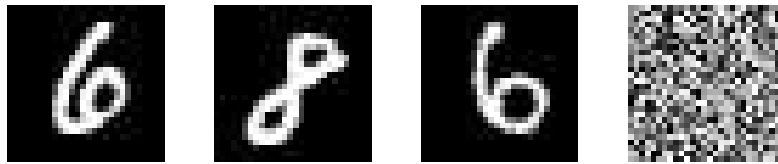


[Next >](#)

1. Introduction

[Bookmark this page](#)

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_conv.py` where you will use convolutional layers to boost performance
 - `part2-twodigit/mlp.py` and `part2-twodigit/conv.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.

Tip: You may work through the Pytorch tutorial in [Introduction to ML Packages \(Part 2\)](#) (posted in the resource section).

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 ▼

- 🗨 [Introduction](#)
Have you noted that the sample images used at the top of this page are spelling 686, which is the course name. I wonder what is the las... 6
- 🗨 [untar files for project 3](#)
Since we already got files for part 2 from project 2, do we need to untar again for project 3? Is it redundant? Please clarify. Thanks. 6
- 🗨 [NOT able to upgrade to certification.](#) 10

< Previous

Next >



edX

[About](#)
[Affiliates](#)
[edX for Business](#)
[Open edX](#)
[Careers](#)
[News](#)

Legal

[Terms of Service & Honor Code](#)
[Privacy Policy](#)
[Accessibility Policy](#)
[Trademark Policy](#)
[Sitemap](#)

Connect

[Blog](#)
[Contact Us](#)
[Help Center](#)
[Media Kit](#)
[Donate](#)



© 2020 edX Inc. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)