

---

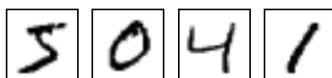
# CS161: FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE

Fall 2016

Assignment 9 (optional) - Due 11:55pm Wednesday, December 7

---

Please submit your solutions on CCLE. The submitted file should be plain text or a formatted PDF file (no scans or pictures). Text files should have lines no longer than 100 characters and should be well-aligned when viewed with a monospace font.



## Abstract

In this assignment, we will use a software tool to automatically learn from data. In particular, we will be able to create a program to recognize handwritten digits from images. This could be very useful for post offices to automatically recognize zip codes. To achieve this, we will use TensorFlow which is an open source software library for numerical computation.

## 1 Installation

You will first need to install TensorFlow on your machine. The required steps can be found here: [https://www.tensorflow.org/versions/r0.11/get\\_started/os\\_setup.html](https://www.tensorflow.org/versions/r0.11/get_started/os_setup.html). The installation assumes a Linux or Mac machine. If you do not have one, please use the virtual box <https://www.virtualbox.org/wiki/Downloads>. We will provide a ubuntu image with password *cs161* for you to download from <https://drive.google.com/file/d/0BxDA1yJ0gaa-V2tmWDdyZDljMUk/view?invite=CJuXiYgM&ts=583e61f5>.

## 2 Tutorial

The assignment is based on the MNIST For ML Beginners TensorFlow tutorial: <https://www.tensorflow.org/versions/r0.11/tutorials/mnist/beginners/index.html>. After installing TensorFlow on your machine, run `mnist_softmax.py`, using the command: `python mnist_softmax.py`, on your command-line interface. Go through the tutorial carefully to understand what every step is doing.

## 3 Assignment

If you installed and ran `mnist_softmax.py` successfully, you should get about 92% accuracy on the test set. This means that we expect our system to classify digits correctly in about 92% of the time. Now you will need to do small changes to the script and rerun it to answer the following questions:

### 3.1 Question 1

Modify the script to print the accuracy of the learned model on the training set. What is the accuracy you got? Why is it different from the test set accuracy?

### 3.2 Question 2

In Line 62 in `mnist_softmax.py`, we loop 1000 times, taking small steps towards our final optimized model. Try changing 1000 to 10, and reprint the accuracy on the test set. What is the accuracy you got? Now try increasing the number of steps to 10000, and report the accuracy in this case. Comment briefly on the results.

### 3.3 Question 3

Lines 40 and 41 in `mnist_softmax.py` are initializing the model:  $W$  and  $b$  with zeros. These are the values that the optimization algorithm (e.g. gradient descent) starts with and then takes small steps towards new values for  $W$  and  $b$  that can perform better in recognizing digits. Try initializing  $W$  and  $b$  with ones rather than zeros (i.e. replace `tf.zeros` by `tf.ones`). What is the test set accuracy in this case? Why do you think it is different?

### 3.4 Optional

Our model was too simple for the problem. We can improve the accuracy by using a larger model. Go through the more advanced tutorial <https://www.tensorflow.org/versions/r0.11/tutorials/mnist/pros/index.html> and try to make changes to `mnist_softmax.py` to improve the accuracy on the test set. Report the best accuracy you got and the changes you made to `mnist_softmax.py`.

## Acknowledgement

Thanks to Khaled S. Refaat for preparing this project. Khaled obtained his PhD in computer science from UCLA in 2015 and is currently at Google X.