Project of Big Data

-by Hao Sun

# Introduction

For this project, the ultimate goal is using the knowledge we learned from the lecture and self-learning to develop an application that can allow user to upload an image of handwritten number to the website. The application will recognize the number and save the data of the file name and the number of it.

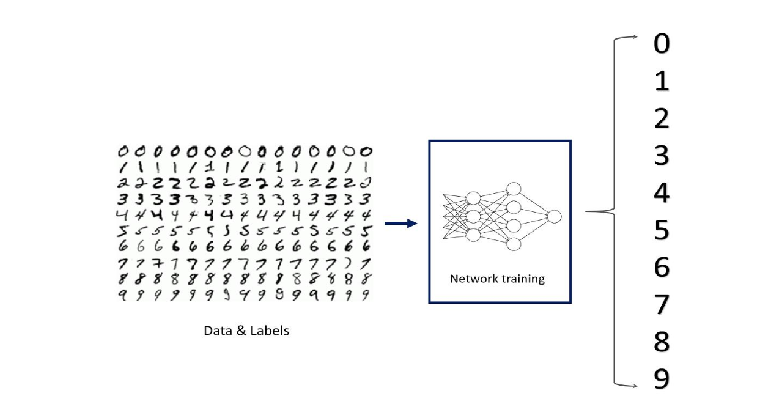
# Brief Description of Final Project

For this project, we are suppose to program an application that will satisfied our purpose using four critical tools: Mnist, Cassandra, flask and Docker. We will deploy our digital handwritten program based on Mnist and flask. At the same time, we will make this program using docker and the data of this program will be stored using Cassandra. I will introduce each of them in detail in follow paragraphs and describe how they function in the application.

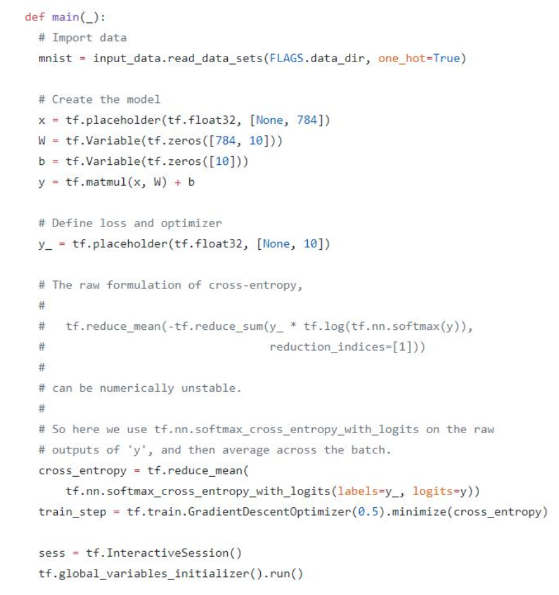
# Mnist

Modified [National Institute of Standards and Technology](https://en.wikipedia.org/wiki/National_Institute_of_Standards_and_Technology) database is a large [database](https://en.wikipedia.org/wiki/Database) of handwritten digits that is commonly used for [training](https://en.wikipedia.org/wiki/Training_set) various [image processing](https://en.wikipedia.org/wiki/Image_processing) systems. The database is also widely used for training and testing in the field of [machine learning](https://en.wikipedia.org/wiki/Machine_learning).

Each picture in the Mnist dataset has 28\*28 pixels and can also be seen as a list with 28\*28=784 numbers. Meanwhile, each number represent the brightness of a specific pixel. By connecting every numbers with the nodes in next layer, seeing every connections as bivariate linear equations, using gradient descent to calculate optimal constant for each equations and connecting nodes with the outcome layers, we can create a neural network. After maybe hundreds or even thousands times of training, we can get a model for our application and move on to the next step of using flask to push our service online.

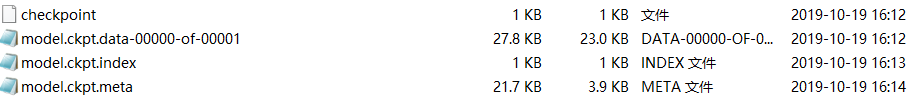


In this program, we could consider using the following code:



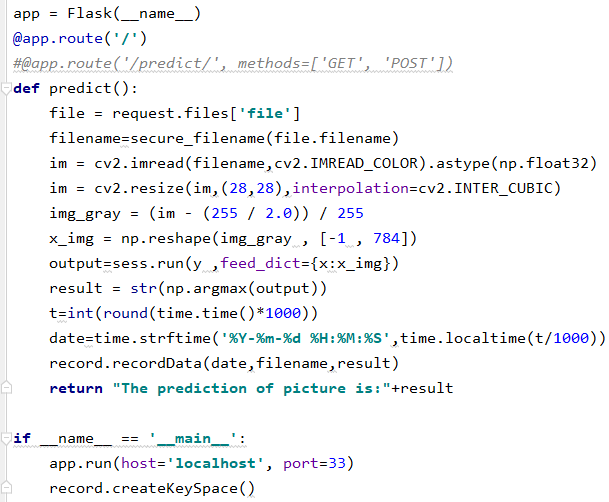


After we run the code above, we can save our model in following format..



## Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, from validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, from validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

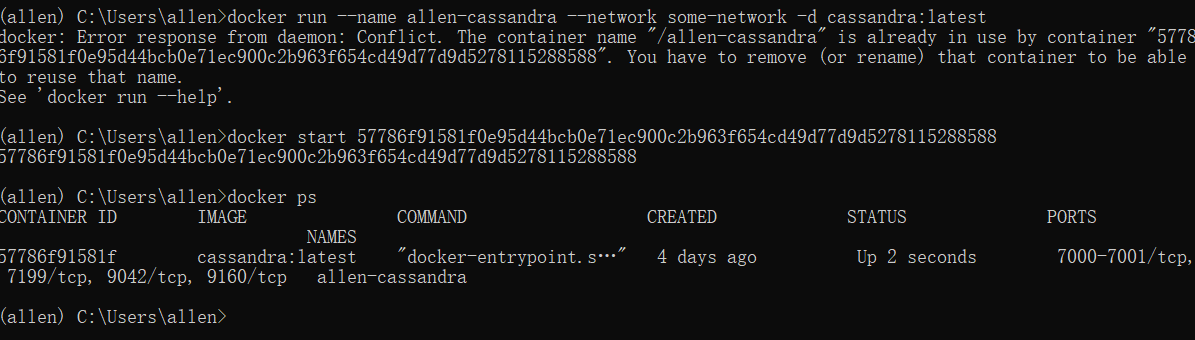


After running the code above, we can simply access our server on “localhost:33”. We can upload the images to the localhost. The router will save the images and requests, the images will be processed by our model and the result will return to our browser.

## Docker

The most impressive application I have learned about during this project is Docker. Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating-system kernel and are thus more lightweight than virtual machines.

In our project, I use docker to create an image for Cassandra and the application to run on.



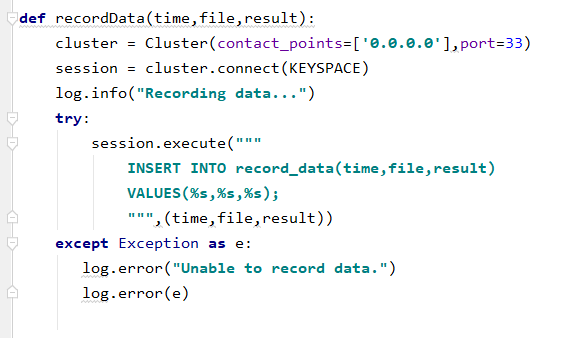
## Cassandra

Cassandra is one of the most important tool in today’s study of big data. Apache Cassandra is a free and open-source, distributed, wide column store, NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple datacenters,[2] with asynchronous replication allowing low latency operations for all clients.

In our project, we use Cassandra to record the name, the time and the result of user’s uploaded images. We define a the table:



And then record the data:

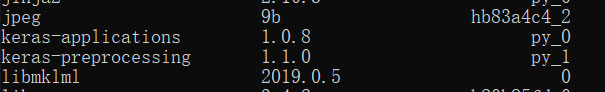
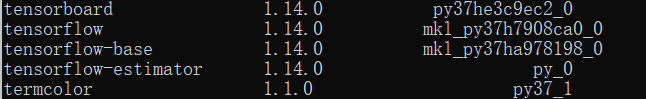


## Running process

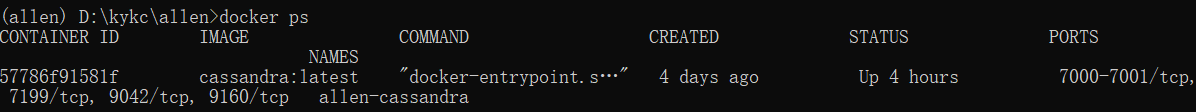
In order to run this application, first we need to activate the conda which have installed all the packages that we need:





(only some of the packages that have installed showed above)

Second we need to run our cassandra image in docker in order to record our data: 

Third we activate our service and make sure it was online

Finally we only need to upload our images and make sure our cassandra network is connected with the cluster.

## Final summary

This project give me a brief overview about big data and a chance to try all these tools. It also make me interested in the future development of big data since the analysis will only become more and more important since the size of data already overwhelm the capacity of human. At the same time the computing power of todays CPU and GPU are able to handle much more data than before. Collecting and analyzing huge amount of data become more and more feasible and necessary. Therefore, there will be more development and advance on today’s study of big data and I believe it can be the combination with Artificial Intelligence which can help to improve the analyzing speed of the data and minimize the size of storing data. For example, machine learning can be use on learning the storing habit of the users to minimize the storing size.