

# CNN training

---

## Introduction

---

1. Folder nn has the necessary .py files: (convolutional layer, pooling layer, flatten layer, fully\_connected layer, activation functions, loss function, SGD). Each layer contains Feed forward process and Backpropagation.
2. cnn\_useclass.py is the file which uses Class, I use regular gradient decent method. This code is quite similar like Duc did in class.
3. cnn\_usedictionary is the file which uses dictionary to save weights and gradients. Besides, I choose SGD for gradient decent method, the hyperparameter refer to CS231n, Lecture 7.
4. The general gradient descent runs quite slow, for large number of datasets, we will choose mini-batch SGD to speed up.
5. Use 4-D array to speed up.
6. Both cnn\_useclass.py and cnn\_usedictionary work, and I write it in a general way, which means we can add mulitple layers in the future.

## About layers

---

1. The Structure of my layer is:  
conv1--->pooling--->flatten--->fullyconnect, which is exactly the same structure Dr.Wei showed us in class.
2. The input image is  $28 \times 28$ , I choose  $5 \times 5$  filter to do the convolutional part. After that I will get a  $24 \times 24$  matrix, and then do the max pooling convolutional, we will get a  $12 \times 12$  matrix.

## Prerequisites

---

1. Python 2.7
2. Numpy 15.1.4
3. Pandas
4. Note: If you get an error about "flip", you need to update your Numpy to the latest version.

## Result

---

1. I choose filters = 16 and the best accuracy I get is 0.942. This code runs for 1 hour.

2. Actually we need to run our code multiple times and calculate the average of the accuracy, I will revise it after finals.
3. To avoid overfitting, we need to add regularization or dropout, I will revise it after finals.