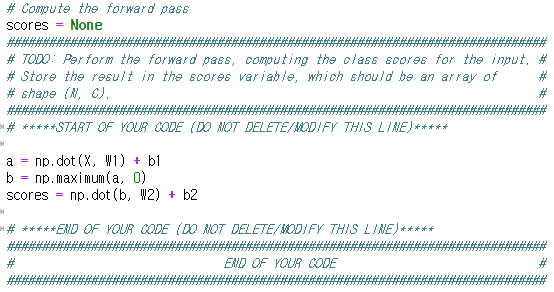
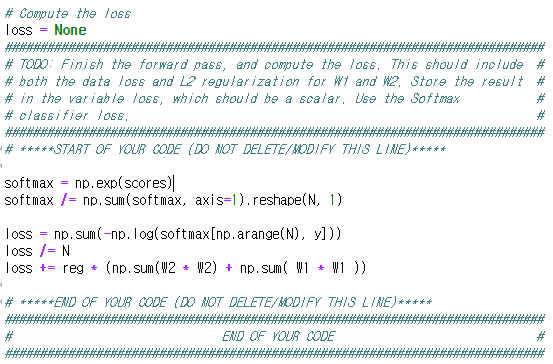
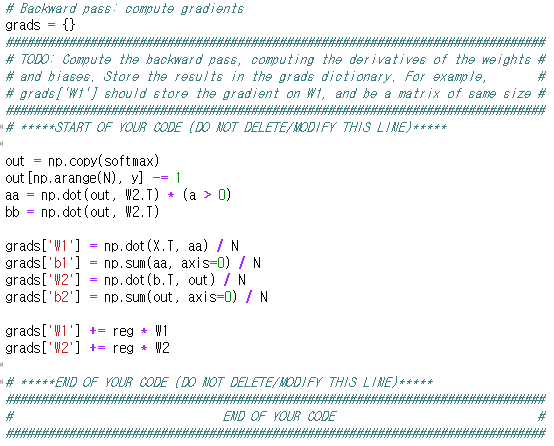
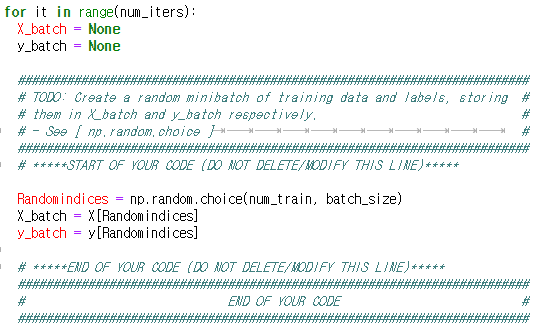
**COSE474 Deep Learning  
Project #1: MLP Implementation**

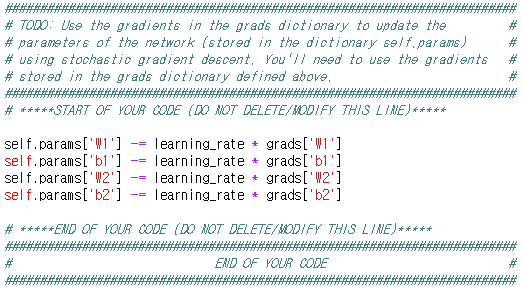
2017320122 김정규

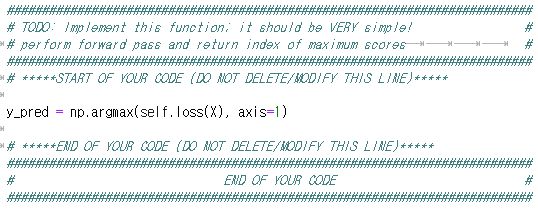
**Code1 – Forward Pass**  
The code above is the part that implements the forward pass. The score of each class is calculated for input data. The result is stored in the score variable.

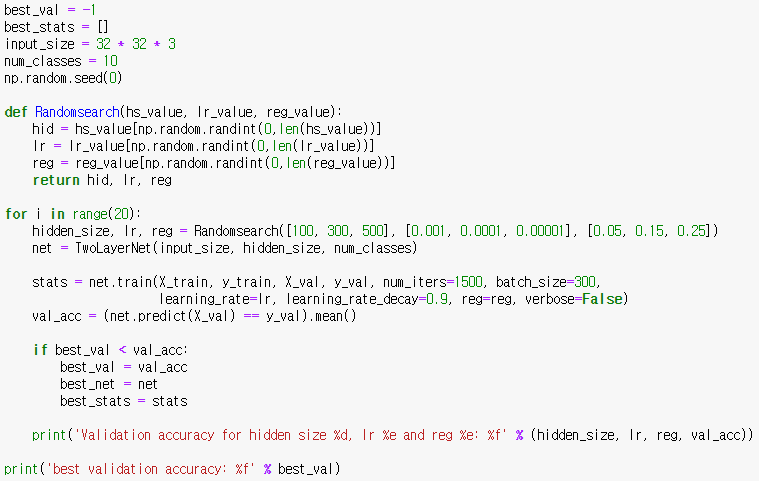
**Code2 – Calculating Loss**  
This is the part where the forward pass is completed and the loss is finally calculated. That means the data loss is calculated, and L2 regularization is calculated for W1 and W2. Softmax is used for loss function.

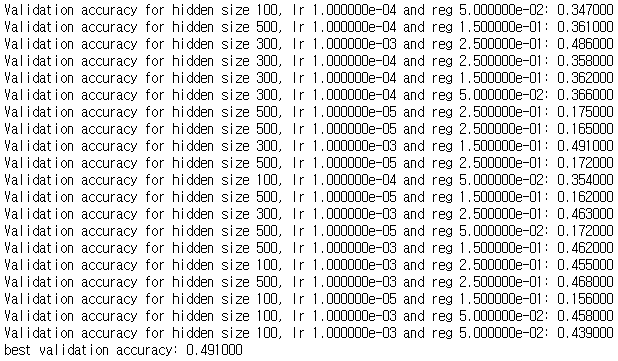
**Code3 – Back Propagation**  
This is the part that implements backpropagation. The learning rate is subtracted from each weight value.

**Code4 – Random Indices**  
The batch\_size number of valid integers are extracted randomly from 0 to num\_train. By using these integers as indices, put these data in X\_batch and y\_batch respectively.

**Code5 - Update**  
  
Multiply the grades calculated from the loss function by the learning\_rate and subtract them from weight and bias, which means to update the value.

**Code 6 – Prediction**  
This is a code for testing using learned parameters. we can get the value which was predicted to be the largest among those other elements using argmax function

**Code 7 - Tuning**  
Based on the functions completed, set the environments and parameters to be applied . Hidden\_size, lr, and reg were set to follow the results in the skeleton file, but they also follow the random order. Additionally, the accuracies obtained in each case are compared with each other. Then the largest value is stored and printed.

**Results**  
This the data obtained by ruunning two\_layer\_net.ipynb. Accuracies obtained through various parameters can be identified. Among them, the best value is 0.491, and the parameters are [hidden size 300], [lr 0.001] and [reg 0.15]