

Overview:

In this part, you are required to understand the whole process of compiling different layers (Convolutional Layer, Fully-Connected Layer, Pooling Layer, Activation Layer, Loss function) of a simple Convolutional Neural Network (CNN) for the visual classification task. And you need to compile your own evaluation code to evaluate the trained CNN to obtain the training and testing results.

The layer definitions have been given in the demo code and please follow the steps to understand the principles of different layers.

The dataset you will utilize for the classification task is a subset from the MNIST dataset. The demo code will randomly select four different categories and **500** training and **100** testing samples for each category. Therefore, the total size of the training and testing samples is **2000** and **400** respectively. The subset training and testing samples will be **shuffled** before providing to you so that you do not need to shuffle the data when doing the training process.

For the evaluation code, the function name, function inputs, and the use of the functions have been given in the demo code. You are required to write the remaining part to make the function work properly and obtain the accuracy and loss for both training and testing samples.

You are required to train the CNN with a fixed epoch number and initialization of parameters. The total epoch number should be **10** and the learning rate should be **0.001**. The batch size for training and testing process is set to **100** and **1** respectively. The number of feature maps in the convolutional layer should be **6** and the size of the filters is set to **5*5**. The size of pooling layer is **2*2** and the **ReLU** activation function is set to default. The number of neurons of the first fully-connected layer is set to **32**. A **cross-entropy loss** with **softmax** activation function is utilized to train the CNN. All those mentioned parameters are set to default values in the demo code.

You are highly suggested to change those abovementioned parameters to have a better understand of the principle of CNN for the visual classification task. However, please reset parameters to the default values to obtain results for the submission. **All the results of submission should be based on the default values.** And you will surely lose points if your results are not based on the default parameter values.

You are suggested to use the build-in Jupyter Notebook to implement your algorithm. You need to take responsibility of any errors caused by the use of any other programming environment.

Note: The loss value should be divided by the number of training/testing samples to normalize its value so that the number of samples do not affect the loss value.

What to submit:

Every student will get their specific training and testing subset samples from the demo code. Please train and test the CNN with your own specific training and testing samples.

Four values, final **training accuracy**, **training loss**, **testing accuracy** and **testing loss after 10 epochs**, should be submitted to the following "**Quiz**" sections.

For the report, you will need to include the same four values submitted to the "Quiz" and also four plots showing **training accuracy vs epochs**, **training loss vs epochs**, **testing accuracy vs epochs**, and **testing loss vs epochs**. Additionally, include your code for the *evaluate()* function along with your report.

Notice:

1. You should compile the evaluation code by yourself.
2. You will not get any points for the project by simply programming here. Please remember you need to submit the results in the "**Quiz**" sections.
3. All the submission results obtained in this project part 3 should be based on the **default settings**.