Ira A. Fulton Schools of Engineering
Arizona State University

CSE 575: Statistical Machine Learning

Classification Using Neural Networks and Deep Learning Project

Purpose

In this project, 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 total points for this project is 10 points.

Objectives

Learners will be able to:

- Understand the process of compiling different layers of CNN.
- Implement and evaluate a CNN for image classification tasks.
- Modify hyper-parameters and observe the effects on training and testing errors.

Technology Requirements

Algorithm:

- Convolutional Neural Network

Resources:

- MNIST dataset

Language:

- Python

Project Description

In this project, we will revisit the Handwritten Digits Recognition task in Project 1, using a convolutional neural network. The basic dataset is the same MNIST dataset from Project 1, but you may choose to use only a subset for training and testing, if speed performance with the entire dataset becomes a bottleneck. For example, you may use only 6000 samples for training (each digit with 600 samples) and 1000 samples for testing (each digit with 100 samples).

The basic requirement of this project is to experiment with a convolutional neural network with the following parameter settings:

1. The input size is the size of the image (28x28).
2. The first hidden layer is a convolutional layer, with 6 feature maps. The convolution kernels are 3x3 in size. Use stride 1 for convolution.
3. The convolutional layer is followed by a max pooling layer. The pooling is 2x2 with stride 1.
4. After max pooling, the layer is connected to the next convolutional layer, with 16 feature maps. The convolution kernels are 3x3 in size. Use stride 1 for convolution.
5. The second convolutional layer is followed by a max pooling layer. The pooling is 2x2 with stride 1.
6. After max pooling, the layer is fully connected to the next hidden layer with 120 nodes and relu as the activation function.
7. The fully connected layer is followed by another fully connected layer with 84 nodes and relu as the activation function, then connected to a softmax layer with 10 output nodes (corresponding to the 10 classes).

We will train such a network with the training set and then test it on the testing set.

You are required to plot the training error and the testing error as a function of the learning epochs. You are also required to change some of the hyper-parameters (the kernel size, the number of feature maps, etc), and then repeat the experiment and plot training and testing errors under the new setting.

These are the minimum requirements. Additional requirements may be added (like experimenting with different kernel sizes, number of feature maps, ways of doing pooling, or even introducing drop-out in training, etc.).

Directions

Accessing Ed Lessons

You will complete and submit your work through Ed Lessons. Follow the directions to correctly access the provided workspace:

1. Go to the Canvas Assignment, "**Submission: Classification Using Neural Networks and Deep Learning Project**".
2. Click the “**Load Submission…in new window**” button.
3. Once in Ed Lesson, select the assignment titled "**Classification Using Neural Networks and Deep Learning Project**".
4. Select a code challenge to work on:
    a. To start the baseline code, click on the “**Analysis of Baseline Code**”
    b. To start the lab, click on the “**Lab: Classification Using Neural Networks and Deep Learning Project**”
    c. To start the result submission, click on the “**Result Submission: Classification Using Neural Networks and Deep Learning Project**”
5. When ready, start working in the notebook for the respective code challenge:
    a. For the baseline code, the notebook is titled “**baseline.ipynb**”
    b. For the lab, the notebook is titled “**project3.ipynb**”
    c. For the result submission, the notebook is titled “**project3_submission.ipynb**”

Baseline Code

The baseline code provides a basic understanding about the different layers of Convolutional Neural Network (CNN) using the keras library.

Required Tasks

1. Run the baseline code and report the accuracy.
2. Change the kernel size to **5*5**, redo the experiment, plot the learning errors along with the epoch, and report the testing error and accuracy on the test set.

3. Change the number of the feature maps in the first and second convolutional layers, redo the experiment, plot the learning errors along with the epoch, and report the testing error and accuracy on the test set.
4. Submit a brief report summarizing the above results in your submission space.

Note: You can change the kernel size up to 5*5 and number of feature maps up to 32 for both the layers.

Report Submission

Draft a report that explains how changing kernel size impacted the results. The report must contain:

- Your full name and student ID number on the first page in the upper left corner
- The accuracy value from the default baseline code
- The accuracy value after modifying the kernel size and feature maps in the first and second convolutional layers
- Plot the learning errors along with the epoch
- A brief overview on how the changes impacted the results

The report must also follow the required format:

- A maximum font size of 12pt
- A maximum length of two (2) pages (8x11 or A4 paper).
- Saved as a PDF (.pdf) file type

Lab

The layer definitions have been given in the 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.

Required Tasks:

1. **Evaluation Code**: The function name, function inputs, and the use of the functions have been given in the 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.

2. 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 the 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 the 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 code.

3. You are highly suggested to change those above-mentioned parameters to have a better understanding 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.

4. Plot the following graphs:
    a. Training and Testing Accuracy vs epochs
    b. Training and Testing Loss vs epochs

You are suggested to use the built-in Jupyter Notebook to implement your algorithm. You need to take responsibility for 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.

Lab Submission

You must complete the tasks mentioned and plot the required graphs in the designated code challenge workspace. Every student will get their specific training and testing subset samples from the code. Please train and test the CNN with your own specific training and testing samples.

Additional requirements are:

- You should compile the evaluation code by yourself.

- You must submit the results in the “Result Submission” code challenge in order to receive credit for your work. You will not get any points for the project by simply programming in the “Lab” workspace.

- All the submission results obtained should be based on the **default settings**

Result Submission

You must complete the “Lab” portion of the project in order to complete this part. From the “Lab”, you will get your specific training and testing subset samples. You need to submit the four values in the code space provided for each value:

- Final training accuracy
- Training loss
- Testing accuracy
- Testing loss after 10 epochs

Submission Directions for Project Deliverables

What to Submit:

You must submit each deliverable for the project through Ed Lessons in their designated code challenge workspaces:

1. **Baseline Code Submission**: A brief report summarizing the results of baseline code
2. **Lab Submission**: Completed evaluate function with final 4 training and testing values and Training and testing accuracy vs epochs; Training and testing loss vs epochs plots.
3. **Result Submission**: You need to submit the four values i.e **final training accuracy, training loss, testing accuracy, testing loss after 10 epochs** in the code space provided for each value.

Notebook Submission

To receive credit for the course, You must complete and submit your work in each code challenge’s notebook provided in the project’s Ed Lesson:

1. Follow the directions provided for each code challenge.
2. When you are ready to submit your completed work, click on either “**Save and Mark Complete**” (Analysis of Baseline Code and Lab) or “**Test**” (Result Submission) at the bottom right of the screen.
3. You will know you have successfully completed the assignment when feedback appears for each test case with a score.

4. If needed: to resubmit the assignment in Ed Lesson
    a. Edit your work in the notebook
    b. Run the code cells again
    c. Click "**Save and Mark Complete**" or "**Test**" at the bottom of the screen

Your submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

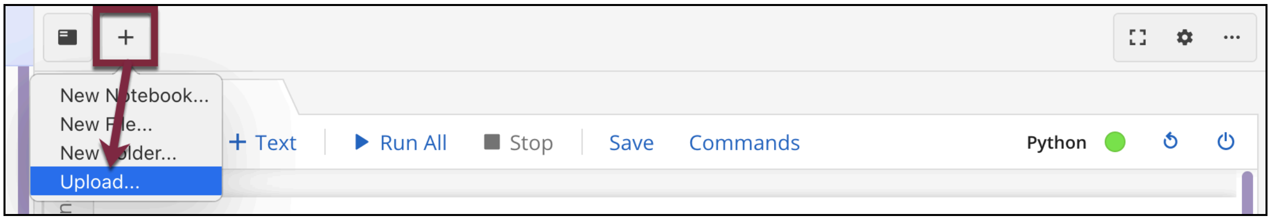
Baseline Code Report Submission

Your report will be manually graded by the course team. You must submit your report in the designated code challenge workspace titled "**Analysis of Baseline Code**".

1. Click the **Plus (+)** icon in the upper left corner of the notebook workspace (second icon from the left)

2. Select "**Upload**"

3. Locate and select your report submission from your device (PDF file only)



4. Your file will appear in a left-pane menu that appears next to the notebook workspace

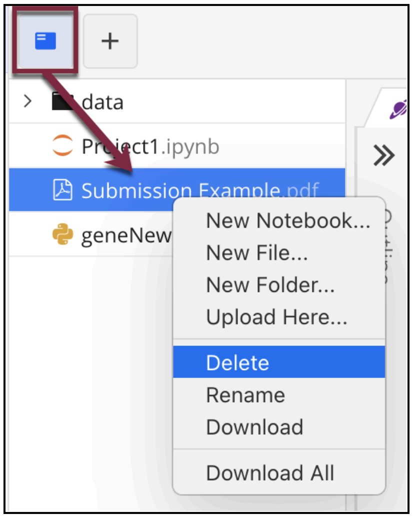
5. Click "**Submit**" in the upper right corner to submit your completed project.

6. If needed: to resubmit the report in Ed Lesson

    a. Click the "**Toggle Files**" icon in the upper left corner of the notebook (first icon from the left)

    b. Locate and right-click on your previous report submission file

    c. Click "**Delete**" to remove it from your attempt and then repeat the upload directions from Step 2



Your latest report submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

Lab Submission

This lab will be manually graded by the course team. You must complete the tasks outlined in the “Lab” code challenge. Then, you will need to submit the results from the lab into the “Result Submission” code challenge.

Reminder: You will not get points for the project by simply programming in the lab.You must submit the results in the “Result Submission” code challenge in order to receive credit for your work.

Result Submission

The Result Submission will be auto-graded. Obtained from the “Lab” submission, you must submit the four values in the code space provided for each value.

- Training accuracy
- Training loss
- Testing accuracy
- Testing loss after 10 epochs

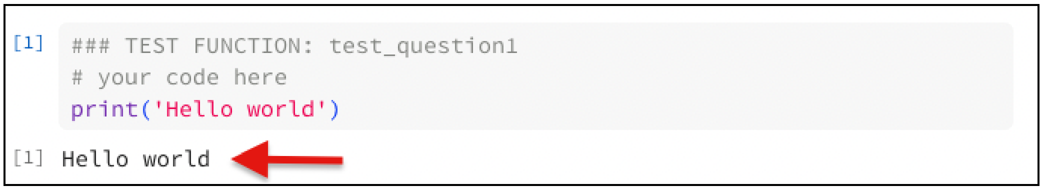
When ready to submit:

1. In order for your answers to be correctly registered in the system, you must place the code for your answers in the cell indicated for each question.

    a. You should submit the assignment with the output of the code in the cell's display area. The display area should contain only your answer to the question with no extraneous information, or else the answer may not be picked up correctly.

    b. Each cell that is going to be graded has a set of comment lines (ex: ### TEST FUNCTION: test_question1) at the beginning of the cell. **This line is extremely important and must not be modified or removed.**

2. After completing the notebook, run each code cell individually or click “**Run All**” at the top to print the outputs.



```
[1]  ### TEST FUNCTION: test_question1
     # your code here
     print('Hello world')

[1]  Hello world
```

3. Click on “**Test**” at the bottom right of the screen.

4. You will know you have successfully completed the assignment when feedback appears for each test case with a score.

5. If needed: to resubmit the assignment in Ed Lesson

    a. Edit your work in the notebook
    b. Run the code cells again
    c. Click “**Test**” at the bottom of the screen

Your submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

Evaluation

The assignment will be evaluate in Ed Lessons and the grades will be automatically applied to the gradebook.