Ira A. Fulton Schools of Engineering
Arizona State University

CSE 575: Statistical Machine Learning

Project 3: Classification Using Neural Networks and Deep Learning

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.

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, Google CoLab

Workspace:

- Google CoLab (see file “CSE 575_Project 3_Intro to Colab.docx” for more details)

Software:

- Google CoLab

Language(s):

- Python

Project Description

In this part, we will revisit the Handwritten Digits Recognition task in Part 1, using a convolutional neural network. The basic dataset is the same MNIST dataset from Part I, 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 part 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

Getting Started:

In addition to the Project 3 Overview Document, review the additional files included (download from the Course Project Overview page in the course):

- CSE 575_Project 3_Intro to Colab
- CSE 575_Project 3_Baseline

For more details about Colab, please go to
https://colab.research.google.com/notebooks/welcome.ipynb

Required Tasks:

1. Read "CSE 575_Project 3_Intro to Colab.docx" to get familiar with the platform.
2. Run the baseline code (see file "CSE 575_Project 3_Baseline") and report the accuracy.
3. 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.
4. 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.
5. Submit a brief report summarizing the above results, along with your code.

Lab: Project 3: Classification Using Neural Networks and Deep Learning

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 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 demo code.

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.

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.

Submission Directions for Project Deliverables

What to Submit:

1. **Result Submission**: Code - please add comments properly to explain what you do.
2. **Report Submission**: A report, including (a) the final learning/testing errors and accuracy values, and (b) the plots for the learning/testing errors and the final classification accuracy.

Result Submission

Quiz: Project 3: Classification Using Neural Networks and Deep Learning Result Submission

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 quiz titled "**Project 3: Classification Using Neural Networks and Deep Learning Result Submission**" located under "Week 7: Graded Coursework".

Report Submission

Graded Assignment: Project 3: Classification Using Neural Networks and Deep Learning Report Submission

For the report, you will need to include the same four values submitted to "Quiz: Project 3: Classification Using Neural Networks and Deep Learning Result Submission" 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 in the lab. Please remember you need to submit the results in the result submission quiz.
3. All the submission results obtained in Project 3 should be based on the **default settings**.

Please submit your code results and report regarding Project 3 to the item titled "**Graded Assignment: Project 3: Classification Using Neural Networks and Deep Learning Report Submission**" located under "Week 7: Graded Coursework".

- Acceptable file types: .pdf or .doc/docx.
- Length of the report: less than 1 page (not including graphs).
- Content of the report: (The following must be included)
    - The plots showing **training accuracy** vs **epochs**, **training loss** vs **epochs**, **testing accuracy** vs **epochs**, and **testing loss** vs **epochs**.

    - Your four values (training accuracy, training loss, testing accuracy, and testing loss) that were submitted in the result submission quiz.

- Code file: .py or .ipynb

Evaluation

Result Submission

Results (accuracy numbers and learning curves)

- Run the baseline code as provided and report the accuracy
- Change the kernel size to 5*5, redo the experiment
- Plot the learning errors along with the epoch
- Report the testing error and accuracy on the test set
- 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
- Report the testing error and accuracy on the test set

Report Submission

- Report summarizes results
- The code