Introduction to Artificial Intelligence (CS470): Assignment 2

Deadline: October 4th, 2022

Setup

Google Colaboratory

Please download a starter code containing Colab notebooks code address.

1. Unzip the starter code `.zip` file. You should see a `CS470_IAI_2022Fall` folder. Create a folder in your personal Google Drive and upload `CS470_IAI_2022Fall` folder to the Drive folder.

2. Each Colab notebook (e.g., files ending in `.ipynb`) corresponds to an assignment problem. In Google Drive, double click on the notebook and select the option to open with Colab.

3. Once you have completed the assignment problem, you can save your edited files back to your Drive and move on to the next problem. Please ensure you are periodically saving your notebook File → Save so that you don't lose your progress.

1. Convolution and Pooling from Scratch [20pts]

In this problem, you will implement convolution and max pooling functions using NumPy. You will then analyze the result on the MNIST dataset for a handwriting digit classification task. You will also compare the result with that using Pytorch. You can now use the Numpy library for network design and construction.

You have to fill your code in the blank section following the "`PLACE YOUR CODE HERE`" comments in the `CNN_assignment_1.ipynb` file. Note that you have to write down all the necessary equations and their derivation processes in your report.

a. Convolution and MaxPooling using NumPy [10pts]

Implement two simple CNN models as follows:

- CNN1: `Input - 2D Convolution - Output`,
- CNN2: `Input - 2D Convolution - 2D MaxPooling - Output`,

where you use three 2D convolution filters using a kernel size of 5 and a 2D max pooling function using a kernel size of 2. In your report, attach the visualization results with your analysis. You must state the difference.

b. Convolution and MaxPooling using NumPy [10pts]

Implement the two CNN models using PyTorch. In your report, attach the visualization results. You must provide whether your implementation using NumPy is the same as that using PyTorch by calculating any errors.

2. Convolutional Neural Networks using PyTorch [40pts]

In this problem, you will implement a convolutional neural network (CNN) on the CIFAR-10 dataset for an image classification task. You can now use the Pytorch library for network design and construction. You have to fill your code in the blank section following the "`PLACE YOUR CODE HERE`" comments in the `CNN_assignment_2.ipynb` file. Note that you have to write down all the necessary equations and their derivation processes in your report.

a. Convolution and MaxPooling layers [20pts]

In this part, you implement a CNN model under the `CNN()` class. The model has a sequential structure:

$$\begin{aligned} Input \rightarrow & Convolution \rightarrow Convolution \rightarrow MaxPool \rightarrow \\ & Convolution \rightarrow Convolution \rightarrow MaxPool \rightarrow \\ & Linear \rightarrow ReLU \rightarrow Linear \rightarrow SoftMax \rightarrow Output, \end{aligned}$$

where we use 32 filters with `kernel_size=3` in each convolution layer and 2D max-pooling layers with kernel_size=2. We also set stride and padding sizes in convolution as 1 and 0, respectively. All other arguments use default values. You will also implement forward and backward passes to optimize CNN

by using stochastic gradient descent (SGD) with the momentum method. Note that your classification accuracy should be over 75% on the test images.

In your report, you first analyze the number of parameters used in each layer and the total number of parameters over the entire model. You have to analytically and empirically validate the numbers. After then training the model with the training dataset, you must report accuracy on the $10,000$ test images and provide a graph of training and testing accuracies over 50 epochs.

b. Convolution and Adaptive AvgPooling layers [20pts]

Implement another CNN model by filling the code under the `CNN()` class. The model has a similar sequential structure replacing MaxPool layers to Adaptive AvgPool layers:

$$\begin{aligned} Input \rightarrow & Convolution \rightarrow Convolution \rightarrow AdaptiveAvgPool \rightarrow \\ & Convolution \rightarrow Convolution \rightarrow AdaptiveAvgPool \rightarrow \\ & Linear \rightarrow ReLU \rightarrow Linear \rightarrow SoftMax \rightarrow Output. \end{aligned}$$

All the other problem requirements are the same as the previous problem. Except for the optimizer, please optimize the CNN model by using the Adam algorithm. Note that your classification accuracy should be over 10% on the test images.

In your report, you must report accuracy on the $10,000$ test images, training-and-testing accuracies over 50 epochs. Then, you have to compare the results of Adaptive AvgPooling (explain why one pooling method improves the performance of the model than the other?)

3. Performance Evaluation and Improvement [40pts]

In this problem, you will implement variants of CNN models and then analyze the performance difference by computing validation accuracies given the image dataset. You have to modify the code in the `CNN_assignment_2.ipynb` file.

a. Comparison of CNN and MLP

Compute the validation accuracies of two models:

- The last model is a CNN with MaxPooling layers with the following structure:

$$Input \rightarrow Convolution \rightarrow MaxPool \rightarrow Convolution \rightarrow$$
$$MaxPool \rightarrow Convolution \rightarrow Linear \rightarrow$$
$$ReLU \rightarrow Linear \rightarrow ReLU \rightarrow Linear \rightarrow SoftMax \rightarrow Output.$$

- The later model is an MLP with ReLU layers from the Problem 1 in assignment 1 (without hyperparameter tuning).

You can decide the number of filters with kernel size for the CNN model, however your classification accuracy should be over 10% on the 10 000 test images. In the report, you must attach a plot of validation accuracy curves from the two models, where the x axis and y axis are accuracy and the number of training epochs (which is 50), respectively. Then, you have to explain the results (e.g., why does one model perform better than the other?)

Submission Guide

a. Submission Requirement

Change your file name to `cs470_yourname_studentID.ipynb` in Colab. Download and save in your machine. Write a report explaining how you implemented, comparing the models and discussing the test performance in PDF file. Generate a zip file of your code and report, then save your zip file as `cs470_yourname_studentID.zip`. Please submit the `.zip` file via KLMS.

Please make sure that the submitted notebooks have been saved and the cell outputs are visible.

b. Academic Integrity Policy

This is homework for each student to do individually. Discussions with other students are encouraged, but you should write your own code and answers. Collaboration on code development is prohibited. There will be given no points in the following cases:

- Plagiarism detection
- Peer cheating
- The incompleteness of the code
- The code does not work