Introduction to Artificial Intelligence (CS470): Assignment 2

Deadline:

Wednesday 12th April, 2023

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_2023_Spring` folder. Create a folder in your personal Google Drive and upload `CS470_IAI_2023_Spring` 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 Shallow Convolution Neural Networks [20pts]

In this problem, you will implement convolution and average 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 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_problem_1.ipynb` file.

1.1 Convolution and Average Pooling using NumPy [10pts]

Implement two simple forward networks as follows:

Input	Input of shape (28,28,3)
Layer 1	5×5 Convolutional Layer with 3 filters
Output	

Table 1: CNN1

Input	Input of shape (28,28,3)
Layer 1	5×5 Convolutional Layer with 3 filters
Layer 2	2×2 Average Pooling Layer with a stride of 2
Output	

Table 2: CNN2

In your report,

- attach the visualization results and
- write down your analysis stating the difference between results (Max. 600 characters).

Note that you must obtain 6 different visualizations.

1.2 Convolution and Average Pooling using PyTorch [10pts]

Implement the above CNN models using PyTorch. In your report,

- attach the visualization results and
- provide whether your implementation using NumPy is the same as that using PyTorch by calculating any errors.

2 Convolutional Neural Networks (CNN) [60pts]

In this part, implement a convolutional neural network (CNN) on the FashionMNIST 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_problem_2.ipynb` file.

2.1 A CNN with MaxPooling layers [20pts]

You must implement a CNN model under the `CNN_Max()` class. The model has a sequential structure:

Input	Input of shape (28,28,1)
Layer 1	3×3 Convolutional Layer with 32 filters
Layer 2	2×2 Max Pooling Layer with a stride of 1
Layer 3	3×3 Convolutional Layer with 64 filters
Layer 4	2×2 Max Pooling Layer with a stride of 1
Layer 5	Flatten Layer.
Layer 6	Dense Layer with 64 output units with ReLU activation.
Layer 7	Dense Layer with 32 output units with ReLU activation.
Layer 8	Dense Layer with 10 output units.
Layer 9	Softmax Layer (No need for PyTorch users)
Output	

where the convolution layer is with bias=True, which needs to be accounted for calculating the number of parameters.

All other arguments use default values. You will also implement forward and backward passes to optimize CNN by using stochastic gradient descent (SGD) with a momentum method. Note that your test accuracy should be over 90.2% on the test images.

In your report,

- analyze the number of parameters used in each layer and the total number of parameters over the entire model considering the input image size,
- attach the graph of training and validation accuracies over 30 epochs,
- attach the capture of the test accuracy on the $10,000$ test images from the ipynb screen.

2.2 Prevention of Overfitting [40pts]

`Overfitting` happens when your model fits too well to the training set. It then becomes difficult for the model to generalize to new examples that were not in the training set. For example, your model

recognizes specific images in your training set instead of general patterns. Your training accuracy will be higher than the accuracy on the validation/test set. To reduce overfitting, you must implement techniques such as followings (but are not limited to) to the CNN model in Problem 2.1:

- Batch Normalization
- Dropout
- Data Augmentation
- other methods or tricks

You can decide to set the parameters for the regularizations (e.g. dropout rate).

In your report,

- attach a graph of training-and-validation accuracies over 30 epochs with a selected technique that handles overfitting,
- report the test accuracy on the $10,000$ test images,
- compare the results of the CNN model without regularization methods (analyze and explain why the selected technique works for preventing overfitting).

3 Comparison of MLP and CNN [20pts]

In this problem, you will compare a CNN model with MLP and analyze the performance difference by computing validation accuracies given the Fashion MNIST image dataset. You have to modify the code in the `CNN_problem_3.ipynb` file.

Compare the validation accuracies of two models:

- A CNN model with MaxPooling layers from Problem 2.1 in this assignment.
- An MLP model with ReLU layers (you can modify as you want).

In your report,

- attach a plot of validation accuracy curves from the two models where the x axis and y axis are the number of training epochs (which is 30) and accuracy, respectively,
- analyze the results (e.g., why does one model perform better than the other?) within one paragraph.

Submission Guide

3.1 Submission Requirement

You have to submit two types of materials: report and code.

- Report: Write a report answering all the questions in the assignments as a **PDF** file.
- Code: Change your code file name to `cs470_yourname_studentID_idx.ipynb` in Colab, where idx is the index of the problem. For example, after modifying the file `CNN_problem_1.ipynb`, save as `cs470_yourname_studentID_1.ipynb`, and for the `CNN_problem.ipynb`, save as `cs470_yourname_studentID.ipynb`. Download and save on your machine. Please make sure that the submitted notebooks have been saved and the cell outputs are visible.

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.

3.2 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