Introduction to Artificial Intelligence (CS470): Assignment 2

Deadline: October 6, 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 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. Note that you have to write down all the necessary equations and their derivation processes on your report.

a. Convolution and MaxPooling using NumPy [10pts]

Implement two simple forward networks as follows:
`CNN1:`

- Input Layer that takes input of shape (28,28,3)
- 5x5 Convolutional Layer with 3 filters
- Output

`CNN2:`

- Input Layer that takes input of shape (28,28,3)
- 5x5 Convolutional Layer with 3 filters
- 2x2 Max Pooling Layer with a stride of 2
- Output

In your report, attach the visualization results with your analysis. You must obtain 6 different visualizations and state the difference of the results.

b. Convolution and MaxPooling using PyTorch [10pts]

Implement the above 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. To compute the errors between two methods, you can create a new cell and you must keep the cell with executed results for the report.

2. Convolutional Neural Networks [60pts]

In this part, 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_problem_2.ipynb` file. Note that you have to write down all the necessary equations and their derivation processes in your report.

a. A CNN with MaxPooling layers [20pts]

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

- Input Layer that takes input of shape (32,32,3)
- 5x5 Convolutional Layer with 6 filters
- 2x2 Max Pooling Layer with a stride of 2
- 5x5 Convolutional Layer with 16 filters
- 2x2 Max Pooling Layer with a stride of 2
- Flatten Layer that reshapes the activations from the last layer to (N,C). N is the batch size and C is the number of features.
- Dense Layer with 120 output units with ReLU activation.
- Dense Layer with 84 output units with ReLU activation.
- Dense Layer with 84 input units to 10 outputs.
- Softmax Layer (No need for PyTorch users)
- Output

All other arguments use default values. You will also implement forward and backward passes to optimize CNN by using stochastic gradient descent (SGD) with momentum method. Note that your classification accuracy should be over 62% 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 validation accuracies over 50 epochs.

b. 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.a:

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

You can decide to set the parameters for the regularizations (e.g. dropout rate). Then, you must report the best accuracy on the $10,000$ test images, and attach a graph of training-and-validation accuracies over 50 epochs for the best technique that handles overfitting. Next, you have to compare the results of the CNN model with different regularization methods by applying them separately (e.g., a CNN with Batch Normalization, a CNN with Dropout, a CNN with Data Augmentation, etc) and in combination (e.g., a CNN model with Batch Normalization and Dropout, a CNN model with Dropout and Data Augmentation, etc)

You have to obtain at least 4 CNN models with each regularization technique and at least 11 CNN models with regularization methods in combination for comparison. If you apply additional regularization methods in more than one, consider all the combinations with others.

Then, analyze and explain why one technique or combination 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 CIFAR-10 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(a.) in this assignment.
- An MLP model with ReLU layers from the Problem 1 in assignment 1 (without hyperparameter tuning).

In the report, you must 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 50) and accuracy, 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_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_2.ipynb, save as cs470_yourname_studentID_2.ipynb. 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