Document History

Date	Changed By	Key changes
25-Jan-2023	MDSR	Document created
28-Jan-2023	MDSR	• Library usage • Preservation of Trained Model • Report Writing • Submission instruction
31-Jan-2023	MDSR	• Added lines in introduction and report writing sections make architecture tuning optional. Also training data can be subsampled. • The changed/added lines are colored in blue.

Bangla Character Recognition Challenge

In this assignment, you will have to implement a convolutional neural network for an image classification task. There will be six basic components in your neural network:

1. Convolution: There will be four (hyper)parameters:
    a. Number of output channels
    b. Filter dimension
    c. Stride
    d. Padding
2. ReLU Activation
3. Max-pooling: There will be two parameters:
    a. Filter dimension
    b. Stride
4. Flattening layer: it will convert a (series of) convolutional filter maps to a column vector.
5. Fully-connected layer: a dense layer. There will be one parameter: output dimension.
6. Softmax: it will convert final layer projections to normalized probabilities.

Write separate classes for each of the aforementioned building blocks. **Vectorize your code whenever possible to speed up training and inference**. Modularize your code well, set up the architecture in one place such that It is trivial to change the model architecture later on **(for possible online and retraining)**. While it is preferable to tune your architecture, this maybe a time consuming effort and is therefore deemed optional. Also, if you are unable to train with the full training set due to limited computing resources, you can subsample.

You will have to implement the back propagation algorithm to train the model. The weights will be updated using mini-batch gradient descent. No deep learning framework is allowed for your implementation. Since the architecture is not fixed, you have to modularize your code in such a way that it works for any architecture that uses the six mentioned modules. To make your implementation efficient, try to pose each operation as matrix multiplication.

For preparing and training your model, write your code in a file named <YourRollNo>_train.py.

Library usage

Purpose	Allowed Packages
Read the images	opencv, pillow
Visualization	matplotlib, seaborn
Progress bar	tqdm
Data manipulation	numpy, pandas
Model saving and loading	Pickle
Performance metrics and statistics	scipy, sklearn

Dataset

1. We will be using the ***Numta Handwritten Bengali Digits*** for this task. Find the data description here: https://www.kaggle.com/competitions/numta/data. The data may not be downloadable from Kaggle, but you can download it from the actual source, which is: https://bengali.ai/wp-content/uploads/datasets/NumtaDB_with_aug.zip. In case, that is performing too slow, download it from here: https://drive.google.com/drive/folders/1iaLxuSN88OyOuHwEbwBzfmwf9gFi_Vqn?usp=share_link
   Then follow the subsequent steps to prepare the data for our task
2. Combine training-a, training-b, training-c datasets to form your training + validation set.
3. Use training-d as your independent test set.

You will also be given a toy dataset that you can use to test whether your implementation of the back propagation algorithm is working correctly or not.

Preservation of Trained Model

You must save your final model in pickle (see: https://docs.python.org/3/library/pickle.html). You should write a separate python script (named: <YourRollNo>_test.py) that can load the pickle file of your trained model and use it to predict labels for query images (i.e., the images for which classification needs to be done) from a specific folder. The path of the folder will be a command line parameter. In the same folder you will prepare a CSV file with 2 columns. The first column contains the input file names (just the name, excluding path) and the second column contains the corresponding predicted digit.

The exact command that we would want to use is:

```
>> python 1705XXX_test.py path_to_folder
```

A sample 1705XXX_prediction.csv file should look like this:

```
FileName,Digit
a00014.png,5
a00039.png,8
```

Report Writing

1. You have to report the training loss, validation loss, validation accuracy and macro-f1 for each full pass over the training set.
2. Prepare graphs for different learning rates (and optionally for different models)
3. Also report the confusion matrix for each such model.
4. Make sure you tune the learning rate (start from 0.001). Select the best model using macro-f1 and report the values of the above-mentioned scores.
5. Finally, for the best (chosen) model, report the independent test performance.

Thrive for good results

You should train hard to get the best results, sky is the limit. Do not overfit, however. During online, a separate independent test set will be used to measure the performance of your model. **The top three performing models in each section will be duly recognized.**

Submission instructions

1. Create a directory with your 7-digit student id as its name.
2. Put the training and test python scripts (2 python files) into the directory created. **Notebook submission is not allowed.**
3. Put the saved model file into the directory created. Name it as <YourRollNumber>_model.pickle.
4. Also create a PDF report, as already described, Name it as <YourRollNumber>_report.pdf, and put it into the same directory.
5. Zip the directory (compress in .zip format) and upload the .zip file on Moodle in the designated assignment submission link.

Honour Code

While you are encouraged to talk to your peers, ask help from teachers, and search relevant resources from the Internet, under no circumstances should you copy code from any source. If found out, you will penalized with due negative marking.