

## 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. 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).

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.

## 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](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](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.

## Report Writing

You have to report the training loss, validation loss, validation accuracy and macro-f1 for each full pass over the training set. Prepare graphs for different models (try different architectures, learning rates etc.) 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.

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

Create a directory with your 7-digit student id as its name. Put the source file (only one source code file is allowed) into the directory created. Also create a PDF report, as already described, and put it into the same directory. Both the source file and the report should be named by your student ID, with appropriate extension. 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 be penalized with due negative marking.