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 layer: there will be four (hyper)parameters: the number of output channels, filter dimension, stride, padding.
2. Activation layer: implement an element-wise ReLU.
3. Max-pooling layer: there will be two parameters: filter dimension, stride.
4. Fully-connected layer: a dense layer. There will be one parameter: output dimension.
5. Flattening layer: it will convert a (series of) convolutional filter maps to a column vector.
6. Softmax layer: it will convert final layer projections to normalized probabilities.

The model architecture will be given in a text file. A sample architecture is shown for your convenience.

```
Conv 6 5 1 2
ReLU
Pool 2 2
Conv 12 5 1 0
ReLU
Pool 2 2
Conv 100 5 1 0
ReLU
FC 10
Softmax
```

You will have to implement the backpropagation algorithm to train the model. The weights will be updated using batch gradient descent, where instead of optimizing with the loss calculated over all training samples, you will update gradients with a subset of the training set (ideally 32 samples) in each step.

You will work with two datasets: MNIST and CIFAR-10. Both datasets are openly available and have 50k-60k samples. Split the evaluation set into half so that you can use 5k samples for validation and 5k samples for test purposes. You will also be given a toy dataset to test whether or not your implementation of the backpropagation algorithm works correctly.

You have to report the validation loss, accuracy, and macro-f1 for each epoch (one pass over the full training set). You will train your model for 5-10 epochs (more if it is runnable in reasonable time). Make sure you tune the learning rate (start from 0.001). Select the best model using macro-f1 and report the above-mentioned scores.

No deep learning framework is allowed for your implementation. No hardware acceleration is required (but allowed if you want to). Since the architecture is not fixed, you have to modularize your code in such a way that it works for any architectures that use the six mentioned layers. To make your implementation efficient, try to pose each operation as matrix multiplication.

Submission instructions: Create a directory with your 7-digit student id as its name. Put the source file (only one file is allowed) into the directory created. Also create a PDF report containing the evaluation scores for the three datasets and put it into the same directory. Both the source file and the report should be named by your student ID. Zip the directory (compress in .zip format. Any other format like .rar, .7z etc. is not acceptable) and upload the .zip file on Moodle in the designated assignment submission link. Failure to follow the above-mentioned submission guideline may result in up to 10% penalty.

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 receive full 100% negative marks.