SMAI Spring-2018
Assignment-2

---

# Convolution Neural Networks [18 marks]

There exist a vast variety of Neural Networks: Feed-forward Network, Radial Basis Network, Hopfield Networks, Recurrent Networks, Convolution Networks, Autoencoders, Generative Adversarial Networks, Deep Residual Networks, Deconvolution Networks etc. (for a more detailed list See here).
At the center of the deep-learning success lie the convolution neural networks and the variants of recurrent neural networks and generative neural networks. In this question we deal with the Convolution Neural Networks (also known as conv-nets). This question is divided into two parts:

1.1 Forward pass of Convolutional Neural Network. [5 + 1 marks]
Implement the forward pass of the neural network (LeNet) as shown in Figure 1 (Do not use any deep learning libraries):

Initialize the weights of the network randomly and (visualize after conv-layers) show output after each layer for any sample image.

Here we do not ask you to implement the backward pass of the convolutional neural network. But interested users can go through the following tutorial for more practical and theoretical details of backpropagation in Convolutional Neural Networks. Blog by Mayank Agarwal, Slides by Zhifei Zhang from University of Tennessee

1.2 Practical aspects in Deep Networks [12 marks]

The task is to analyze result variations due to parameter changes: batch-size, activation function (tanh, sigmoid, relu), learning-rate, number of convolutional-filters and number of (convolutional) layers. For this problem we will use the CIFAR-10 dataset. The following codebase (ipython notebook) is provided for a quickstart: Google Drive. Detailed questions can be found in the shared ipython notebook.

This part will be graded on the basis of understanding of the code and the ability to understand the variations due to the practical parameters as reported above (and not just merely running the code or copying the results).

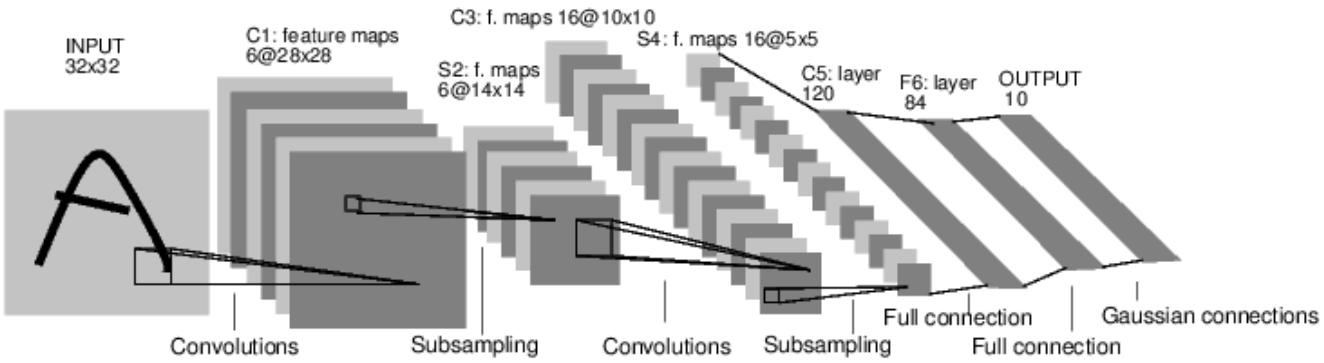


Figure 1: LeNet Architecture is [CONV-POOL-CONV-POOL-FC-FC]. Conv filters are 5x5, applied at stride 1. Pooling layers are 2x2 applied at stride 2

> For this subpart of the question you are allowed to use other libraries like tensorflow, keras, pytorch etc. (Code is also provided in *keras*: for experimentation.)

Write a report based on your results and observations.

# 3 Layer Neural Network [12 Marks ]

Implement a simple 3 layer feed-forward neural network for multi-class classification problem. Implement back-propagation for training the parameters. Do not use any standard Neural Network library (write the code from scratch). Write a report based on your results and observations due to variations of parameters.

1. **Activation Function:** Use 2 different nonlinear activation function in hidden layer. Why did you use these function?
2. **Loss Function:** Use cross-entropy loss.
3. **Hidden Layer:** Vary the number of units in hidden layer.
4. **Stopping Condition:** While training what stopping condition are you using and why?

5. **DataSets:**

   - Learn a 3 class classifier for **Dermatology dataset** for classes psoriasis, seboreic dermatitis, lichen planus. (Divide data in train and test set and report the findings).
   - Learn 4 class classifier for **Pendigit recognition dataset** for any four digits. (Use pendigit.tra for training and pendigit.tes for testing)

## Report

1. Clearly report your experiments and results.
2. Report 5-fold cross validation accuracy for each combination of parameters (namely, number of hidden nodes and activation function)