

## Basics of ML and Deep Learning

**Objective :** The assignment aims at familiarizing you with numpy and the basics of ML and feed-forward neural networks in python. We will be working mostly on the ipython notebooks for the assignment. You can refer <http://cs231n.github.io/ipython-tutorial/> for installing the required software for working with ipython notebooks and getting started with it.

### Assignment 1a : Multiple linear regression model

This part of the assignment consist of a python notebook that depicts the outline of using multiple linear regression to learn a linear function to map input data to the output. The function should be learned with the objective to minimize the **Mean Square Error (MSE) loss** between the predicted output and the ground truth data.

The dataset for training and testing is provided in the assignment itself ('Concrete\_Data.csv')

### Assignment 1b : Feedforward Neural Network model

This part of the assignment consists of learning a feedforward neural network to classify three types of flowers on the basis of certain input features. As in the previous assignment, the objective is to decrease the **categorical cross-entropy loss** between the predicted output and the ground truth data.

The dataset for training and testing is provided in the assignment itself ('Iris\_Data.csv')

For both the two assignments, you have to fill all the missing parts which is indicated using question marks (?). Please don't declare any function other than given in the notebook. The functions have their usual meanings and is described below in brief. Some of the functions are partially implemented and some of them are left for you to complete it.

Since the objectives are similar, we describe the common functions you need to write for both these cases (i.e Assignment 1a and 1b).

- `shuffle_dataset()` : Simple function to shuffle the X and Y dataset. The dataset has to be shuffled **in place**. **(0.5 + 0.5)**
- `__init__()` : Initialize the linear regression and the neural network models by specifying the biases and the weight matrices **(2 + 5)**
- `forward()` : Perform a forward pass by taking the entire dataset as the input for both the models and calculates the output. Compute the predicted scores (for the LR model) and labels ( for the NN models). **(3 + 8 )**
- `backward()` : Computes the gradients for every parameter of the network and update the corresponding parameter using gradient descent step. This is also called a

backward pass as a prediction has been made and now the network is updated accordingly, so as to minimize the loss function. **(5 + 10)**

- `MSE_loss()` : It computes the Mean Squared Loss between the predicted output and ground truth data. **(3)**
- `crossEntropy_loss()`: It computes the categorical cross entropy loss between the predicted output and ground truth data. **(5)**
- Accuracy: Computes the accuracy between the predicted and actual values for the NN model **(3)**.
- Compute the values of the following:
  - Final training loss **(1+1)**
  - Final test loss **(1+1)**
  - Final accuracy of the NN model **(1)**

Submission :

You have to submit **two ipython notebooks** with the .ipynb extension for the two assignments . Please make sure that all the cells are running. The ipynb notebooks with the skeleton code are given for your convenience. Please make the requisite changes there and submit. Outline of the results mentioned above has been implemented in the notebook, you just have to complete the blanks.

Please submit the completed file named as Assignment1a\_<RollNo>.ipynb and Assignment1b\_<RollNo>.ipynb

**No report is required.**