# Assignment #2: CIFAR-10 Image Classification using Fully Connected Neural Network

**Submit Assignment**

---

**Due** Feb 17 by 11:59pm          **Points** 40          **Submitting** a file upload          **File Types** zip

---

In this assignment, you are going to implement a one hidden layer fully connected neural network using Python from the given skeleton code **mlp_skeleton.py** on Canvas (find in the Files tab). This skeleton code forces you to write linear transformation, ReLU, sigmoid cross-entropy layers as separate classes. You can add to the skeleton code as long as you follow its class structure. Given N training examples in 2 categories $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, your code should implement backpropagation using the cross-entropy loss (see Assignment 1 for the formula) on top of a sigmoid layer: (e.g.

$p(c_1|x) = \frac{1}{1+\exp(-f(x))}, p(c_2|x) = \frac{1}{1+\exp(f(x))}$ ), where you should train for an output

$f(x) = \mathbf{w}_2^\top g(\mathbf{W}_1^\top \mathbf{x} + b) + c$. $g(x) = \max(x, 0)$ is the ReLU activation function (note Assignment #1 used a sigmoid activation but here it's ReLU), $\mathbf{W}_1$ is a matrix with the number of rows equal to the number of hidden units, and the number of columns equal to the input dimensionality.

**Finish the above project and write a report (in pdf) with following questions:**

**Please put the report(in pdf) and the source code into a same zip file, "firstname_lastname_hw2.zip". Submit this zip file on Canvas. You have to make sure your code could run and produce reasonable results!**

1) Write a function that evaluates the trained network (5 points), as well as computes all the subgradients of $W_1$ and $W_2$ using backpropagation (5 points).

2) Write a function that performs stochastic mini-batch gradient descent training (5 points). You may use the deterministic approach of permuting the sequence of the data. Use the momentum approach described in the course slides.

3) Train the network on the attached 2-class dataset extracted from CIFAR-10: (data can be found in the **cifar-2class-py2.zip** file on Canvas.). The data has 10,000 training examples in 3072 dimensions and 2,000 testing examples. For this assignment, just treat each dimension as uncorrelated to each other. Train on all the training examples, tune your parameters (number of hidden units, learning rate, mini-batch size, momentum) until you reach a good performance on the testing set. What accuracy can you achieve? (20 points based on the report).

(4) Training Monitoring: For each epoch in training, your function should evaluate the training objective, testing objective, training misclassification error rate (error is 1 for each example if misclassifies, 0 if correct), testing misclassification error rate (5 points).

(5) Tuning Parameters: please create three figures with following requirements. Save them into jpg format:

i) test accuracy with different number of batch size

ii)test accuracy with different learning rate

iii) test accuracy with different number of hidden units

(6) Discussion about the performance of your neural network.