**Ahmad El Rouby**

**900143559**

**Spring 2018**

**Machine Learning**

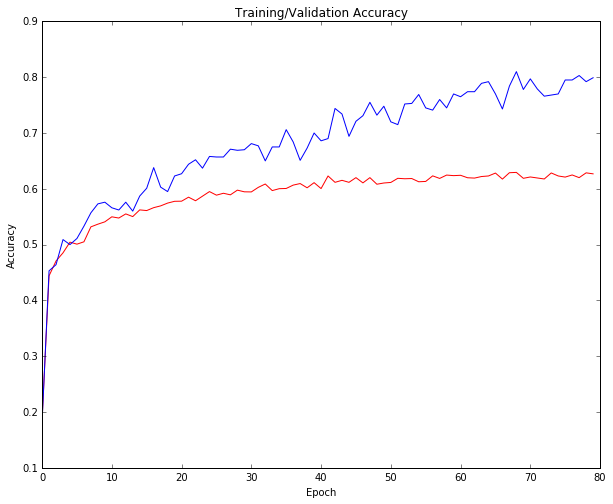
**Assignment 2**

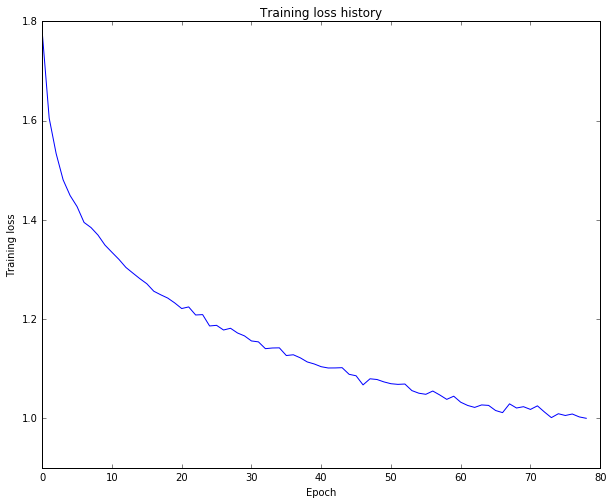
**Part 1**

* Source Code is attached in the folder with random seed (3) named “test\_nn\_note.ipynb”

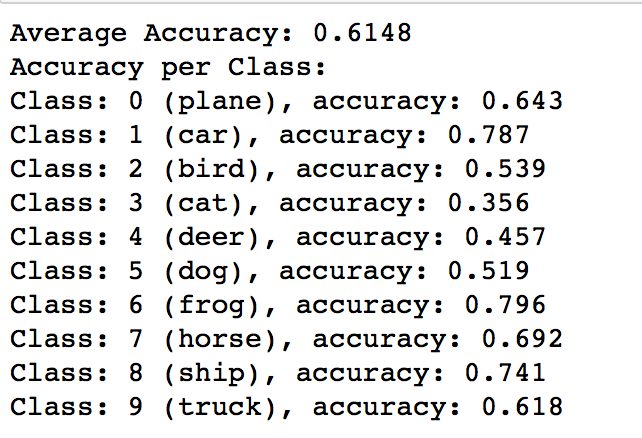
**Part 2**

1. I divided all test and training data images by the standard deviation of the training set and subtracted the mean of the training set as well. I also perform real-time data augmentation by apply random transformation to the training batch in each step to create diversity in the training set.
2. I tried different architectures including pyramid and bottleneck. I used to change learning rate and regularization according to the training statistics. Whenever, overfitting occurred with the training accuracy and loss are far away from the validation ones, I increased the regularization rate. I also tried different learning rates to find the one that gives best accuracy and choose 0.001
3. I didn’t stop training because my implementation saves the best parameters according to the validation accuracy (Blue is for training and Red is for Validation)





1. CCRn for the fully connected Neural Network:



CCRn for the LLS Classifier

Class: 0 ( plane ) , accuracy: 0.469

Class: 1 ( car ) , accuracy: 0.445

Class: 2 ( bird ) , accuracy: 0.207

Class: 3 ( cat ) , accuracy: 0.177

Class: 4 ( deer ) , accuracy: 0.243

Class: 5 ( dog ) , accuracy: 0.285

Class: 6 ( frog ) , accuracy: 0.449

Class: 7 ( horse ) , accuracy: 0.426

Class: 8 ( ship ) , accuracy: 0.508

Class: 9 ( truck ) , accuracy: 0.428

1. ACCR: I achieved 61.5% accuracy using my own implementation. The notebook file (test\_nn\_note) has the saved values for proof. I also got 72% accuracy using a the keras implementation and the .h5 File can also be found in this folder.