COL 341: Assignment 3

Notes:

- This assignment has one part - Image Classification using ConvNet Challenge.
- You need to make multiple submissions on the following dates; **Oct 10, 17, 24 & 31 and Nov 7 & 14**.
- This assignment can be done in teams of size **two or less**.
- Mention the team details here. The deadline for reporting the team info is 8th Oct.
- Include a *running* report of maximum 5 pages explaining what you did. Continue appending all changes to the previously submitted version.
- You should use Python for all your programming solutions.
- Your assignments will be auto-graded, make sure you test your programs before submitting. We will use your code to train the model on training data and predict on test set.
- Input/output format, submission format and other details are included. Your programs should be modular enough to accept specified parameters.
- Submit your code, report and predictions
- You should submit work of your own. You should cite the source, if you choose to use any external resource. You will be awarded F grade or DISCO in case of plagiarism.
- No buffer days are allowed for this assignment.
- Submit only a zip file.
- Data is available at this link

1. **Image Classification (50 points, Due date: 10^{th} November, 2019)**
   In this problem, we will use Convolution Neural Network to classify given images of CIFAR 100 dataset.

   (a) The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image comes with a ”fine” label (the class to which it belongs) and a ”coarse” label (the superclass to which it belongs).

   (b) You only have to predict the **fine class labels**.

   (c) Each row in the file corresponds to one image with second last and last entry being the coarse label and fine label respectively.

   (d) Since this is a competition, you are allowed to use all forms of feature engineering and any neural network architecture.

   (e) Experiment with the architectures and report your results and observations.

   (f) Report your results and observations.

   For this problem you will NOT only be evaluated on the accuracy obtained but also on the reported analysis of the best performing architecture and a demo.

   Evaluation:

   - Accuracy will be used as evaluation criterion.

- Marks will be based on your results and report. Note that you will not get any mark in case your code fails to run.
- To reward consistency, you'll be required to make a submission a on every 7th day, and we'll use it to compute your Weekly Average Performance (WAP). Marks will be given on the basis of this WAP and final score on a test set.
- Your demo score will be the harmonic mean of the demo scores of both members.

Submission Instructions:

- All the codes should be in Python 3.5 or higher.
- Please specify all the dependencies in a README file.
- You have to submit the prediction file along with your submission. We'll run all your codes again and if your predictions do not match with the submitted prediction file, there will be a heavy penalty/disciplinary action.
- **You will use Keras to train the CNN**.
- Your submission should include one file called **compete.py** which should run the training process with the best set of parameters. It should be runnable using the command: *python compete.py trainfile.csv testfile.csv output.txt* and write the predictions for test data (1 per line) in *output.txt*.
- Your code should complete within **6 hours**.
- For any error in the code or layout of output file, a penalty will be applied depending on the issue.
- Report the results obtained with all the architectures but elaborately explain only the one with best results.

Extra Readings (highly recommended):

(a) CIFAR 100: Information

(b) Papers with Code

(c) SOTA: CIFAR 100

Bonus Marks

(a) If you're model achieves 2% better results than global state-of-the-art (till the time of moderation), you will get **25 bonus marks** for the report. You'll also be awarded full marks across **all** assignments.

(b) If you wish to submit your code for this part, you'll be given 24hrs to finish your code.