Assignment 1:

- User Keras to work on these problems
- Due date for assignment : Oct 11th 11.59 pm
- Try your code locally and submit your code through github + Colab
- This is an individual assignment; No collaboration allowed.
- Send a report with results and analysis to analyticsneu@gmail.com

Data to be used: http://www.cs.utoronto.ca/~kriz/cifar.html

Preparation: Review: https://github.com/fchollet/keras/blob/master/examples/mnist_mlp.py

Problem 1: (50 points)

Using Keras, build a MLP to classify the CIFAR-10 dataset. Note that each record is of size 1*3072. Starting with the MNIST example code, build a MLP to classify the data into the 10 classes.

Modify the following parameters and discuss the effect of changing parameters on loss and accuracy.

1. No of epochs
2. Batch size
3. Network configuration
    a. Number of neurons in a layer
    b. Number of layers
4. Learning rate
5. Activation functions
6. Dropout rates

Ensure you are building the model with the training data set and validating against the provided test data set.

- You are expected to provide a recommendation for the best model you would recommend for classification. Which model (with parameter values) would you choose and why?
- Comment on how good your model is ? Does it overfit/underfit data ? What could you do to improve the model?

We are going to continue using the CIFAR-10 dataset for Part 2 but this time around, we will use Keras to build a CNN network.

Start with this as your base code:
https://github.com/fchollet/keras/blob/master/examples/cifar10_cnn.py

See https://keras.io/getting-started/faq/ to get answers to commonly asked questions.

Part A:

I have pre-selected some of the topics I would like you to review in the Keras documentation. You will review this section and change various parameters for your given topic and discuss your findings when you change these parameters (keeping others constant).

No	Goal	Topic
1	Review what preprocessing you could do with images; Discuss data augmentation	https://keras.io/preprocessing/image/ Review https://medium.com/towards-data-science/image-augmentation-for-deep-learning-using-keras-and-histogram-equalization-9329f6ae5085 https://machinelearningmastery.com/image-augmentation-deep-learning-keras/ for some details
2	Choose different optimizers (SGD, Adam, RMSprop etc. and compare and contrast the optimizers) Which one would you recommend and why?	https://keras.io/optimizers/
3	Use the sk-learn's grid search api to change multiple parameters	https://keras.io/scikit-learn-api/
4	Experiment with different parameters you can experiment with a conv-2d layer	https://keras.io/layers/convolutional/#conv2d
5	Discuss and use different types of layers that you could use for the problem	https://keras.io/layers/core/ https://keras.io/layers/pooling/

	See `Dense, Dropout, Activation, Flatten`	
6	Discuss and use the different activation and pooling layers function	https://keras.io/activations/ https://keras.io/layers/advanced-activations/ https://keras.io/layers/pooling/
7	Discuss different Metrics you could use. Create your own custom metric	https://keras.io/metrics/
8	GPU option vs CPU option Try with CPU on your machine. Also try using Amazon's infrastructure with a GPU and discuss your findings Try different callbacks including the Tensorboard callback	See https://hackernoon.com/keras-with-gpu-on-amazon-ec2-a-step-by-step-instruction-4f90364e49ac for a tutorial https://keras.io/callbacks/
9	Try different layer configurations for deeper network topologies	See https://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/ for examples
10	Discuss the use of Noise and Normalization layers. Also try writing your own layer	https://keras.io/layers/noise/ https://keras.io/layers/writing-your-own-keras-layers/ https://keras.io/layers/normalization/
11	Functional API	https://keras.io/getting-started/functional-api-guide/

Deliverables: by Thursday midnight (Oct 11th)

1. 5 minute youtube video to be shared through discussion board
2. Code through github + Colab
3. A 2-page report discussing your finding

Note: You will share these three with the entire class. This is not your final model but you are testing one set of parameters out of many to find what happens starting with the base code.

1 babel.n@husky.neu.edu
2 bhimarayasinghekar.m@husky.neu.edu
3 cui.wenq@husky.neu.edu
4 dai.shi@husky.neu.edu
5 davda.g@husky.neu.edu
6 emmanuel.di@husky.neu.edu
7 kaushik.pr@husky.neu.edu
8 movva.r@husky.neu.edu
9 nagare.p@husky.neu.edu
10 perveez.s@husky.neu.edu
11 thirumalaisamy.d@husky.neu.edu

Part B:

You should read it from a configuration file rather than hardcoding parameters. Start from the base case code (https://github.com/fchollet/keras/blob/master/examples/cifar10_cnn.py) and abstract out constants to a configuration file (You are free to choose the format of the configuration file. It could be another .py file, csv, json etc.). Please convert the code to a jupyter notebook

Starting from the base case, try different experiments to see if you can get better accuracy. Design your model such that you can easily change parameters and rerun experiments.

Discuss at-least 3 experiments with different parameters and your best model. Also comment on which parameters you thought helped better accuracy compared to that of the base case.

Deliverables: (By Oct 11th midnight)

1. A new version of your code.
2. The new configuration and outputs
3. Discussion on your design, experiments and outputs in a 1-2-page report