Problem Set 06
Data Mining and Machine Learning – Spring 2016
Due date: 2016-04-08 13:00

All assignments must be uploaded to the assignments tab in ClassesV2 (notice that this is **not** the dropbox) by the date and time specified. Make sure that you follow the instructions exactly as described. You may discuss problem sets with others, but must write up your own solutions. This means that you should have no need to look at other's final written solutions.

You need to turn in all of your solutions as a zip compressed file, named `netid_pset06.zip`, with your actual netid filled in in all lower case letters. This archive should contain the following two files:

- `pset06.pdf`

- `pset06.py`

The python file will **not** be run or autograded, but is just for showing your work for the assignment. The pdf file should contain results and answers to the questions below.

General instructions

For this problem set you will use python and keras library to fit models to the CIFAR-10 dataset. To make the time to compute these functions feasible on standard hardware, we will subset to only using the first 3 classes. There is a starter code with some functions that you may find useful here:

    http://www.stat.yale.edu/~tba3/psets/pset06/pset06_starter.py

Throughout the assignment, unless otherwise noted, use the following learning parameters:

- a validation split of 20% of the training data

- batch size of 32

- 25 epochs, with early stopping using a patience of 2

- RMSprop learning algorithm, default settings

- cost: 'categorical_crossentropy'

- 'relu' activation functions

- dropout (tuning parameter of 0.2) following every hidden layer

- a final softmax layer

- no weight regularization

You must upload your python script, but we will not be autograding it so you do not need to worry about making it run on our machines. It is just serves to show your work.

Do not save this assignment to the last minute! These models will take a while to run, depending on your hardware some could take upwards of 5 minutes per epoch.

I. Width of the model

Fit several neural networks with one hidden layer, using a number of hidden nodes equal to: (2, 8, 32, 128, 512). Present the classification rate on the test dataset for each of these as either a table or graph. Describe the pattern you see and explain why the classification rate may change this way with an increase in the width of the model. Also explain any anomalies you observe (and don't worry if you don't see any; you may or may not due to randomness in the fitting process).

II. Depth of the model

Fit neural networks with hidden layers having 512 nodes, for 1, 2, 3, 4, and 5 layers. Present the classification rate on the test dataset for each of these as either a table or graph. Describe the pattern you see and explain why the classification rate may change this way with an increase in the width of the model.

III. Freezing layers

Build a neural network with 5 hidden layers of 512 nodes each, but do so by training each layer one at a time and freezing the weights before including the next layer. Save the test classification rates at each stage and present them as a table. How do these rates compare to the classification rates in part II? What technique does this approach resemble that we used in the first half of the semester? How could we implement a compromise between the model in part II and the model here?

IV. Autoencoder

In this section, you will create an autoencoder which builds a neural network that attempts to reconstruct its input. You should not use a final softmax layer, and you should change the objective function to 'mean_squared_error'.

Fit neural networks with one hidden layer, with a number of hidden nodes equal to: (32 128, 512, 1024). Present the mean squared error rate on the test dataset for each of these as either a table or graph. Describe the pattern you see and explain why the mean squared error may change this way with an increase in the width of the model. How do these compare to the pattern in part I? Explain why you might expect such a relationship.

V. Autoencoder as pretraining

Repeat question III (give classification rates, relate to other questions, and explain the pattern you see) but use the weights learned from question IV with 1024 hidden nodes as the (frozen) first layer. Do the results here surprise you at all? What technique does this approach resemble that we used in the first half of the semester?

VI. Experiment

Finally, using the results from the simulations in the first 5 questions, try to blend the benefits (and avoid the pitfalls) from each to a construct a new predictive model. You may (but do by no means need to do all of these) use alternative activation functions, optimization methods, weights, or combination of freezing layers and layer sizes and depths. You should **not** use convolution, however, as we will look at that in the next problem set. Explain the model you used, why you choose to this model based on the results from the other questions, and give the final validation prediction rate.