Problem Set 07

Data Mining and Machine Learning – Spring 2016
Due date: 2016-04-15 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_pset07.zip`, with your actual netid filled in in all lower case letters. This archive should contain the following two files:

- `pset07.pdf`
- `pset07.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. There is a starter code with some functions that you may find useful here:

`http://www.stat.yale.edu/~tba3/psets/pset07/pset07_starter.py`

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

- batch size of 32
- 25 epochs, with early stopping using a patience of 2
- RMSprop learning algorithm, default settings
- cost layer: 'categorical_crossentropy'
- 'relu' activation functions
- dropout (tuning parameter of 0.5) 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 10 minutes per epoch.*

I. Convolution model kernel size

Using the the two class version of the CIFAR-10 dataset, fit a model that consists of a 2D-convolution and 2-by-2 max pooling with 'relu' activation function, followed by a dense hidden layer with 512 nodes (don't forget the 'relu' activation and 0.5 dropout), followed by the output layer (complete with softmax). Run the convolution layer with 32 filters using three kernel sizes: 1, 3, and 5. How does the classification rate compare to those from the last problem set? Present the classification rates and try to explain why you see the pattern than you see. Explain in your own words what the kernel of size 1 is actually doing.

II. Convolution model as autoencoder

Repeat question I but use the model as an autoencoder (make sure to turn off the softmax and use mean squared error as a cost function). How does the pattern mimic and/or differ from that seen in question I?

III. Freezing the features

Using only a kernel of size 3, take the two models from part I and II and freeze the convolution layer. Refit the top dense layer, but in both cases do so on the 2 class classification task. How do the testing and training rates change and compare to one another? Give an explanation for why you might see such results.

IV. Transfer learning

Repeat question III, but now use the entire CIFAR-10 dataset. To be clear, the feature level is trained on only the subsampled data, frozen, and then used as an input to a single dense layer on the entire dataset. How does this rate compare to the rates you had on problem set 5?

V. Double convolution

Using a kernel of size 3, repeat question I, but now use two successive convolution layers. Note that there should be an activation layer between them, but no max pooling or dropout until the end. How does the result from this model compare to the result from using only one convolution?

VI. Extensions

Keeping the feature levels fixed as in part II, try to modify the architecture of the learning algorithm from part III to achieve the best classification rates you can on the entire CIFAR-10 dataset. For example, change the learning algorithm to plain vanilla SGD, add another dense layer, use regularization, and/or increase the number of hidden nodes. Describe your model and give the final classification rates.