October 15 2pm

Homework 4 Deep Learning
Due Saturday 11:59 pm EST October 29
Partner Homework

Neural networks have been growing rapidly in popularity over the past 5 years due proportionally to their rise in performance against other classifiers. Deep learning and convolutional neural networks in particular have become top performers, particularly with image data. This homework aims to give you some hands-on experience with both image analysis and deep learning.

The goals of this assignment are:

- Gain familiarity with a standard deep learning software package so you can use it
- Try out some image processing
- Compare deep learning methods against other approaches in terms of speed and performance
- Explore architectures and parameters for both deep learning and convolution neural networks

The sections this week will be of significant help in getting you started. Friday's (10/14) second paper presentation included some aspects of CNN as well. It is an active research area and plenty of discussions and research online. We hope all of these will help you get started and inspire you. Be sure to cite code (excluding section) that you have leveraged. We do expect that you and your partner will develop your own understanding through coding the various methods to learn how the approaches work.

1) **Download the CIFAR-10 data set**
   The images are available from https://www.cs.toronto.edu/~kriz/cifar.html. This data set contains 60000 32x32 color images with 10 different labeled classes. It is divided into 50000 training and 10000 test images. All training and tuning must be performed with this training set. All testing will be performed on this test set. This means that you may not include any additional training data sets and you may not ever train/tune on your test set.

2) **Select a deep learning package**.
   We strongly recommend TensorFlow since that will be covered in the section, but we are open to other packages. We will likely not be able to provide much support for alternatives.

3) **Establish baseline runs [20 points]**
   3a) Train and test a generic deep neural network on the data. Record the training time (your choice of approaches) and the accuracy on the test set.

   3b) Train and test a generic convolutional network on the data. Record the training time (your choice of approaches) and the accuracy on the test set.

3c) Train and test a non-NN algorithm with reasonable performance on the data. I would suggest not SVM for speed reasons. I would suggest avoiding Naïve Bayes and Decision Tree as they generally lack accuracy, but might be worth a try as a supplement. As before, record the training time (your choice of approaches) and the accuracy on the test set.

3d) Comment on the differences and perhaps plot the results

4) Assess the sensitivity of the test set to training data size [10]

4a) Randomly divide the training data into at least 3 pieces (like cross-validation): A, B, C, … but do so in a way that preserves the an equal fraction of the 10 categories of images across sets. That is, each set {A,B, etc.} should have the same number of category 1, 2,…10 images within it.

4b) Train your classifiers in part 3 on each of the following, A, AB, etc. and test on the full test set

4c) Plot and comment on how quickly the classifiers learn as compared to 3) using the full data set

5) Assess the sensitivity of the test set to distortions [15 pts]

5a) Create at least one additional test sets by distorting the test set in one manner (not the training data). Possible choices include:

- Adding random noise to the images of different levels
- Shifting the brightness or contrast by different levels
- Flipping or transposing the images
- Randomly settting different fractions of the pixels to an intermediate value
- Obscuring the top k rows of the images to an intermediate value (different levels of k)

These can be done on the fly in testing or separately. TensorFlow includes some options in their tf.images package but similar results could be obtained using GIMP, for example.

5b) Repeat the baseline run analysis in 3) on the distorted images

5c) Comment on what you learned

6) Compare the tuning of hyperparameters across architectural models [25 points]

6a) Select at least 2 architectural models where an architectural model can include:

- arrangement of layers in a CNN
- number of hidden layers
- number of nodes in hidden layers
- order of pooling/convolution/etc. in CNN, and/or
- CNN vs. deep NN

6b) Compare the architectures in terms of speed & accuracy across ≥ 2 hyperparameters such as:

- gradient descent optimization methods such as momentum, ADAM, Adagrad, etc.
- normalization effects
- regularization
- dropout

6c) Plot the results and comment on what you learned. Compare the errors (or accuracy) as a function of epoch.

7) Maximize your performance with your deep neural network or the CNN [30 pts]

a. Tune the parameters
b. Discover what works best and walk us through what you tried
c. Submit the performance as a %accuracy and as a 10x10 table in the format showing the counts of the test images actual vs. predicted.

	Predicted 1	Predicted 2…	Predicted10
Actual1	883	3	40
Actual2…	23	523	23
Actual10	1		999

8) Exploratory points [optional]

Exploratory points will be given if you go beyond these methods. We are open to your creative ideas for this part that could include different data sets, different types of networks, a significantly deeper analysis into the tuning perhaps, etc. Include a separate section highlighting what you have done that is worthy of exploratory points so we don't overlook your effort.

Grading Plan:

Like HW3, we will look at a combination of the performance semi-quantitatively and the approaches that you used to guide your search qualitatively. We feel this combination is consistent with the goals of the assignment. Be sure to walk us through what you tried in achieving good results, especially for Part 8.