

Assignment 5 – Convolutional Neural Networks and MNIST with Tensorflow.

Assignment overview. This assignment is designed to introduce you to the convolutional neural networks and their implementation with Tensorflow. The first question is just a basic implementation of a 2-dimensional convolution with the example of an edge detector. Your next task is to go through the basic Tensorflow tutorials provided by Google and then to experiment with some variations on the MNIST dataset.

Submission. As usual, submit your assignment as jupyter file on Brightspace.

Submission deadline. Tuesday, March 22, 2:00 pm.

Late submission policy. No late assignments are excepted.

Academic Integrity. Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at

If you have a question: Teaching Assistants (TAs) will be present during the labs to help you with any questions you may have. If you still have questions, feel free to email me at tt@cs.dal.ca.

Questions:

1. [10 marks] Implement an edge filter to filter an image. The convolution should be implemented with basic array processing without the use of a convolution function or an existing edge filter. You need to look up how to import and image into a numpy array. Show the original and the filtered image within your Jupyter file.
2. [40 marks] Go through the Tensorflow tutorials, specifically https://www.tensorflow.org/get_started/premade_estimators and <https://www.tensorflow.org/tutorials/layers>. A slightly modified (simplified) version of the basic convolutional network for the MNIST data set is provided in the file test_MNIST_cnn.py. Get familiar with the code provided. Compare it to the code provided in the Tensorflow website https://github.com/tensorflow/tensorflow/blob/r1.6/tensorflow/examples/tutorials/layers/cnn_mnist.py. In the provided program you should be able to change the number of classes to speed up running in case your system is slow. Also, you may need to use smaller batches.
 - 2.1. [10 marks, 5 marks for Grads] Modify the code to have only one convolutional layer by removing the second convolutional layer. Save your program in the MNIST_conv1.ipynb.
 - 2.2. [10 marks, 5 marks for Grads] Modify the code MNIST_conv2.ipynb to have 3 convolutional layers. Add another convolutional layer with filter size (5 by 5) after the second convolutional layer. Save this code in MNIST_conv3.ipynb.
 - 2.3. [20 marks, 10 marks for Grads] Evaluate the performance of networks with 1, 2 or 3 layers and report on your findings.
 - 2.4. Grad Students only [20 marks] Add label-noise to the training data set and train the two-layer model. To make mislabeled data you can simply shuffle some of the labels. For example, if you want to make 10 percent of the data mislabeled, you can permute/shuffle %10 of the labels. Evaluate the performance of the network with different rates of label noise (%10, %25, %50, %75, %100) applied. Illustrate your result in a plot. Explain the performance of your network with %100 noise, specifically the difference between the training and test error.