Assignment 6– 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. **Provide your answer as Jupyter files (.ipynb).**

Submission. Create a folder called ML_Assignment6 and put all the files inside the folder. Compress this folder to create either ML_Assignment6.zip or ML_Assignment6.rar. Submit this compressed folder as your assignment submission on Brightspace.

Submission deadline. Tuesday, Nov 14, 10:00 pm.

Late submission policy. If submitted after the due date, the penalty will be 10% per day.

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 https://www.dal.ca/academicintegrity

Python: We will be using Python for the programming exercises based on scientific Python libraries like

- numpy - mainly useful for its *N*-dimensional array objects.
- matplotlib - 2D plotting library producing publication quality figures.
- Jupyter Notebook - an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.

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. **[20 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 exciting edge filter. **Provide your answer as Jupyter file.**
2. **[80 marks]** Go through the Tensorflow tutorials, specifically
   https://www.tensorflow.org/get_started/get_started
   and
   https://www.tensorflow.org/get_started/mnist/beginners
   and
   https://www.tensorflow.org/get_started/mnist/pros
   Then answer the following questions. **Provide your answers as Jupyter files (.ipynb)**.

2.1. **[20 marks, 10 marks for Grads]** Get familiar with the code provided in the tutorial with 2 convolutional layers and save it in **MNIST_conv2.ipynb**. Try to run this code. If the code is running into an error message, try to track the error message. You may need to use batches to test the model. In the provided code, they test the whole data set at once. If your system has not enough memory, you can use batches of the test data (check how it is done in the training phase).

2.2. **[20 marks, 10 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**.

2.3. **[20 marks, 10 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**.

2.4. **[20 marks, 10 marks for Grads]** Evaluate the performance of networks with 1, 2 or 3 layers and **report** on your findings.

2.5. **Grad Students only [20 marks]** Add 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.

2.6. **Grad Students only [20 marks]** Evaluate the performance of the network with different rates of 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.