Data Analytics for Cybersecurity

Fall 2021

Homework 3 (75 points)

Instructions:

- Create your Jupyter notebook to solve the tasks below. Make sure your notebook is well documented with Markdown cells
- Submit your Jupyter notebook to Gradescope by the start of class on October 8, 2021.

TASKS:

1. Create a dataset class suitable for loading the HW3 dataset files (hw3-letters-test-images-idx3-ubyte, hw3-letters-test-labels-idx1-ubyte, hw3-letters-train-images-idx3-ubyte, hw3-letters-train-labels-idx1-ubyte) into your Jupyter notebook on Google colab (colab.research.google.com)
    a. You probably want to mount your Google drive and keep the files there
    b. The label files have a 4 byte integer for the number of items, followed by 1 unsigned byte per label
    c. The image files have a 4 byte integer for the number of images, followed by a 4 byte integer for number of row, followed by a 4 byte integer for number of columns, followed by 1 unsigned byte per pixel (0-255, white -> black)
    d. Create a child class for torch.utils.data.Dataset
    e. NOTE: I have deliberately changed the file format/data files from the MNIST dataset used by Yann LeCun. You should write your own load method to read the file instead of using a library that parses the IDX format.
2. Plot at least 20 of the images using matplotlib.pyplot.imshow (note they may appear rotated 90 degrees clockwise- that's OK. If it bothers you transpose the matrix)
3. Train a convolutional neural network to recognize the 26 letters (note the labels are 1-26 instead of 0-25).
    a. https://github.com/erykml/medium_articles/blob/master/Computer%20Vision/lenet5_pytorch.ipynb may be helpful, but that code works on images that are 32x32 (the homework images are 28x28 so you'll need to create a different architecture)
    b. You should be at least 90% accurate on the validation set. Don't overtrain your model!
4. Compare the time to train the model with and without a hardware accelerator
    a. You can use a comment to indicate the timed output since you can't easily switch back and forth in the middle of the code
5. Output a confusion matrix of the test set with model.

GRADING:

5 points – loads hw3 files
15 points – creates a proper child class of Dataset
10 points – plots at least 20 images using pyplot
10 points – trains a neural network that is at least 90% accurate on validation (test) set.
15 points – demonstrates speedup by using hardware accelerator
10 points – outputs confusion matrix
10 points – uses Markdown appropriately to create an easy to read program with properly labeled results