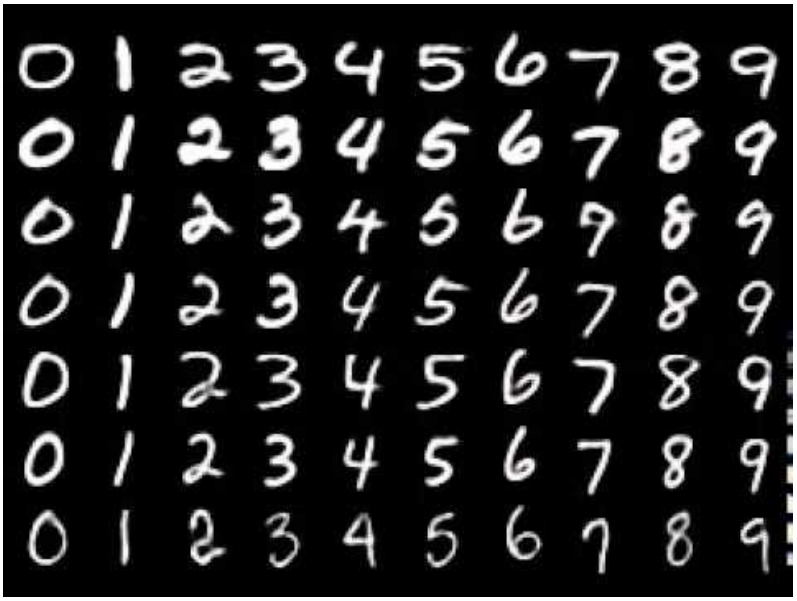
Kratos – Autonomous Subsystem QSTP-2021

Week 3

Now we shift our attention to the object detection vertical of our subsystem. In recent years, most of the state of the art systems for object detection involves application of Deep Learning, one of the techniques of Machine Learning, so this week will be devoted to learning some theory behind deep learning, a very popular ML framework called pytorch for building ML models, and using all this to build a model that can recognize hand written digits



First, we begin with learning some theory behind how the Deep Learning architecture works. Some useful resources for the same include:

3blue1brown videos: https://www.youtube.com/watch?v=aircAruvnKk: A playlist of 4 videos that offer a great introduction to the field of Deep Learning.

CS 231n Notes: https://cs231n.github.io/
Note: This is a very detailed but almost sufficent look into how deep learning and Convolutional architecture works. Try to understand the overview of all the topics, do it at your own pace as this is also carried on to the 4th week assignment, once you feel you are comfortable with the idea a bit, you can jump to pytorch documentations and keep coming back to it.

No need to get bogged down by complex calculations, while understanding them is necessary , you won't normally need to work with the formulas,s so focus on the concepts

Video playlist:

Important python prerequisites:
Numpy
Matplotlib

Pytorch Tutorials:
https://pytorch.org/tutorials/beginner/basics/intro.html

Kaggle Courses: A good resource to pick up machine learning in general if you wish to: https://www.kaggle.com/learn

TASK:

Your 3rd task requires you to build any model (recommended CNN model, can also train a fully connected neural network incase of any issues) using the pytorch ML framework for python to classify the MNIST dataset. If you are already comfortable with some other framework, then it should be easy to pick up pytorch anyway.

You can use google colab to utilize GPU for faster training. If not, the script should be written on jupyter notebook either way. It’s pretty easy to use so shouldn’t be a hassle.

Submission

Your final submission should print out the model’s accuracy after each training loop on the test dataset and in the end contain a predict function which takes as input an image and outputs its label, along with a demonstration of the function on 5-6 randomly chosen images from the test-dataset, in which you display the image, it’s original label and it’s predicted label.