EEE/AME 515 HOMEWORK ASSIGNMENT 2

Due on Wednesday, September 27th at 11:59 pm

1 Front Page of the Report

You must use either Latex or Microsoft Word or Jupyter notebooks to typeset your reports. In addition, please present relevant code and figures to explain your method to others. Think about this like a job, you need to explain to your supervisor/boss what you did without dumping a whole bunch of code, answering questions in comments (big NO), and using labeled figures with captions.

Please include a cover page with your name, ASU ID, and indicate which problems have been solved, and which problems need extensions of one month. Please provide a justification for each extension (i.e. "I solved problems a, b, but having difficulty with the getting real data for part c").

Problem 1. *(Image Classification on CIFAR-10)*

For this problem, you will need to use either Google Colab or Sol/Agave server. I highly recommend Sol/Agave if you are planning to use it for your project, as you can practice running code on the server. Click on the ASU Research Computing resources link for instructions how to sign up for an account.

1. This problem will get you used to PyTorch. First complete this tutorial to train a LeNet Style architecture on CIFAR-10. Make sure to train on the GPU, and report the best accuracy you can achieve (you can run for as many epochs as you like). **Tutorial link:** `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#sphx-glr-beginner-blitz-cifar`
   You also can use other PyTorch tutorials to get yourself familiar with the language (highly recommend the MNIST tutorial if its your first time with PyTorch).

2. **Dataset augmentation:** One technique to improve the accuracy of your classifier is to show more diverse data through dataset augmentation. Implement dataset augmentation with rotations, translations, flipping, and adding a small amount of noise to the images. Show some examples of the augmented training data, and then report the performance of this technique on yoru LeNet-style architecture. How much did performance improve?

3. **Resnet:** Implement Residual blocks from the "Resnet" architecture from a well-known paper: `https://arxiv.org/pdf/1512.03385.pdf`. Show that your new architecture with residual connections outperforms the previous network architecture. You can find a version of the Resnet pytorch model here: `https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py`. Report your final performance.

4. We will replicate the experiment of dataset bias featured in `https://arxiv.org/pdf/1911.11834.pdf`. Create two datasets: (1) Take five of the ten classes, and convert all the images in those classes to grayscale, and then stack the same gray image 3 times into the RGB channels. Leave the other five classes the same. (2) Convert all data to grayscale, stack the same gray image 3 times into the RGB channels. Report the accuracy of the previous neural network on Datasets (1) and (2). Which one performs better?

Problem 2. *DC-GAN*

1. Follow the PyTorch tutorial for DCGANs: `https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html`. Try to train your DCGAN to get as best performance as possible. Show the resulting output images of your trained GAN.

2. **New Dataset of Colored Squares:** Create your own dataset of 64x64 images, with black backgrounds and colored squares. The dataset should have squares of different colors, sizes, and orientations. Create a dataset of at least 1000 images. Save these images when you generate them.

3. Train DC-GAN on your dataset of colored squares. How well does it perform? Does it learn to create squares? If not, keep increasing the size of the dataset until it does.

4. **Collect a Dataset of Favorite Animal:** Now pick a favorite animal of yours. Collect a small dataset of 500-1000 images of this animal, resize all the images to the same size of 64 x 64. To make collecting images easier, you can try the following extension: `https://chrome.google.com/webstore/detail/download-all-images/ifipmflagepipjokmbdecpmjbibjnakm?hl=en`. Or any other resources that you can find on the web to download images in bulk.

5. Train DC-GAN on your new dataset. Try various tricks to improve performance (either add more data, dataset augmentation, different learning rates, network architecture/sizes). There's no right answer here, just try to get some performance or document any mode collapse you get.