
Do not cheat. The assignment is for learning purposes and hence just aim to understand the role of different components used in the network while training and testing.

If you are an expert in coding or feel comfortable changing the code in a way to generate an entirely new architecture, you are free to do so. Otherwise, code will be uploaded on the UBLearns which you need to utilize for Q: 1-2.

Q1. Play with the hyper-parameters such as learning rate, batch size, optimizer, and layers to generate good quality images.

Prepare a 5 minutes video explaining what you have observed by changing any or few or all hyper-parameters. Your observation can be in the form of generated images (no need to add/show all, a few best representative images are fine) or anything you think is important such as image quality, time, etc.

Few hyper-parameters locations: Image Normalization: cell 3, Optimizer type, learning, layers: cell 4, and Epochs: cell 7 last line

Q2. Calculate the image quality generated in each variation you have applied in your code and see what impact in terms of image quality you can observe. You can use the following tool for that or anything you want:

<https://pypi.org/project/image-quality/>

Q3. Apply PCA on the CIFAR-10 dataset to reduce the dimensionality of the images and use the reduced dimension data for classification. For classification, you can use any classification architecture you have built so far for any assignment. However, make sure you can compare and contrast the performance of the same architecture when you use the classifier on original data vs. on reduced dimensional data. The comparisons can be done using metrics such as accuracy, errors, training-testing time, etc.

Make sure you are not using a test set in any kind of training.