

SUBJECT: Deep Learning and Computer Vision
SUBJECT CODE: ECS795P

Online Assessment-2

NAME: Ganesh Kumaran Masilamani
STUDENT NO: 200434339

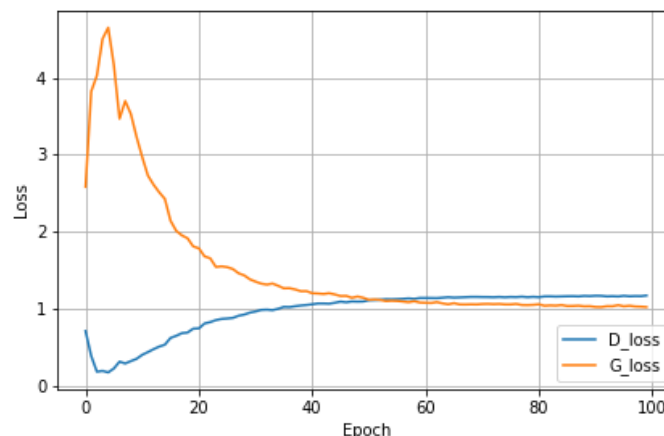
Question to be answered (Q2.4)

How do D-loss and G-loss change during training? Visualize how the D-loss and D-loss change during training and explain why.

ANSWER:

When training the discriminator, D-Loss is generated it classifies both the real images and the generated images from the generator which is otherwise known as fake data . The discriminator penalizes itself for predicting wrongly, a Real image as the fake one which is generated by the generator network. The weights of the discriminator are updated using gradient ascent. Meanwhile, **when training the generator**, The generator samples the latent random noise and produces the output from the latent noise which produces the fake images, Next, the output goes through the discriminator, there the generator tries to fool the discriminator by showing the fake images as real ones, there occurs the minmax game. The weights of the generator are updated using gradient descent. Even in the situation the discriminator predicts real and the fake images based on its network ability. Finally G-loss is then calculated from the classification of the discriminator , generator gets rewarded if it successfully fools the discriminator otherwise it will be penalized if discriminator predicts correctly.

VISUALIZATION OF D-LOSS AND G-LOSS:



From the plot we can infer that D-loss and G-loss gradually converge, at one point both the loss become ideal and fits properly, as the D-loss and G-loss produces a good fit curve.

Exercise to be conducted (E1.4)

Change the training epoch to 200, run it.

CODING:

```
if __name__ == '__main__':
    from six.moves import urllib
    opener = urllib.request.build_opener()
    opener.addheaders = [('User-agent', 'Mozilla/5.0')]
    urllib.request.install_opener(opener)
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    data_dir = './MNIST_data/'
    save_dir = './MNIST_GAN_results/'
    image_save_dir = './MNIST_GAN_results/results'

    # create folder if not exist
    if not os.path.exists(save_dir):
        os.mkdir(save_dir)
    if not os.path.exists(image_save_dir):
        os.mkdir(image_save_dir)

    # training parameters
    batch_size = 100
    learning_rate = 0.0002
    epochs = 200
```

I tired to run 200 epochs but only I can able to reach 100 epochs within this 2 hours of time due to more time consuming for each epoch in google colaboratory which take more than a minute for each epoch. Therefore, I hereby attach my final output produced for 100th epoch.



Epoch 100