IE 643: Assignment #4

Name: Vamsi Krishna Reddy Satti **Roll Number:** 160050064

Notes about Code

- My architecture is same as that of DCGAN and chose the person category for training.
- You can train the model by using the command (assuming `IE643_160050064_assignment4_main.py` is in the current directory). Use the `--checkpoint` flag to restart training from a checkpoint

  ```
  python IE643_160050064_assignment4_main.py --data ../natural_images [--checkpoint .pth]
  ```

  - All logs, outputs and checkpoints will be saved in a directory `outputs/[TIMESTAMP]`.
- A pretrained model `a4_pretrained.pt` is also given for inference and verification. To generate plot and images, run

  ```
  python IE643_160050064_assignment4_plot.py --checkpoint a4-pretrained.pt --output-dir '.'
  ```

  - Due to limitation in size of moodle submission, the pretrained model can be downloaded from this URL: https://www.cse.iitb.ac.in/~vamsikrishna/ie643/a4-pretrained.pt
- The code has been tested to be working on Python 3.7.4 on a Windows 10 machine.

My training procedure

- I had trained the model in two steps since I was having stability issues directly training larger networks which are common while training GANs.
- I had first trained a smaller Discriminator (by removing the last `Conv2d + Batchnorm2d + LeakyReLU` layers from Discriminator) and then trained the Generator. Later, I added this last layer to increase the Discriminator's capacity and then trained it to stable convergence of training.
- That is the reason all the plots show a distinct kink at near 15000 iterations and infact the Discriminator loss decreases further after this increased capacity.

Observations and Answers

1(c)

I update the Generator twice for each update in Discriminator for stable training. I had used the validation set to determine the quality of Discriminator by seeing how well it classifies the validation set images as being real. All the final parameters after tuning can be seen in default argument values in `IE643_160050064_assignment4_main.py`. Note that though I had trained for

quite many iterations in anticipation for improvement, it is indeed not necessary to actually train them for so many epochs in practice.

After my hyperparameters were finally decided I trained the GAN on complete dataset without validation set, which is a very common procedure to do.

1(d), 1(e) and 1(f)

The convergence of GANs is quite different from the other NNs like simple Feed-Forward networks. In case of GANs, both the Generator and Discriminator might be improving even thought the net loss might more or less remain the same. This is due to the fact that Generator loss is affected by improvement in Discriminator's ability and *vice versa*. The plots depict no unusual behavior and is similar to plots of many DCGAN implementations trained across various datasets.

1(g)

On visual inspection, the quality of the generated images is quite good, given a relatively smaller model compared to the state-of-the-art models in recent times. I'm optimistic a slightly more rigorous search of hyperparameters may have lead to slightly improved results, which anyway cannot be quantitatively decided but instead checked through human inspection.