DLCV 2023: Assignment 2
Deadline: 20th March

Instructions

1. Your submission should be a zip file containing the codes (.py or .ipynb), readme.txt file, and your report in pdf format.
2. Please include comments in your code. The readme.txt file should contain information on the organisation of your files, packages used along with their versions, python version, and instructions on running the code.
3. While you are encouraged to go through online repositories to learn best practices and tricks, please avoid directly copying from somewhere.
4. Please submit a report on your observations, results, plots, and analysis in pdf format. This assignment carries 25% weightage for code and 75% weightage for your analysis in the report.

Q1. [**Transformer networks**]
In this question, you will implement a vision transformer based image classification model using pytorch.

a) Implement a basic version of vision transformer (https://arxiv.org/pdf/2010.11929.pdf), that first divides an image into patches and then passes them through a set of multihead self attention modules to perform classification. Please check out the details in the paper.
b) **[Experiment 1]** Train this model on the CIFAR-10 dataset for 10-class classification. Keep the number of attention layers to be 4 for all the experiments.
c) **[Experiment 2]** Try out different patch sizes (like 4x4, 8x8, 16x16). You can divide the image into both overlapping and non-overlapping patches.
d) [Experiment 3] Report model performance by varying the number of attention heads.
e) **[Experiment 4]** Perform classification by using the CLS token from different layers of the model.

Create a detailed report of all the experiments and analyses.

Q2. [**Generative Adversarial Networks**]
Train a Generative Adversarial Network (GAN) to generate images using the AFHQ-Wild dataset. Use a Deep Convolutional GAN (DCGAN) architecture for 64x64 resolution images (ref: https://arxiv.org/pdf/1511.06434.pdf) for

training the model. You can reduce the number of filters/ layers to speed up training.

Note: ***You can download the dataset from https://github.com/clovaai/stargan-v2 and select the AFHQ-dog subset to train for your experiments. Clone the repository and download the data using the command*** `bash download.sh afhq-dataset`.

a. [**Experiment 1**] Plot the generator loss, discriminator loss, and discriminator accuracy. Report your observations on how the GAN training is progressing.
b. [**Experiment 2**] Play with hyperparameters (for GAN training) such as learning rate, learning rate schedule, number of training iterations, optimizer, number of convolutional layers, or any other hyperparameters you can think of! Report your observations (you can select 3 hyperparameters from this list for your analysis).
c. [**Experiment 3**] Implement the mapping network from z-space to w-space from StyleGAN2 (http://arxiv.org/abs/1912.04958). Keep the dimensions same in the z and w-spaces. Note that the mapping network is a 4-layer MLP network. Use AdaIN operation to condition the generator on w-latent code from the StyleGAN2 paper.
d. [**Experiment 4**] Traverse in the latent space of GAN and report your observations and images generated. Sample two latent codes and perform linear interpolation between them. Generate images by the interpolated latent codes and report your observations. Do this experiment for both z and w space.
e. [**Bonus**] Think of changes in architecture to improve either the quality/diversity of generated images. Report your results and present an analysis (irrespective of whether your idea worked or not!).