ASSIGNMENT 2

(100 POINTS) DUE BY MONDAY, JANUARY 18, 2020, 11:30 AM

Assignment Policy:

- No late submission is allowed.
- Collaboration is okay, however, students must submit assignment in their own words. Plagiarism in any form will not be accepted.
- All questions carry equal points.

Question 1 (GENERATIVE ADVERSARIAL NETWORKS (GAN))**.** Let the latent variable, $z \sim U[-1,1]$ be the input to the generator. The real samples are drawn uniformly from a bimodal Gaussian distribution with means, $\mu_1 = (0, 5, 0.5)$ and $\mu_2 = (-0.5. -0.5)$. Let the variance of the Gaussians be same $\sigma = 0.10$. Aim is to learn to generate a bimodal Gaussian.

- Implement a GAN with fully-connected layers for both generator and discriminator. Use the Wasserstein Loss (or Wasserstein Loss with gradient penalty) to train it[1]. After every 500 generator iterations, sample 2000 points from the generator and plot a density estimation plot to see if the GAN learns the multimodal distribution. You can use python's package seaborn for the plot [2]
- Report your hyperparameters, learning strategy and the final density plot at the end of training.

Question 2 (VARIATIONAL AUTOENCODERS (VAE) [1])**.** In this problem, we will design a VAE for learning a low dimensional representation for CIFAR-10 [2] dataset with the following architectures for the encoder, E_ϕ and decoder, D_θ.

TABLE 1. Encoder and Decoder Architectures for CIFAR-10

Encoder, (E_ϕ)	Decoder, (D_θ)
$\boldsymbol{x} \in \mathbb{R}^{32\times32\times3}$	$\boldsymbol{z} \in \mathbb{R}^{128}$
$\rightarrow$ Conv$_{64,4,2}$ $\rightarrow$ ReLU	$\rightarrow$ FC$_{1024}$ $\rightarrow$ BN $\rightarrow$ ReLU
$\rightarrow$ Conv$_{128,4,2}$ $\rightarrow$ BN $\rightarrow$ ReLU	$\rightarrow$ FC$_{8\times8\times128}$ $\rightarrow$ BN $\rightarrow$ ReLU
$\rightarrow$ Flatten $\rightarrow$ FC$_{1024}$ $\rightarrow$ BN $\rightarrow$ ReLU	$\rightarrow$ Reshape$_{8\times8\times128}$
$\rightarrow$ FC$_{128}$	$\rightarrow$ TCONV$_{128,4,2}$ $\rightarrow$ BN $\rightarrow$ ReLU
	$\rightarrow$ TCONV$_{64,4,2}$ $\rightarrow$ BN $\rightarrow$ ELU
	$\rightarrow$ CONV$_{3,4,1}$ $\rightarrow$ Sigmoid

CONV$_{n,k,s}$ denotes a convolutional layer with n kernels of size k and stride size s. TCONV$_{n,k,s}$ denotes a transpose convolutional layer with n kernels of size k and stride size s. FC$_n$ denotes a Fully Connected layer with n neurons. BN denotes a batch normalization layer.

- **Task I - Generation** : Train the model using the standard training split of 50000 samples (without labels). Use to final trained model to compute reconstruction Fréchet Inception Distance (FID) on the test split consisting of 10000 samples. Using the final model compute the generation FID using 10000 generated samples and the test split.

[1] https://github.com/igul222/improved_wgan_trainingcode

[2] https://seaborn.pydata.org/generated/seaborn.kdeplot.html

- **Task II - Classification via Latent Space** : Design a single layer classifier on top of the 64-dimensional latent representation learnt at the output of the encoder. Here you use the learned latent space along with the labels provided in the CIFAR dataset. Train the single layer classifier and report the classification accuracy. Comment on the classification accuracy.
- **Task III - Improved Classification via Latent Space (open-ended)** : Propose an unsupervised modification on top of the VAE designed in problem 1 to improve the classification accuracy. What is the reconstruction FID and generation FID of this modified model? Comment on the findings.

Question 3 (VAE-GAN [3])**.** Implement the proposed model in [3]. Use the same architecture for encoder and decoder as described in Question 2. For the discriminator use the following architecture.

Table 2. Discriminator Architecture for CIFAR-10

Discriminator, (D_κ)
$\boldsymbol{x} \in \mathbb{R}^{32\times32\times3}$
$\rightarrow \mathrm{Conv}_{64,4,2} \rightarrow \mathrm{ReLU}$
$\rightarrow \mathrm{Conv}_{128,4,2} \rightarrow \mathrm{ReLU}$
$\rightarrow \mathrm{Flatten} \rightarrow \mathrm{FC}_{1024} \rightarrow \mathrm{ReLU}$
$\rightarrow \mathrm{FC}_1$

- Compare the reconstruction and generation FID scores of VAE-GAN with the models in the Tasks I of Question 2. Comment on the findings.
- Again as before (Task II of Question 2), design a single layer on top of the encoded representations. Report and compare the classification accuracy. Comment on the findings.
- Apply the modification developed in Task III of Question 2 on top of VAE-GAN and compute the FID scores and classification accuracy. Comment on the findings.

Question 4 (Optimality of the Fixed Prior)**.** In this question, we will argue about optimality or not of the fixed prior, which we discussed in the class, through two toy examples.

- If the true data distribution $p_d()$ is not Gaussian, then under the assumption of Gaussian Decoder, $p_\theta(|) \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$ a Gaussian prior, $q_{\psi_0}() \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, a standard VAE-based model with Gaussian encoder cannot reach the optimum value in ELBO maximization and hence cannot maximize the likelihood.
- If the true data distribution $p_d()$ is Gaussian, then under the assumption of Gaussian Decoder, $p_\theta(|) \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$ and Gaussian Encoder, $q_\phi(|) \sim \mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi)$, an VAE based generative model cannot reach the optimum value in ELBO with a non Gaussian prior $q_\psi()$.

References

[1] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[2] A. Krizhevsky, “Learning multiple layers of features from tiny images,” tech. rep., 2009.

[3] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” in *Proc. of ICML*, 2016.