CS294D: Assignment 4

February 21, 2024

Due: Wednesday March 1st, 11:59pm
Submit by the **blackboard system**

By turning in this assignment, I agree with the KAUST student protocol and declare that all of this is my own work.
Requirements:

- Put all files in a zip file and the file should be named LastnameFirstname_Assignment4.zip
- It is an individual assignment, independent write up, and submission in your own hand is required for credit.
- The work should be written in a **clear** way.
- Late submission will receive credit penalty.

1 Questions (60 points)

Please answer the following questions in 2-4 sentences each:

Q1 (10 points) GDPP [1] targets increasing the diversity of the generated data to match real data. GDPP is based on Determinants Point Processes [2], which is an elegant probabilistic model that measures how representative a selected subset S is for the entire set: $P(S \subseteq Y) \propto det(L_S)$, where L_S is a diversity kernel on the subset S. Given a subset of data S and a feature extraction function $\phi(\cdot)$, GDPP constructs L_S as: $L_S = \Phi(S)^T \Phi(S)$. What does this kernel measure and how it is influenced when the selected subset are similar versus different from one another?

Q2 (10 points) Given the diversity kernel of the real data batch L_{D_B} and the diversity kernal of the fake data batch L_{S_B}, GDPP constructs a diversity magnitude loss $\mathcal{L}_m$ to minimize the difference of the eigenvalues between L_{D_B} and L_{S_B}, and a diversity structure loss $\mathcal{L}_s$ to minimize the difference of eigenvectors between L_{D_B} and L_{S_B} as follows:

$$
\begin{aligned}
\mathcal{L}_m &= \sum_i ||\lambda^i_{real} - \lambda^i_{fake}||_2 \\
\mathcal{L}_s &= -\sum_i \hat{\lambda}^i_{real} \cos(v^i_{real}, v^i_{fake})
\end{aligned}
\tag{1}
$$

where λ^i_{real} and λ^i_{fake} denote the i-th eigenvalue of L_{D_B} and L_{S_B}, respectively. v^i_{real} and v^i_{fake} are the correspond eigenvectors. Why we need these 2 losses $\mathcal{L}_m$ and $\mathcal{L}_s$?

Q3 (10 points) Among the improvements to the original GAN framework, Wasserstein GAN [3] proposes some of the most interesting developments.

1. Explain how Wasserstein GAN resolves the training instability issues faced by vanilla GANs related to mode collapse.

2. What additional assumption is made over the discriminator function? Explain why it is important in the context of enforcing a smooth training process.

Q4 (10 points) Unlike normal GAN models that can only generate images randomly, StyleGAN [4] provides control over the style of the generated images. One of the most important components in StyleGAN [4] is the Adaptive Instance Normalization (AdaIN). What is the key difference between AdaIN and the standard Instance Normalization (IN) [5]? How is the mapping network designed and how does it influence AdaIN operations?

Q5 (10 points) Normalizing flows [6] aim to approximate intricate distributions by composing a series of invertible learnable functions. Give the theoretical basis behind their principle and explain briefly.

Q6 (10 points) GANs have demonstrated their proficiency at generating novel, realistic and diverse samples matching a training distribution. But can they be used to edit an input (non-generated) image, *without* being explicitly trained for it? Explain how a trained GAN model could be used to edit a photo. What limitations would such a method have? (*Hint:* [7])

2 Coding Assignment: Flow-based models (40 points)

Please refer to the attached Jupyter notebook.

Bonus. By fine-tuning a random z latent using regular backpropagation, show how real input images (sampled from MNIST) can be embedded into the latent space of your trained GAN model from HMW3. Implement and showcase a few examples.

References

[1] Mohamed Elfeki, Camille Couprie, Morgane Riviere, and Mohamed Elhoseiny. Gdpp: Learning diverse generations using determinantal point process, 2019.

[2] Alex Kulesza. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[5] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017.

[6] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, November 2021.

[7] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space?, 2019.