CS283: Assignment 4

February 17, 2022

Due: Sunday February 27th, 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) High-resolution image generation is a difficult task for GAN model. As an effective solution to this task, Progressive GAN [3] is trained to generate low-resolution images first as an easier task. Then, the generation resolution is gradually progressively increased through the whole training process. What is the purpose of the"toRGB", "fromRGB", and the α parameter in Progressive GAN (Hint: see Fig 2 in the paper)?

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 is the Adaptive Instance Normalization (AdaIN). What is the key difference between the AdaIN, and the standard Instance Normalization (IN) [5]? How is the mapping Network and how does it influence AdaIN operations?

Q5 (10 points) In StyleGAN2 [6], the authors revise the architecture of the original StyleGAN to obtain better performance. List 3 architecture changes here.

Q6 (10 points) Data Efficient GAN [7] utilizes image augmentation to support training with fewer image data. The authors show that augmenting the real images only or Discriminator only do not perform well. Explain why.

2 Coding Assignment (40 points)

In this coding assignment, you are requested to implement the Conditional GAN to make your model generates some MNIST images given the numbers you want. In detail, you need to implement the generator and the discriminator part. You will notice that the framework here is quite similar to assignment 3. But now, both the generator and the discriminator additionally take the numbers you want as input. You can build upon your last implementation. Fill the code in the file model.py and train the model. The entrance of this project is the file GAN.ipynb. The training loop can be found in train.py

References

[1] Mohamed Elfeki, Camille Couprie, Morgane Riviere, and Mohamed Elhoseiny. Gdpp: Learning diverse generations using determinantal point processes. In *International Conference on Machine Learning*, pages 1774–1783. PMLR, 2019.

[2] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[5] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[6] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[7] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.