CS283: Assignment 1

January 24, 2022

Due: Sunday February 6th, 11:59pm
Submit by the **blakcboard 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_Assignment1.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 Problem 1: Generative adversarial networks (60 points)

In this problem, suppose that we will implement a generative adversarial network (GAN) that models a high-dimensional data distribution $p_{data}(x)$, where $x \in \mathbb{R}^n$. To do so, we will define a generator $G_\theta : \mathbb{R}^k \to \mathbb{R}^n$; we obtain samples from our model by first sampling a k-dimensional random vector $z \sim \mathcal{N}(0, I)$ and then returning $G_\theta(z)$.
We will also define a discriminator $D_\phi : \mathbb{R}^n \to (0, 1)$ that judges how realistic the generated images $G_\theta(z)$ are, compared to samples from the data distribution $x \sim p_{data}(x)$. Because its output is intended to be interpreted as a probability, the last layer of the discriminattor is frequently the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

There are several common variants of the loss functions used to train a generative adversarial network (GAN). They can all be described as a procedure where we alternately perform a gradient descent step on $L_D(\phi;\theta)$ with respect to ϕ to train the discriminator D_ϕ, and a gradient descent step on $L_G(\theta;\phi)$ with respect to θ to train the generator G_θ:

$$\underset{\phi}{min}\, L_D(\phi;\theta) \qquad\qquad \underset{\theta}{min}\, L_G(\phi;\theta)$$

In our lecture, we talked about the following losses, where the discriminator's loss is given by:

$$L_D(\phi;\theta) = -\mathbb{E}_{x\sim p_{data}(x)}[\log D_\phi(x)] - \mathbb{E}_{z\sim\mathcal{N}(0,I)}[\log(1 - D_\phi(G_\theta(z)))]$$

and the generator's loss is given by the **minimax loss**:

$$L_G^{minimax}(\theta;\phi) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[\log(1 - D_\phi(G_\theta(z)))]$$

1. **(20 points)** the minimax loss for L_G suffers from **vanishing gradient** problem. In terms of the discriminator's logits, the **minimax loss** is

$$L_G^{minimax}(\theta;\phi) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[\log(1 - \sigma(h_\phi(G_\theta(z))))]$$

Show that the derivative of $L_G^{minimax}$ with respect to θ is approximately 0 if $D(G_\theta(z)) \approx 0$, or equivalently, if $h_\phi(G_\theta(z)) \ll 0$. You may use the fact that $\sigma'(x) = \sigma(x)(1-\sigma(x))$. Why is this problematic for the training of the generator when the discriminator successfully identifies a fake sample $G_\theta(z)$?

2. **(20 points)** To solve this vanishing gradient problem, we usually replace $L_G^{minimax}$ with other loss functions such as non-saturating loss L_G^{nsgan} [1] and more other forms of loss functions can be found in [2] . You may plot different loss functions including minimax loss and non-saturating loss to show the contrast. You also need to explain why non-saturating loss can avoid vanishing gradient problem.

$$L_G^{nsgan}(\theta;\phi) = -\mathbb{E}_{z\sim\mathbb{N}(0,I)}[\log D_\phi(G_\theta(z))]$$

3. **(20 points)** Coding practice: You are going to fill out the missing code blocks in ***CS283_homework1_GAN.ipynb*** and build a GAN model to convert 1D uniform noise distribution to a uniform 1D normal data distribution. The ***readme.txt*** is provided for environment setup.

2 KL-Divergence (30 points)

1. **(30 points)** Let $\mu : \mathcal{X} \to \mathbb{R}^k$, $\sigma : \mathcal{X} \to \mathbb{R}^k$ and let $q(z|x) = \mathcal{N}(z; \mu(x), (\sigma^2(x)))$. Suppose that $p(z) = \mathcal{N}(z; 0, I)$. Show that

$$D_{KL}\left(q\left(z|x\right) \|\, p(z)\right) = \tfrac{1}{2}\textstyle\sum_{i=1}^{k}(\sigma_i^2(x) + \mu(x)_i^2 - 1 - \log\sigma_i^2(x))$$

Note that the KL divergence between two distributions P and Q is defined as

$$D_{KL}\left(p(x)\|\, q(x)\right) = \textstyle\int p(x)\log(\frac{p(x)}{q(x)})\,\mathrm{d}x.$$

3 Creative Adversarial Networks (20 points)

In Creative Adversarial Netwoks (CANs)[3], an additional loss is added to encourage producing novel images with the loss defined as

$$\begin{aligned}&\min_G \max_D V(D,G) = \\ &\quad \mathbb{E}_{x,\hat{c}\sim p_{data}}[\log D_r(x) + \log D_c(c=\hat{c}|x)] + \\ &\quad \mathbb{E}_{z\sim p_z}[\log(1-D_r(G(z))) - \sum_{k=1}^{K}\big(\frac{1}{K}\log(D_c(c_k|G(z))+ \\ &\qquad\qquad\qquad (1-\frac{1}{K})\log(1-D_c(c_k|G(z)))], \end{aligned} \tag{1}$$

where x and $\hat{c}$ are a real image and its corresponding art style label from the data distribution p_{data}. $D_r(\cdot)$ is the transformation function that tries to discriminate between real art and generated images. $D_c(\cdot)$ is the the function that discriminates between different style categories and estimates the style class posteriors (i.e., $D_c(c_k|\cdot) = p(c_k|\cdot)$).

1. **(10 points)** Comment on the use of Binary Cross Entropy(BCE) in the deviation loss summed up over each style class (i.e. terms in $\sum_{k=1}^{K}(\cdots)$ above). What part of the loss function enables the model not to produce meaningless art images since they may also have high entropy.

2. **(10 points) Bonus*** Why Multiclass Cross Entropy improved the performance over BCE in Holistic-CAN[4] (CAN-H)?

References

[1] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets.

[2] Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Are gans created equal? a large-scale study. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 698–707, 2018.

[3] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*, 2017.

[4] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. Design: Design inspiration from generative networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.