Advanced Deep Learning
Homework 2

Abbas Nosrat
810199294

April 23, 2022

Contents

1 Generative Adversarial Networks

1.1 Stage 1

- The loss function used in cyclegan is consisted of three parts:
  - **Adversarial Loss**: This loss is formulated as

    $$\mathcal{L}_{\mathcal{GAN}} = \mathbb{E}_{y \sim P_{data}(y)} \left[(D(y) - 1)^2 \right] + \mathbb{E}_{X \sim P_{data}(X)} \left[(D(G(X)))^2 \right]$$

    This loss is applied to the output of the discriminator and it is basically a classification loss which gives a hard 1 label to the real images and a hard zero label to the generated images which tries to train the discriminator to distinguish between fake and real images. Optimizing this loss for the generator, trains the generator to generate images that will increase the discriminator loss .This loss is responsible for the adversarial nature of cyclegan and without this loss, the generators and discriminators cannot be trained. There are two $\mathcal{L}_{\mathcal{GAN}}$ terms corresponding to the two pairs of generator and discriminators.
  - **Cycle Consistency Loss:** This loss is and L1 loss between the input image and the output of the second generator. It tries to learn the networks to return the image to its original state after passing it through both generators. In other words

    $$G_2(G_1(X)) \approx X \Rightarrow \mathcal{L}_{cycle} = \|X - G_2(G_1(X))\|$$

    The general idea of GANs is that the network can learn any mapping from the input domain to the target domain. However, the network may learn to change the spatial properties of the image such that the output image looks nothing like the input image. In order to prevent such phenomena, cycle consistency loss is utilized. Utilization of this loss, enforces the network to only apply the minimum change required to go from the input domain to the target domain. By removing this term from the objective function, the network may learn to change the image too much.
  - **Identity Loss:** This loss ensures that if an image from the target domain is given as input to the generator, the image remains unchanged. This loss has no positive effect on training of the networks and in some cases it is better to be removed.
- The authors of cycleGAN used MSE loss instead of cross entropy due to training stability. Optimizing MSE loss is more stable than BCE and would result in generation of better quality images.
- If an objective is consisted of multiple terms, it is generally represented as a weighted sum. By increasing the weight for the the more important terms and/or decreasing for the less important ones, one can ensure better training and fulfillment of the desired objective.

- Yes. The generator uses a hierarchical architecture. It reduces dimensions at first and then increases them to match the target image. Any CNN architecture following the same pattern can be used. A good example of this type of networks can be UNET. For the discriminator part, any classifier architecture can be used since it is a classification problem. However, the results may be worse than PatchGan. Another approach can be the use of Siamese networks. The objective can be reformulated. In the new formulation, Instead of classification loss for the discriminator, Triplet loss or Contrastive loss can be used. By changing the loss in this way, the discriminator tries to create distance between real and fake images and the generator tries to get its images among the real images.

1.2 Stage 2

- The model has been trained for 20 epochs due to lack of computational power and time. The identity loss was removed to reduce computations and save time. The results are demonstrated in figure 1:

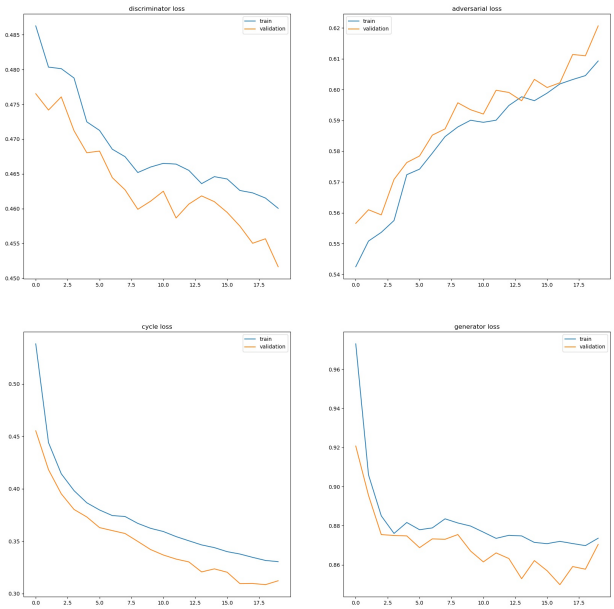

Figure 1: Training logs for CycleGan

As witnessed in Figure 1, the adversarial loss is increasing but all other

losses are decreasing. The increase in adversarial loss can have two possible causes:

1. The Generator may be too powerful compared to discriminator.
2. It may be due to the stochastic nature of training and further training may remedy this problem.

- Here are a few images generated by the model: As demonstrated in Figure

(a) Generated images from A dataset(horse to zebra)

(b) Generated images from B dataset(zebra to horse)

Figure 2: generated images by model

2, the model has better learned to turn horses to zebras than vice versa. This can be fixed with further training.

2 Variational Auto Encoder

2.1 Stage 1

- if the decoder is sufficiently powerful, then the training objective can be solved with a dumb strategy: the encoder always produces $p(z)$ regardless of the data, and the decoder always produces $p(x)$ regardless of z. The paper Neural Discrete Representation Learning (from van den Oord and others from deepmind) calls this "posterior collapse".
  Due to posterior collapse, VAEs cannot be controled since the decoder ignores the latent space. This results in the VAE only producing a limited set of outputs and not being able to generalize

- In a GAN models, the generator may be able to learn one or a few pluseble outputs which can decive the discriminator. However this is not desirable. This phenomena can be remedied if the discriminator learns to distinguish between real and fake inputs. However if the discriminator gets stuck in a local minimum, it will not be able to solve the loophole that the generator has discoverd and thus resulting in the GAN only generating a few different output images. This phenomena is called “mode collapse”. The difference between “model collapse” and “posterior collapse” is that the former is cauesed due to under parameterized discriminator but the latter is caused by over parameterized decoder.

2.2 Stage 2

- The VQ-VAE loss is

$$L = log(p(x|z_q(x)) + \|sg\left[z_e(x)\right] - e\|_2^2 + \beta \left\|z_e(x) - sg[e]\right\|_2^2$$

  The first term is the reconstruction loss which basically can be simplified to an L2 loss between the input and the output assuming a Gaussian data likelihood. This term is responsible for training the encoder and the decoder(it is the auto encoder loss) and by removing this term the encoder and the decoder cannot be trained.
  The second term is pushing the code-book vectors to their corresponding encoded vectors. SG is an abbreviation for stop gradient which means detaching the argument from the computational graph. The embeddings optimize this term hence by removing this part, training of the embeddings is disrupted.
  The last term pushes the encoded vectors to the code-book vectors(the conjugate operation of the second term somehow). This term is a part of training loss for the encoder and removing it will negatively effect the training of the encoder part.

- Auto-regressive models such as pixelCNN, generate outputs by utilizing their previous outputs. For example, pixelCNN uses a receptive field to measure an estimate of the distribution of surrounding generated/given

pixels and generates the most likely pixel based on the estimated distribution. In case of VQ-VAE, a pixelCNN generates the index of code book vectors and after completion of the matrix, VQ-VAE gets the corresponding vectors from the code book and generates the novel image.

- The prior distribution $p(z)$ is a uniform distribution and the posterior distribution $p(z|x)$ is a single vector. This is due to discretization of the posterior with the code-book vectors. If KL divergence between the prior and the posterior is computed, the result would be

$$1 \times \log\left(\frac{1}{\frac{1}{k}}\right) = \log(k)$$

k is the length of the prior distribution.

2.3 Stage 3

- Figure 3 demonstrates training logs for 40 epochs: It can be seen that

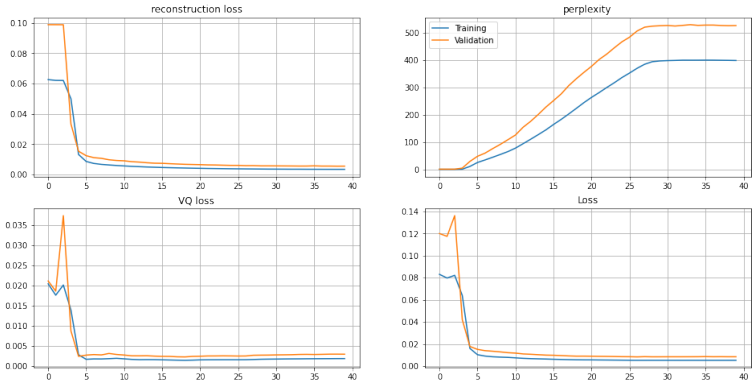

Figure 3: Training logs for VQ-VAE

both reconstruction loss and VQ loss are decreasing and the perplexity, which is the utilization rate of the code-book vectors, is increasing. after 40 epochs everything will converge. Hence, 40 epochs was enough training. A UMAP abstraction of the embedding vectors is illustrated in Figure 4:

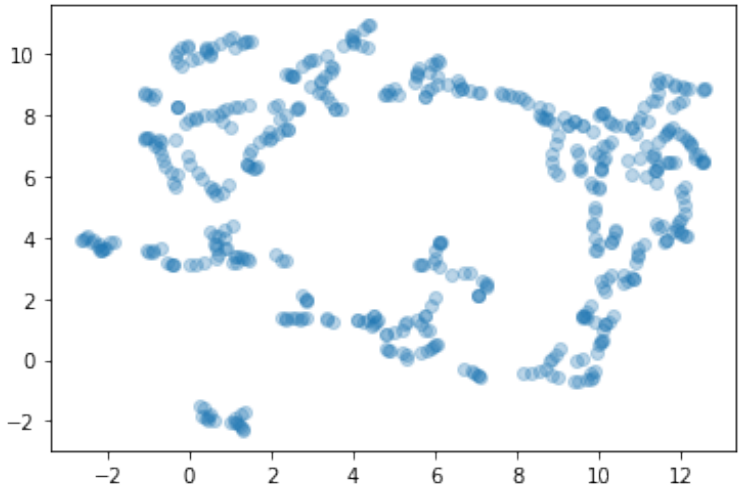

Figure 4: UMAP representation of embedding vectors

As clearly demonstrated in figure 4, there is a nice separation in the embedding space which is a result of vector quantization.

- Some reconstructed images are presented in Figure 5:

(a) Original (b) Reconstructed

Figure 5: Original vs reconstructed images from validation set