CS5787: Exercises 3

`https://github.com/eminalparslan/CS-5787-Assignments/tree/main`

Mingshu Liu
ml2837

Emin Arslan
eaa84

1 Theory: Question 1 [40 pts]

(a) When $\theta \neq 0$

- For KL divergence: since there is no overlap between p and q, the distance will be $D_{KL}(p||q) = \infty$ since the log in the function is not defined here.
- For JS divergence:

$$\begin{aligned} D_{JS}(p||q) &= \frac{1}{2}D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q||\frac{p+q}{2}) \\ &= \frac{1}{2}\int p(x) \cdot log(\frac{p(x)}{\frac{p(x)}{2}}) + \frac{1}{2}\int q(x) \cdot log(\frac{q(x)}{\frac{q(x)}{2}}) \\ &= \frac{1}{2}\int p(x) \cdot log(2) + \frac{1}{2}\int q(x) \cdot log(2) \\ &= log(2) \end{aligned}$$

- For Wasserstein Distance: Every point on p must be moved from 0 to θ, so the minimum distance will be θ and therefore, $W(p,q) = \theta$

(b) When $\theta = 0$

- For KL divergence: Since $\theta = 0$, we can tell p(x) = q(x) and $D_{KL}(p||q) = 0$
- For JS divergence: $\frac{1}{2}D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q||\frac{p+q}{2}) = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 0 = 0$
- For Wasserstein Distance: We don't need to move since the two distributions are identical, so $W(p,q) = 0$

(c) KL divergence becomes infinite when the distributions P and Q are disjoint, and JS divergence will reach its maximum value in this case. However, Wasserstein distance remains well-defined for disjoint distributions and provides a meaningful measure of the distance between them. Meanwhile, the Wasserstein distance considers the shape and the location of the distributions while the KL and JS divergence primarily focus on differences in probability mass at specific points without considering the cost of moving mass between points.

2 Theory: Question 2 [10 pts]

For the LSTM model, we will calculate the sequence sequentially, which means the model will compute the hidden state based on the previous hidden state and current input, so the total calculation needed is $N \cdot d^2$, where d is the dimension of the hidden state.
For the transformer model, the model will process the entire sequence in parallel order. The system will compute the output and input which will be N^2, and there will be several layers with each layer in a feedforward neural network in addition to the self-attention, so the total complexity will be $N^2 \cdot d \cdot L$
LSTM is best for short to moderate sequence lengths and moderate dimensions. The transformer model is best for parallelization is beneficial and the model dimension is large, but can be computationally intensive for every long sequence due to quadratic scaling on N.,

3 Practical: Question 3 [5 pts]

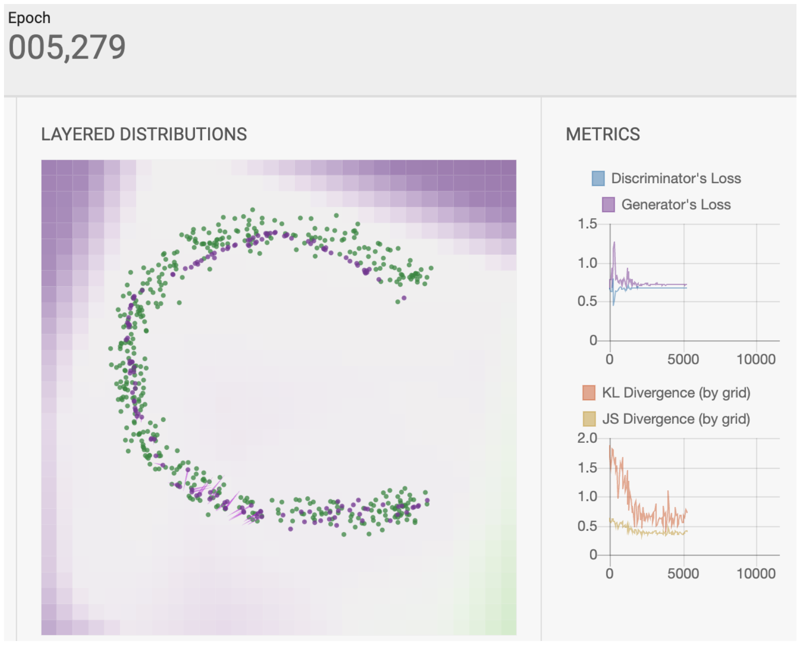


Figure 1: Trained GAN converged after 5000 epochs.

4 Practical: Question 4 [20 pts]

We trained a VAE to extract features from the data and produce a vector for each image in the learned latent space. We then used a SVM classifier with the RBF (Radial basis function) filter to classify each data point's latent space representation. We trained the SVC on sets of 100, 600, 1000, and 3000 labeled data points. The results are in the table below

N	Accuracy
100	0.6500
600	0.8667
1000	0.8800
3000	0.9317

5 Practical: Question 5 [25 pts]

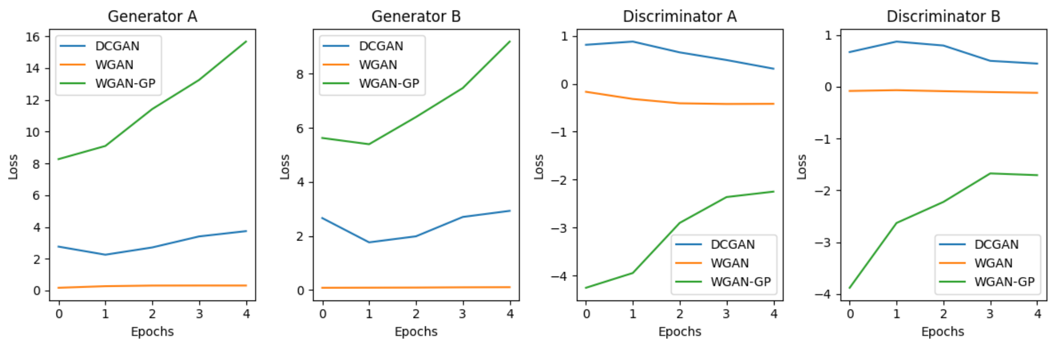
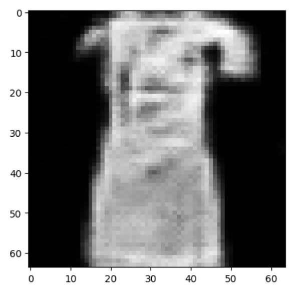
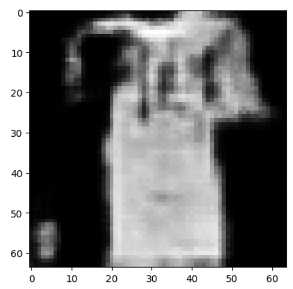
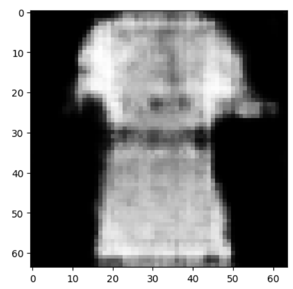
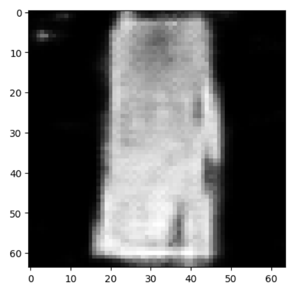


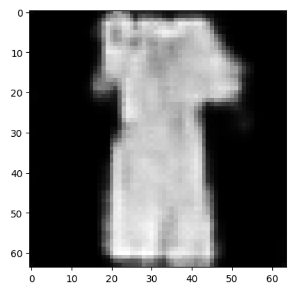
Figure 2: Plots for loss for each type of GAN for architectures A and B.


(a) DCGAN - shirt


(b) WGAN - shirt


(c) WGAN-GP - shirt


(d) DCGAN - dress


(e) WGAN - dress

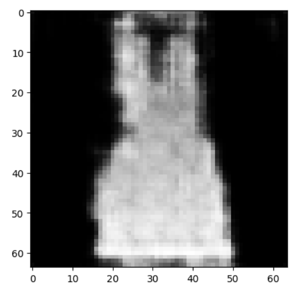
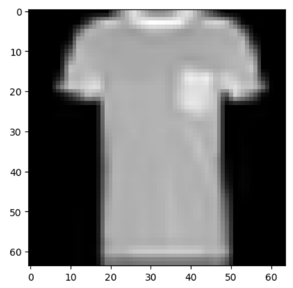

(f) WGAN-GP - dress

Figure 3: Outputs for each GAN architecture.


(a) shirt

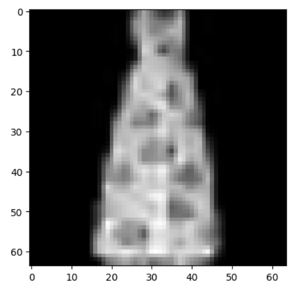

(b) dress

Figure 4: Images from original dataset.

The base architecture and hyperparameters are the following: latent dimension size of 100, learning rate of 2e-4, the generator has 4 layers of conv-transpose, batch norm (layer norm for WGAN-GP), and ReLU activation, and the discriminator has 4 layers of conv, batch norm, and LeakyRelu. The two architectures we chose differ in their base filter count: architecture A has 64 base filters in the conv layers while architecture B has 32.

We had to run DCGAN multiple times (2-3 runs at least) to get a successful model that converged properly. WGAN and WGAN-GP did not have this problem. WGAN-GP's loss went up as training continued, but nevertheless produced better results. This might be because the discriminator was able to get better at distinguishing between the fake and real images, passing on this information to the generator.

Observing the outputs, both DCGAN and WGAN would confuse the difference between shirts and dresses, where the dresses might have features similar to those of shirts and vice versa. WGAN-GP is the only model that output dresses that looked liked dresses consistently.