Deep learning- Assignment 3

Tamar Mano 206295917
Orgad Shlishman 303142897

Theoretical part

1. Suppose we have two 2-dimensional probability distributions, P and Q:

- $\forall(\boldsymbol{x},\boldsymbol{y}) \in \boldsymbol{P}, \boldsymbol{x} = \mathbf{0}\ \boldsymbol{and}\ \boldsymbol{y}\sim\boldsymbol{U}(\mathbf{0},\mathbf{1})$
- $\forall(\boldsymbol{x},\boldsymbol{y}) \in \boldsymbol{Q}, \boldsymbol{x} = \boldsymbol{\theta},\ \mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1}, \boldsymbol{and}\ \boldsymbol{y}\sim\boldsymbol{U}(\mathbf{0},\mathbf{1})$

θ is a given constant. Obviously, when $\theta \neq 0$, there is no overlap between P and Q.

a. For the case where $\theta \neq 0$, calculate the distance between P and Q by each of the three measurements:

i. KL:

$$D_{KL}(p||q) = \int_{x\epsilon X}\int_{y\in Y} p(x,y)\cdot\log\frac{p(x,y)}{q(x,y)}dx\,dy =$$
$$= \int_{y=0}^{1} p(0,y)\cdot\log\frac{p(0,y)}{q(0,y)}dy + \int_{y=0}^{1} p(\theta,y)\cdot\log\frac{p(\theta,y)}{q(\theta,y)}dy =$$
$$= \int_{y=0}^{1} 1\cdot\log\frac{1}{0}dy + \overbrace{\int_{y=0}^{1} 0\cdot\log\frac{0}{1}dy}^{=0} = \int_{y=0}^{1} 1\cdot\log\frac{1}{0}dy = \infty$$
$$\boxed{D_{KL}(p||q) = \infty}$$

ii. JS:

$$D_{KL}(p||\frac{p+q}{2}) = \int_{x\epsilon X}\int_{y\in Y} p(x,y)\cdot\log\frac{p(x,y)}{\frac{p(x,y)+\ q(x,y)}{2}}dx\,dy =$$
$$= \int_{x\epsilon X}\int_{y\in Y} p(x,y)\cdot\log\frac{2\cdot p(x,y)}{p(x,y)+\ q(x,y)}dx\,dy =$$
$$= \int_{y=0}^{1} p(0,y)\cdot\log\frac{2\cdot\ p(0,y)}{p(0,y)+\ q(0,y)}dy + \int_{y=0}^{1} p(\theta,y)\cdot\log\frac{2\cdot\ p(\theta,y)}{p(\theta,y)+\ q(\theta,y)}dy$$
$$=$$
$$= \int_{y=0}^{1} 1\cdot\log\frac{2\cdot 1}{1+\ 0}dy + \overbrace{\int_{y=0}^{1} 0\cdot\log\frac{2\cdot\ 0}{0+\ 1}dy}^{=0} = \int_{y=0}^{1}\log 2\,dy = \log 2$$

$$D_{KL}(q||\frac{p+q}{2}) = \int_{x\epsilon X}\int_{y\in Y} q(x,y)\cdot\log\frac{q(x,y)}{\frac{p(x,y)+\ q(x,y)}{2}}dx\,dy =$$
$$= \int_{x\epsilon X}\int_{y\in Y} q(x,y)\cdot\log\frac{2\cdot q(x,y)}{p(x,y)+\ q(x,y)}dx\,dy =$$
$$= \int_{y=0}^{1} q(0,y)\cdot\log\frac{2\cdot\ q(0,y)}{p(0,y)+\ q(0,y)}dy + \int_{y=0}^{1} q(\theta,y)\cdot\log\frac{2\cdot\ q(\theta,y)}{p(\theta,y)+\ q(\theta,y)}dy$$
$$=$$

$$= \overbrace{\int_{y=0}^{1} 0 \cdot \log\frac{2 \cdot 0}{1 + 0} dy}^{=0} + \int_{y=0}^{1} 1 \cdot \log\frac{2 \cdot 1}{0 + 1} dy = \int_{y=0}^{1} \log 2 \, dy = \log 2$$

$$D_{JS}(p||q) = \frac{1}{2} D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2} D_{KL}(q||\frac{p+q}{2}) =$$

$$= \frac{1}{2}\log 2 + \frac{1}{2} log2 = log2$$

$$\boxed{D_{JS}(p||q) = log2}$$

iii. Wasserstein Distance:

$W(p,q) = \inf_{\gamma \sim \pi(p,q)} E_{(x,y)\sim\gamma} [\, \|x - y\|]$.

We first note that x, y here has different meaning than the x, y of before.

A better formula with the same x-y meaning like before is:

$W(p,q) = \inf_{\gamma \sim \pi(p,q)} E_{(P(x,y),Q(x,y))\sim\gamma} [\, \|P(x,y) - Q(x,y)\|]$

this distance measures the minimal cost of "moving" P to become

Q from the probability distribution graph above we see that the

$\|P(x,y) - Q(x,y)\| = \|\theta\|$, thus $W(p,q) = \inf_{\gamma \sim \pi(p,q)} E_{(x,y)\sim\gamma} [\, \|\theta\| \,] = \|\theta\|$.

We can lose the absolute value since ($0 \leq \theta \leq 1$). Hence $\boxed{W(p,q) = \theta}$

b. Repeat section "a" for the case where $\theta = 0$

i. KL:

Now, p(x, y)=q(x, y). Therefore,

$$D_{KL}(p||q) = \int_{x\epsilon X}\int_{y\in Y} p(x,y) \cdot \log\frac{p(x,y)}{q(x,y)} dx\, dy = \int_{x\epsilon X}\int_{y\in Y} p(x,y) \cdot \log 1 \, dx\, dy$$
$$= 0$$

$$\boxed{D_{KL}(p||q) = 0}$$

ii. JS:

Now, p(x, y)=q(x, y). Therefore $\frac{p+q}{2} = \mathrm{p(x,y)} = q(x,y)$ and then,

$$D_{KL}(p||\frac{p+q}{2}) = D_{KL}(p||p) = \int_{x\epsilon X}\int_{y\in Y} p(x,y) \cdot \log\frac{p(x,y)}{p(x,y)} dx\, dy = 0$$
$$D_{KL}(q||\frac{p+q}{2}) = D_{KL}(q||q) = \int_{x\epsilon X}\int_{y\in Y} q(x,y) \cdot \log\frac{q(x,y)}{q(x,y)} dx\, dy = 0$$

$$D_{JS}(p||q) = \frac{1}{2}D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q||\frac{p+q}{2}) = 0$$

$$\boxed{D_{JS}(p||q) = 0}$$

iii. Wasserstein Distance:

Now, p(x,y)=q(x,y). Therefore, $\|P(x,y) - Q(x,y)\| = 0$ hence,

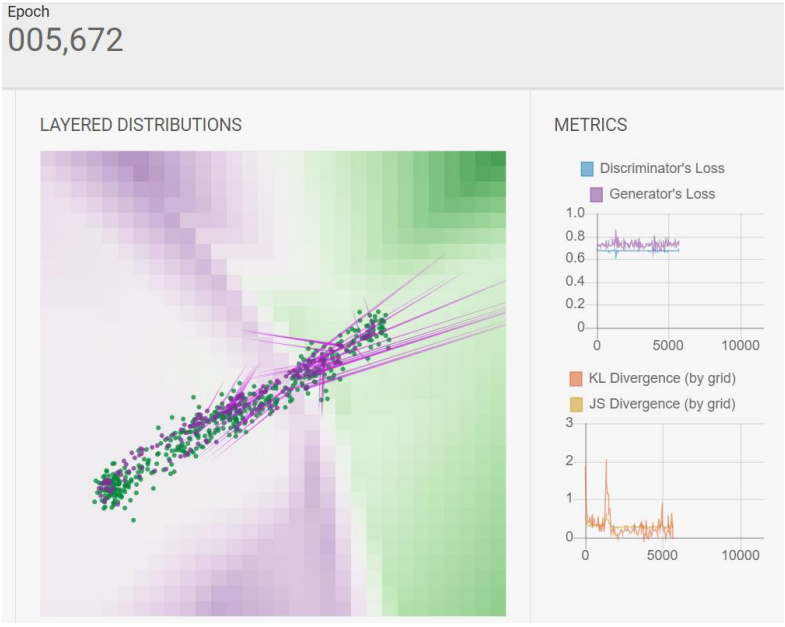
$W(p,q) = \inf_{\gamma \sim \pi(p,q)} E_{(x,y)\sim\gamma}\,[\,0\,] = 0$

$$\boxed{W(p,q) = 0}$$

c. Following your answers, what is the advantage of Wasserstein Distance over the previous two?

<u>Wasserstein over KL divergence:</u>
KL divergence is not symmetric (meaning $D_{KL}(p||q) \neq D_{KL}(q||p)$ for $\neq q$), where Wasserstein is ($W(p,q) = W(q,p)$ for $p \neq q$). In addition, in section we saw that sometimes the KL divergence we got was infinity distance in case of disjoint distribution, a far way value from the real distance, hence KL divergence is not a good metric to choose.
<u>Wasserstein over JS divergence:</u>
Although JS divergence is symmetric metric, it is still fails to provide a meaningful value when the two distributions as disjoint. In addition looking at the JS distance we got for $\theta \neq 0$ and for $\theta = 0$, there is a sudden jump in the JS distance value (from log 2 to 0), not differentiable at $\theta = 0$, which makes it hard for the gradient descents algorithm stability, thus, making it hard to train.

2. GAN Lab results:



In the figure above, we can see in purple the fake samples, and in green, the real samples. We can see in the visualization that the samples are close. Also, on the KL/JS Divergence graph, we can observe small values, after ~5,000 epochs, which indicates that the similarity between the probability distributions of Real & Fake samples is high.

The long gradients, indicate that update process of the weights, to help the model converge is taking big steps, which leads to better performance.

The discriminator's loss represents how well the discriminator is performing in distinguishing between real and fake samples. A lower discriminator loss indicates that the discriminator is becoming more effective at distinguishing between real and fake samples. So, in our case, the discriminator can still improve.

Similarly, the generator's loss measures how well the generator is performing in generating samples that can fool the discriminator. So, here too, the generator can improve and become better.

Practical part

3. In this question, we trained a VAE network using the fashionMNIST training data set. We then used the decoder output (the latent variable representation) on the test dataset as features to train an SVC model. The SVC model uses an RBF (radial basis function) kernel and a set seed for reproducibility.

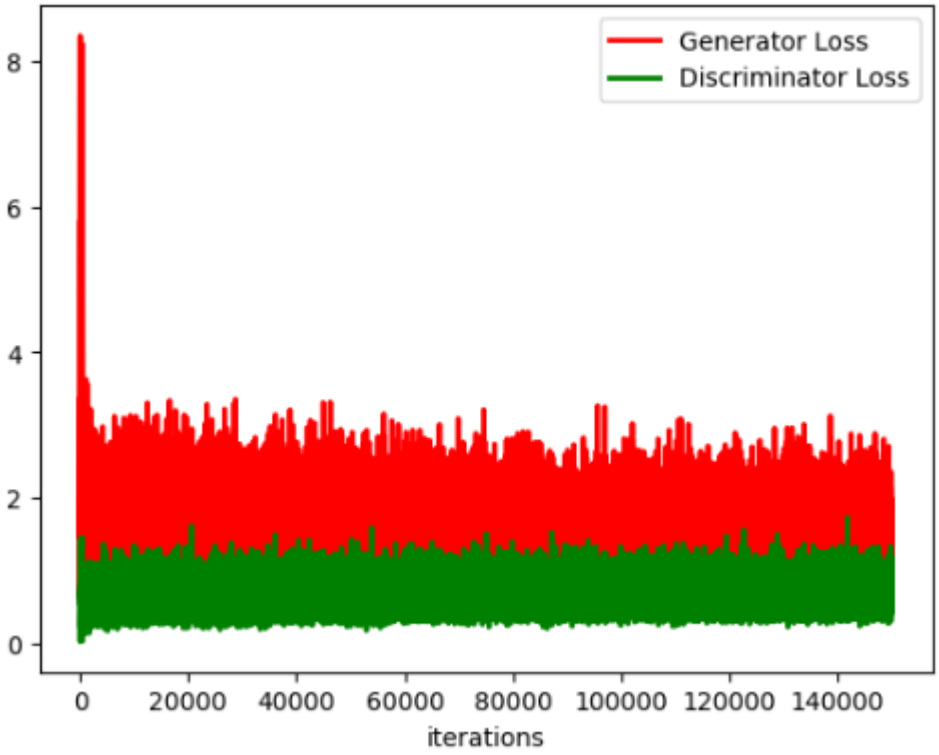
We trained the network for 15 epochs using a learning rate of 0.001. We tested the outputs of the VAE for various labels:

Although information is clearly missing from some of the images, they resemble their label to a good extent. At this point in training, we used the latent variable to train the SVM model. The SVM model results for each training size:

	sample_size	accuracy
0	100	0.570
1	600	0.780
2	1000	0.824
3	3000	0.843

The accuracy of the test data improves when we use a larger sample size, as is expected. However, the accuracy even with just 600 samples is 78%, which shows that the latent variable representation allows for strong classification using a very small amount of data.

4. In this question we've implemented DCGAN & WGAN based on "Improved Training of Wasserstein GANs" (Can be seen here: https://arxiv.org/pdf/1704.00028.pdf) and modified it so it could work with FashionMNIST dataset.
    a. DCGAN
        i. Loss function as a function of the iterations for training



        ii. Real images:

iii. Fake images – we will present fake images along the training (every 5 epochs), so the process of improving the generator could be seen:

- Epoch 1

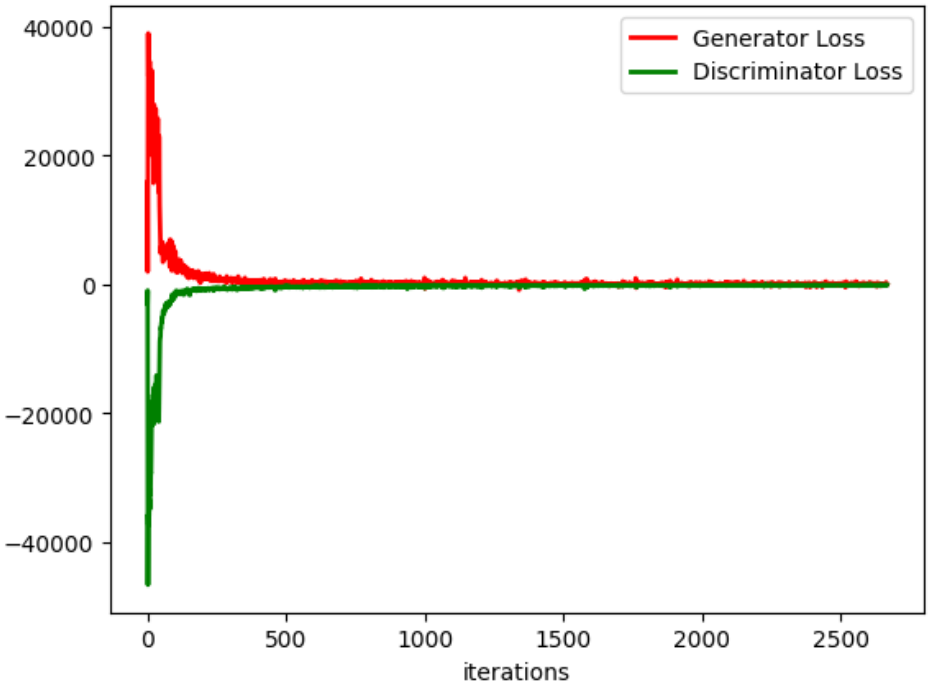
- Epoch 5

- Epoch 10

- Epoch 15

b. WGAN

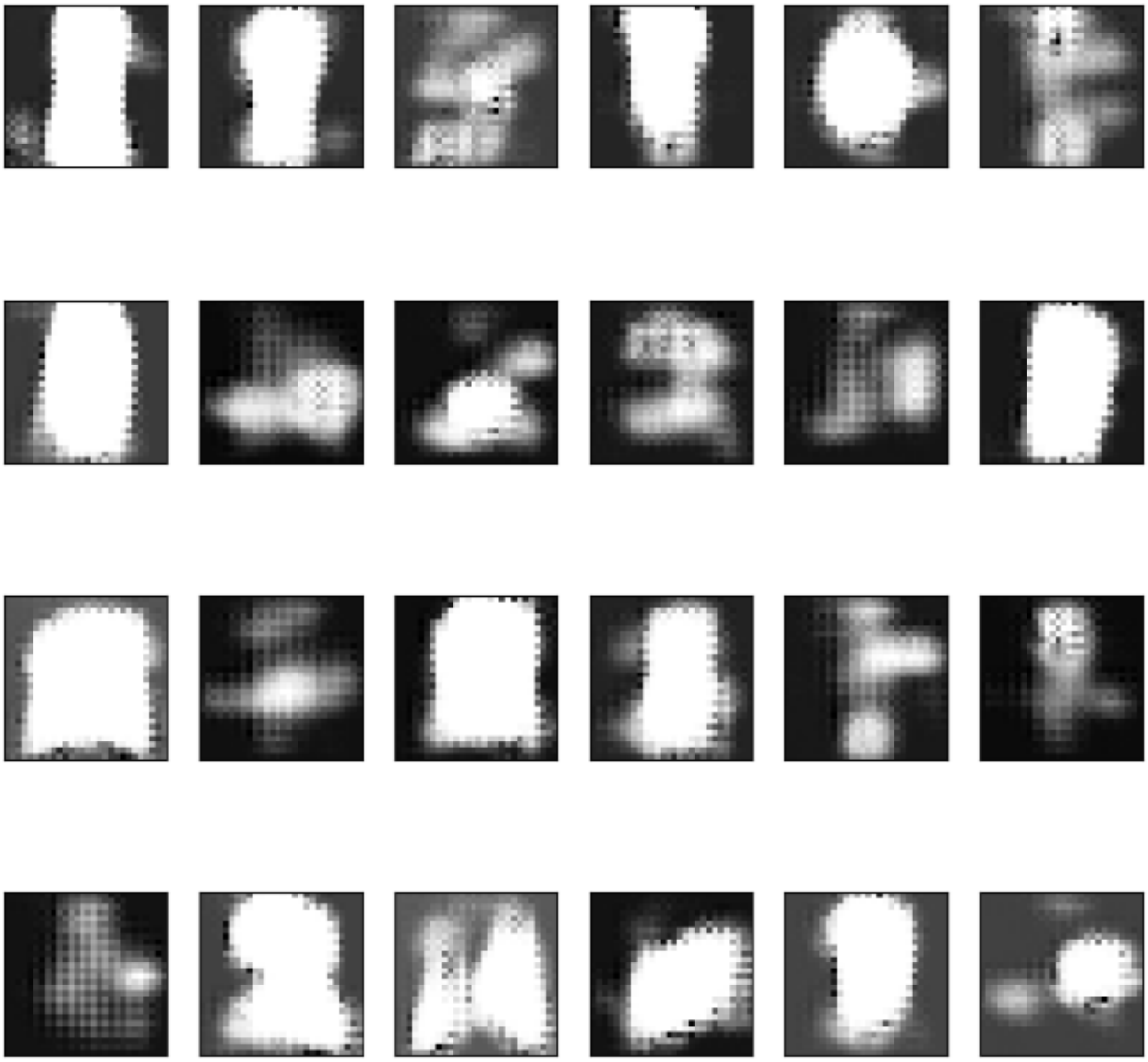
i. Loss function as a function of the iterations for training



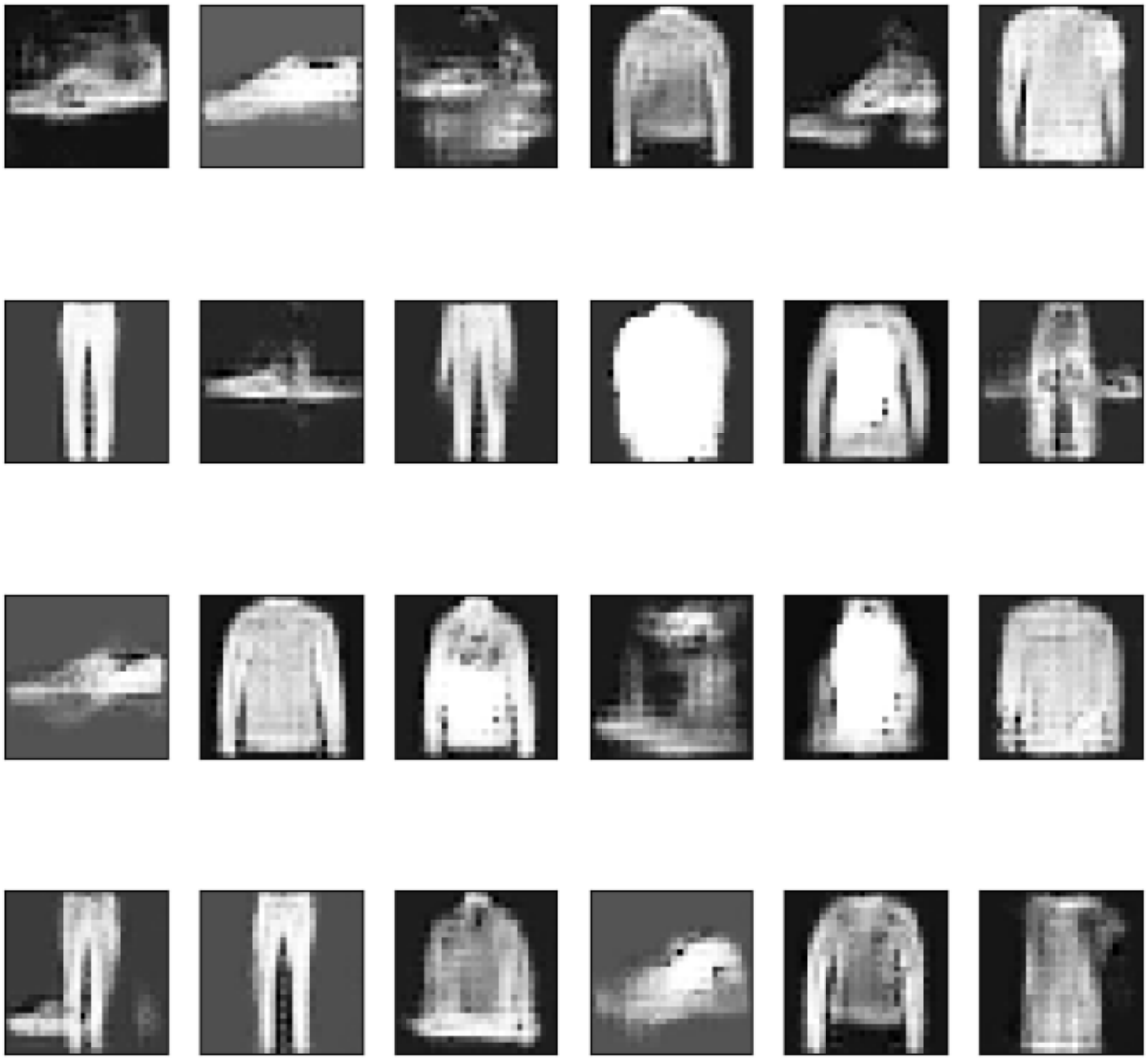
ii. Real images

iii. Fake images – we will present fake images along the training (every 5 epochs), so the process of improving the generator could be seen:

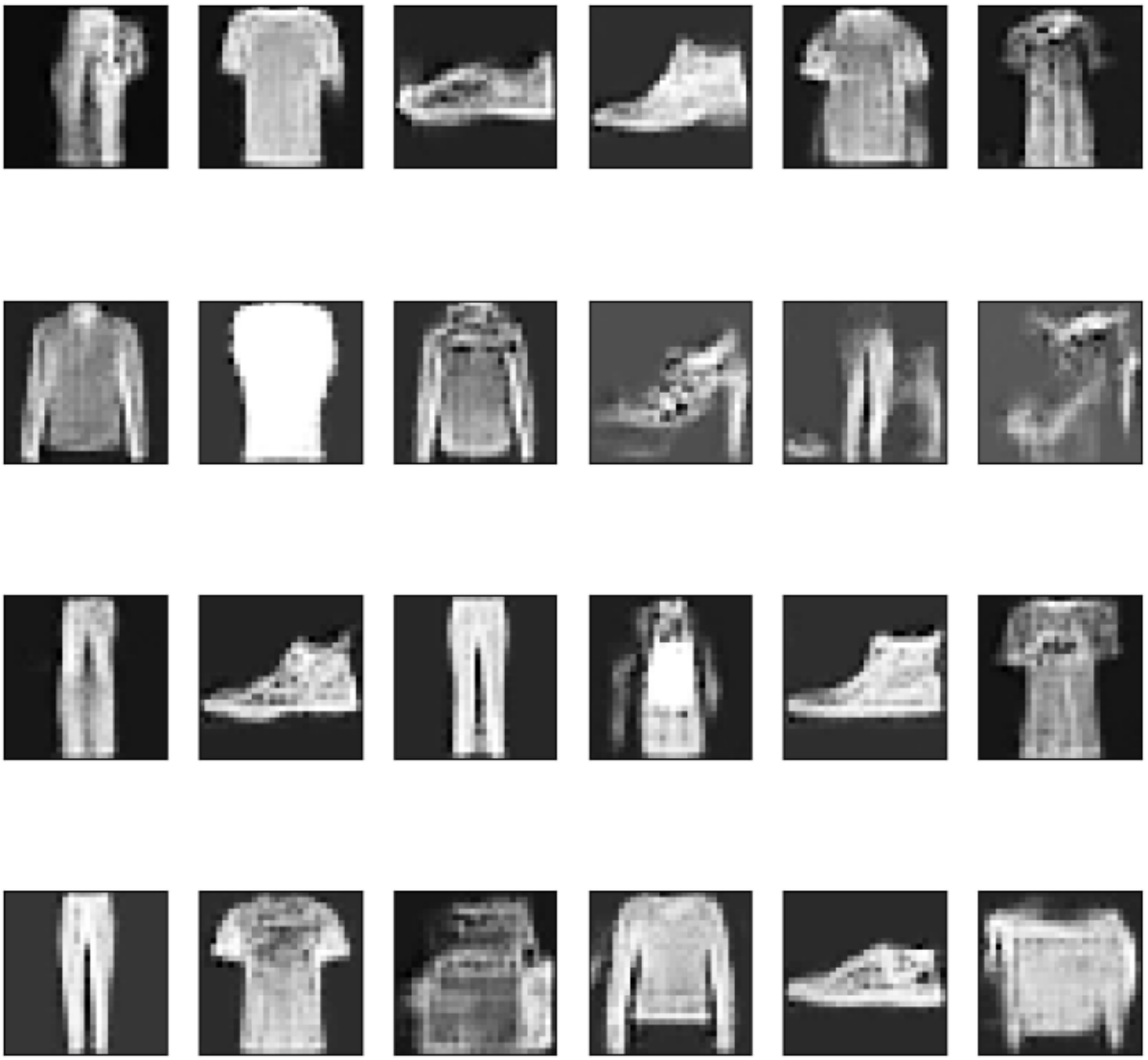
- Epoch 1



- Epoch 5



- Epoch 10



- Epoch 15