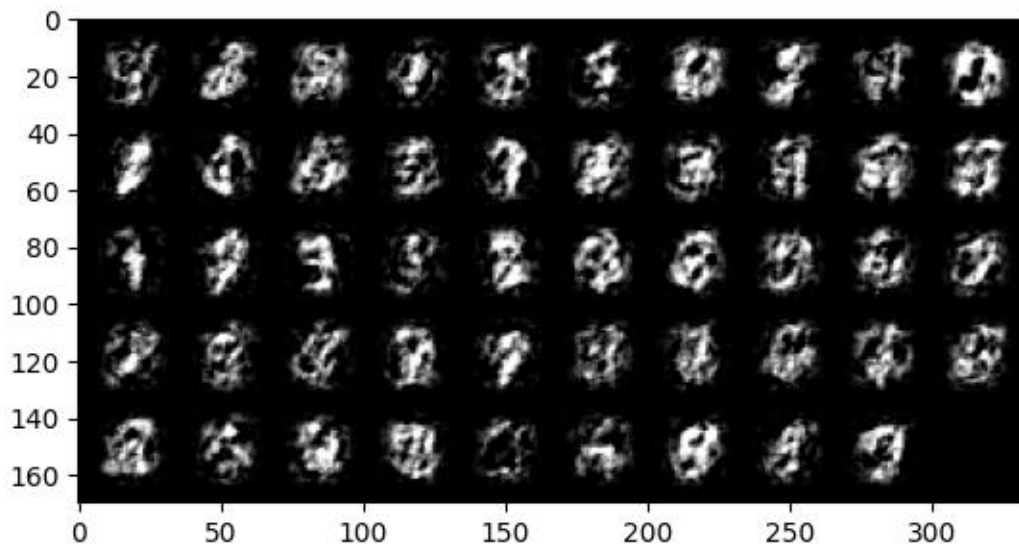


DATA303 HW2: MNIST with Gaussians and VAE

데이터과학과 2021320322 윤민서

1. Learn a multivariate Gaussian and generate samples.

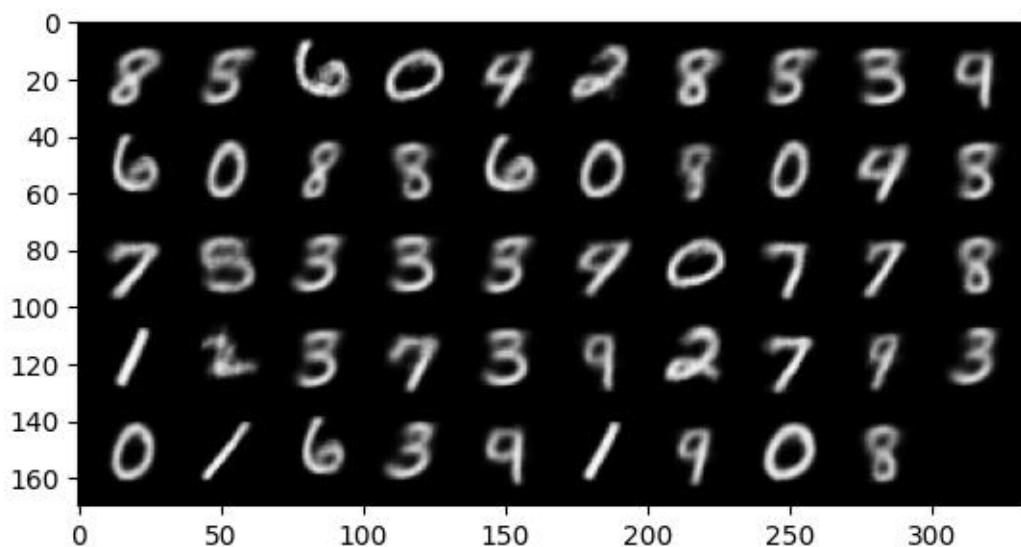
Image_Generation_by_Gaussian.ipynb 실행 결과



2. Learn VAE and generate samples.

- a. Train Model A. VAE with Conv2d, ConvTranspose2d, kernel_size = 4, stride = 2, latent_dims = 2

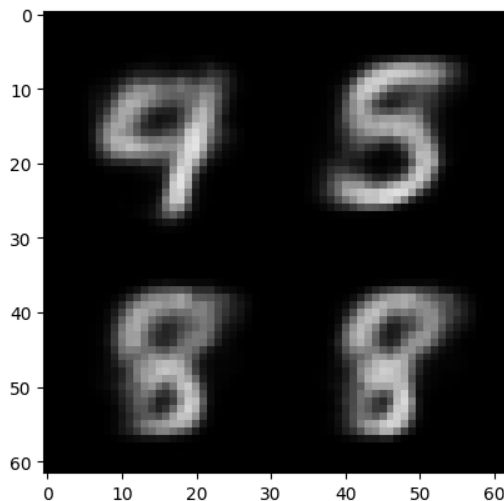
Image_Generation_by_VAE.ipynb - Model A VAE reconstruction



b. Train Model B. Find a model that yields the “checkerboard artifacts”.

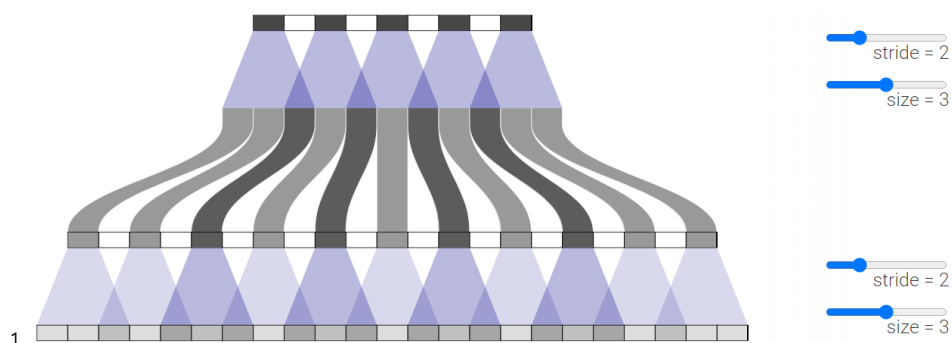
Image_Generation_by_VAE.ipynb - Model B 실행 결과

Checkerboard artifacts를 확인하기 위해서 이미지를 확대하여 생성했다.



Checkerboard artifacts를 생성하기 위해서 Decoder에 있는 nn.ConvTranspose2d의 parameter를 다음과 같이 수정하였다.

```
self.conv2 = nn.ConvTranspose2d(in_channels=c*2, out_channels=c,  
kernel_size=3, stride=2, padding=1, output_padding=1)  
self.conv1 = nn.ConvTranspose2d(in_channels=c, out_channels=1,  
kernel_size=3, stride=2, padding=1, output_padding=1)
```



Decoder의 마지막 layer에서 kernel_size가 stride로 나누어떨어지지 않으면 checkerboard artifacts가 관찰된다.

Decode가 완료된 이후에 image의 size를 다시 28 x 28로 만들어 주기 위해서 output_padding parameter를 설정해 주었다.

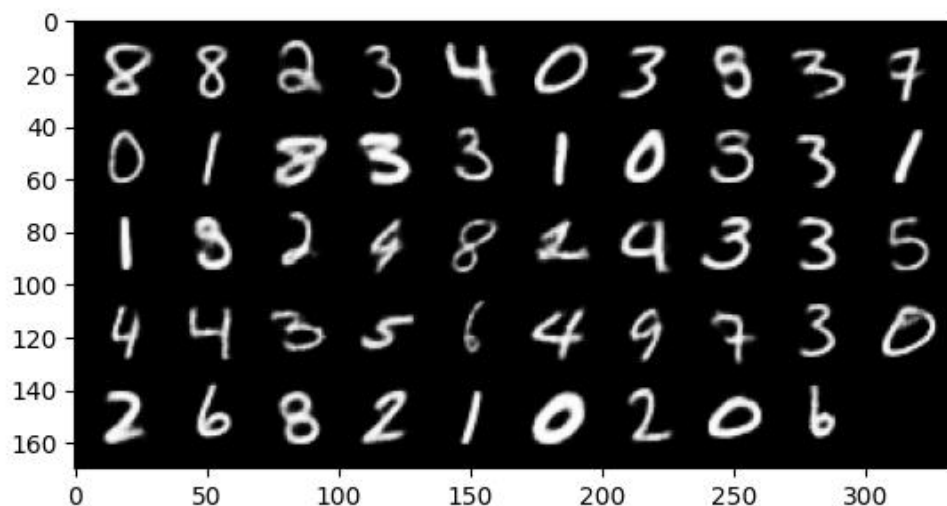
Transpose Convolution의 공식

$$\text{output_size} = (\text{input_size} - 1) * \text{stride} - 2 * \text{padding} + \text{kernel_size}$$
에 의하여 7 x 7 이미지를 추가적인 padding 없이 28 x 28로 만들기는 어렵다.

¹ <https://distill.pub/2016/deconv-checkerboard/>

- c. Train Model C. VAE with Conv2d, ConvTranspose2d, kernel_size = 3, stride = 2, latent_dims = 10

Image_Generation_by_VAE.ipynb - Model C VAE reconstruction



```
self.conv1 = nn.Conv2d(in_channels=1, out_channels=c, kernel_size=3, stride=2, padding=1)
self.conv2 = nn.Conv2d(in_channels=c, out_channels=c*2, kernel_size=3, stride=2, padding=1)
```

Encoder의 코드 수정 사항이다. kernel_size를 4에서 3으로 줄이더라도 아래 Convolution의 공식에 의하여 똑같이 이미지의 가로, 세로 크기가 각각 반으로 줄어들기 때문에 다른 곳을 수정할 필요는 없다.

```
output_size = floor((input_size + 2 * padding - kernel_size) / stride + 1)
```

Decoder는 Model B의 Decoder와 똑같이 수정하였다.

- d. Compare Model A, B, C.

	Model A	Model B	Model C
# of parameters	308357	250565	343317
Average Reconstruction error	18988.374413	19385.497398	12331.945211

Model C에서 latent space의 dimension이 높아지면서 reconstruction error가 유의미하게 감소한 것을 알 수 있다.

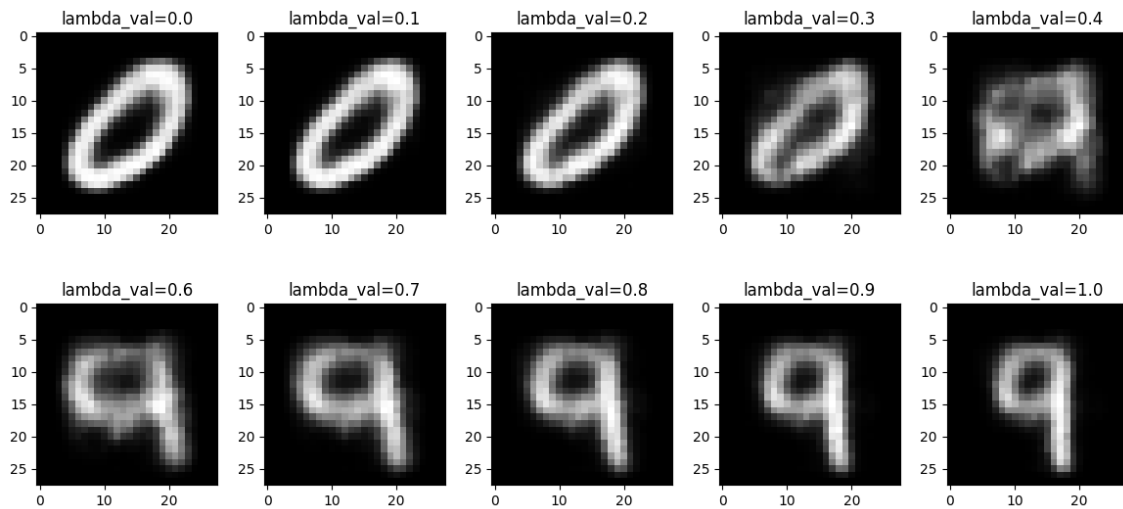
latent space의 dimension이 높아질수록 parameter 개수가 많아진다. 차원을 kernel_size 대신에 padding을 사용하여 맞춰 주는 경우에 parameter 개수가 적어진다.

Model A는 Decoder의 kernel_size = 4가 stride = 2로 나누어떨어지는 반면에 Model B, C는 Decoder의 kernel_size = 3이 stride = 2로 나누어떨어지지 않기 때문에 이미지를 확대해서 출력했을 때 checkerboard artifact를 관찰할 수 있었다.

e. Interpolation in Latent Space

- Interpolate 0 and 9 in latent space. (Model A)

Image_Generation_by_VAE.ipynb



- Run cell multiple times: "Interpolation in Latent Space", explain why different results are obtained.

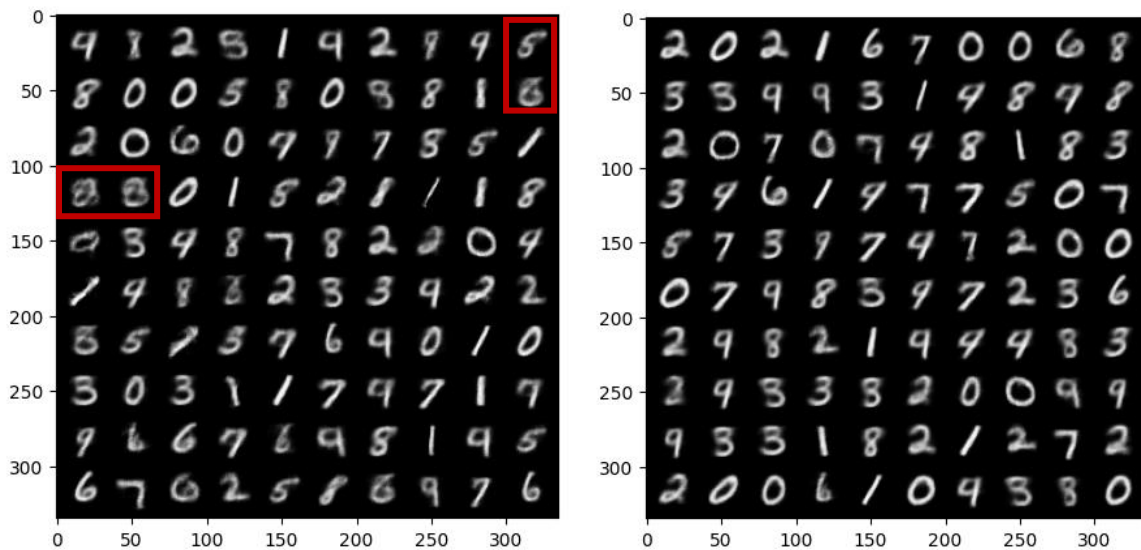
`test_dataloader` 에 정의된 `DataLoader` 함수의 parameter 중 `shuffle=True` 는 데이터를 섞어 주는 역할을 한다. 따라서 `digits` 리스트에 들어가는 데이터는 실행할 때마다 달라지고 이에 따라 다른 결과가 관찰될 수 있다.

f. Sample Latent Vector from "Prior" (VAE as Generator) (Model A)

- Compare the prior and the distribution of training data in the latent space.

Prior와 Training data의 distribution을 비교하기 위해 KL Divergence를 계산하였고, 여러 번 실행한 결과 약 800 정도로 계산된다. 정확한 기준은 알 수 없지만 어느 정도 두 분포 간의 차이가 있는 것으로 생각할 수 있다.

- Sample Latent Vector from estimated Gaussian.



왼쪽은 $N(0, 1)$ 에서 sampling한 latent vector를 reconstruction한 결과이고, 오른쪽은 estimated distribution에서 같은 과정을 수행한 결과이다. 왼쪽 이미지는 오른쪽 이미지에 비해서 숫자라고 판독할 수 없는 것이 비교적 많다. 몇 가지 예시를 빨간 직사각형으로 표시하였고, 이외에도 여러 가지 예시들이 있다.