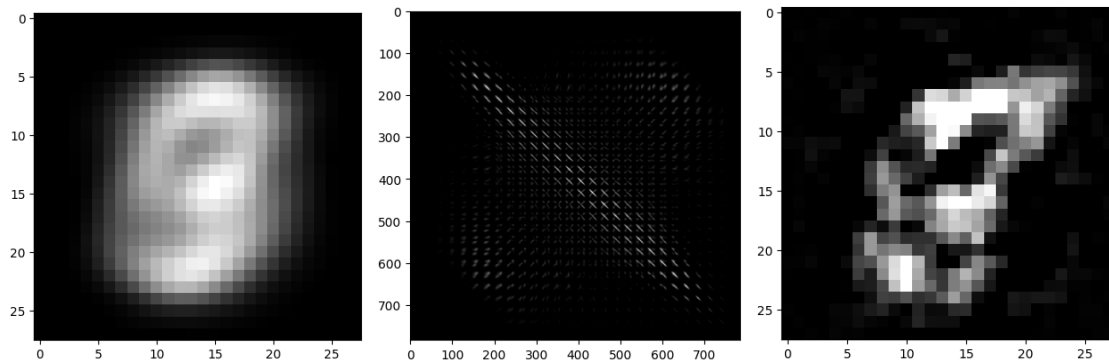


## Assignment 2 - MNIST with Gaussians and VAE

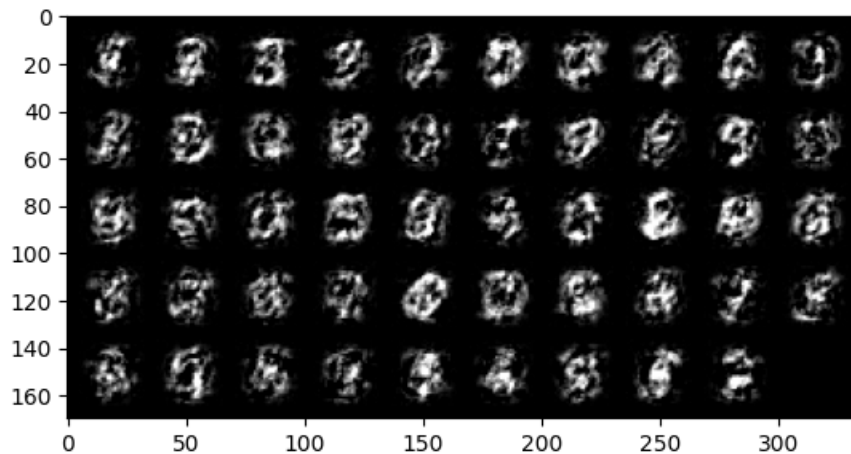
2021320117 Bae Minseong

### 1. Learn a multivariate Gaussian and generate samples

수업 Official Github (<https://github.com/mlvlab/data303>)의 “Image\_Generation\_by\_Gaussian.ipynb” 파일을 Google Colab 상에서 실행시켜 다음과 같은 결과를 얻을 수 있었다.

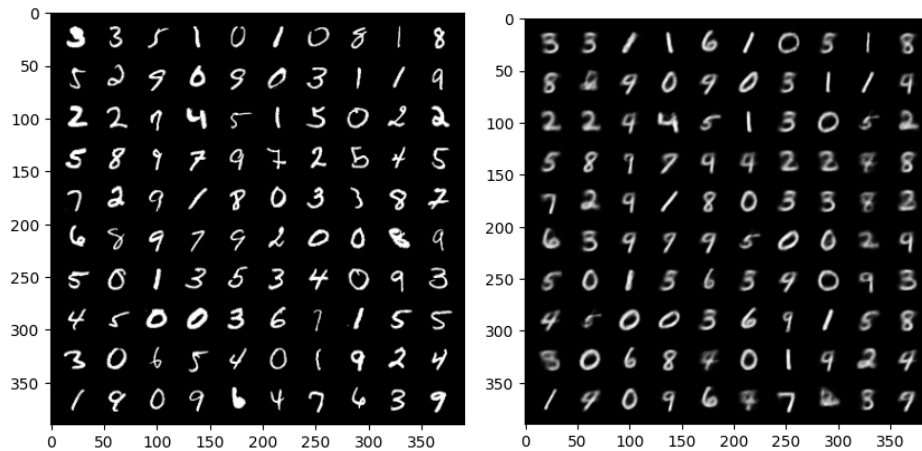


위 그림은 왼쪽에서부터 순서대로 training data 의 mean, covariance matrix, 그리고 mean 과 변형된 covariance matrix 에 대한 multivariate Gaussian 으로부터 sampling 하여 만들어진 generated sample 1 개이다. 아래의 그림은 generated sample 을 50 개 나열한 것이다.



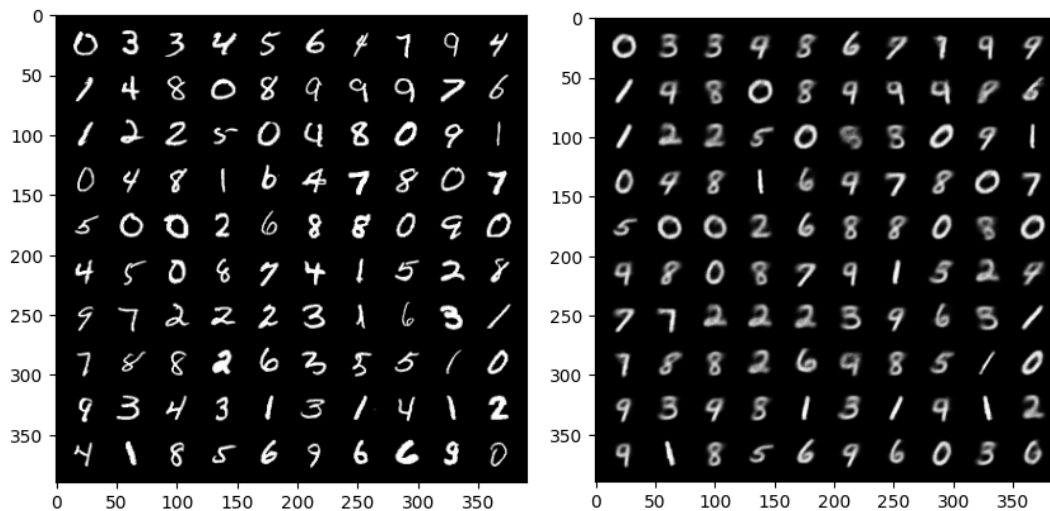
## 2. Learn VAE and generate samples

a. Train model A (VAE with Conv2d / ConvTranspose2d, kernel\_size = 4, stride = 2, latent\_dims = 2)

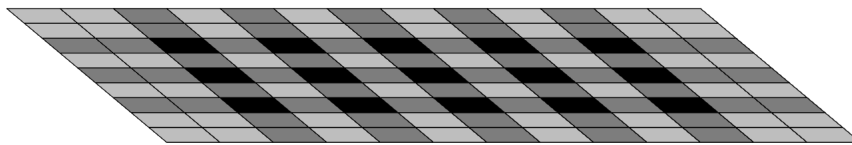


Model A 는 수업 Official Github (<https://github.com/mlvlab/data303>)의 "Image\_Generation\_by\_VAE.ipynb" 파일을 기본 parameter (latent\_dims = 2) 로 실행시켜 Google Colab 상에서 실행시켜 다음과 같은 결과를 얻을 수 있었다. VAE definition 부분에서 kernel\_size = 4, stride = 2 로 세팅해서 실행시켰고, 왼쪽이 original image 이고 오른쪽이 reconstructed image 이다.

b. Train model B (Find a model that yields the "checkerboard artifacts")

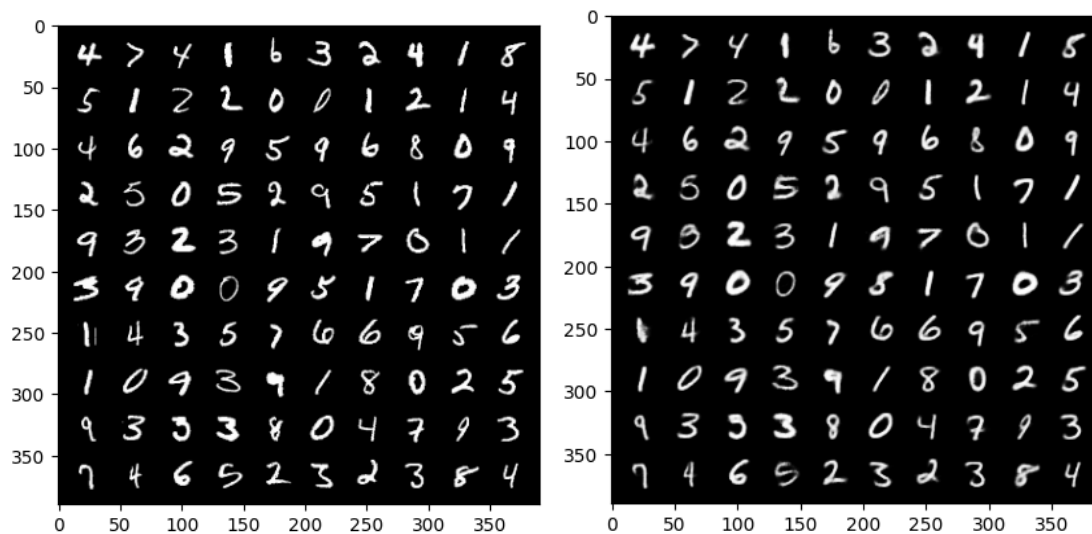


<https://distill.pub/2016/deconv-checkerboard/>의 설명에 따르면, Transpose Convolution 을 활용해 deconvolution 을 진행할 때 stride 와 filter size 에 따라 filter 가 적용되어 upsampling 되는 과정에서 overlap 이 발생하면서 checkerboard 형태의 image 가 생성되는 문제가 발생할 수 있고, 이런 문제는 kernel size 가 stride 로 나누어떨어지도록 만듦으로써 완화시킬 수 있다고 한다. 따라서 checkerboard artifact 를 의도적으로 발생시키기 위해 VAE decoder 의 ConvTranspose2d 함수의 kernel\_size 를 5 로 조정하고, image 의 size 를 7x7 과 14x14 로 맞추기 위해 padding 을 2 로 조정하고 output\_padding 을 1만큼 추가하였다. 여기서, stride 의 크기인 2로 나누어떨어지지 않는 kernel size는 3이나 5가 가능한데, overlap 을 더 키우기 위해 5를 선택했다. 원본 이미지와 해당 모델을 통해 reconstruction 된 결과는 위와 같다.



(3x3의 kernel size로 deconvolution을 2D에서 진행했을 때의 결과, 색이 진할수록 overlap이 크다.)

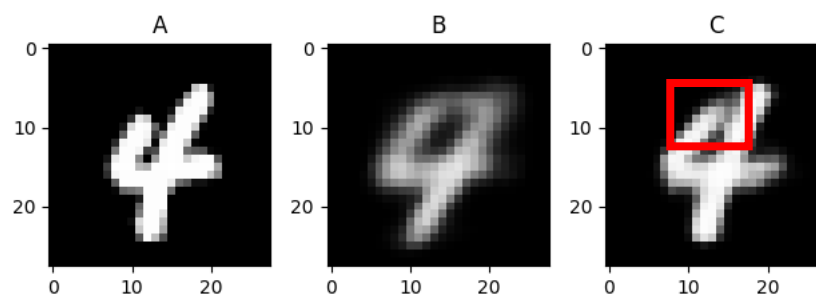
c. Train model C (VAE with Conv2d / ConvTranspose2d, kernel\_size = 3, stride = 2, latent\_dims = 10)



Model A의 hyperparameter에서 latent\_dims를 10으로 바꾸고, VAE encoder의 Conv2d와 decoder의 ConvTranspose2d 함수의 kernel\_size를 3으로 조정하고, image의 size를 7x7과 14x14로 맞추기 위해 output\_padding을 1만큼 추가하였다. 원본 이미지와 해당 모델을 통해 reconstruction된 결과는 위와 같다.

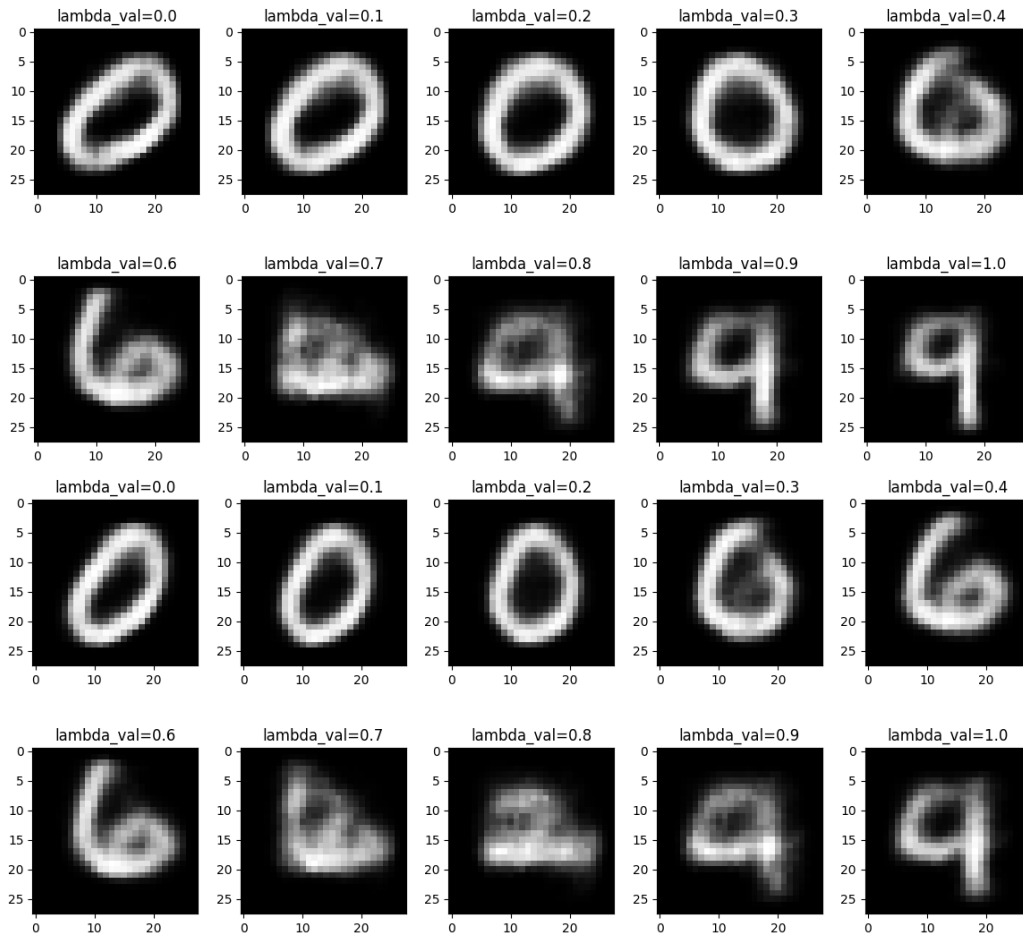
d. Compare model A, B, and C

A, B와 C의 결과를 비교하기 위해 같은 데이터에 대해 세 가지 모델을 적용시킨 결과를 비교할 수 있는 새로운 코드를 작성하였다. 이를 실행시킨 결과는 다음과 같다.



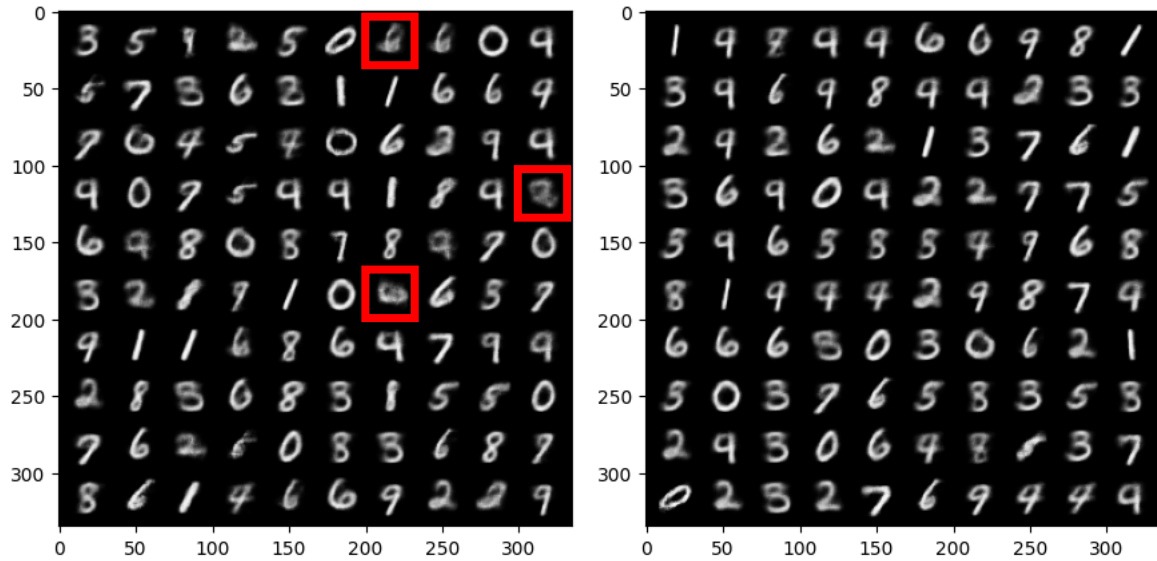
모델 B의 결과는 다른 모델들에 비해 reconstruction된 결과에 checkerboard 형태가 의도한대로 강하게 나타났음을 확인할 수 있다. A와 C를 비교해보았을 때, 모델 A의 경우 모델 C보다 더 선명하고 checkerboard 형태가 덜 나타난 이미지가 생성되었는데, 이는 앞에서 서술했듯이 kernel size를 stride 크기로 나누어 떨어지도록 설정하는 것이 checkerboard artifact를 줄이는데 도움이 되는데, A와 C 모두 stride = 2이지만 A의 kernel size는 4이고 C의 kernel size는 3이기 때문이다. 하지만 C의 latent space의 dimension이 더 크기 때문에 더 많은 정보를 담고 있고 이에 따라 빨간색 박스 안의 부분처럼 reconstruction된 결과에 추가적인 요소가 존재한다.

## e. Interpolation in Latent Space



Model A 를 활용하여 latent space 에서 0 과 9 를 나타내는 두 latent vector 의 interpolation 을 진행했을 때의 reconstruction 결과를 다음과 같이 확인해볼 수 있다. Lambda 의 값에 따라 0 과 9 의 interpolation 비율이 정해지기 때문에 lambda 의 값이 클수록 형태가 9 에 가깝고, 작을수록 0 에 가까운 것을 볼 수 있다. 그리고 0.5~0.6 의 lambda 값, 즉 0 과 9 를 비슷한 가중치로 interpolate 한 latent vector 를 decoding 한 결과 6 과 비슷한 형태를 보인다. 위와 아래의 이미지들은 같은 코드를 두 번 실행했을 때의 결과인데, 두 결과가 다르게 나온 것을 볼 수 있다. 이는 interpolation 을 수행하기 위해 test\_dataloader 에서 이미지를 가져오는 과정에서 dataloader 의 shuffle 변수가 True 이기 때문에 해당 함수를 실행시킬 때마다 다른 image batch 가 interpolation 에 사용되기 때문이다.

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



왼쪽의 이미지는  $N(0, 1)$  (standard normal distribution)으로부터 sampling 한 latent vector 에 decoder 를 적용시켜 생성한 이미지이고, 오른쪽의 이미지는 기존의 model A 를 이용해 train data 의 하나의 batch 를 통해 만들어낸 reconstructed image 이다.

Latent space 상에서 training data 의 distribution 과 prior (normal distribution)을 비교하기 위해서 VAE encoder 의 결과값인 latent\_mu 와 latent\_logvar 을 이용해 두 distribution 의 KL divergence 값을 계산해보면, 약 800~830 의 값이 나온다. 오른쪽의 reconstruction 결과에 비해 왼쪽의 random sampling 을 통한 generated image 의 경우 빨간색 박스와 같이 숫자의 형태로 보기 매우 어려운 결과들이 몇 개 존재한다.