

변이형 인코더 (VAE)

먼저 이 장에서 발생할 수 있는 오류가 크게 두 가지 있으니 해결 방법을 알아봅니다.

만약 .ipynb 파일을 GDL_code 파일 안에 생성하지 않았다면,

첫번째는 경로오류 문제로 아래와 같은 메세지가 뜹니다.

No module name named 'utils'

No module name named 'models'

[WinError 3] 지정된 경로를 찾을 수 없습니다: 'run/vae/0001_digits'

이는 모두 GDL_code 파일에 있는 utils 파일, models 파일, data 파일, run 파일을 불러 올 수 없어서 생기는 경로오류 문제입니다.



이를 해결하려면,

첫번째 방법,

GDL_code 파일에 있는 utils 파일, models 파일, data 파일, run 파일을 복사 후
생성한 파일이 있는 경로와 동일한 곳에 붙여넣기 하는 방법과

두번째 방법,

GDL_code 파일 안에 .ipynb 파일을 생성하여 작업하는 방법 두 가지가 있겠습니다.

다른 하나는 라이브러리가 설치되지 않는 문제로 아래와 같은 메시지가 발생합니다. (다음 슬라이드 사진 첨부)

Failed to import `pydot`. Please install `pydot`. For example with `pip install pydot`.

`pydot` failed to call GraphViz. Please install GraphViz (<https://www.graphviz.org/>) and ensure that its executables are in the \$PATH.

Anaconda Prompt (생성한 가상환경 이름) 으로 들어갑니다.

그러면 자동으로 생성한 가상환경으로 activate 된 프롬프트 화면을 볼 수 있고,

여기서 다음과 같은 명령어를 입력하여 사전에 오류를 방지합니다.

```
pip install pydot==1.4.1
```

```
conda install GraphViz
```



최근에 추가한 앱



Node.js



Node.js command prompt



Install Additional Tools for Node.js

확장 ▾

A



Acrobat Reader DC



AhnLab



Anaconda3 (64-bit)
새로 설치됨



Anaconda Navigator (anaconda3)



Anaconda Powershell Prompt (an...



Anaconda Powershell Prompt (tes...
새로 설치됨



Anaconda Prompt (anaconda3)



Anaconda Prompt (testGAN)



Jupyter Notebook (anaconda3)



Jupyter Notebook (testGAN)

만들기



일정



그림판 3D



사진



Office



흐림

24°

24°
17°

서울



메모장

■ Anaconda Prompt (testGAN)

```
(testGAN) C:\Users\User>pip install pydot==1.4.1
```

```
Collecting pydot==1.4.1
```

```
  Using cached pydot-1.4.1-py2.py3-none-any.whl (19 kB)
```

```
Requirement already satisfied: pyparsing>=2.1.4 in c:\users\user\anaconda3\envs\testgan\lib\site-packages (from pydot==1.4.1) (2.4.7)
```

```
Installing collected packages: pydot
```

```
Successfully installed pydot-1.4.1
```

(testGAN) C:\Users\User>conda install GraphViz

Collecting package metadata (current_repodata.json): done

Solving environment: done

==> WARNING: A newer version of conda exists. <==

current version: 4.8.2

latest version: 4.8.3

Please update conda by running

\$ conda update -n base -c defaults conda

Package Plan

environment location: C:\Users\User\anaconda3\envs\testGAN

added / updated specs:

- graphviz

The following packages will be downloaded:

package	build	
graphviz-2.38	hfd603c8_2	29.3 MB
Total:		29.3 MB

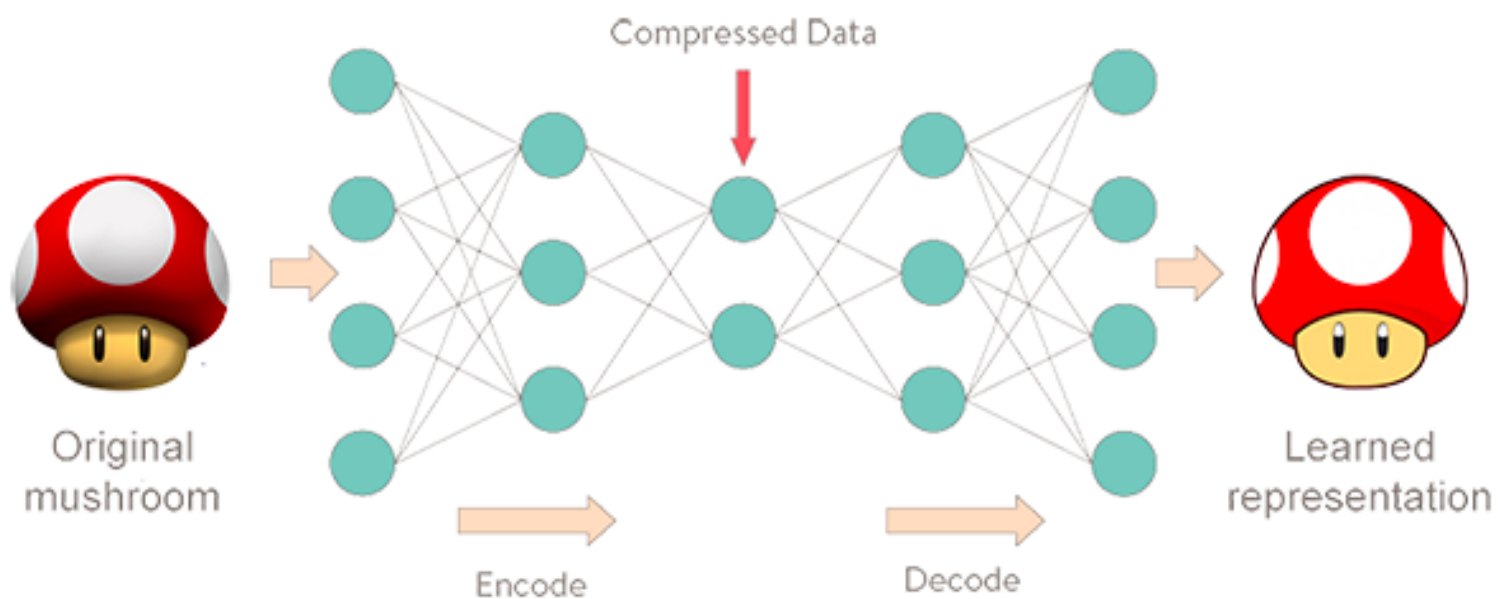
The following NEW packages will be INSTALLED:

graphviz pkgs/main/win-64::graphviz-2.38-hfd603c8_2

Proceed ([y]/n)? y

이제 VAE(변이형 오토인코더; Variation AutoEncoder)를 알아보기 전에

기본적인 AE(오토인코더; AutoEncoder)를 만들어 보고 이를 확장하여 VAE를 구현해 봅니다.



오토인코더

라이브러리 импорт

In [1]:

```
import os

from utils.loaders import load_mnist
from models.AE import Autoencoder
```

Using TensorFlow backend.

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint16 = np.dtype [("quint16", np.uint16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint32 = np.dtype [("qint32", np.int32, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
np_resource = np.dtype [("resource", np.ubyte, 1)]
```

매개변수 설정

```
In [2]: # 실행 매개변수
SECTION = 'vae'
RUN_ID = '0001'
DATA_NAME = 'digits'
RUN_FOLDER = 'run/{}/'.format(SECTION)
RUN_FOLDER += '_'.join([RUN_ID, DATA_NAME])

if not os.path.exists(RUN_FOLDER):
    os.mkdir(RUN_FOLDER)
    os.mkdir(os.path.join(RUN_FOLDER, 'viz'))
    os.mkdir(os.path.join(RUN_FOLDER, 'images'))
    os.mkdir(os.path.join(RUN_FOLDER, 'weights'))

MODE = 'build' #'load' #
```

데이터 적재 (Mnist 데이터 사용)

```
In [3]: (x_train, y_train), (x_test, y_test) = load_mnist()

Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
11493376/11490434 [=====] - 9s 1us/step
```

신경망 구조 정의

```
In [4]: AE = Autoencoder(
    input_dim = (28,28,1)
    , encoder_conv_filters = [32,64,64, 64]
    , encoder_conv_kernel_size = [3,3,3,3]
    , encoder_conv_strides = [1,2,2,1]
    , decoder_conv_t_filters = [64,64,32,1]
    , decoder_conv_t_kernel_size = [3,3,3,3]
    , decoder_conv_t_strides = [1,2,2,1]
    , z_dim = 2
)

if MODE == 'build':
    AE.save(RUN_FOLDER)
else:
    AE.load_weights(os.path.join(RUN_FOLDER, 'weights/weights.h5'))
```

WARNING:tensorflow:From C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.

Instructions for updating:

Colocations handled automatically by placer.

인코더의 모양

In [5]: AE.encoder.summary()

Layer (type)	Output Shape	Param #
encoder_input (InputLayer)	(None, 28, 28, 1)	0
encoder_conv_0 (Conv2D)	(None, 28, 28, 32)	320
leaky_re_lu_1 (LeakyReLU)	(None, 28, 28, 32)	0
encoder_conv_1 (Conv2D)	(None, 14, 14, 64)	18496
leaky_re_lu_2 (LeakyReLU)	(None, 14, 14, 64)	0
encoder_conv_2 (Conv2D)	(None, 7, 7, 64)	36928
leaky_re_lu_3 (LeakyReLU)	(None, 7, 7, 64)	0
encoder_conv_3 (Conv2D)	(None, 7, 7, 64)	36928
leaky_re_lu_4 (LeakyReLU)	(None, 7, 7, 64)	0
flatten_1 (Flatten)	(None, 3136)	0
encoder_output (Dense)	(None, 2)	6274
Total params: 98,946		
Trainable params: 98,946		
Non-trainable params: 0		

디코더의 모양

In [6]: AE.decoder.summary()

Layer (type)	Output Shape	Param #
decoder_input (InputLayer)	(None, 2)	0
dense_1 (Dense)	(None, 3136)	9408
reshape_1 (Reshape)	(None, 7, 7, 64)	0
decoder_conv_t_0 (Conv2DTran	(None, 7, 7, 64)	36928
leaky_re_lu_5 (LeakyReLU)	(None, 7, 7, 64)	0
decoder_conv_t_1 (Conv2DTran	(None, 14, 14, 64)	36928
leaky_re_lu_6 (LeakyReLU)	(None, 14, 14, 64)	0
decoder_conv_t_2 (Conv2DTran	(None, 28, 28, 32)	18464
leaky_re_lu_7 (LeakyReLU)	(None, 28, 28, 32)	0
decoder_conv_t_3 (Conv2DTran	(None, 28, 28, 1)	289
activation_1 (Activation)	(None, 28, 28, 1)	0
Total params: 102,017		
Trainable params: 102,017		
Non-trainable params: 0		

오토인코더 훈련

```
In [7]: LEARNING_RATE = 0.0005  
        BATCH_SIZE = 32  
        INITIAL_EPOCH = 0
```

```
In [8]: AE.compile(LEARNING_RATE)
```

```
In [*]: AE.train(  
    x_train[:1000]  
    , batch_size = BATCH_SIZE  
    , epochs = 200  
    , run_folder = RUN_FOLDER  
    , initial_epoch = INITIAL_EPOCH  
)
```

```
1000/1000 [=====] - 5s 5ms/step - loss: 0.0401
```

```
Epoch 00079: saving model to run/vae/0001_digits#weights/weights.h5
```

```
Epoch 80/200
```

```
1000/1000 [=====] - 5s 5ms/step - loss: 0.0402
```

```
Epoch 00080: saving model to run/vae/0001_digits#weights/weights.h5
```

```
Epoch 81/200
```

```
1000/1000 [=====] - 5s 5ms/step - loss: 0.0402
```

```
Epoch 00081: saving model to run/vae/0001_digits#weights/weights.h5
```

```
Epoch 82/200
```

```
1000/1000 [=====] - 5s 5ms/step - loss: 0.0398
```

```
Epoch 00082: saving model to run/vae/0001_digits#weights/weights.h5
```

```
Epoch 83/200
```

```
1000/1000 [=====] - 5s 5ms/step - loss: 0.0400
```

```
Epoch 00083: saving model to run/vae/0001_digits#weights/weights.h5
```

```
Epoch 84/200
```

오토인코더 분석

Note: 최신 버전의 라이브러리를 사용하기 때문에 책의 내용과 결과가 다를 수 있습니다

라이브러리 импорт

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import os
from scipy.stats import norm

from models.AE import Autoencoder
from utils.loaders import load_mnist, load_model
```

Using TensorFlow backend.

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint16 = np.dtype [("quint16", np.uint16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint32 = np.dtype [("qint32", np.int32, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
np_resource = np.dtype [("resource", np.ubyte, 1)]
```

```
In [2]: # run params
SECTION = 'vae'
RUN_ID = '0001'
DATA_NAME = 'digits'
RUN_FOLDER = 'run/{}'.format(SECTION)
RUN_FOLDER += '_'.join([RUN_ID, DATA_NAME])
```

데이터 적재

```
In [3]: (x_train, y_train), (x_test, y_test) = load_mnist()
```

모델 구조 불러오기

```
In [4]: AE = load_model(Autoencoder, RUN_FOLDER)
```

WARNING:tensorflow:From C:\Users\User\anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.

원본 그림 재구성

```
In [5]: n_to_show = 10
np.random.seed(88)
example_idx = np.random.choice(range(len(x_test)), n_to_show)
example_images = x_test[example_idx]

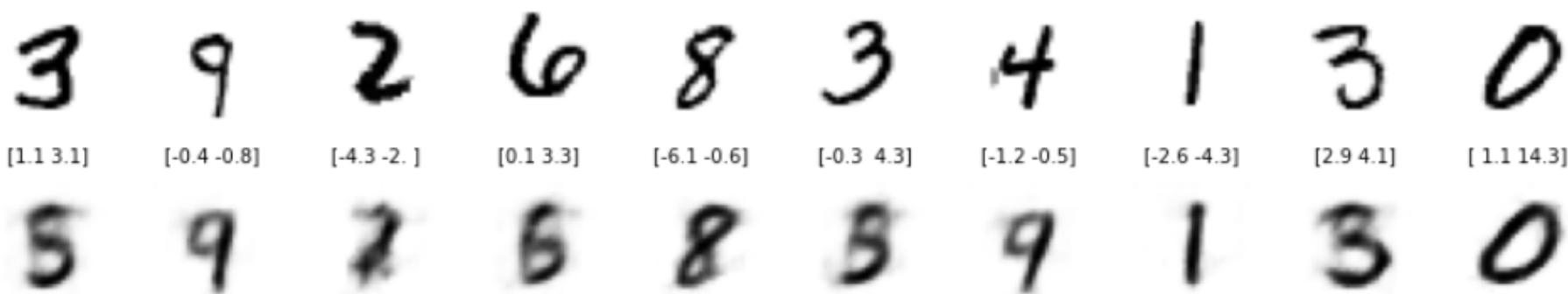
z_points = AE.encoder.predict(example_images)

reconst_images = AE.decoder.predict(z_points)

fig = plt.figure(figsize=(15, 3))
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(n_to_show):
    img = example_images[i].squeeze()
    ax = fig.add_subplot(2, n_to_show, i+1)
    ax.axis('off')
    ax.text(0.5, -0.35, str(np.round(z_points[i],1)), fontsize=10, ha='center', transform=ax.transAxes)
    ax.imshow(img, cmap='gray_r')

for i in range(n_to_show):
    img = reconst_images[i].squeeze()
    ax = fig.add_subplot(2, n_to_show, i+n_to_show+1)
    ax.axis('off')
    ax.imshow(img, cmap='gray_r')
```



엔 코더 씨의 전시벽

```
In [6]: n_to_show = 5000
        grid_size = 15
        figsize = 10

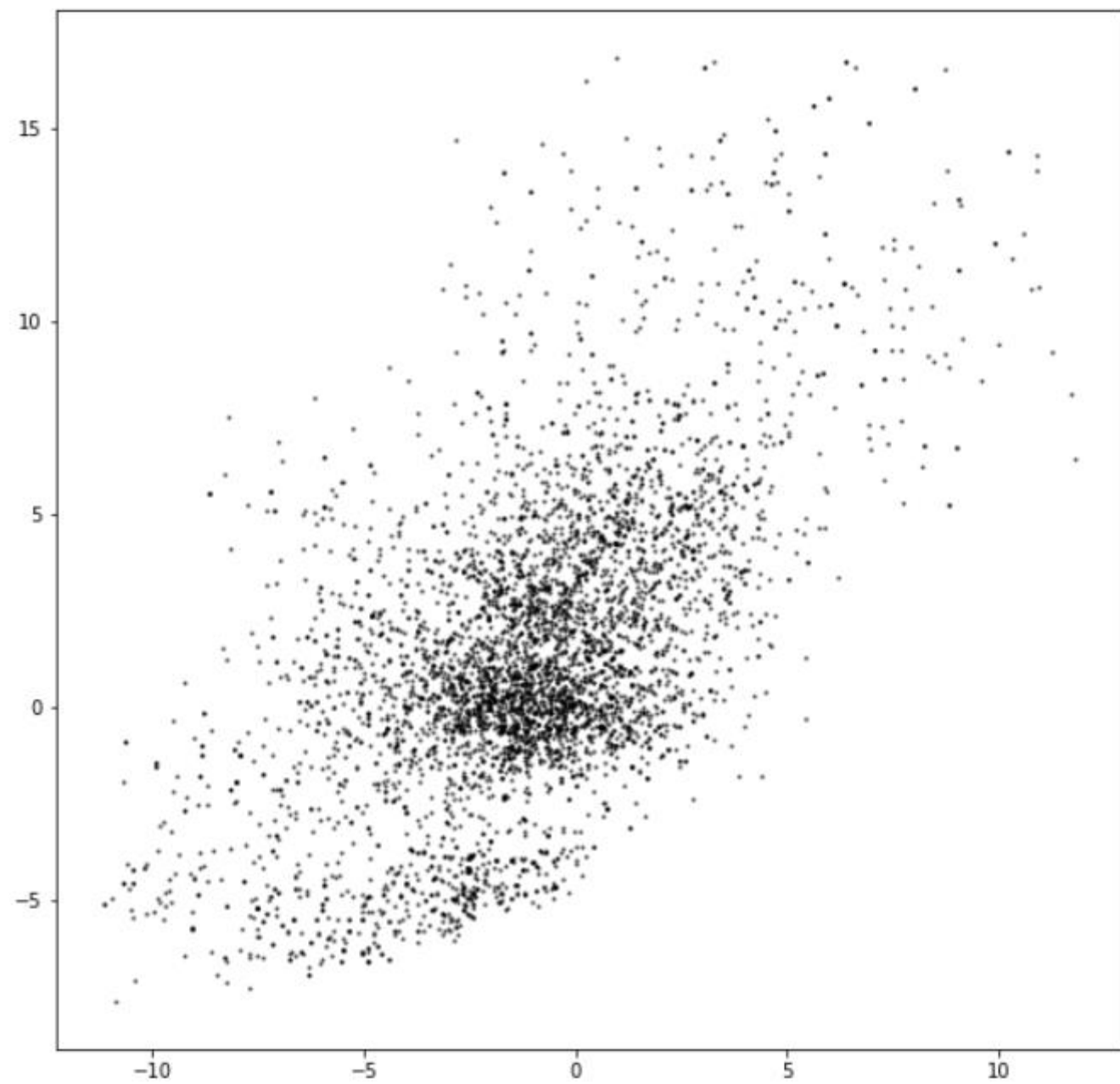
        example_idx = np.random.choice(range(len(x_test)), n_to_show)
        example_images = x_test[example_idx]
        example_labels = y_test[example_idx]

        z_points = AE.encoder.predict(example_images)

        min_x = min(z_points[:, 0])
        max_x = max(z_points[:, 0])
        min_y = min(z_points[:, 1])
        max_y = max(z_points[:, 1])

        plt.figure(figsize=(figsize, figsize))
        plt.scatter(z_points[:, 0], z_points[:, 1], c='black', alpha=0.5, s=2)

        plt.show()
```



새로 생성한 미술 전시품

```
In [7]: grid_size = 15
figsize = 10

plt.figure(figsize=(figsize, figsize))
plt.scatter(z_points[:, 0], z_points[:, 1], c='black', alpha=0.5, s=2)

grid_size = 10
grid_depth = 3
figsize = 15

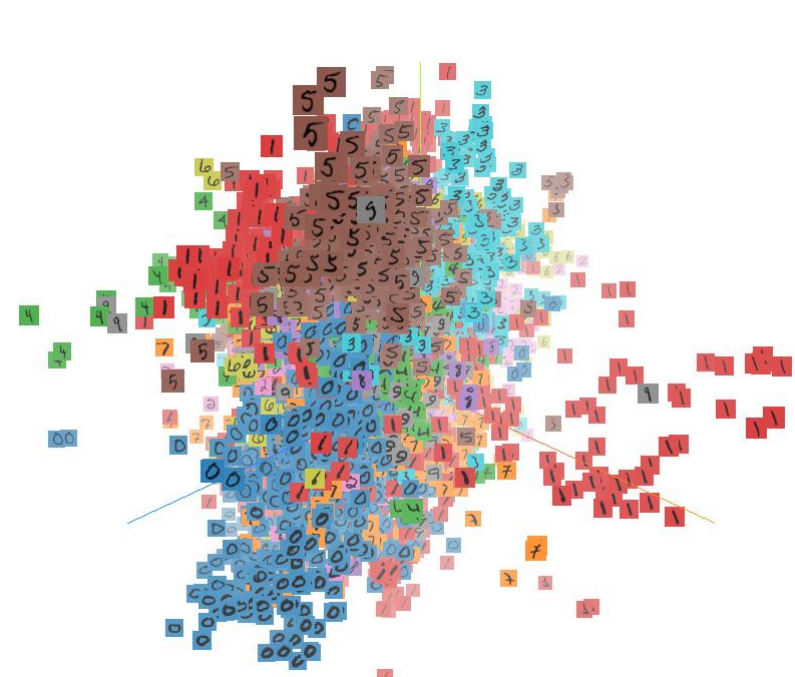
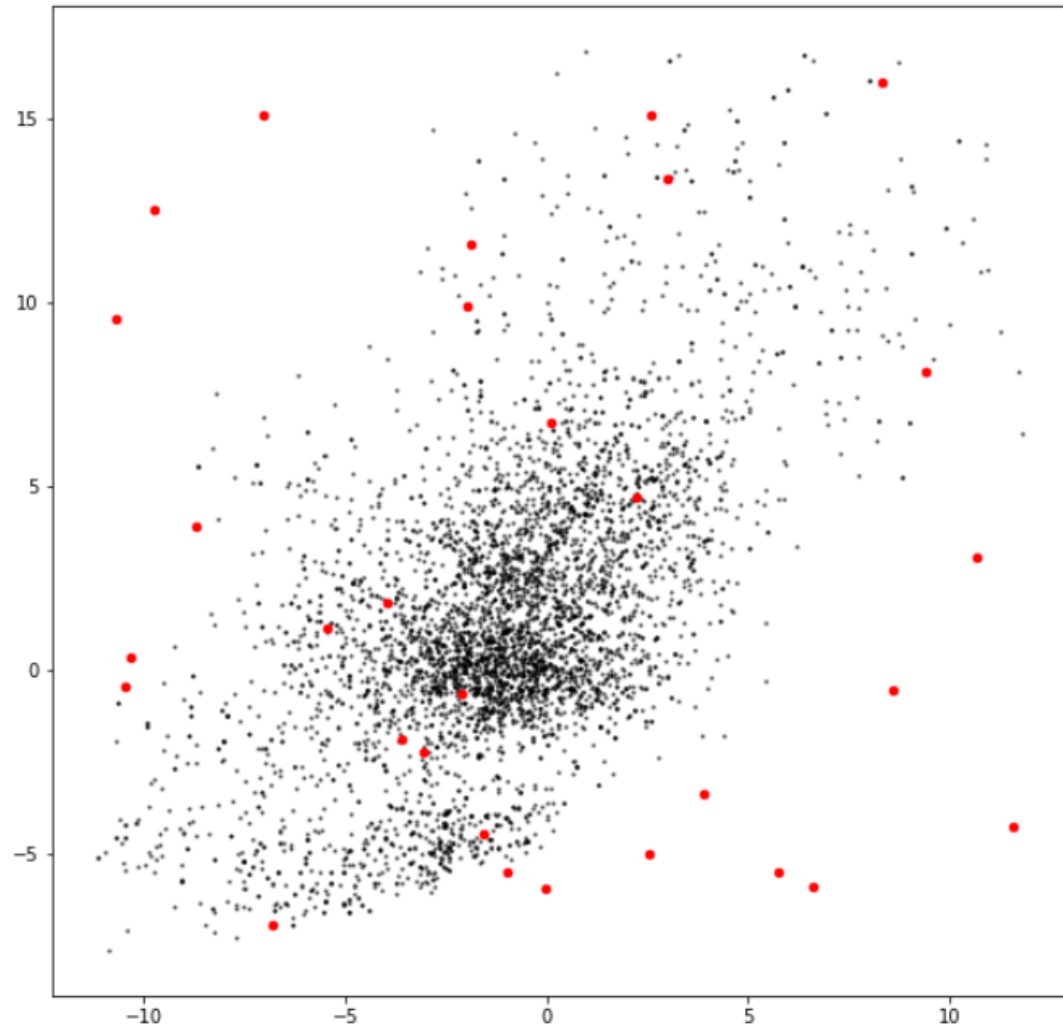
x = np.random.uniform(min_x, max_x, size = grid_size * grid_depth)
y = np.random.uniform(min_y, max_y, size = grid_size * grid_depth)
z_grid = np.array(list(zip(x, y)))
reconst = AE.decoder.predict(z_grid)

plt.scatter(z_grid[:, 0], z_grid[:, 1], c='red', alpha=1, s=20)
plt.show()

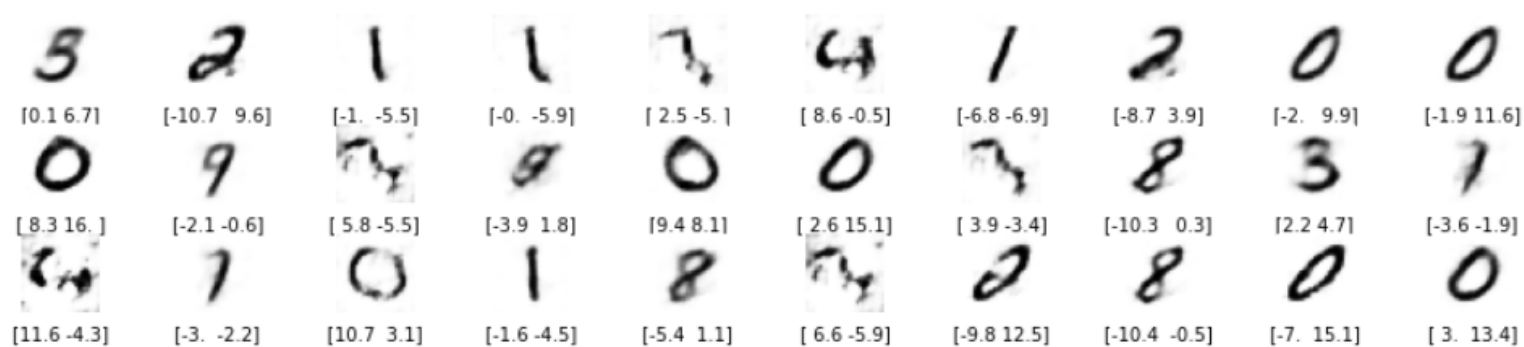
fig = plt.figure(figsize=(figsize, grid_depth))
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(grid_size*grid_depth):
    ax = fig.add_subplot(grid_depth, grid_size, i+1)
    ax.axis('off')
    ax.text(0.5, -0.35, str(np.round(z_grid[i], 1)), fontsize=10, ha='center', transform=ax.transAxes)

    ax.imshow(reconst[i, :, :, 0], cmap = 'Greys')
```



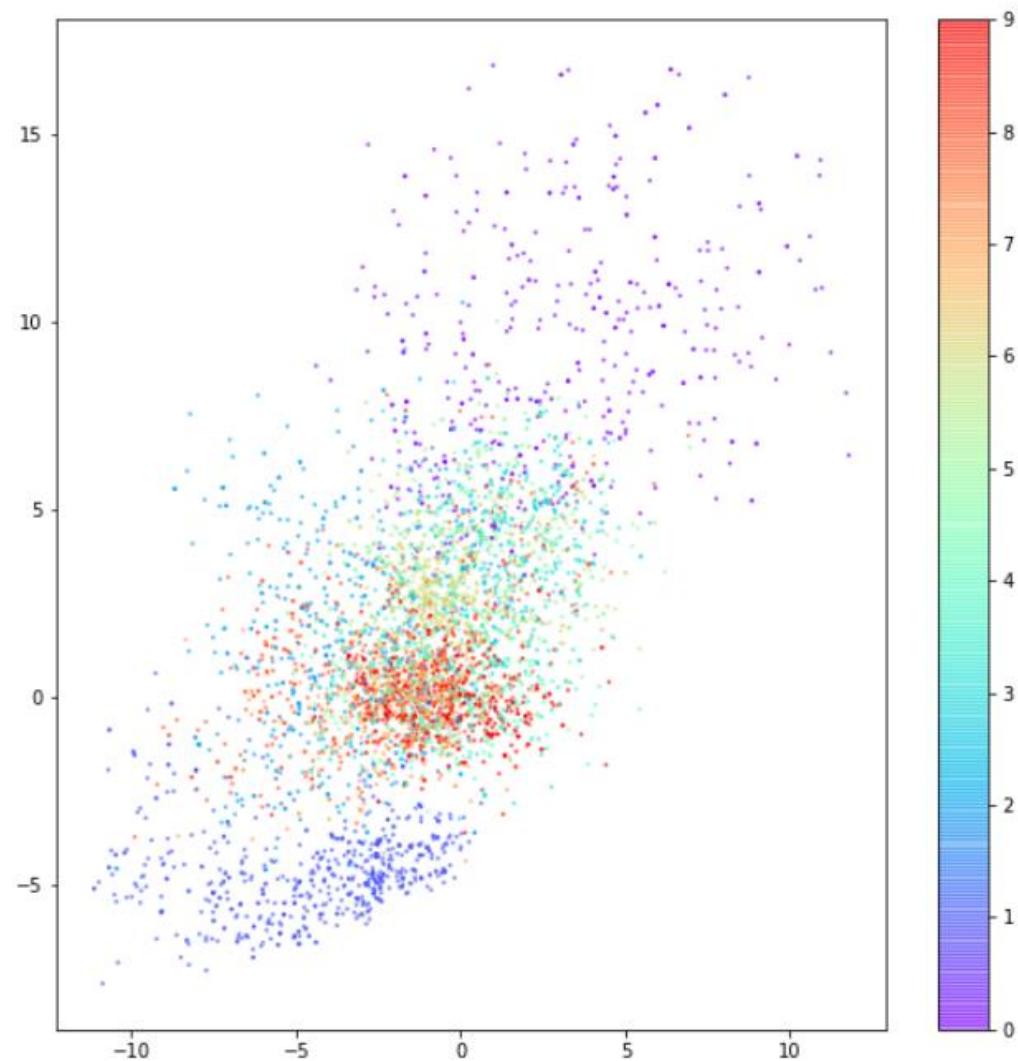
3차원 공간의 표현 (내용과 무관)



숫자별 색깔 산점도 (scatter plot)

In [8]: `figsize = 10`

```
plt.figure(figsize=(figsize, figsize))  
plt.scatter(z_points[:, 0], z_points[:, 1], cmap='rainbow', c=example_labels, alpha=0.5, s=2)  
plt.colorbar()  
  
plt.show()
```



임의의 위치 (-5,-5) (0,5) (0,0) (6,-8)에 있는 한 점의 숫자를 추출

```
In [9]: figsize = 10

plt.figure(figsize=(figsize, figsize))
plt.scatter(z_points[:, 0], z_points[:, 1], cmap='rainbow', c=example_labels, alpha=0.5, s=2)
plt.colorbar()

bad_examples = np.array([[ -5, -5], [0, 5], [0, 0], [6, -8]])
plt.scatter(bad_examples[:, 0], bad_examples[:, 1], c='black', alpha=1, s=20)

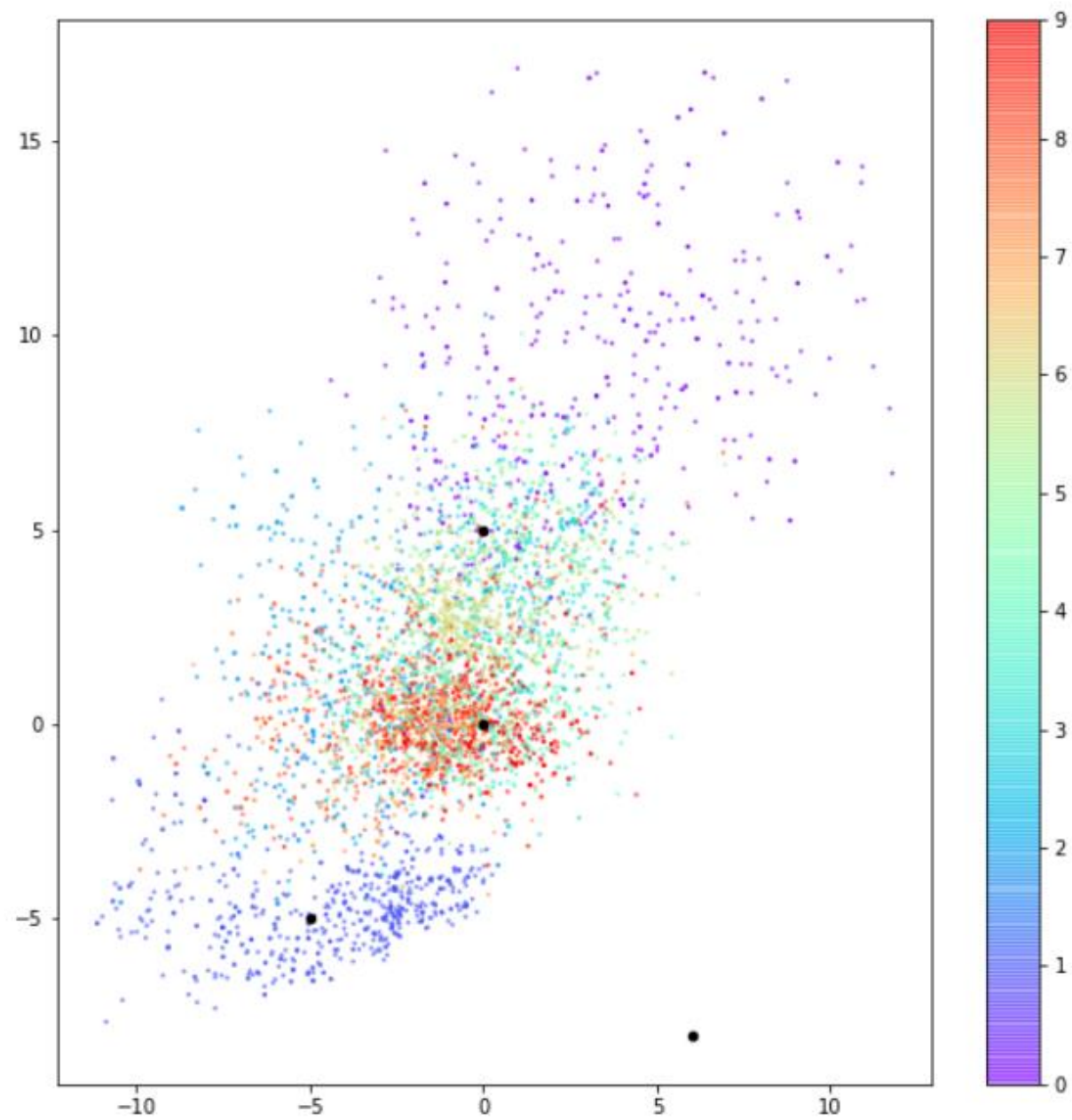
plt.show()

reconst = AE.decoder.predict(bad_examples)

fig = plt.figure(figsize=(figsize, grid_depth))
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(4):
    ax = fig.add_subplot(grid_depth, grid_size, i+1)
    ax.axis('off')

    ax.imshow(reconst[i, :, :, 0], cmap = 'Greys')
```



1 3 9 π_2

전체적으로 분포된 점 위치에서의 숫자 분포도

```
In [10]: n_to_show = 5000
grid_size = 20

example_idx = np.random.choice(range(len(x_test)), n_to_show)
example_images = x_test[example_idx]
example_labels = y_test[example_idx]

z_points = AE.encoder.predict(example_images)

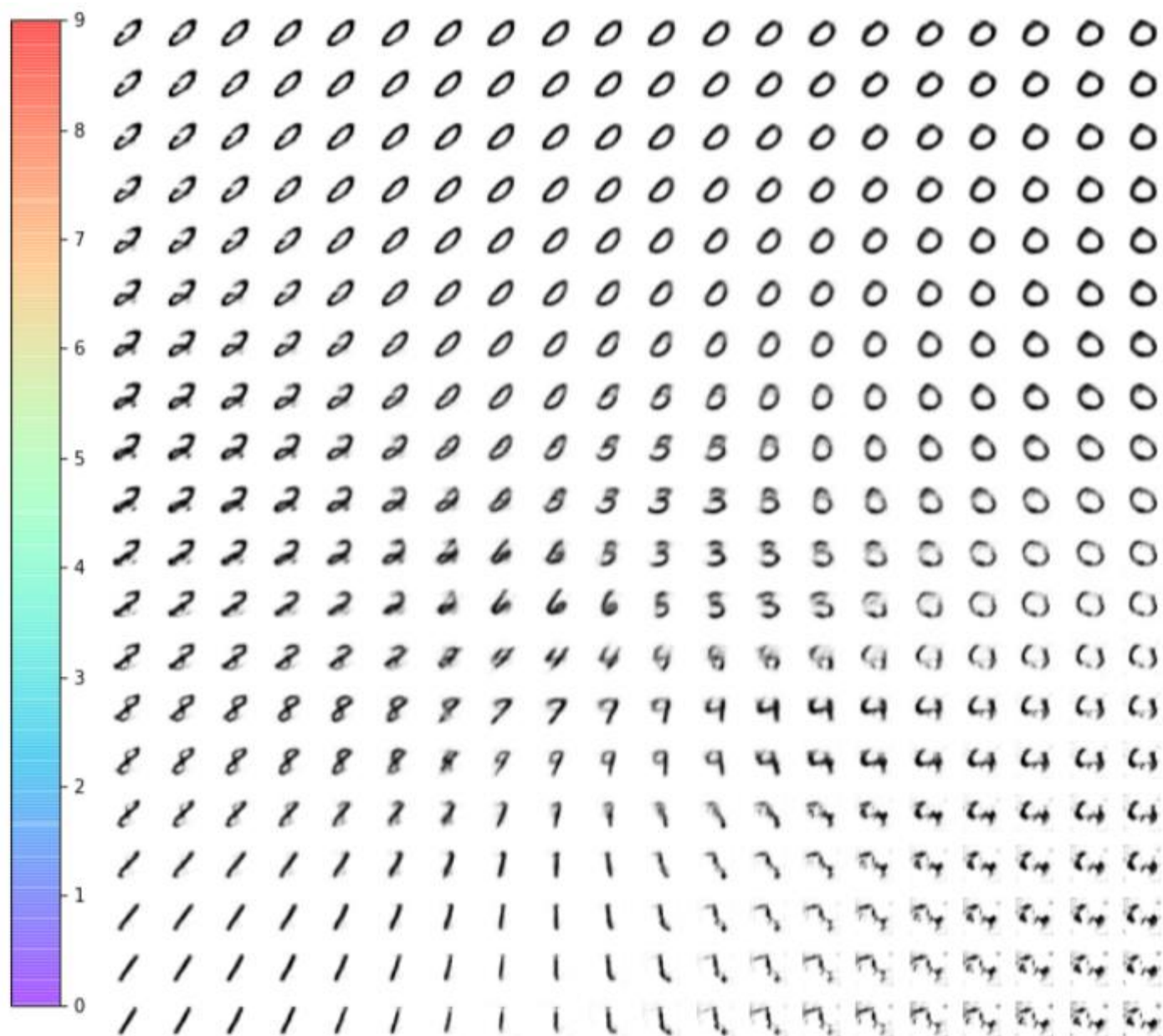
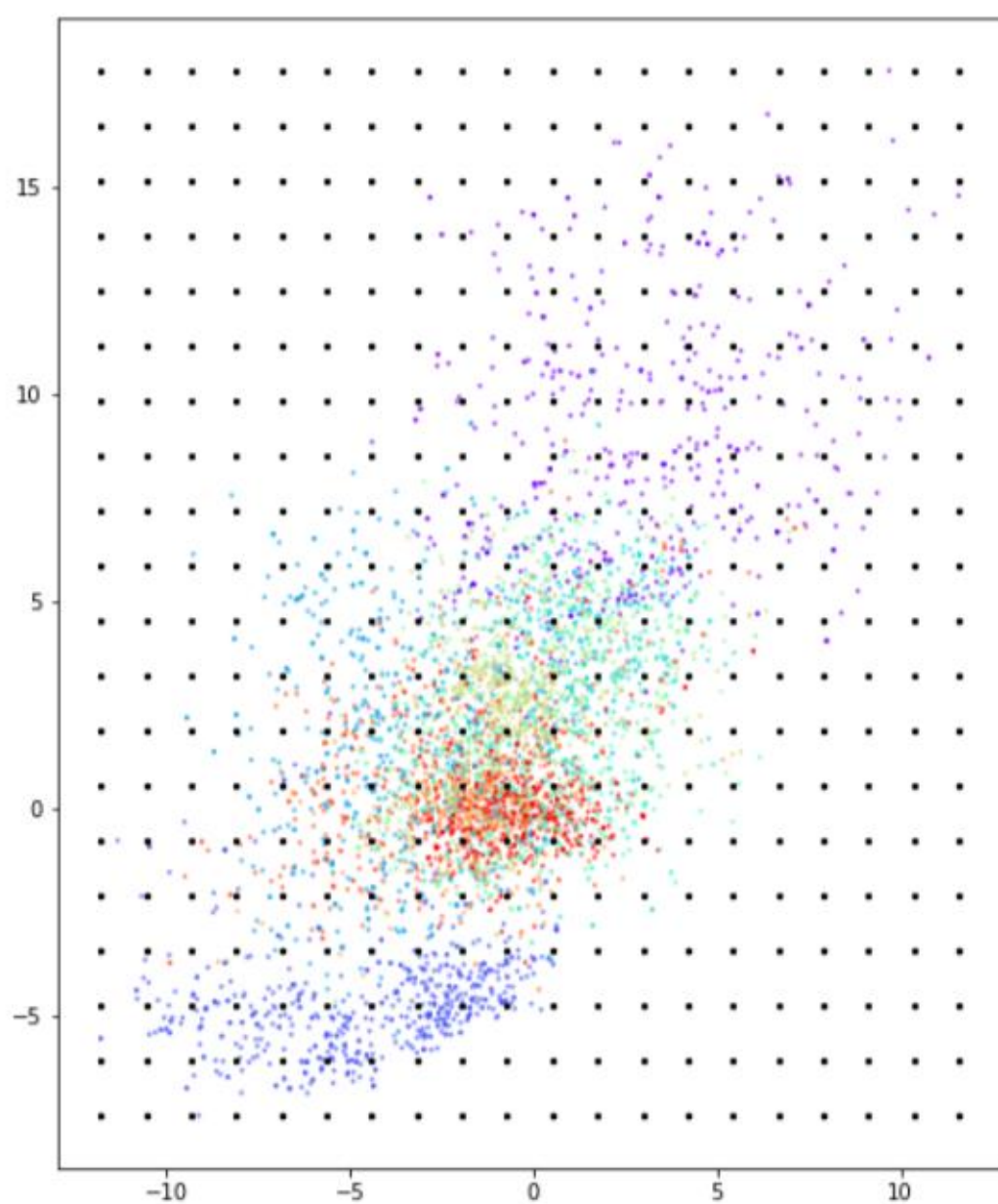
plt.figure(figsize=(figsize, figsize))
plt.scatter(z_points[:, 0], z_points[:, 1], cmap='rainbow', c=example_labels,
            alpha=0.5, s=2)
plt.colorbar()

# x = norm.ppf(np.linspace(0.05, 0.95, 10))
# y = norm.ppf(np.linspace(0.05, 0.95, 10))
x = np.linspace(min(z_points[:, 0]), max(z_points[:, 0]), grid_size)
y = np.linspace(max(z_points[:, 1]), min(z_points[:, 1]), grid_size)
xv, yv = np.meshgrid(x, y)
xv = xv.flatten()
yv = yv.flatten()
z_grid = np.array(list(zip(xv, yv)))

reconst = AE.decoder.predict(z_grid)

plt.scatter(z_grid[:, 0], z_grid[:, 1], c='black', #, cmap='rainbow', c=example_labels,
            alpha=1, s=5)
plt.show()

fig = plt.figure(figsize=(figsize, figsize))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
for i in range(grid_size**2):
    ax = fig.add_subplot(grid_size, grid_size, i+1)
    ax.axis('off')
    ax.imshow(reconst[i, :, :, 0], cmap='Greys')
```

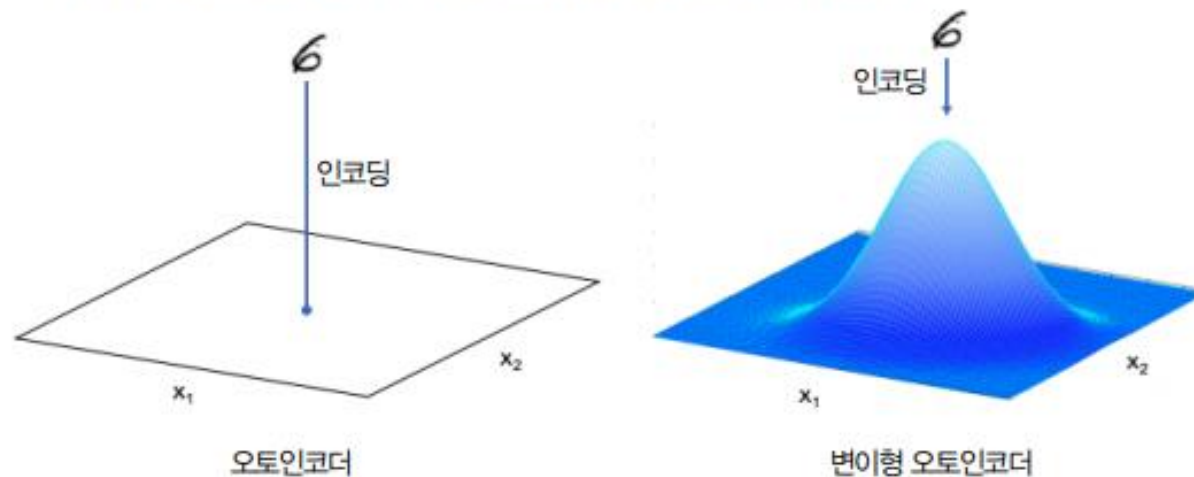
이제 VAE(변이형 오토인코더; Variation AutoEncoder)로 변경해 봅니다.

바꾸어야 할 부분은 인코더와 손실함수 딱 두 군데입니다.

AE에는 각 이미지가 잠재 공간의 한 포인트에 직접 매핑됩니다.

VAE는 아래 그림처럼 이미지가 잠재 공간에 있는 포인트 주변의 다변수 정규분포에 매핑됩니다.

그림 3-11 오토인코더(AE)와 변이형 오토인코더(VAE)에 있는 인코더의 차이점



라이브러리 импорт

```
In [1]: import os

from models.VAE import VariationalAutoencoder
from utils.loaders import load_mnist
```

Using TensorFlow backend.

```
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype [("qint8", np.int8, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint8 = np.dtype [("quint8", np.uint8, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype [("qint16", np.int16, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint16 = np.dtype [("quint16", np.uint16, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint32 = np.dtype [("qint32", np.int32, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    np_resource = np.dtype [("resource", np.ubyte, 1)]
```

```
In [2]: # 실행 매개변수
SECTION = 'vae'
RUN_ID = '0002'
DATA_NAME = 'digits'
RUN_FOLDER = 'run/{}/'.format(SECTION)
RUN_FOLDER += '_'.join([RUN_ID, DATA_NAME])

if not os.path.exists(RUN_FOLDER):
    os.mkdir(RUN_FOLDER)
    os.mkdir(os.path.join(RUN_FOLDER, 'viz'))
    os.mkdir(os.path.join(RUN_FOLDER, 'images'))
    os.mkdir(os.path.join(RUN_FOLDER, 'weights'))

mode = 'build' # 'load' #
```

데이터 적재

```
In [3]: (x_train, y_train), (x_test, y_test) = load_mnist()
```

모델 만들기

```
In [4]: vae = VariationalAutoencoder(
    input_dim = (28,28,1)
    , encoder_conv_filters = [32,64,64, 64]
    , encoder_conv_kernel_size = [3,3,3,3]
    , encoder_conv_strides = [1,2,2,1]
    , decoder_conv_t_filters = [64,64,32,1]
    , decoder_conv_t_kernel_size = [3,3,3,3]
    , decoder_conv_t_strides = [1,2,2,1]
    , z_dim = 2
)

if mode == 'build':
    vae.save(RUN_FOLDER)
else:
    vae.load_weights(os.path.join(RUN_FOLDER, 'weights/weights.h5'))
```

WARNING:tensorflow:From C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.

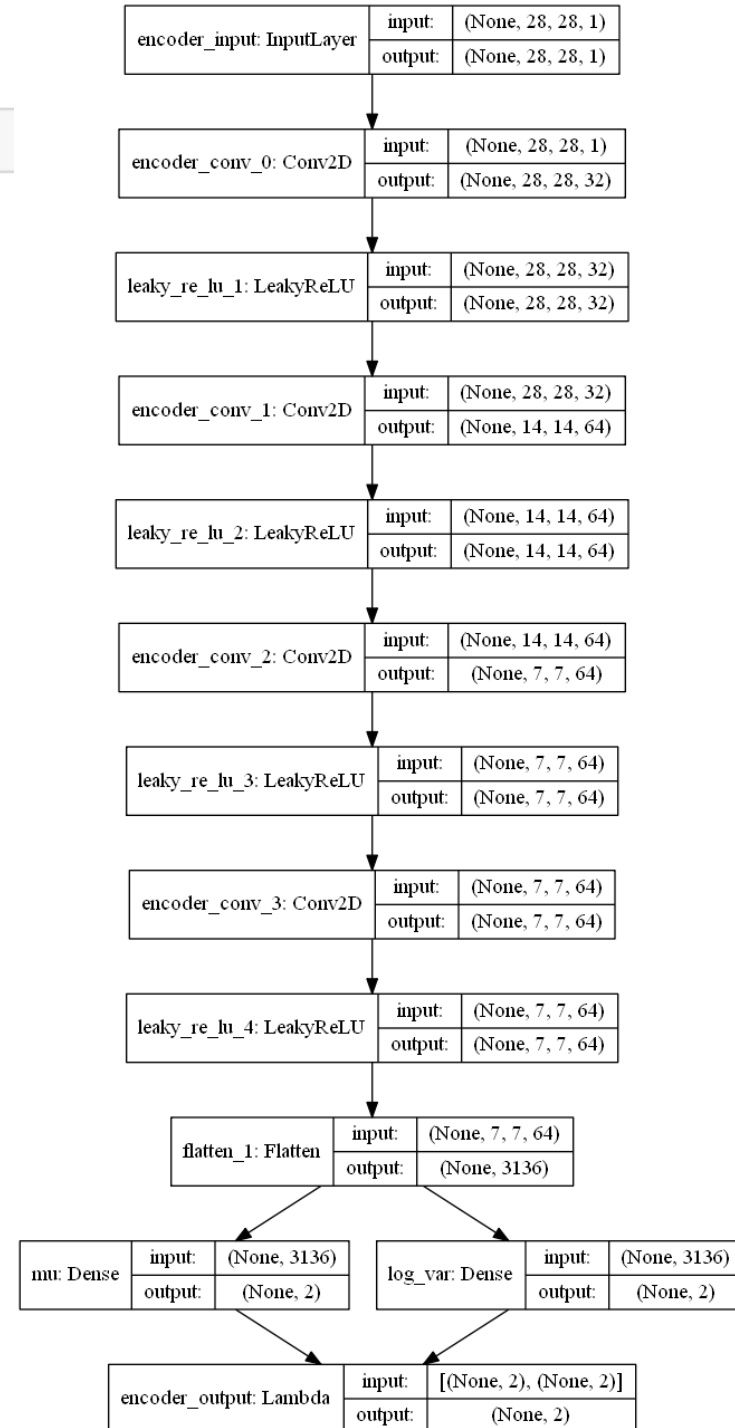
Instructions for updating:

Colocations handled automatically by placer.

인코더 모양

In [5]: `vae.encoder.summary()`

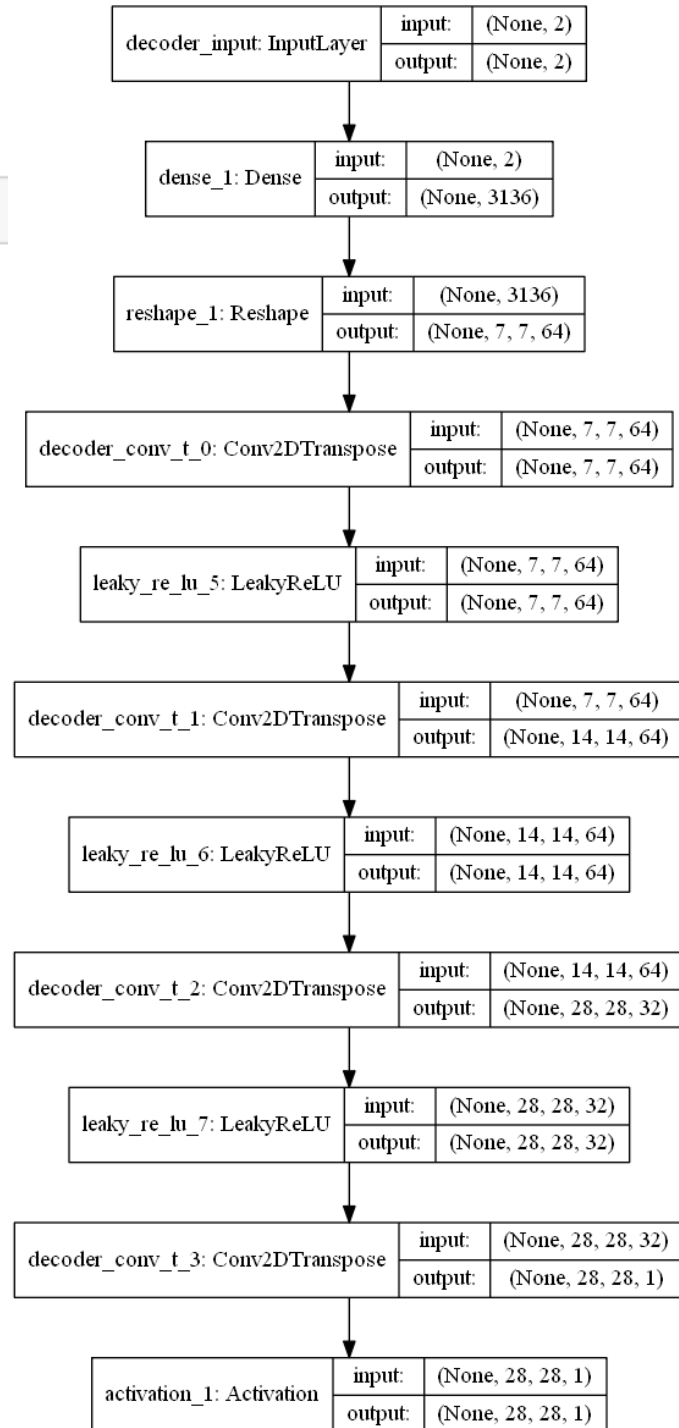
Layer (type)	Output Shape	Param #	Connected to
encoder_input (InputLayer)	(None, 28, 28, 1)	0	
encoder_conv_0 (Conv2D)	(None, 28, 28, 32)	320	encoder_input[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 28, 28, 32)	0	encoder_conv_0[0][0]
encoder_conv_1 (Conv2D)	(None, 14, 14, 64)	18496	leaky_re_lu_1[0][0]
leaky_re_lu_2 (LeakyReLU)	(None, 14, 14, 64)	0	encoder_conv_1[0][0]
encoder_conv_2 (Conv2D)	(None, 7, 7, 64)	36928	leaky_re_lu_2[0][0]
leaky_re_lu_3 (LeakyReLU)	(None, 7, 7, 64)	0	encoder_conv_2[0][0]
encoder_conv_3 (Conv2D)	(None, 7, 7, 64)	36928	leaky_re_lu_3[0][0]
leaky_re_lu_4 (LeakyReLU)	(None, 7, 7, 64)	0	encoder_conv_3[0][0]
flatten_1 (Flatten)	(None, 3136)	0	leaky_re_lu_4[0][0]
mu (Dense)	(None, 2)	6274	flatten_1[0][0]
log_var (Dense)	(None, 2)	6274	flatten_1[0][0]
encoder_output (Lambda)	(None, 2)	0	mu[0][0] log_var[0][0]
Total params: 105,220			
Trainable params: 105,220			
Non-trainable params: 0			



디코더 모양

In [7]: `vae.decoder.summary()`

Layer (type)	Output Shape	Param #
decoder_input (InputLayer)	(None, 2)	0
dense_1 (Dense)	(None, 3136)	9408
reshape_1 (Reshape)	(None, 7, 7, 64)	0
decoder_conv_t_0 (Conv2DTran	(None, 7, 7, 64)	36928
leaky_re_lu_5 (LeakyReLU)	(None, 7, 7, 64)	0
decoder_conv_t_1 (Conv2DTran	(None, 14, 14, 64)	36928
leaky_re_lu_6 (LeakyReLU)	(None, 14, 14, 64)	0
decoder_conv_t_2 (Conv2DTran	(None, 28, 28, 32)	18464
leaky_re_lu_7 (LeakyReLU)	(None, 28, 28, 32)	0
decoder_conv_t_3 (Conv2DTran	(None, 28, 28, 1)	289
activation_1 (Activation)	(None, 28, 28, 1)	0
Total params: 102,017		
Trainable params: 102,017		
Non-trainable params: 0		



모델 훈련

```
In [7]: LEARNING_RATE = 0.0005  
        R_LOSS_FACTOR = 1000
```

```
In [8]: vae.compile(LEARNING_RATE, R_LOSS_FACTOR)
```

```
In [9]: BATCH_SIZE = 50  
        EPOCHS = 200  
        PRINT_EVERY_N_BATCHES = 100  
        INITIAL_EPOCH = 0
```

```
In [10]: vae.train(  
        x_train  
        , batch_size = BATCH_SIZE  
        , epochs = EPOCHS  
        , run_folder = RUN_FOLDER  
        , print_every_n_batches = PRINT_EVERY_N_BATCHES  
        , initial_epoch = INITIAL_EPOCH  
        )
```

Epoch 00197: saving model to run/vae/0002_digits#weights/weights.h5

Epoch 198/200

60000/60000 [=====] - 243s 4ms/step - loss: 42.6997 - vae_r_loss: 37.0541 - vae_kl_loss: 5.6456

Epoch 00198: saving model to run/vae/0002_digits#weights/weights-198-42.70.h5

Epoch 00198: saving model to run/vae/0002_digits#weights/weights.h5

Epoch 199/200

60000/60000 [=====] - 251s 4ms/step - loss: 42.6619 - vae_r_loss: 37.0226 - vae_kl_loss: 5.6394

Epoch 00199: saving model to run/vae/0002_digits#weights/weights-199-42.66.h5

Epoch 00199: saving model to run/vae/0002_digits#weights/weights.h5

Epoch 200/200

60000/60000 [=====] - 259s 4ms/step - loss: 42.6702 - vae_r_loss: 37.0302 - vae_kl_loss: 5.6400

Epoch 00200: saving model to run/vae/0002_digits#weights/weights-200-42.67.h5

Epoch 00200: saving model to run/vae/0002_digits#weights/weights.h5

변이형 오토인코더(VAE) 분석

라이브러리 임포트

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import os
from scipy.stats import norm

from models.VAE import VariationalAutoencoder
from utils.loaders import load_mnist, load_model
```

Using TensorFlow backend.

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint16 = np.dtype [("quint16", np.uint16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint32 = np.dtype [("qint32", np.int32, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
np_resource = np.dtype [("resource", np.ubyte, 1)]
```

```
In [2]: # run params
SECTION = 'vae'
RUN_ID = '0002'
DATA_NAME = 'digits'
RUN_FOLDER = 'run/{}/'.format(SECTION)
RUN_FOLDER += '_'.join([RUN_ID, DATA_NAME])
```


데이터 적재

```
In [3]: (x_train, y_train), (x_test, y_test) = load_mnist()
```

모델 만들기

```
In [4]: vae = load_model(VariationalAutoencoder, RUN_FOLDER)
```

WARNING:tensorflow:From C:\Users\User\anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.

원본 그림 재구성

```
In [5]: n_to_show = 10
np.random.seed(88)
example_idx = np.random.choice(range(len(x_test)), n_to_show)
example_images = x_test[example_idx]

z_points = vae.encoder.predict(example_images)

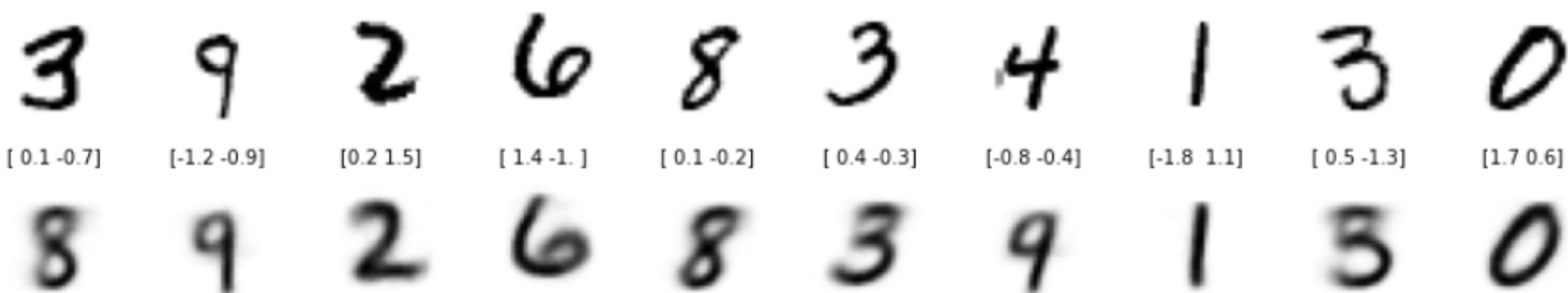
reconst_images = vae.decoder.predict(z_points)

fig = plt.figure(figsize=(15, 3))
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(n_to_show):
    img = example_images[i].squeeze()
    sub = fig.add_subplot(2, n_to_show, i+1)
    sub.axis('off')
    sub.text(0.5, -0.35, str(np.round(z_points[i],1)), fontsize=10, ha='center', transform=sub.transAxes)

    sub.imshow(img, cmap='gray_r')

for i in range(n_to_show):
    img = reconst_images[i].squeeze()
    sub = fig.add_subplot(2, n_to_show, i+n_to_show+1)
    sub.axis('off')
    sub.imshow(img, cmap='gray_r')
```



엔 코더 씨의 전시벽

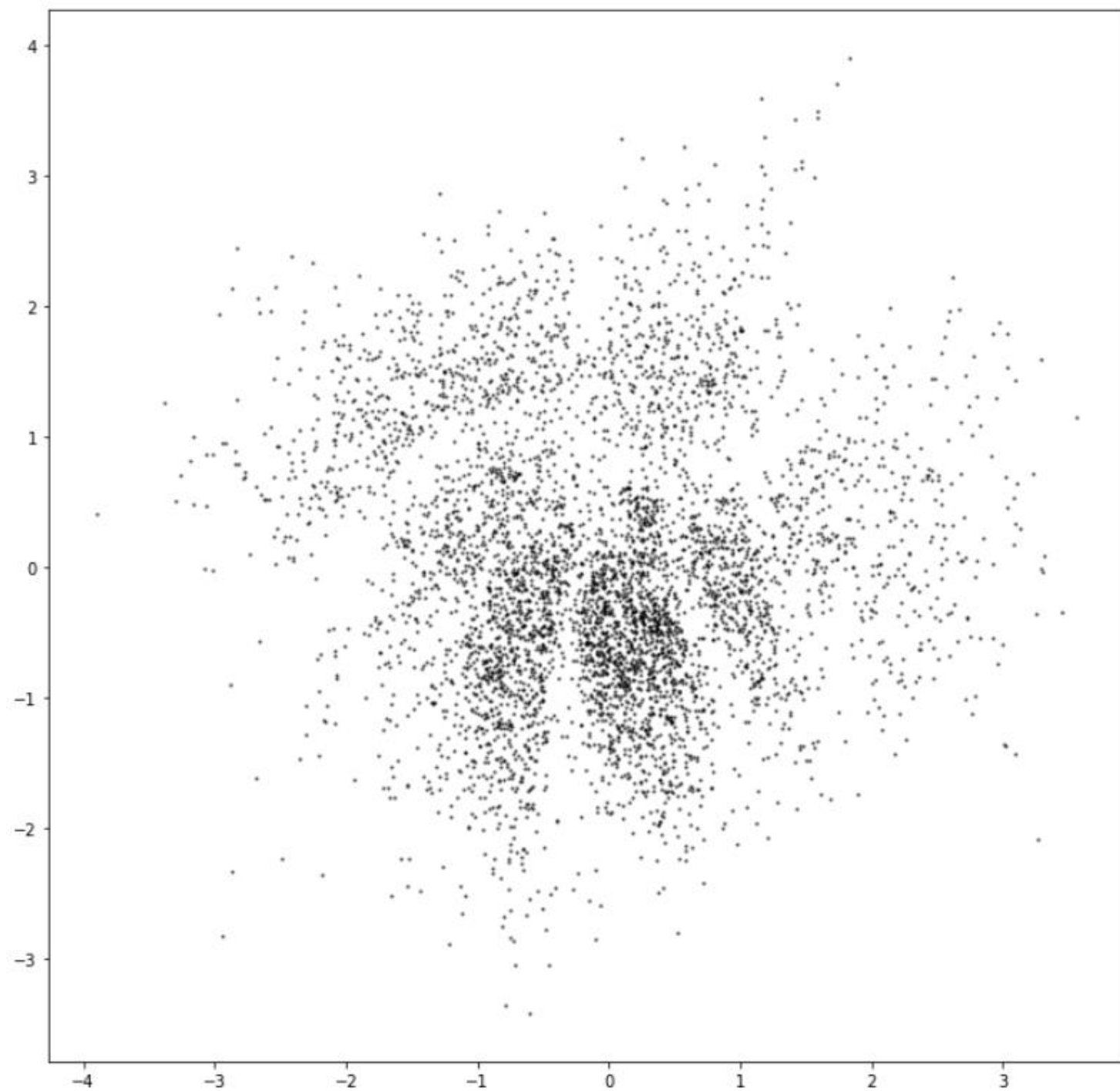
```
In [6]: n_to_show = 5000
        figsize = 12

        example_idx = np.random.choice(range(len(x_test)), n_to_show)
        example_images = x_test[example_idx]
        example_labels = y_test[example_idx]

        z_points = vae.encoder.predict(example_images)

        min_x = min(z_points[:, 0])
        max_x = max(z_points[:, 0])
        min_y = min(z_points[:, 1])
        max_y = max(z_points[:, 1])

        plt.figure(figsize=(figsize, figsize))
        plt.scatter(z_points[:, 0], z_points[:, 1], c='black', alpha=0.5, s=2)
        plt.show()
```



새로 생성한 미술 전시품

```
In [7]: figsize = 10
plt.figure(figsize=(figsize, figsize))
plt.scatter(z_points[:, 0], z_points[:, 1], c='black', alpha=0.5, s=2)

grid_size = 15
grid_depth = 2
figsize = 15

x = np.random.normal(size = grid_size * grid_depth)
y = np.random.normal(size = grid_size * grid_depth)

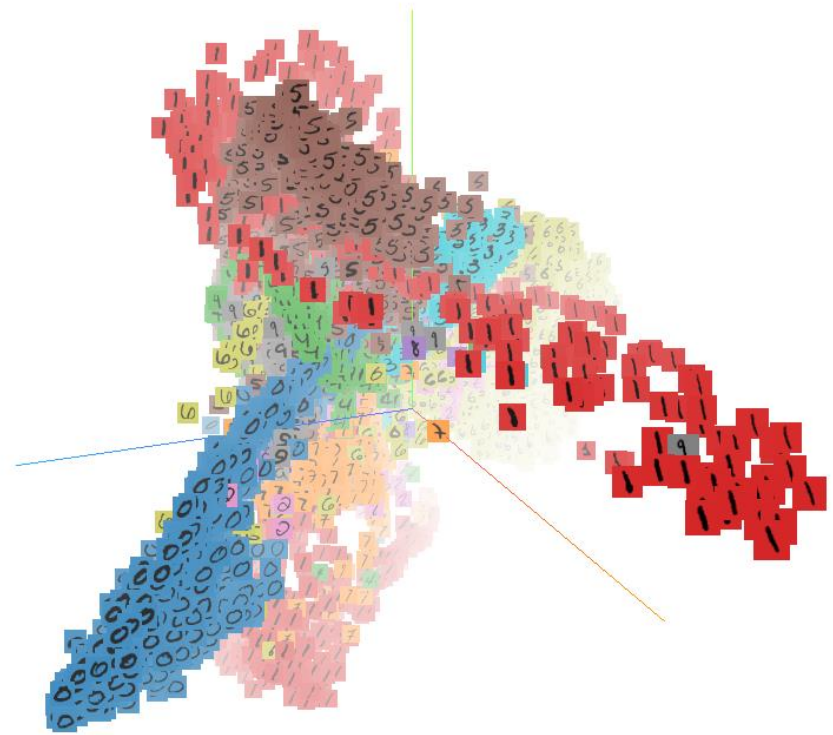
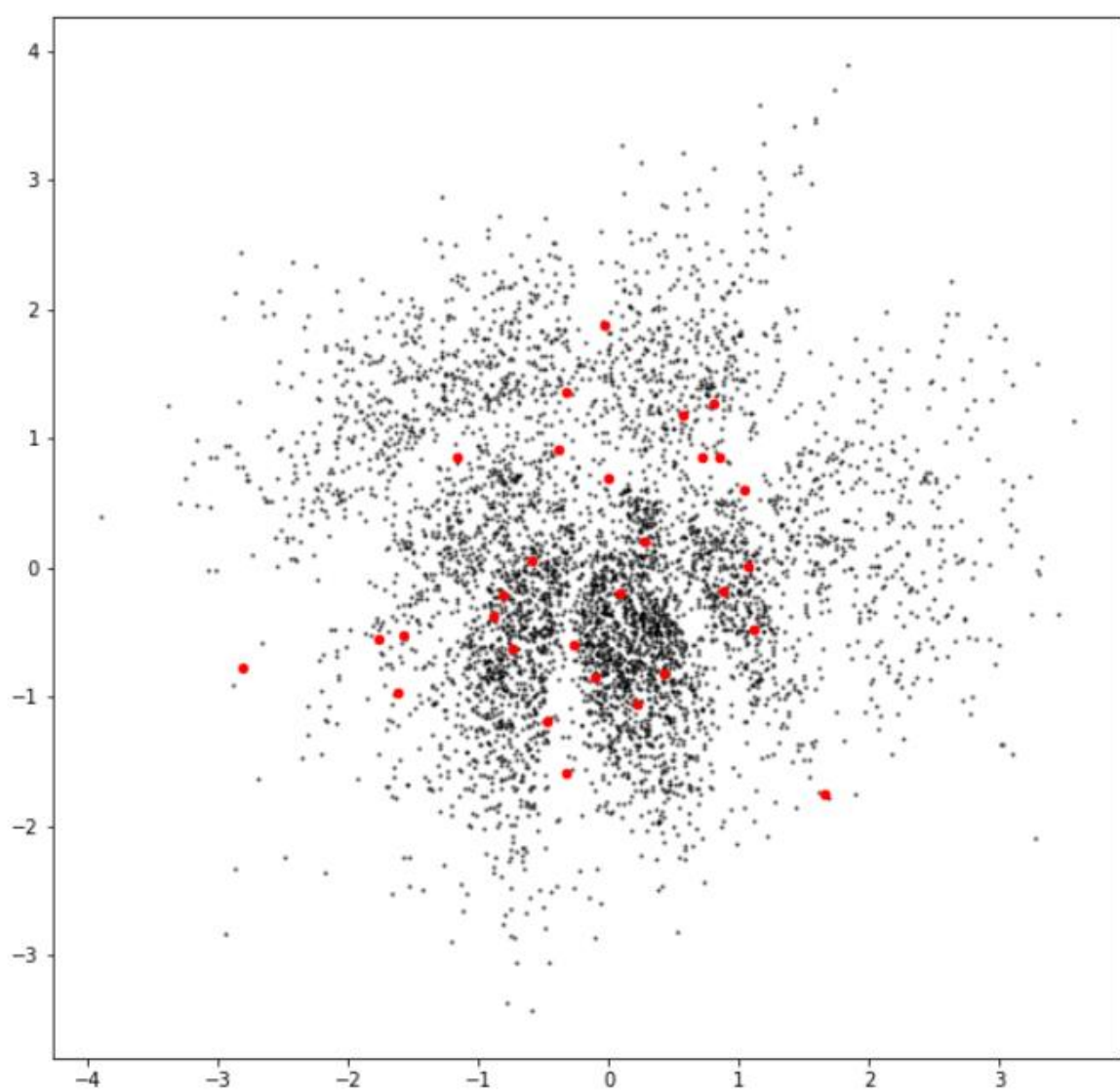
z_grid = np.array(list(zip(x, y)))
reconst = vae.decoder.predict(z_grid)

plt.scatter(z_grid[:, 0], z_grid[:, 1], c = 'red', alpha=1, s=20)
plt.show()

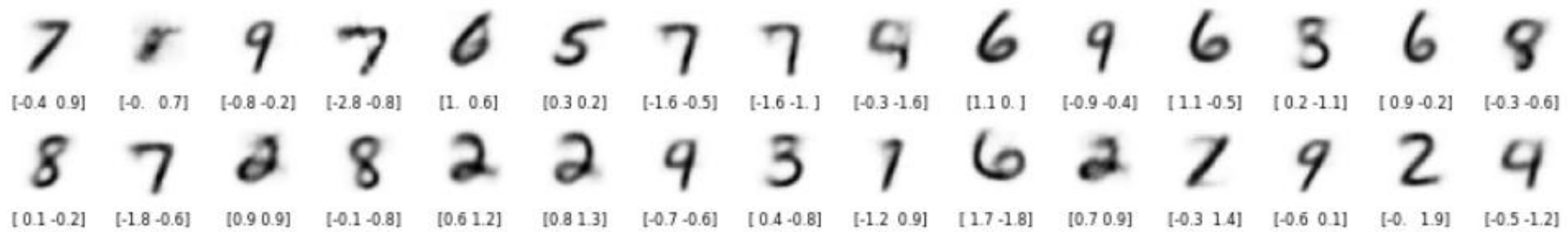
fig = plt.figure(figsize=(figsize, grid_depth))
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(grid_size*grid_depth):
    ax = fig.add_subplot(grid_depth, grid_size, i+1)
    ax.axis('off')
    ax.text(0.5, -0.35, str(np.round(z_grid[i],1)), fontsize=8, ha='center', transform=ax.transAxes)

    ax.imshow(reconst[i, :, :, 0], cmap = 'Greys') ## 숫자별 색깔 산점도 (scatter plot)
```



3차원 공간의 표현 (내용과 무관)



숫자별 색깔 산점도 (scatter plot)

```
In [8]: n_to_show = 5000
        grid_size = 15
        fig_height = 7
        fig_width = 15

        example_idx = np.random.choice(range(len(x_test)), n_to_show)
        example_images = x_test[example_idx]
        example_labels = y_test[example_idx]

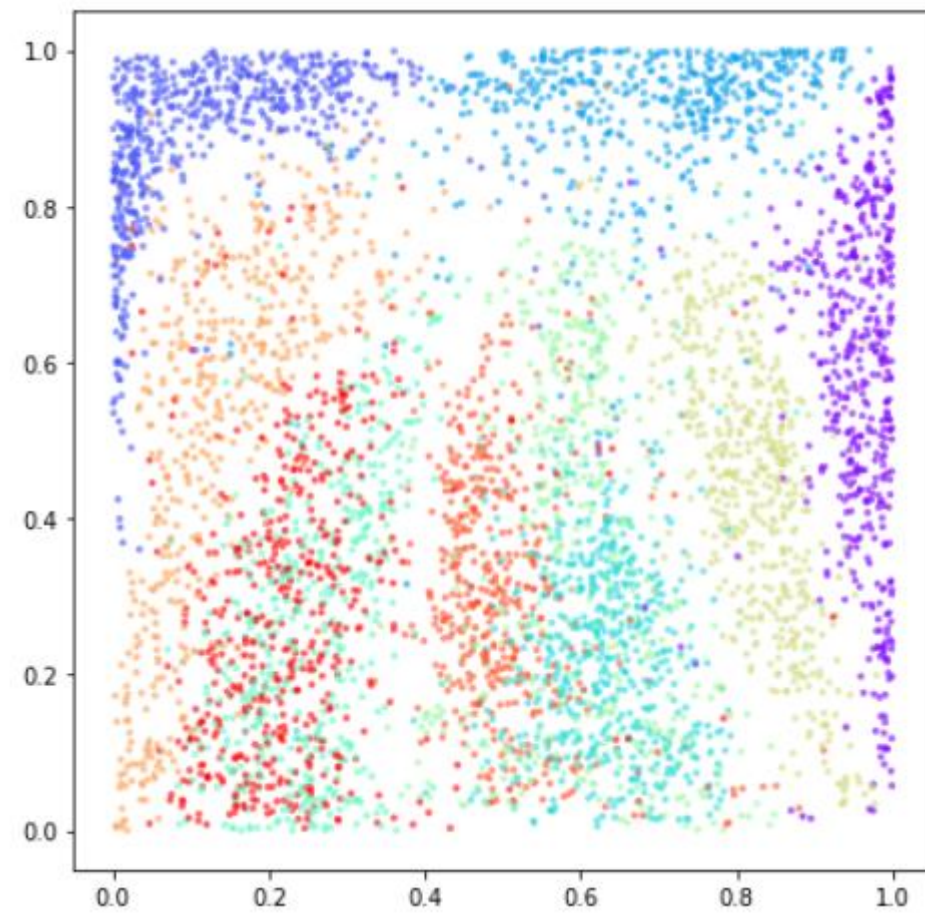
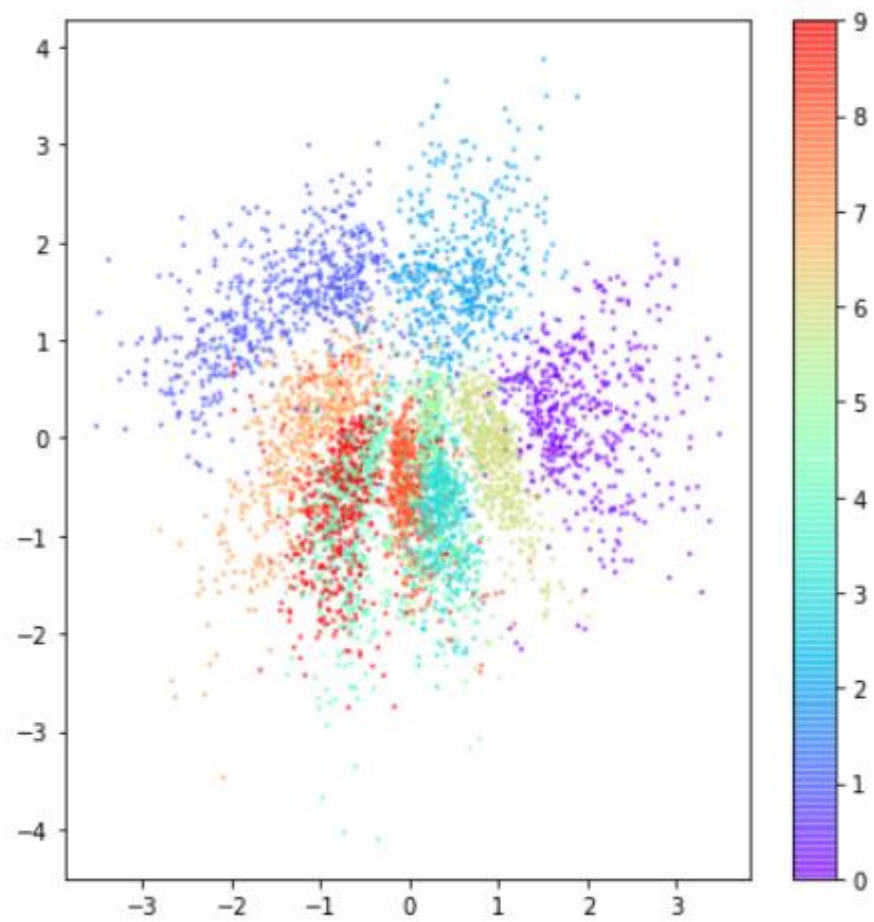
        z_points = vae.encoder.predict(example_images)
        p_points = norm.cdf(z_points)

        fig = plt.figure(figsize=(fig_width, fig_height))

        ax = fig.add_subplot(1, 2, 1)
        plot_1 = ax.scatter(z_points[:, 0], z_points[:, 1], cmap='rainbow', c=example_labels,
                           , alpha=0.5, s=2)
        plt.colorbar(plot_1)

        ax = fig.add_subplot(1, 2, 2)
        plot_2 = ax.scatter(p_points[:, 0], p_points[:, 1], cmap='rainbow', c=example_labels,
                           , alpha=0.5, s=5)

        plt.show()## 전체적으로 분포된 점 위치에서의 숫자 분포도
```



전체적으로 분포된 점 위치에서의 숫자 분포도

```
In [9]: n_to_show = 5000
        grid_size = 20
        figsize = 10

        example_idx = np.random.choice(range(len(x_test)), n_to_show)
        example_images = x_test[example_idx]
        example_labels = y_test[example_idx]

        z_points = vae.encoder.predict(example_images)

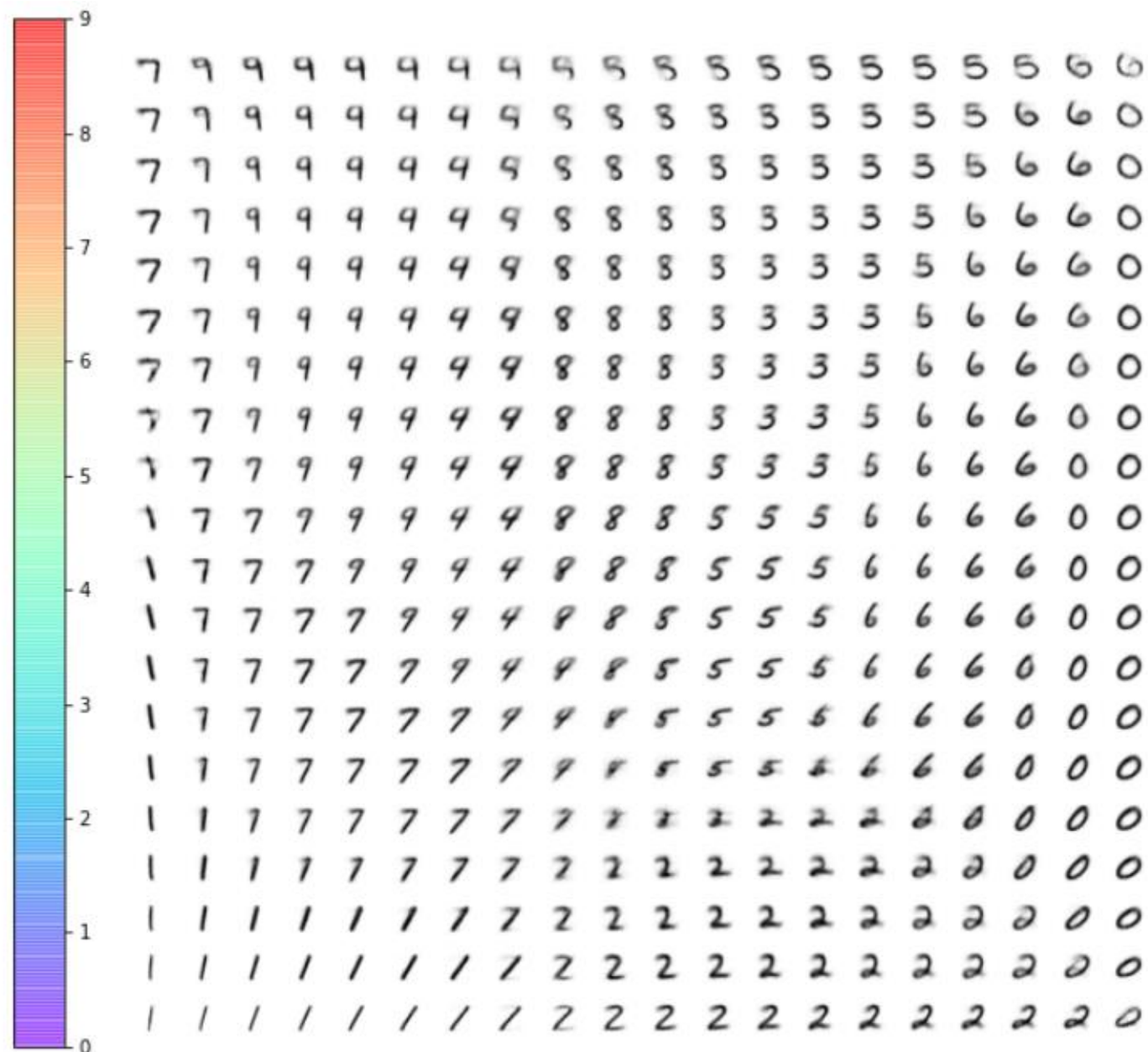
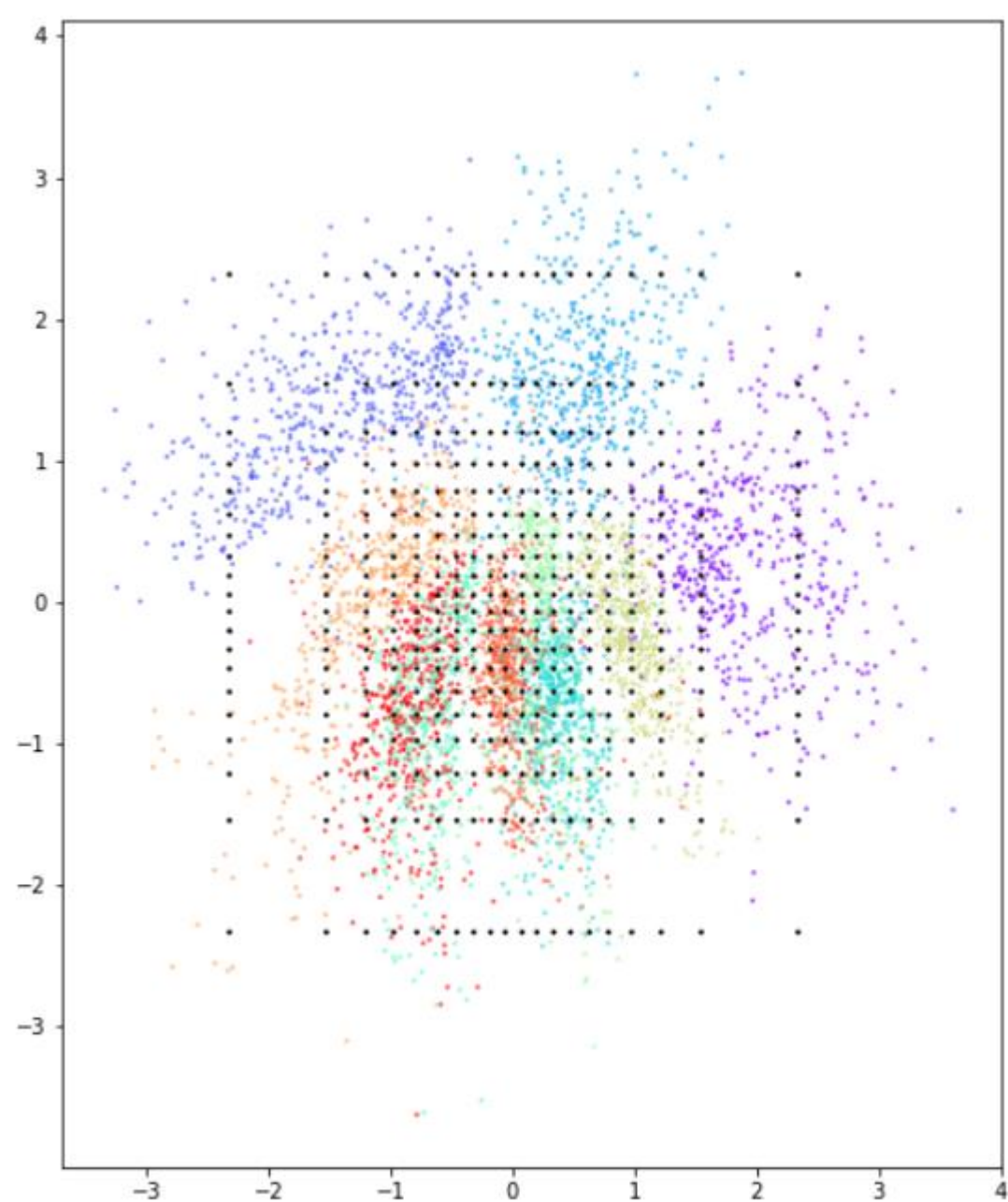
        plt.figure(figsize=(figsize, figsize))
        plt.scatter(z_points[:, 0], z_points[:, 1], cmap='rainbow', c=example_labels,
                    , alpha=0.5, s=2)
        plt.colorbar()

        x = norm.ppf(np.linspace(0.01, 0.99, grid_size))
        y = norm.ppf(np.linspace(0.01, 0.99, grid_size))
        xv, yv = np.meshgrid(x, y)
        xv = xv.flatten()
        yv = yv.flatten()
        z_grid = np.array(list(zip(xv, yv)))

        reconst = vae.decoder.predict(z_grid)

        plt.scatter(z_grid[:, 0], z_grid[:, 1], c='black', #, cmap='rainbow', c=example_labels,
                    , alpha=1, s=2)
        plt.show()

        fig = plt.figure(figsize=(figsize, figsize))
        fig.subplots_adjust(hspace=0.4, wspace=0.4)
        for i in range(grid_size**2):
            ax = fig.add_subplot(grid_size, grid_size, i+1)
            ax.axis('off')
            ax.imshow(reconst[i, :, :, 0], cmap='Greys')
```



변이형 오토인코더 훈련 - 얼굴 데이터셋

라이브러리 импорт

```
In [1]: import os
from glob import glob
import numpy as np

from models.VAE import VariationalAutoencoder
from keras.preprocessing.image import ImageDataGenerator
```

Using TensorFlow backend.

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint16 = np.dtype [("quint16", np.uint16, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint32 = np.dtype [("qint32", np.int32, 1)]
```

C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
np_resource = np.dtype [("resource", np.ubyte, 1)]
```

```
In [2]: # run params
section = 'vae'
run_id = '0001'
data_name = 'faces'
RUN_FOLDER = 'run/{}/'.format(section)
RUN_FOLDER += '_'.join([run_id, data_name])

if not os.path.exists(RUN_FOLDER):
    os.mkdir(RUN_FOLDER)
    os.mkdir(os.path.join(RUN_FOLDER, 'viz'))
    os.mkdir(os.path.join(RUN_FOLDER, 'images'))
    os.mkdir(os.path.join(RUN_FOLDER, 'weights'))

mode = 'build' #'load' #

DATA_FOLDER = './data/celeb/'
```

데이터 적재

CelebA 사이트에서 `img_align_celeba.zip` 파일을 다운로드 받은 후 `data/celeb/` 폴더 안에서 압축을 해제하세요. `data/celeb/img_align_celeba` 폴더에 이미지가 저장되어야 합니다.

`list_attr_celeba.csv` 파일은 깃허브에 포함되어 있으므로 다운로드 받을 필요가 없습니다.

```
In [3]: INPUT_DIM = (128,128,3)
        BATCH_SIZE = 32

        filenames = np.array(glob(os.path.join(DATA_FOLDER, '*/*.jpg')))

        NUM_IMAGES = len(filenames)
```

```
In [4]: data_gen = ImageDataGenerator(rescale=1./255)

        data_flow = data_gen.flow_from_directory(DATA_FOLDER
                                                , target_size = INPUT_DIM[:2]
                                                , batch_size = BATCH_SIZE
                                                , shuffle = True
                                                , class_mode = 'input'
                                                , subset = "training"
                                                )
```

Found 202599 images belonging to 1 classes.

모델 만들기

```
In [5]: vae = VariationalAutoencoder(
        input_dim = INPUT_DIM
        , encoder_conv_filters=[32,64,64, 64]
        , encoder_conv_kernel_size=[3,3,3,3]
        , encoder_conv_strides=[2,2,2,2]
        , decoder_conv_t_filters=[64,64,32,3]
        , decoder_conv_t_kernel_size=[3,3,3,3]
        , decoder_conv_t_strides=[2,2,2,2]
        , z_dim=200
        , use_batch_norm=True
        , use_dropout=True)

if mode == 'build':
    vae.save(RUN_FOLDER)
else:
    vae.load_weights(os.path.join(RUN_FOLDER, 'weights/weights.h5'))
```

WARNING:tensorflow:From C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.

Instructions for updating:

Colocations handled automatically by placer.

WARNING:tensorflow:From C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.

Instructions for updating:

Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.

```

FileNotFoundError                                Traceback (most recent call last)
<ipython-input-5-32454e8f9bb7> in <module>
    12
    13 if mode == 'build':
--> 14     vae.save(RUN_FOLDER)
    15 else:
    16     vae.load_weights(os.path.join(RUN_FOLDER, 'weights/weights.h5'))

~\VAE\models\VAE.py in save(self, folder)
    176         ], f)
    177
--> 178     self.plot_model(folder)
    179
    180

~\VAE\models\VAE.py in plot_model(self, run_folder)
    230
    231     def plot_model(self, run_folder):
--> 232         plot_model(self.model, to_file=os.path.join(run_folder, 'viz/model.png'), show_shapes = True, show_layer_names = True)
    233         plot_model(self.encoder, to_file=os.path.join(run_folder, 'viz/encoder.png'), show_shapes = True, show_layer_names = True)
e)
    234         plot_model(self.decoder, to_file=os.path.join(run_folder, 'viz/decoder.png'), show_shapes = True, show_layer_names = True)
e)

~\anaconda3\envs\testGAN\lib\site-packages\keras\tutils\vis_utils.py in plot_model(model, to_file, show_shapes, show_layer_names, rankd
ir)
    136     else:
    137         extension = extension[1:]
--> 138     dot.write(to_file, format=extension)

~\anaconda3\envs\testGAN\lib\site-packages\pydot.py in write(self, path, prog, format, encoding)
    1816     else:
    1817         s = self.create(prog, format, encoding=encoding)
-> 1818         with io.open(path, mode='wb') as f:
    1819             f.write(s)
    1820     return True

```

FileNotFoundError: [Errno 2] No such file or directory: 'run/vae/0001_faces\\viz/model.png'

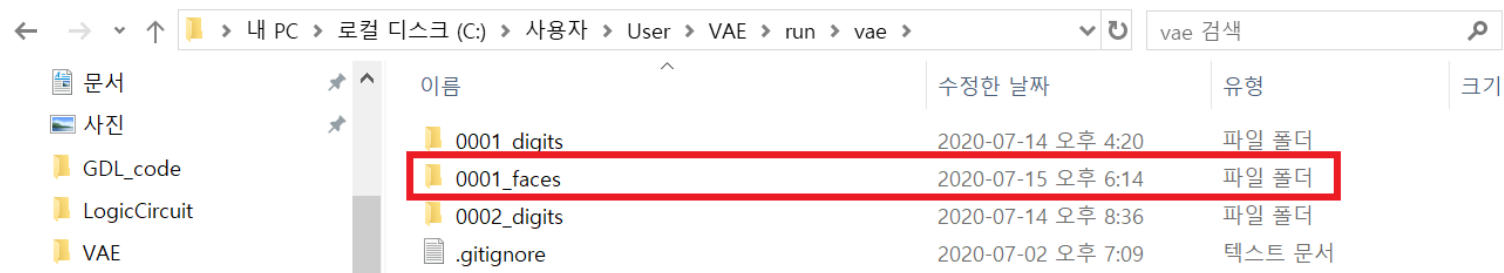
여기서 위와 같은 오류가 발생할 수 있는데,

이는 파일이나 디렉토리를 찾을 수 없어서 발생하는 오류로 run 파일을 그대로 복사해 왔기 때문에 발생합니다.

C:\Users\User\VAE\run\vae 이나 본인이 run 파일을 복사 해 놓은 곳의 vae 파일에서

0001_faces 파일을 삭제해 준 후 커널을 켜다운 후에 다시 시작을 합니다.

44번 슬라이드 혹은 라인 2번에서 다시 파일을 생성 할 테니 과감하게 삭제해도 무방합니다.

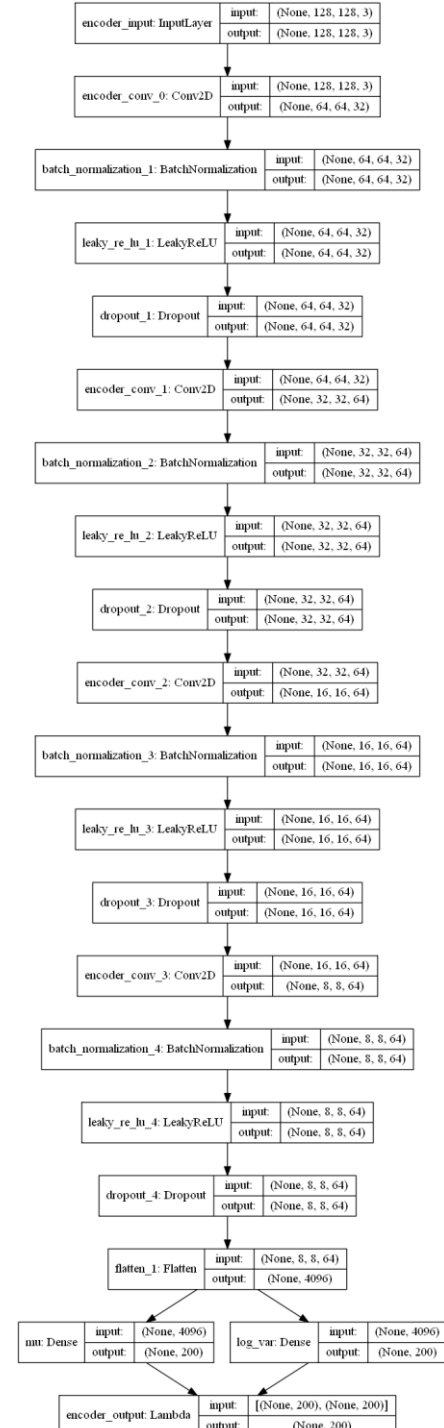


인코더 모양

```
In [6]: vae.encoder.summary()
```

Layer (type)	Output Shape	Param #	Connected to
encoder_input (InputLayer)	(None, 128, 128, 3)	0	
encoder_conv_0 (Conv2D)	(None, 64, 64, 32)	896	encoder_input[0][0]
batch_normalization_1 (BatchNor	(None, 64, 64, 32)	128	encoder_conv_0[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 64, 64, 32)	0	batch_normalization_1[0][0]
dropout_1 (Dropout)	(None, 64, 64, 32)	0	leaky_re_lu_1[0][0]
encoder_conv_1 (Conv2D)	(None, 32, 32, 64)	18496	dropout_1[0][0]
batch_normalization_2 (BatchNor	(None, 32, 32, 64)	256	encoder_conv_1[0][0]
leaky_re_lu_2 (LeakyReLU)	(None, 32, 32, 64)	0	batch_normalization_2[0][0]
dropout_2 (Dropout)	(None, 32, 32, 64)	0	leaky_re_lu_2[0][0]
encoder_conv_2 (Conv2D)	(None, 16, 16, 64)	36928	dropout_2[0][0]
batch_normalization_3 (BatchNor	(None, 16, 16, 64)	256	encoder_conv_2[0][0]
leaky_re_lu_3 (LeakyReLU)	(None, 16, 16, 64)	0	batch_normalization_3[0][0]
dropout_3 (Dropout)	(None, 16, 16, 64)	0	leaky_re_lu_3[0][0]
encoder_conv_3 (Conv2D)	(None, 8, 8, 64)	36928	dropout_3[0][0]
batch_normalization_4 (BatchNor	(None, 8, 8, 64)	256	encoder_conv_3[0][0]
leaky_re_lu_4 (LeakyReLU)	(None, 8, 8, 64)	0	batch_normalization_4[0][0]
dropout_4 (Dropout)	(None, 8, 8, 64)	0	leaky_re_lu_4[0][0]
flatten_1 (Flatten)	(None, 4096)	0	dropout_4[0][0]
mu (Dense)	(None, 200)	819400	flatten_1[0][0]
log_var (Dense)	(None, 200)	819400	flatten_1[0][0]
encoder_output (Lambda)	(None, 200)	0	mu[0][0] log_var[0][0]

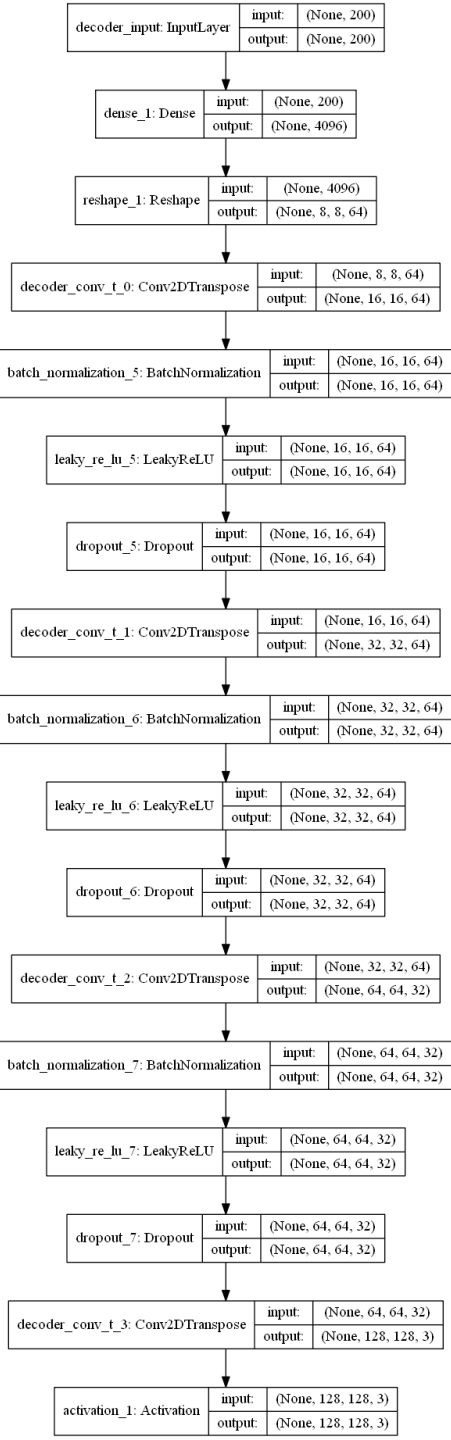
=====
Total params: 1,732,944
Trainable params: 1,732,496
Non-trainable params: 448
=====



디코더 모양

```
In [7]: vae.decoder.summary()
```

Layer (type)	Output Shape	Param #
decoder_input (InputLayer)	(None, 200)	0
dense_1 (Dense)	(None, 4096)	823296
reshape_1 (Reshape)	(None, 8, 8, 64)	0
decoder_conv_t_0 (Conv2DTran	(None, 16, 16, 64)	36928
batch_normalization_5 (Batch	(None, 16, 16, 64)	256
leaky_re_lu_5 (LeakyReLU)	(None, 16, 16, 64)	0
dropout_5 (Dropout)	(None, 16, 16, 64)	0
decoder_conv_t_1 (Conv2DTran	(None, 32, 32, 64)	36928
batch_normalization_6 (Batch	(None, 32, 32, 64)	256
leaky_re_lu_6 (LeakyReLU)	(None, 32, 32, 64)	0
dropout_6 (Dropout)	(None, 32, 32, 64)	0
decoder_conv_t_2 (Conv2DTran	(None, 64, 64, 32)	18464
batch_normalization_7 (Batch	(None, 64, 64, 32)	128
leaky_re_lu_7 (LeakyReLU)	(None, 64, 64, 32)	0
dropout_7 (Dropout)	(None, 64, 64, 32)	0
decoder_conv_t_3 (Conv2DTran	(None, 128, 128, 3)	867
activation_1 (Activation)	(None, 128, 128, 3)	0
Total params: 917,123		
Trainable params: 916,803		
Non-trainable params: 320		



모델 훈련

주의: 이 훈련은 시간이 오래 걸립니다. 깃허브에 훈련된 모델이 포함되어 있으므로 아래 셀에서 VAE를 직접 훈련하지 않아도 `03_06_vae_faces_analysis.ipynb` 노트북을 실행할 수 있습니다.

```
In [8]: LEARNING_RATE = 0.0005
R_LOSS_FACTOR = 10000
EPOCHS = 200
PRINT_EVERY_N_BATCHES = 100
INITIAL_EPOCH = 0
```

```
In [9]: vae.compile(LEARNING_RATE, R_LOSS_FACTOR)
```

```
In [*]: vae.train_with_generator(
    data_flow
    , epochs = EPOCHS
    , steps_per_epoch = NUM_IMAGES / BATCH_SIZE
    , run_folder = RUN_FOLDER
    , print_every_n_batches = PRINT_EVERY_N_BATCHES
    , initial_epoch = INITIAL_EPOCH
)
```

WARNING: tensorflow: From C:\Users\User\anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.cast instead.

Epoch 1/200

5/6331 [.....] - ETA: 8:14:58 - loss: 1312.2517 - vae_r_loss: 1111.0498 - vae_kl_loss: 201.2020

시간상의 문제로 훈련된 파일을 사용하려면

C:\Users\User\GDL_code\run\vae\0001_faces\weights에 있는 weights.h5 파일을 복사한 후
본인이 이전에 run 파일을 붙여넣기 한 곳에 \run\vae\0001_faces\weights에
weights.h5 파일을 붙여넣습니다.

e.g.) C:\Users\User\VAE\run\vae\0001_faces\weights

VAE 분석 - 얼굴 데이터셋

라이브러리 импорт

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import os
from scipy.stats import norm
import pandas as pd

from models.VAE import VariationalAutoencoder
from utils.loaders import load_model, ImageLabelLoader
```

Using TensorFlow backend.

```
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype [("qint8", np.int8, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype [("qint8", np.uint8, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype [("qint16", np.int16, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype [("qint16", np.uint16, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint32 = np.dtype [("qint32", np.int32, 1)]
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    np_resource = np.dtype [("resource", np.ubyte, 1)]
```

```
In [2]: # run params
section = 'vae'
run_id = '0001'
data_name = 'faces'
RUN_FOLDER = 'run/{}/'.format(section)
RUN_FOLDER += '_'.join([run_id, data_name])

DATA_FOLDER = './data/celeb/'
IMAGE_FOLDER = './data/celeb/img_align_celeba/'
```

데이터 적재

```
In [3]: INPUT_DIM = (128,128,3)

att = pd.read_csv(os.path.join(DATA_FOLDER, 'list_attr_celeba.csv'))

imageLoader = ImageLabelLoader(IMAGE_FOLDER, INPUT_DIM[:2])
```

```
In [4]: att.head()
```

Out[4]:

	image_id	5_o_Clock_Shadow	Arched_Eyebrows	Attractive	Bags_Under_Eyes	Bald	Bangs	Big_Lips	Big_Nose	Black_Hair	...	Sideburns	Smiling	Str:
0	000001.jpg	-1	1	1	-1	-1	-1	-1	-1	-1	...	-1	1	
1	000002.jpg	-1	-1	-1	1	-1	-1	-1	1	-1	...	-1	1	
2	000003.jpg	-1	-1	-1	-1	-1	-1	1	-1	-1	...	-1	-1	
3	000004.jpg	-1	-1	1	-1	-1	-1	-1	-1	-1	...	-1	-1	
4	000005.jpg	-1	1	1	-1	-1	-1	1	-1	-1	...	-1	-1	

5 rows × 41 columns



모델 만들기

```
In [5]: vae = load_model(VariationalAutoencoder, RUN_FOLDER)
```

WARNING:tensorflow:From C:\Users\User\anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.

Instructions for updating:

Colocations handled automatically by placer.

WARNING:tensorflow:From C:\Users\User\anaconda3\envs\testGAN\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.

Instructions for updating:

Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.

얼굴 이미지 재구성

```
In [6]: n_to_show = 10

data_flow_generic = imageLoader.build(att, n_to_show)

example_batch = next(data_flow_generic)
example_images = example_batch[0]

z_points = vae.encoder.predict(example_images)

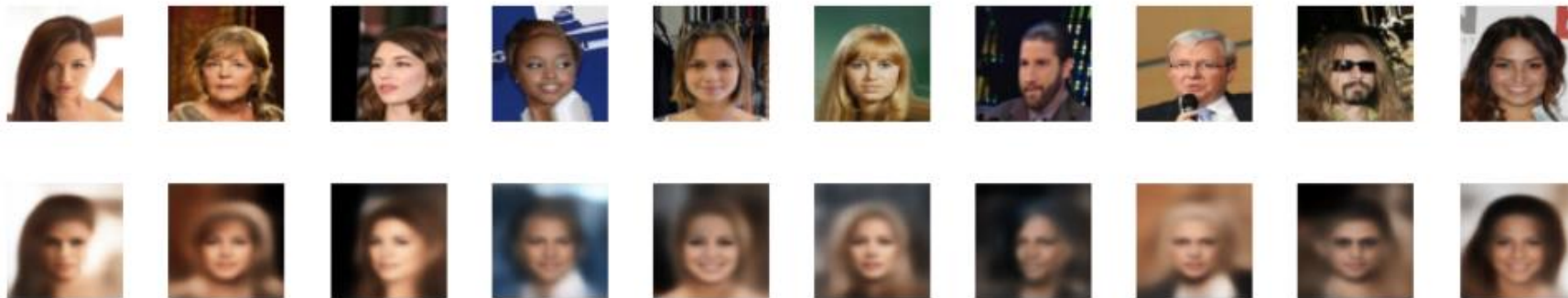
reconst_images = vae.decoder.predict(z_points)

fig = plt.figure(figsize=(15, 3))
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(n_to_show):
    img = example_images[i].squeeze()
    sub = fig.add_subplot(2, n_to_show, i+1)
    sub.axis('off')
    sub.imshow(img)

for i in range(n_to_show):
    img = reconst_images[i].squeeze()
    sub = fig.add_subplot(2, n_to_show, i+n_to_show+1)
    sub.axis('off')
    sub.imshow(img)
```

Found 202599 validated image filenames.



잠재 공간 분포

```
In [7]: z_test = vae.encoder.predict_generator(data_flow_generic, steps = 20, verbose = 1)

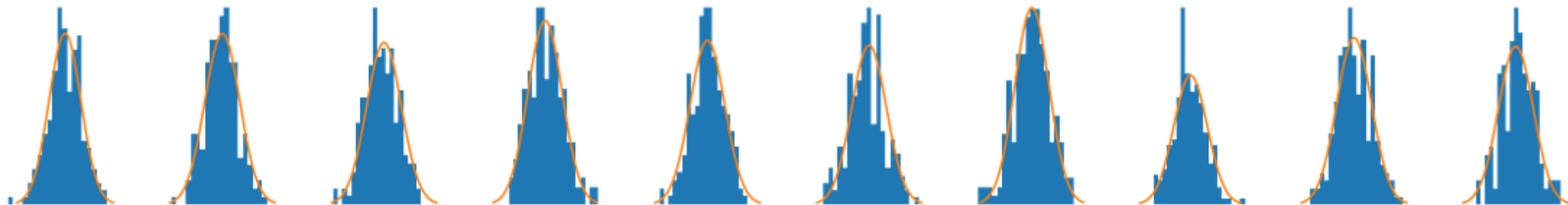
x = np.linspace(-3, 3, 100)

fig = plt.figure(figsize=(20, 20))
fig.subplots_adjust(hspace=0.6, wspace=0.4)

for i in range(30):
    ax = fig.add_subplot(5, 10, i+1)
    ax.hist(z_test[:,i], density=True, bins = 20)
    ax.axis('off')
    ax.text(0.5, -0.35, str(i), fontsize=10, ha='center', transform=ax.transAxes)
    ax.plot(x, norm.pdf(x))

plt.show()
```

20/20 [=====] - 5s 244ms/step



0

1

2

3

4

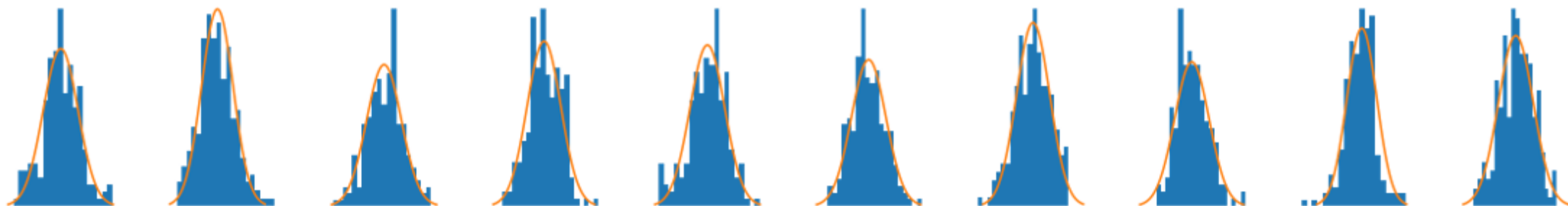
5

6

7

8

9



10

11

12

13

14

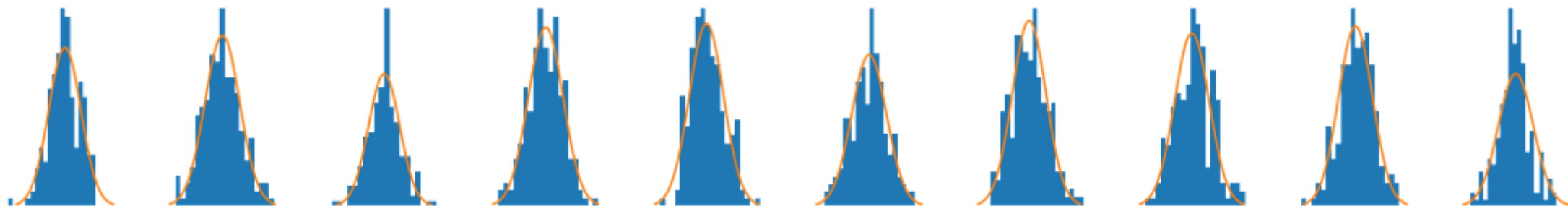
15

16

17

18

19



20

21

22

23

24

25

26

27

28

29

새로 생성한 얼굴

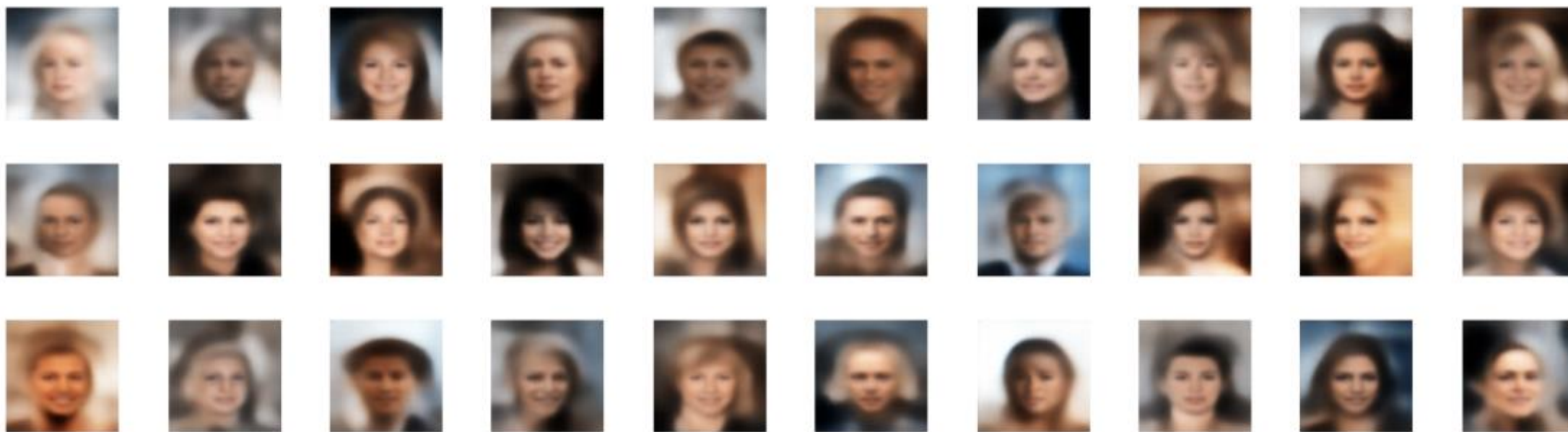
```
In [8]: n_to_show = 30

znew = np.random.normal(size = (n_to_show, vae.z_dim))

reconst = vae.decoder.predict(np.array(znew))

fig = plt.figure(figsize=(18, 5))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
for i in range(n_to_show):
    ax = fig.add_subplot(3, 10, i+1)
    ax.imshow(reconst[i, :, :, :])
    ax.axis('off')

plt.show()
```



일부 특징(mouth_open, male, eyeglasses) 벡터를 추출

In [11]: BATCH_SIZE = 500

```
mouth_open_vec = get_vector_from_label('Mouth_Slightly_Open', BATCH_SIZE)
male_vec = get_vector_from_label('Male', BATCH_SIZE)
eyeglasses_vec = get_vector_from_label('Eyeglasses', BATCH_SIZE)
```

Found 202599 validated image filenames.

label: Mouth_Slightly_Open

images : POS move : NEG move : distance : Δ distance

243	: 1.022	: 1.061	: 1.641	: 1.641
484	: 0.6	: 0.616	: 1.319	: -0.322
719	: 0.333	: 0.365	: 1.271	: -0.048
944	: 0.261	: 0.253	: 1.24	: -0.03
1199	: 0.205	: 0.17	: 1.175	: -0.066
1433	: 0.175	: 0.146	: 1.151	: -0.023
1677	: 0.134	: 0.126	: 1.143	: -0.009
1925	: 0.129	: 0.115	: 1.111	: -0.032
2160	: 0.105	: 0.098	: 1.099	: -0.012
2407	: 0.096	: 0.084	: 1.089	: -0.011
2653	: 0.09	: 0.082	: 1.07	: -0.019
2897	: 0.083	: 0.075	: 1.071	: 0.0
3154	: 0.073	: 0.069	: 1.062	: -0.009
3399	: 0.063	: 0.067	: 1.059	: -0.003
3649	: 0.07	: 0.06	: 1.054	: -0.005
3885	: 0.053	: 0.054	: 1.048	: -0.007
4109	: 0.051	: 0.051	: 1.039	: -0.009
4365	: 0.047	: 0.05	: 1.031	: -0.008
4638	: 0.05	: 0.039	: 1.021	: -0.01
4853	: 0.047	: 0.047	: 1.018	: -0.003
5104	: 0.045	: 0.041	: 1.023	: 0.006
5357	: 0.044	: 0.039	: 1.025	: 0.001
5614	: 0.038	: 0.041	: 1.022	: -0.003

Found the Mouth_Slightly_Open vector

Found 202599 validated image filenames.

label: Male

images : POS move : NEG move : distance : Δ distance

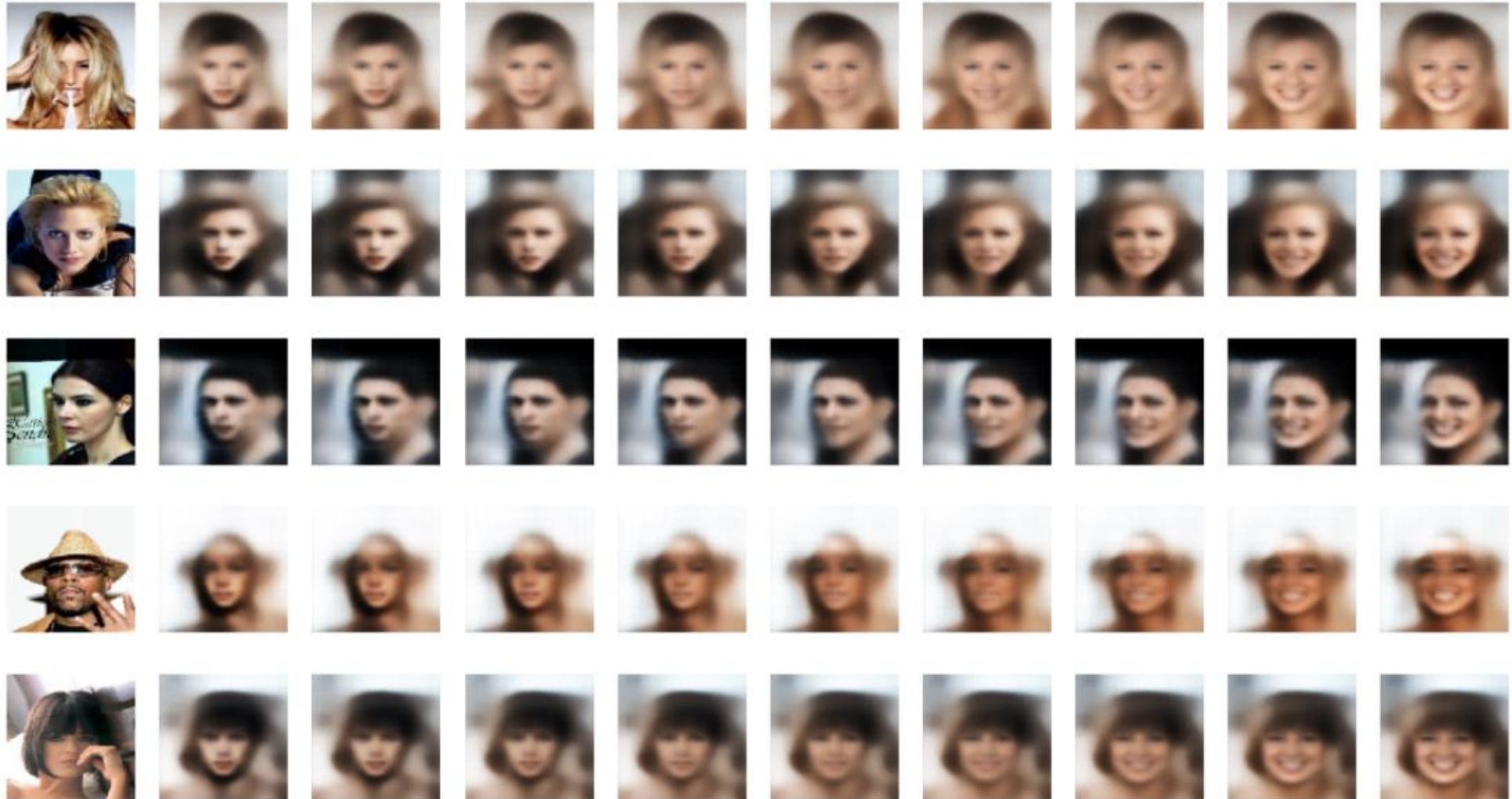
223	: 1.2	: 1.04	: 1.819	: 1.819
430	: 0.713	: 0.615	: 1.557	: -0.262
643	: 0.402	: 0.368	: 1.408	: -0.149
866	: 0.277	: 0.231	: 1.356	: -0.052
1070	: 0.217	: 0.174	: 1.302	: -0.054
1288	: 0.175	: 0.15	: 1.309	: 0.007
1497	: 0.154	: 0.125	: 1.289	: -0.02

```
In [12]: print('Mouth Open Vector')
         add_vector_to_images(mouth_open_vec)

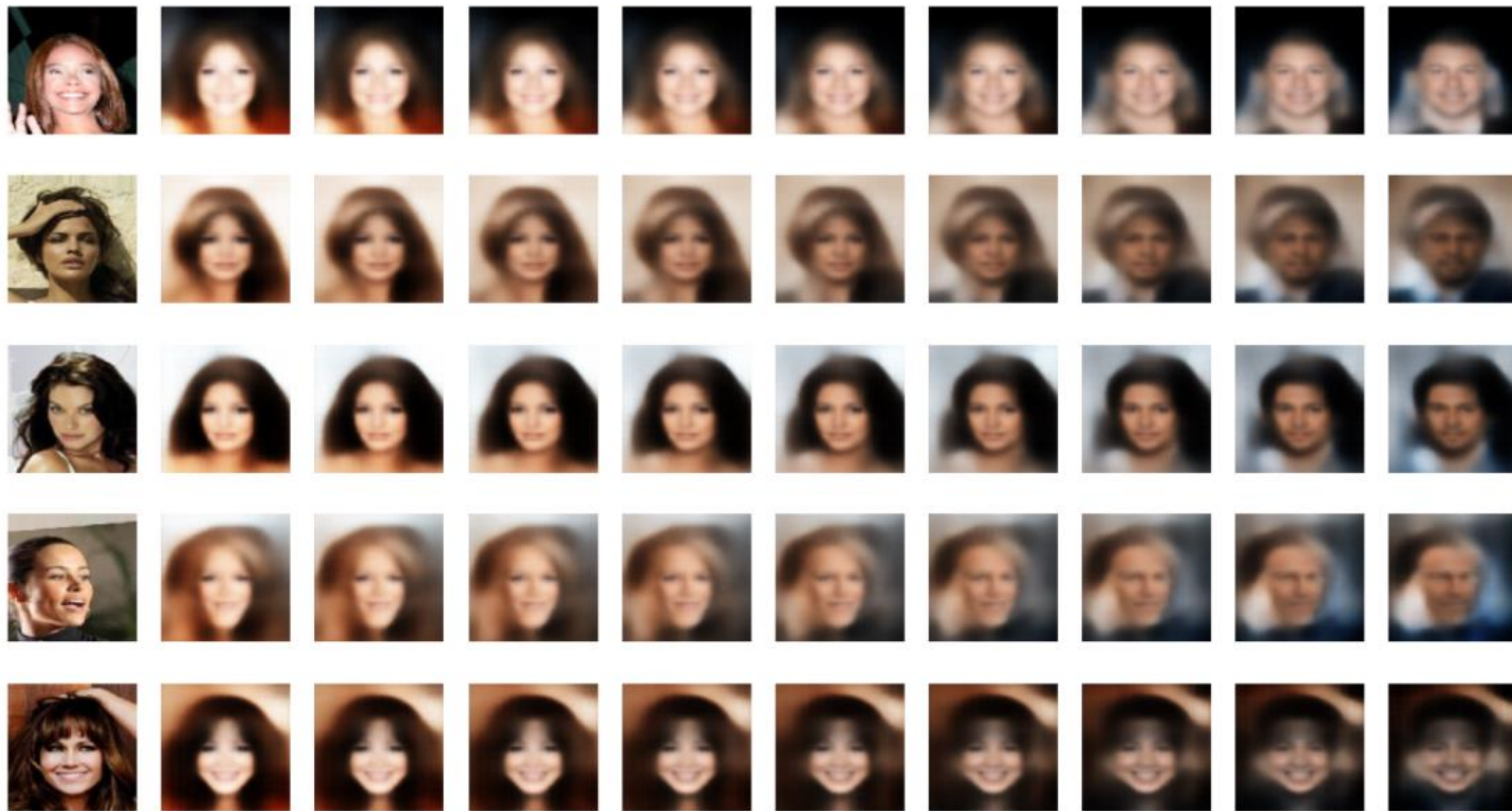
         print('Male Vector')
         add_vector_to_images(male_vec)

         print('Eyeglasses Vector')
         add_vector_to_images(eyeglasses_vec)
```

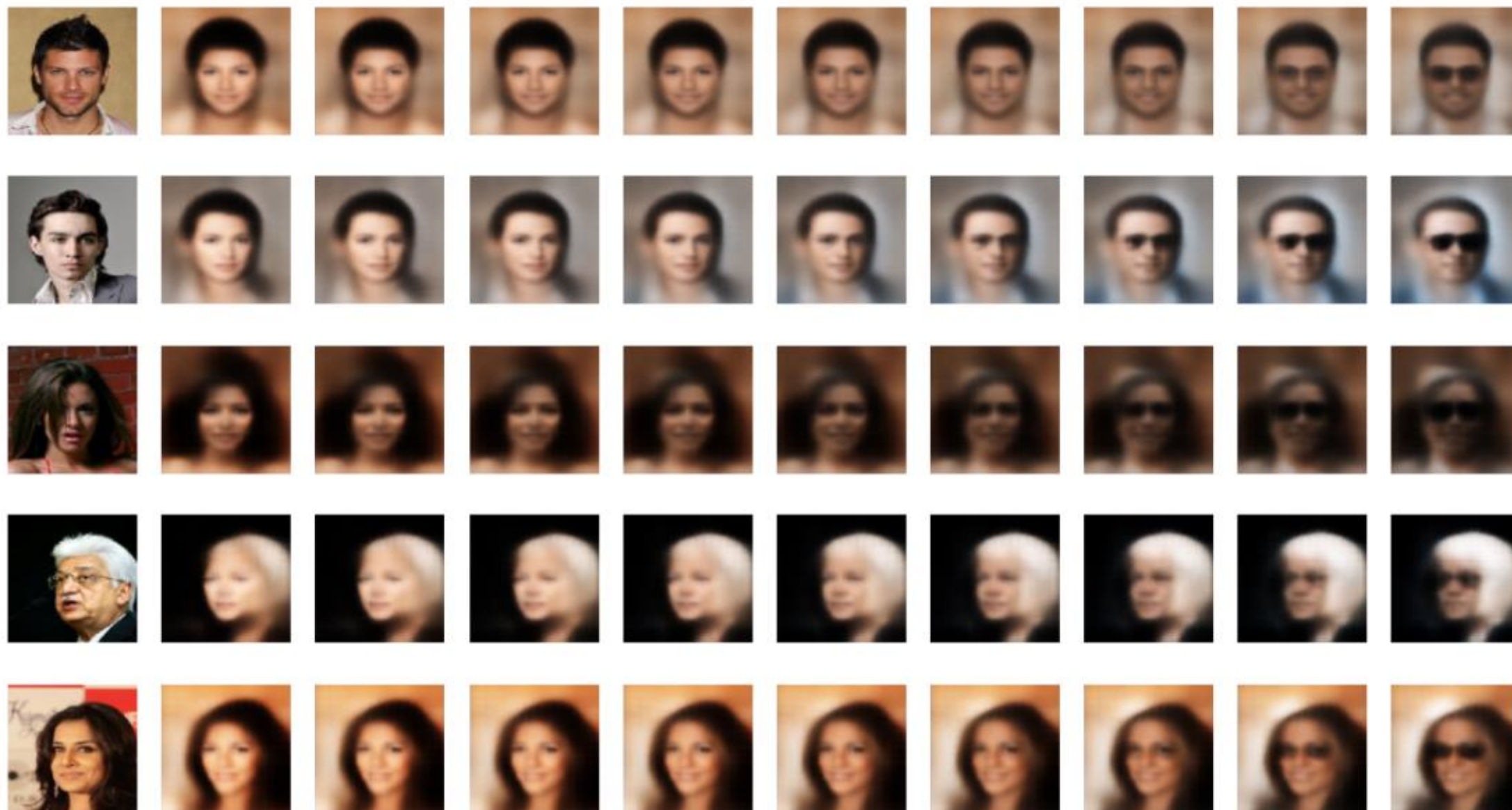
Mouth Open Vector



Male Vector



Eyeglasses Vector



```

In [13]: def morph_faces(start_image_file, end_image_file):

    factors = np.arange(0,1,0.1)

    att_specific = att[att['image_id'].isin([start_image_file, end_image_file])]
    att_specific = att_specific.reset_index()
    data_flow_label = imageLoader.build(att_specific, 2)

    example_batch = next(data_flow_label)
    example_images = example_batch[0]
    example_labels = example_batch[1]

    z_points = vae.encoder.predict(example_images)

    fig = plt.figure(figsize=(18, 8))

    counter = 1

    img = example_images[0].squeeze()
    sub = fig.add_subplot(1, len(factors)+2, counter)
    sub.axis('off')
    sub.imshow(img)

    counter+=1

    for factor in factors:

        changed_z_point = z_points[0] * (1-factor) + z_points[1] * factor
        changed_image = vae.decoder.predict(np.array([changed_z_point]))[0]

        img = changed_image.squeeze()
        sub = fig.add_subplot(1, len(factors)+2, counter)
        sub.axis('off')
        sub.imshow(img)

        counter += 1

    img = example_images[1].squeeze()
    sub = fig.add_subplot(1, len(factors)+2, counter)
    sub.axis('off')
    sub.imshow(img)

    plt.show()

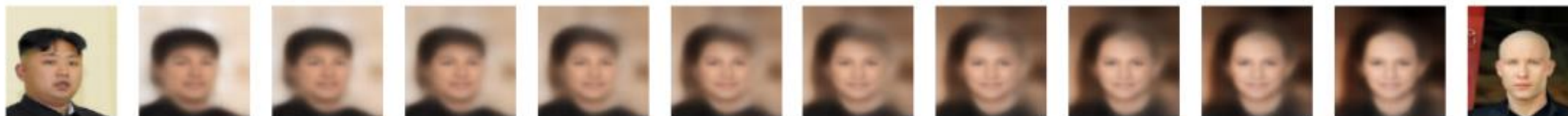
```



```
In [14]: start_image_file = '026192.jpg'
end_image_file = '000115.jpg'

morph_faces(start_image_file, end_image_file)
```

Found 2 validated image filenames.



```
In [15]: start_image_file = '077969.jpg'
end_image_file = '026677.jpg'

morph_faces(start_image_file, end_image_file)
```

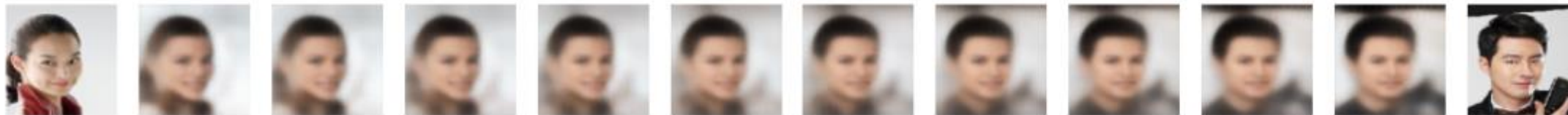
Found 2 validated image filenames.



```
In [16]: start_image_file = '078621.jpg'
end_image_file = '026130.jpg'

morph_faces(start_image_file, end_image_file)
```

Found 2 validated image filenames.



수고하셨습니다.