In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers
import time
%load_ext tensorboard
```

Load data

In [ ]:
```
def load_data():
  (train_images, train_labels), (valid_images, valid_labels) = tf.keras.datasets.mnist
.load_data()
  # preprocess the images
  train_images = (train_images.astype(np.float32) - 127.5)/127.5 # standardize to [-1,
1]
  train_images = train_images.reshape(train_images.shape[0], 28, 28, 1)

  valid_images = (valid_images.astype(np.float32) - 127.5)/127.5 # standardize to [-1,
1]
  valid_images = valid_images.reshape(valid_images.shape[0], 28, 28, 1)
  return (train_images, train_labels, valid_images, valid_labels)
```

In [ ]:
```
train_images, train_labels, valid_images, valid_labels = load_data()
train_images.shape, valid_images.shape
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mn
ist.npz
11493376/11490434 [==============================] - 0s 0us/step
```

Out[ ]:
```
((60000, 28, 28, 1), (10000, 28, 28, 1))
```

Generator

In [ ]:

```
# define a function to build generator
def build_generator():
  model = tf.keras.Sequential([
          layers.Dense(128, use_bias=False, input_shape=(100,)),
          layers.BatchNormalization(),
          layers.LeakyReLU(0),

          layers.Dropout(0.2),

          layers.Dense(7*7*256, use_bias=False),
          layers.BatchNormalization(),
          layers.LeakyReLU(0),
          layers.Reshape((7,7,256)),

          layers.Conv2DTranspose(128, (5,5), strides=1, padding='same', use_bias=False
), # 7,7,128
          layers.BatchNormalization(),
          layers.LeakyReLU(0),

          layers.Conv2DTranspose(64, (5,5), strides=2, padding='same', use_bias=False
), #14,14,64
          layers.BatchNormalization(),
          layers.LeakyReLU(0),

          layers.Conv2DTranspose(1, (5,5), strides=2, padding='same', use_bias=False),
#28,28,1
          layers.Activation('tanh')
  ])

  return model

g = build_generator()
g.summary()
```

Model: "sequential_5"

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 128)	12800
batch_normalization_10 (Batc	(None, 128)	512
leaky_re_lu_12 (LeakyReLU)	(None, 128)	0
dropout_6 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 12544)	1605632
batch_normalization_11 (Batc	(None, 12544)	50176
leaky_re_lu_13 (LeakyReLU)	(None, 12544)	0
reshape_2 (Reshape)	(None, 7, 7, 256)	0
conv2d_transpose_6 (Conv2DTr	(None, 7, 7, 128)	819200
batch_normalization_12 (Batc	(None, 7, 7, 128)	512
leaky_re_lu_14 (LeakyReLU)	(None, 7, 7, 128)	0
conv2d_transpose_7 (Conv2DTr	(None, 14, 14, 64)	204800
batch_normalization_13 (Batc	(None, 14, 14, 64)	256
leaky_re_lu_15 (LeakyReLU)	(None, 14, 14, 64)	0
conv2d_transpose_8 (Conv2DTr	(None, 28, 28, 1)	1600
activation_2 (Activation)	(None, 28, 28, 1)	0

Total params: 2,695,488
Trainable params: 2,669,760
Non-trainable params: 25,728

Discriminator

In [ ]:
```
# define a function to build discriminator
def build_discriminator():
  model = tf.keras.Sequential([
            layers.Conv2D(128, (5,5), strides=2, padding='same',
                          input_shape=(28,28,1)), #14,14,64
            layers.LeakyReLU(0.2),

            layers.Conv2D(64, (5,5), strides=2, padding='same'), #7,7,128
            layers.BatchNormalization(),
            layers.LeakyReLU(0.2),
            layers.Dropout(0.2),

            layers.Flatten(),
            layers.Dense(128, activation='relu'),
            layers.Dropout(0.2),
            layers.Dense(1, activation='sigmoid')
  ])
  return model

d = build_discriminator()
d.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 14, 14, 128)       3328
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 14, 14, 128)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 7, 7, 64)          204864
_________________________________________________________________
batch_normalization_4 (Batch (None, 7, 7, 64)          256
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 7, 7, 64)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 7, 7, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 3136)              0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               401536
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 129
=================================================================
Total params: 610,113
Trainable params: 609,985
Non-trainable params: 128
_________________________________________________________________
```

DCGAN

In [ ]:
```
# define a function to combine generator and discriminator --> GAN
def build_gan(discriminator_lr, generator_lr):
  generator = build_generator()

  discriminator = build_discriminator()
  discriminator.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=discriminator
_lr, beta_1=0.5),
                        loss='binary_crossentropy')

  gan = tf.keras.Sequential([generator, discriminator])
  gan.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=generator_lr, beta_1=0.
5),
              loss='binary_crossentropy')

  return gan, generator, discriminator
```

Training GAN

In [ ]:
```
# set discriminator to trainable or not
def set_to_trainable(model, trainable):
    for layer in model.layers:
        layer.trainable = trainable

# show 64 generated images in square
def show_images(generated_images):
    n_images = len(generated_images)
    cols = 8
    rows = n_images//cols

    plt.figure(figsize=(8, 8))
    for i in range(n_images):
        img = generated_images[i,:,:,0]*127.5+127.5
        ax = plt.subplot(rows, cols, i+1)
        plt.imshow(img, cmap='gray')
        plt.xticks([])
        plt.yticks([])
    plt.tight_layout()
    plt.show()

# Transform train_on_batch return value to dict
def named_logs(model, logs):
  result = {}
  for l in zip(['training_loss','validation_loss'], logs):
    result[l[0]] = l[1]
  return result
```

In [ ]:
```
# the seed would be repeatedly used for image output at different epoch
np.random.seed(2020)
seed = np.random.normal(loc=0, scale=1, size=(64, 100))
```

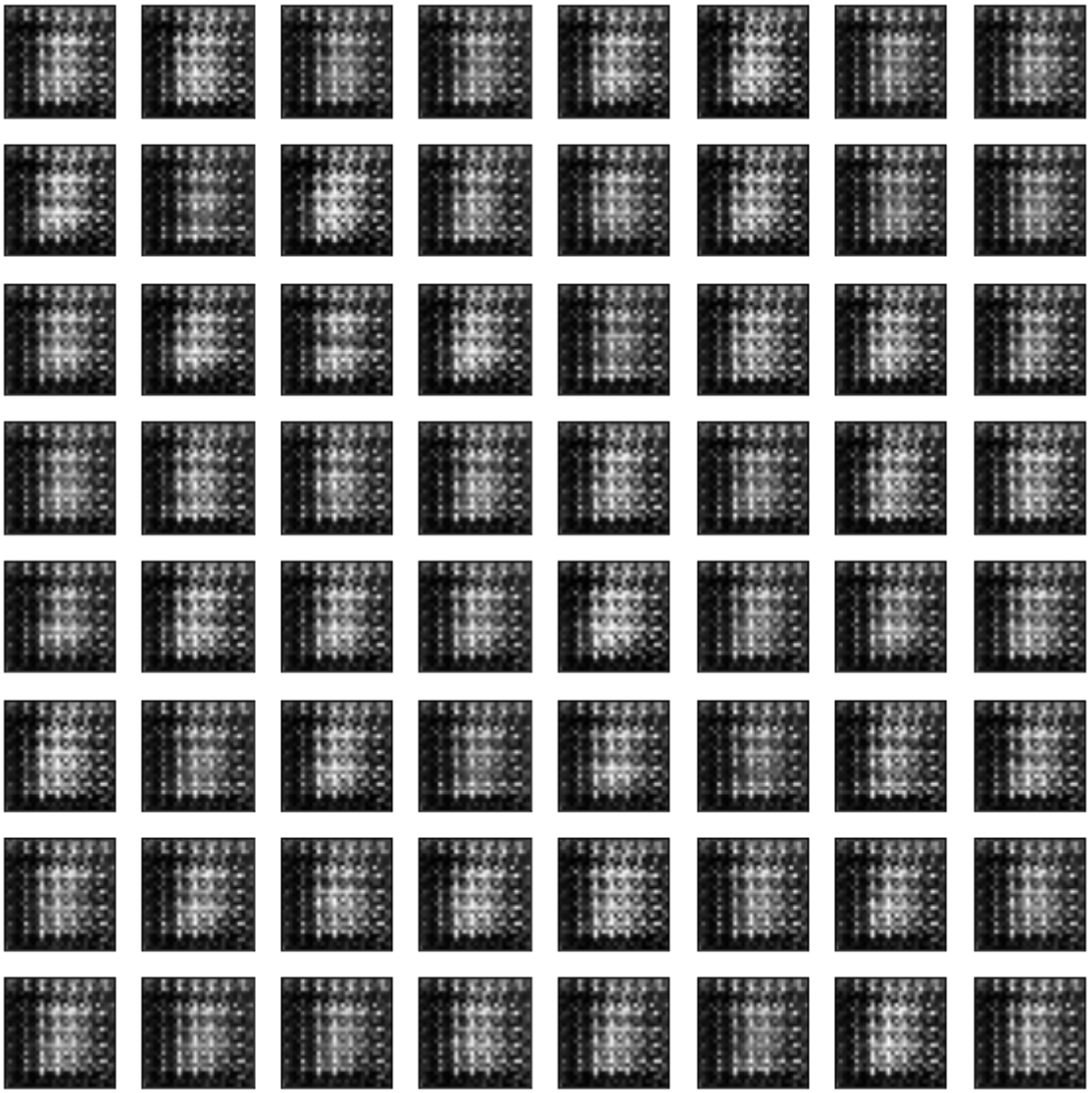
In [ ]:
```
!rm -rf ./logs_mnist/
```

In [ ]:
```
def train(generator_lr,
          discriminator_lr,
          smooth=0.1,
          epochs=50,
          batch_size=128,
          val_size=36):

    start = time.time()

    # labels for the batch size and the validation size
    y_train_real, y_train_fake = np.ones([batch_size, 1]), np.zeros([batch_size, 1])
    y_val_real,  y_val_fake  = np.ones([val_size, 1]), np.zeros([val_size, 1])

    # create a GAN, a generator and a discriminator
    gan, generator, discriminator = build_gan(discriminator_lr, generator_lr)

    # prepare for tensorboard
    tb_dis = tf.keras.callbacks.TensorBoard(
        log_dir='./logs_mnist/discriminator',
        histogram_freq=1,
        write_graph=True)
    tb_dis.set_model(discriminator)

    tb_gan = tf.keras.callbacks.TensorBoard(
        log_dir='./logs_mnist/gan',
        histogram_freq=1,
        write_graph=True)
    tb_gan.set_model(gan)

    # iterate epochs to train the GAN
    num_batches = len(train_images)//batch_size
    for e in range(epochs):
      for i in range(num_batches):
        # real images in a batch
        X_batch_real = train_images[i*batch_size:(i+1)*batch_size]

        # generate a batch of fake images using random vectors
        random_vectors = np.random.normal(loc=0.0, scale=1.0, size=(batch_size, 100))
        X_batch_fake = generator.predict_on_batch(random_vectors)

        # train the discriminator
        set_to_trainable(discriminator, True)
        dis_loss_train = discriminator.train_on_batch(X_batch_real, y_train_real-0.1)
        dis_loss_train += discriminator.train_on_batch(X_batch_fake, y_train_fake)

        # train the GAN (i.e. train the generator while keeping discriminator fixed)
        set_to_trainable(discriminator, False)
        # set the target of fake images to 1 because we want the loss between fake ima
ges and 1 (label for real image) to be minimized
        # i.e. we want the fake images to be real
        gen_loss_train = gan.train_on_batch(random_vectors, y_train_real)

        # keep noticed by the training process
        if (i+1) % 50 == 0:
          print('This is the {}/{} of epoch {}.'.format(i+1, num_batches, e+1))

      # compute loss on validation set
      X_val_real = valid_images[np.random.choice(len(valid_images), size=val_size, rep
lace=False)]

      random_vectors = np.random.normal(loc=0.0, scale=1.0, size=(val_size, 100))
```

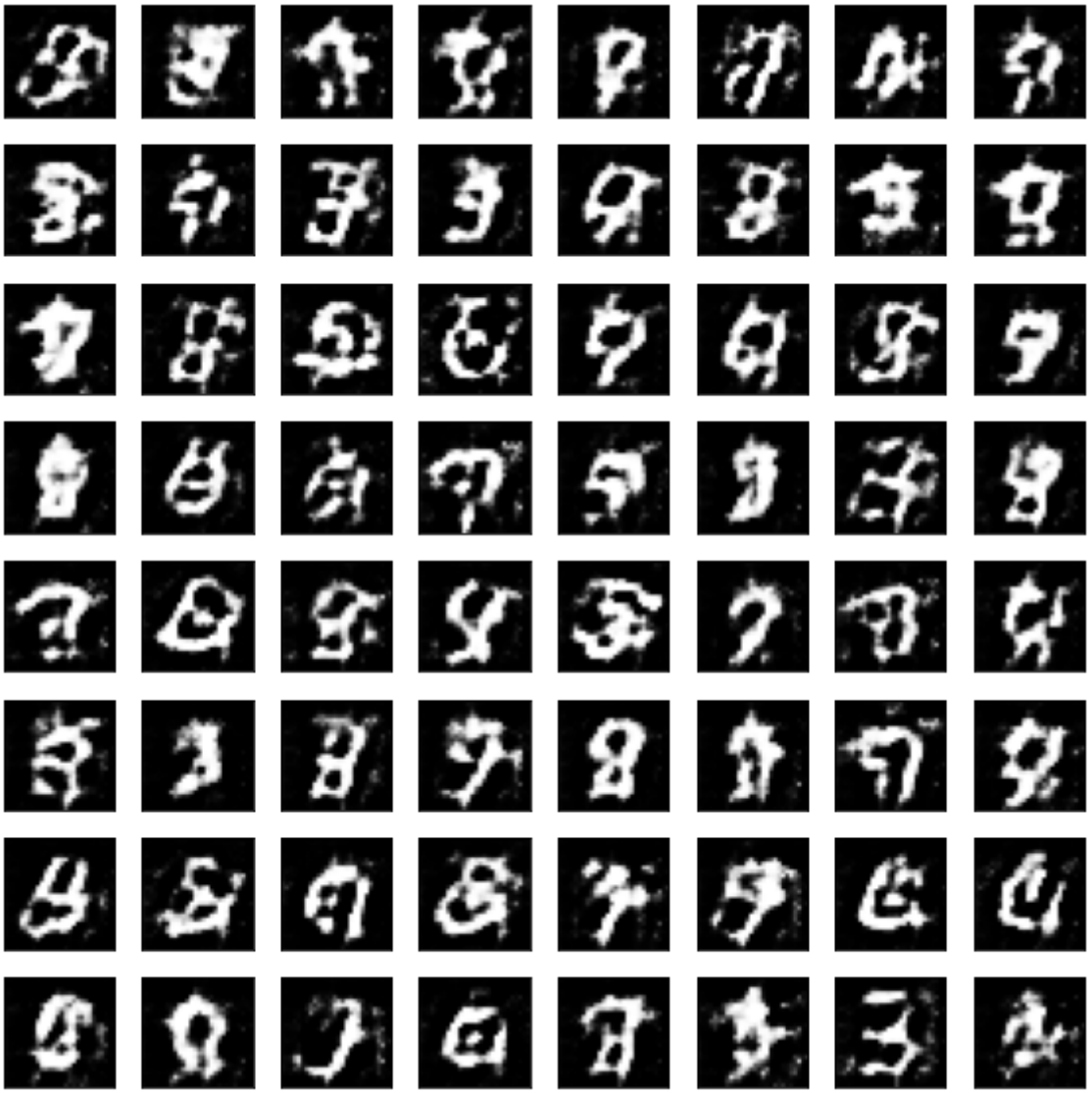
```
      X_val_fake = generator.predict_on_batch(random_vectors)

      dis_loss_val = discriminator.test_on_batch(X_val_real, y_val_real)
      dis_loss_val += discriminator.test_on_batch(X_val_fake, y_val_fake)
      gen_loss_val = gan.test_on_batch(random_vectors, y_val_real)

      print("Epoch: {}/{} Discriminator Loss: {:.4f}  Generator Loss: {:.4f}".format
(e+1, epochs, dis_loss_val, gen_loss_val))

      tb_dis.on_epoch_end(e, named_logs(discriminator, [dis_loss_train, dis_loss_val
]))
      tb_gan.on_epoch_end(e, named_logs(gan, [gen_loss_train, gen_loss_val]))

      if e==0 or (e+1) % 5 == 0:
        imgs = generator.predict_on_batch(seed)
        show_images(imgs)

    end = time.time()
    print(end-start)
    return generator
```

In [ ]:
```
generator_mnist_improved = train(generator_lr=1e-4,
          discriminator_lr=0.001, epochs=200)
```

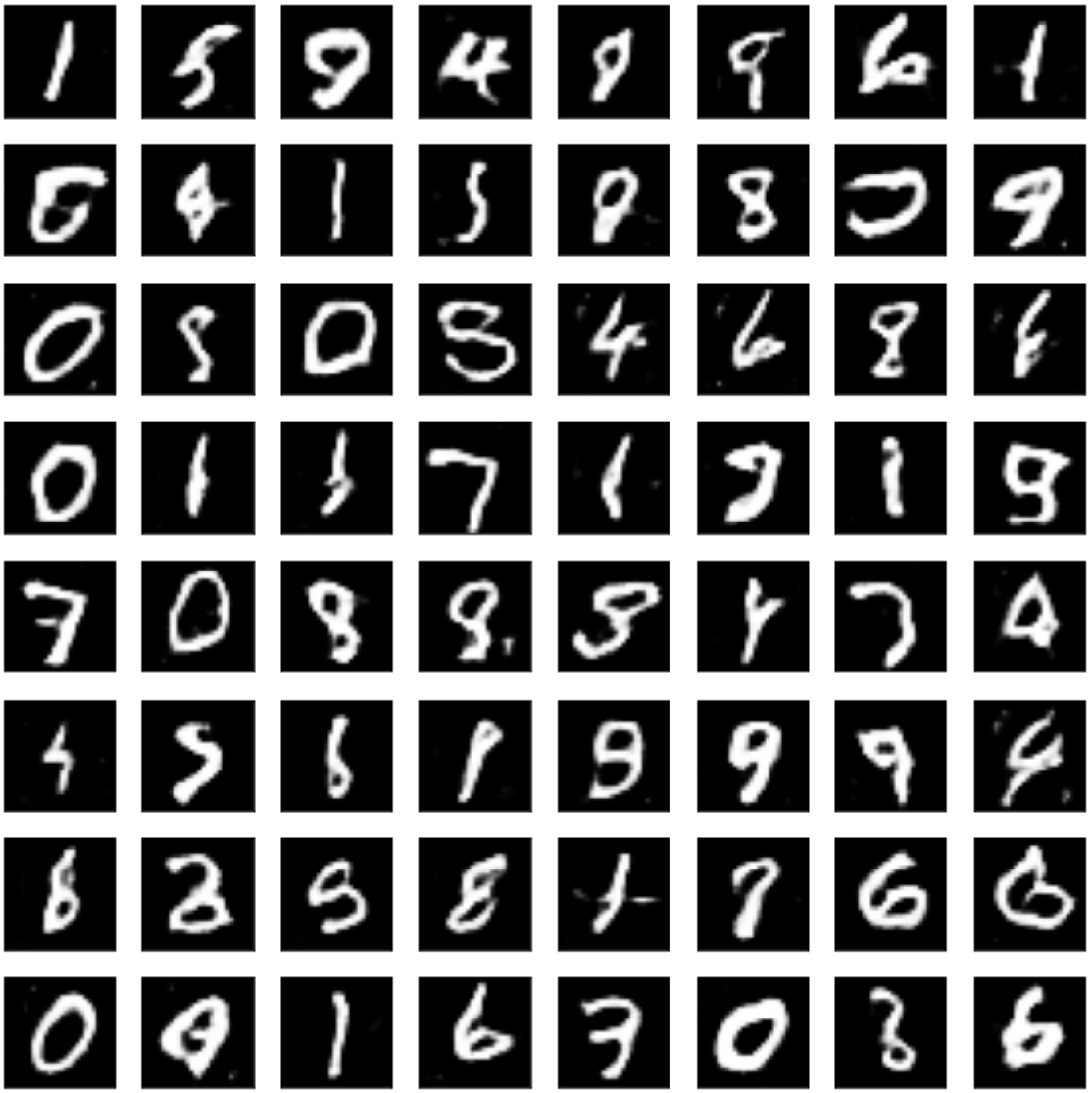
```
This is the 50/468 of epoch 1.
This is the 100/468 of epoch 1.
This is the 150/468 of epoch 1.
This is the 200/468 of epoch 1.
This is the 250/468 of epoch 1.
This is the 300/468 of epoch 1.
This is the 350/468 of epoch 1.
This is the 400/468 of epoch 1.
This is the 450/468 of epoch 1.
Epoch: 1/200 Discriminator Loss: 1.9430  Generator Loss: 1.3722
```



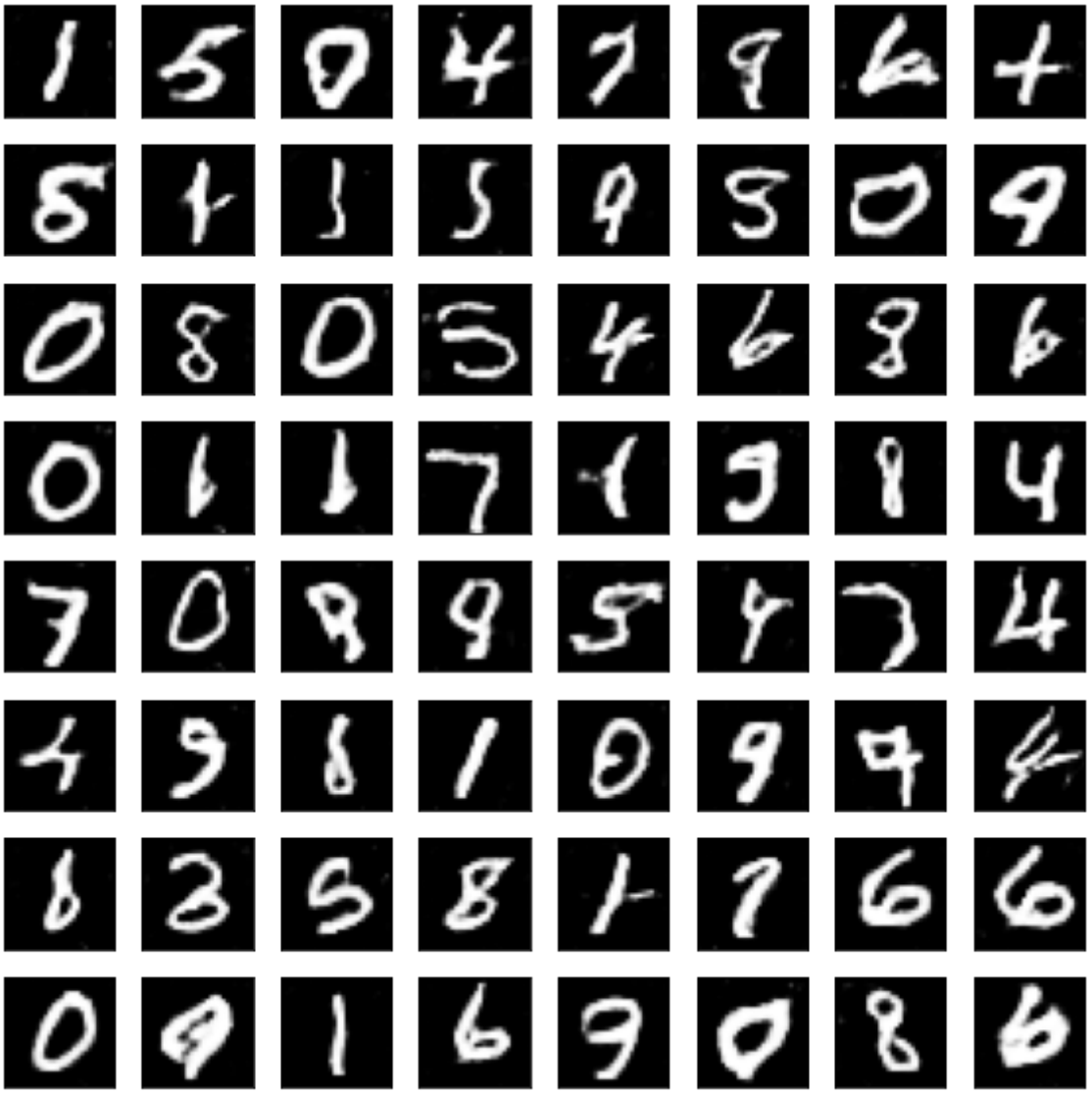
```
This is the 50/468 of epoch 2.
This is the 100/468 of epoch 2.
This is the 150/468 of epoch 2.
This is the 200/468 of epoch 2.
This is the 250/468 of epoch 2.
This is the 300/468 of epoch 2.
This is the 350/468 of epoch 2.
This is the 400/468 of epoch 2.
This is the 450/468 of epoch 2.
Epoch: 2/200 Discriminator Loss: 7.2388  Generator Loss: 6.5259
This is the 50/468 of epoch 3.
This is the 100/468 of epoch 3.
This is the 150/468 of epoch 3.
This is the 200/468 of epoch 3.
This is the 250/468 of epoch 3.
This is the 300/468 of epoch 3.
This is the 350/468 of epoch 3.
This is the 400/468 of epoch 3.
This is the 450/468 of epoch 3.
Epoch: 3/200 Discriminator Loss: 0.8849  Generator Loss: 0.9347
This is the 50/468 of epoch 4.
This is the 100/468 of epoch 4.
This is the 150/468 of epoch 4.
This is the 200/468 of epoch 4.
This is the 250/468 of epoch 4.
This is the 300/468 of epoch 4.
This is the 350/468 of epoch 4.
This is the 400/468 of epoch 4.
This is the 450/468 of epoch 4.
Epoch: 4/200 Discriminator Loss: 2.9156  Generator Loss: 0.0660
This is the 50/468 of epoch 5.
This is the 100/468 of epoch 5.
This is the 150/468 of epoch 5.
This is the 200/468 of epoch 5.
This is the 250/468 of epoch 5.
This is the 300/468 of epoch 5.
This is the 350/468 of epoch 5.
This is the 400/468 of epoch 5.
This is the 450/468 of epoch 5.
Epoch: 5/200 Discriminator Loss: 1.9230  Generator Loss: 0.2046
```



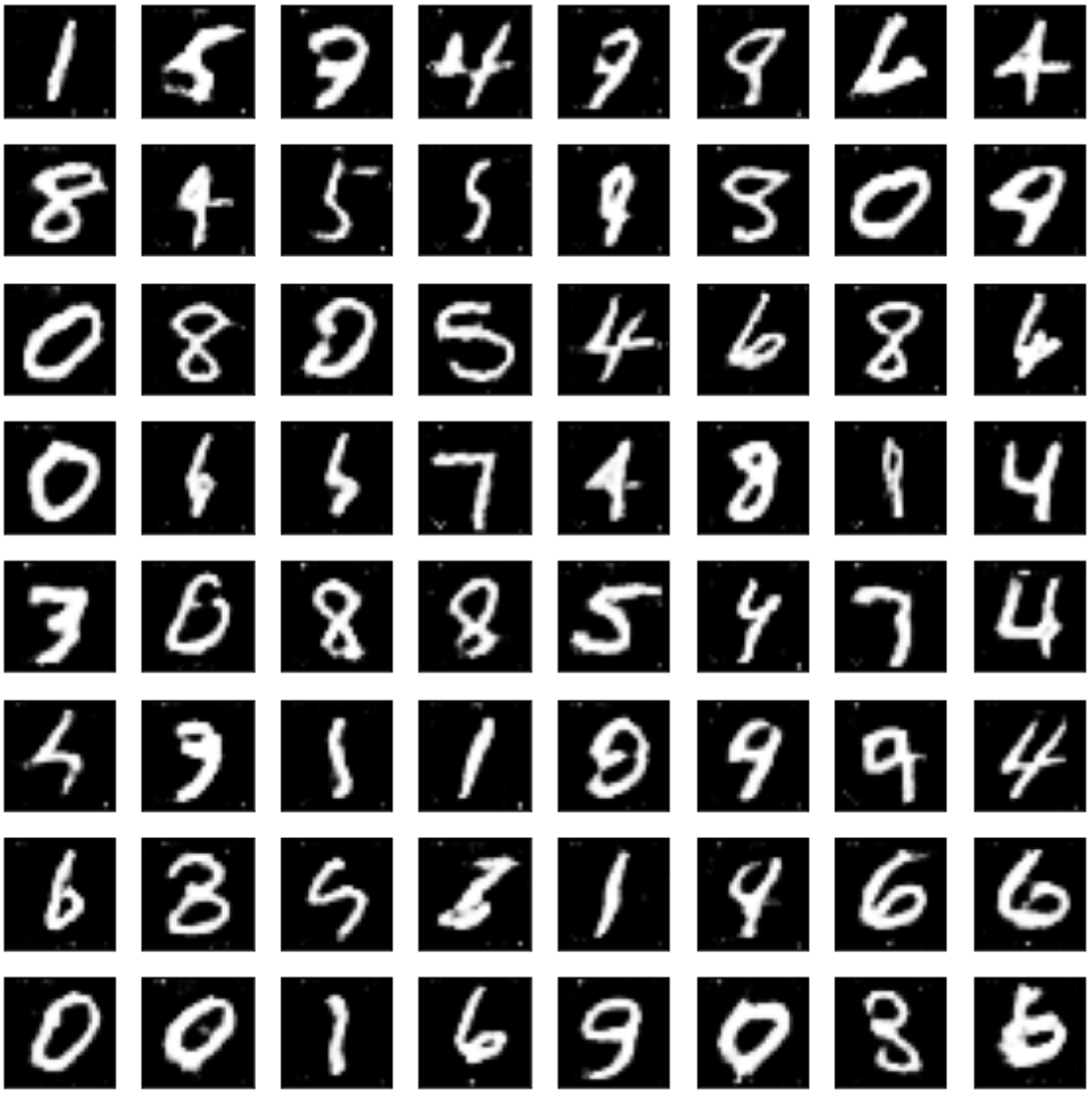
```
This is the 50/468 of epoch 6.
This is the 100/468 of epoch 6.
This is the 150/468 of epoch 6.
This is the 200/468 of epoch 6.
This is the 250/468 of epoch 6.
This is the 300/468 of epoch 6.
This is the 350/468 of epoch 6.
This is the 400/468 of epoch 6.
This is the 450/468 of epoch 6.
Epoch: 6/200 Discriminator Loss: 9.2274  Generator Loss: 0.0002
This is the 50/468 of epoch 7.
This is the 100/468 of epoch 7.
This is the 150/468 of epoch 7.
This is the 200/468 of epoch 7.
This is the 250/468 of epoch 7.
This is the 300/468 of epoch 7.
This is the 350/468 of epoch 7.
This is the 400/468 of epoch 7.
This is the 450/468 of epoch 7.
Epoch: 7/200 Discriminator Loss: 6.5957  Generator Loss: 0.0022
This is the 50/468 of epoch 8.
This is the 100/468 of epoch 8.
This is the 150/468 of epoch 8.
This is the 200/468 of epoch 8.
This is the 250/468 of epoch 8.
This is the 300/468 of epoch 8.
This is the 350/468 of epoch 8.
This is the 400/468 of epoch 8.
This is the 450/468 of epoch 8.
Epoch: 8/200 Discriminator Loss: 1.1748  Generator Loss: 0.6341
This is the 50/468 of epoch 9.
This is the 100/468 of epoch 9.
This is the 150/468 of epoch 9.
This is the 200/468 of epoch 9.
This is the 250/468 of epoch 9.
This is the 300/468 of epoch 9.
This is the 350/468 of epoch 9.
This is the 400/468 of epoch 9.
This is the 450/468 of epoch 9.
Epoch: 9/200 Discriminator Loss: 3.3667  Generator Loss: 0.0543
This is the 50/468 of epoch 10.
This is the 100/468 of epoch 10.
This is the 150/468 of epoch 10.
This is the 200/468 of epoch 10.
This is the 250/468 of epoch 10.
This is the 300/468 of epoch 10.
This is the 350/468 of epoch 10.
This is the 400/468 of epoch 10.
This is the 450/468 of epoch 10.
Epoch: 10/200 Discriminator Loss: 7.8567  Generator Loss: 0.0007
```



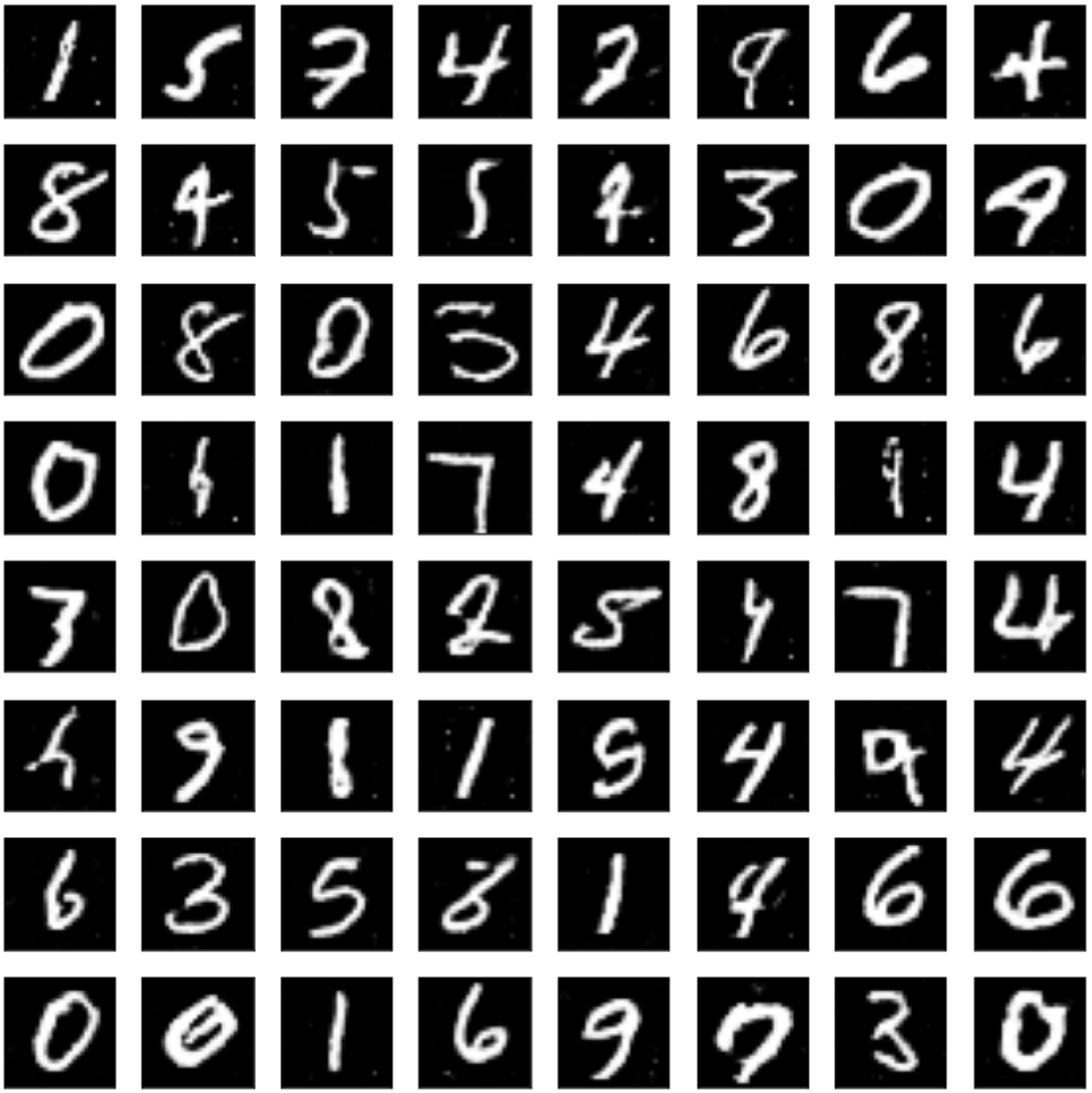
```
This is the 50/468 of epoch 16.
This is the 100/468 of epoch 16.
This is the 150/468 of epoch 16.
This is the 200/468 of epoch 16.
This is the 250/468 of epoch 16.
This is the 300/468 of epoch 16.
This is the 350/468 of epoch 16.
This is the 400/468 of epoch 16.
This is the 450/468 of epoch 16.
Epoch: 16/200 Discriminator Loss: 4.9882  Generator Loss: 0.0398
This is the 50/468 of epoch 17.
This is the 100/468 of epoch 17.
This is the 150/468 of epoch 17.
This is the 200/468 of epoch 17.
This is the 250/468 of epoch 17.
This is the 300/468 of epoch 17.
This is the 350/468 of epoch 17.
This is the 400/468 of epoch 17.
This is the 450/468 of epoch 17.
Epoch: 17/200 Discriminator Loss: 0.1286  Generator Loss: 8.8101
This is the 50/468 of epoch 18.
This is the 100/468 of epoch 18.
This is the 150/468 of epoch 18.
This is the 200/468 of epoch 18.
This is the 250/468 of epoch 18.
This is the 300/468 of epoch 18.
This is the 350/468 of epoch 18.
This is the 400/468 of epoch 18.
This is the 450/468 of epoch 18.
Epoch: 18/200 Discriminator Loss: 1.5127  Generator Loss: 0.3977
This is the 50/468 of epoch 19.
This is the 100/468 of epoch 19.
This is the 150/468 of epoch 19.
This is the 200/468 of epoch 19.
This is the 250/468 of epoch 19.
This is the 300/468 of epoch 19.
This is the 350/468 of epoch 19.
This is the 400/468 of epoch 19.
This is the 450/468 of epoch 19.
Epoch: 19/200 Discriminator Loss: 1.5344  Generator Loss: 2.9284
This is the 50/468 of epoch 20.
This is the 100/468 of epoch 20.
This is the 150/468 of epoch 20.
This is the 200/468 of epoch 20.
This is the 250/468 of epoch 20.
This is the 300/468 of epoch 20.
This is the 350/468 of epoch 20.
This is the 400/468 of epoch 20.
This is the 450/468 of epoch 20.
Epoch: 20/200 Discriminator Loss: 3.2209  Generator Loss: 0.0603
```



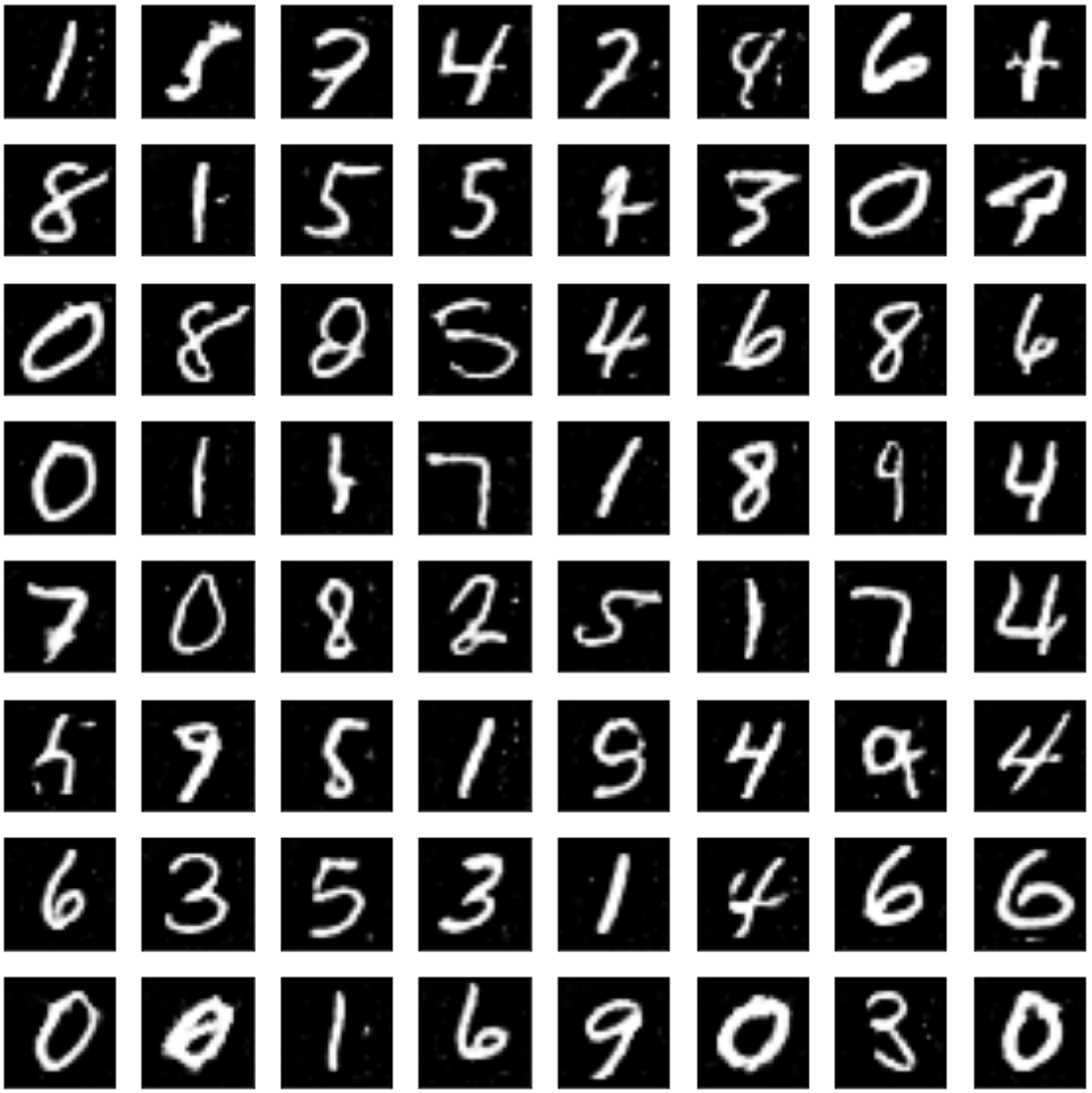
```
This is the 50/468 of epoch 46.
This is the 100/468 of epoch 46.
This is the 150/468 of epoch 46.
This is the 200/468 of epoch 46.
This is the 250/468 of epoch 46.
This is the 300/468 of epoch 46.
This is the 350/468 of epoch 46.
This is the 400/468 of epoch 46.
This is the 450/468 of epoch 46.
Epoch: 46/200 Discriminator Loss: 5.9879  Generator Loss: 6.7624
This is the 50/468 of epoch 47.
This is the 100/468 of epoch 47.
This is the 150/468 of epoch 47.
This is the 200/468 of epoch 47.
This is the 250/468 of epoch 47.
This is the 300/468 of epoch 47.
This is the 350/468 of epoch 47.
This is the 400/468 of epoch 47.
This is the 450/468 of epoch 47.
Epoch: 47/200 Discriminator Loss: 1.8670  Generator Loss: 2.3351
This is the 50/468 of epoch 48.
This is the 100/468 of epoch 48.
This is the 150/468 of epoch 48.
This is the 200/468 of epoch 48.
This is the 250/468 of epoch 48.
This is the 300/468 of epoch 48.
This is the 350/468 of epoch 48.
This is the 400/468 of epoch 48.
This is the 450/468 of epoch 48.
Epoch: 48/200 Discriminator Loss: 1.6252  Generator Loss: 4.0493
This is the 50/468 of epoch 49.
This is the 100/468 of epoch 49.
This is the 150/468 of epoch 49.
This is the 200/468 of epoch 49.
This is the 250/468 of epoch 49.
This is the 300/468 of epoch 49.
This is the 350/468 of epoch 49.
This is the 400/468 of epoch 49.
This is the 450/468 of epoch 49.
Epoch: 49/200 Discriminator Loss: 1.4940  Generator Loss: 0.4953
This is the 50/468 of epoch 50.
This is the 100/468 of epoch 50.
This is the 150/468 of epoch 50.
This is the 200/468 of epoch 50.
This is the 250/468 of epoch 50.
This is the 300/468 of epoch 50.
This is the 350/468 of epoch 50.
This is the 400/468 of epoch 50.
This is the 450/468 of epoch 50.
Epoch: 50/200 Discriminator Loss: 1.8510  Generator Loss: 2.3981
```



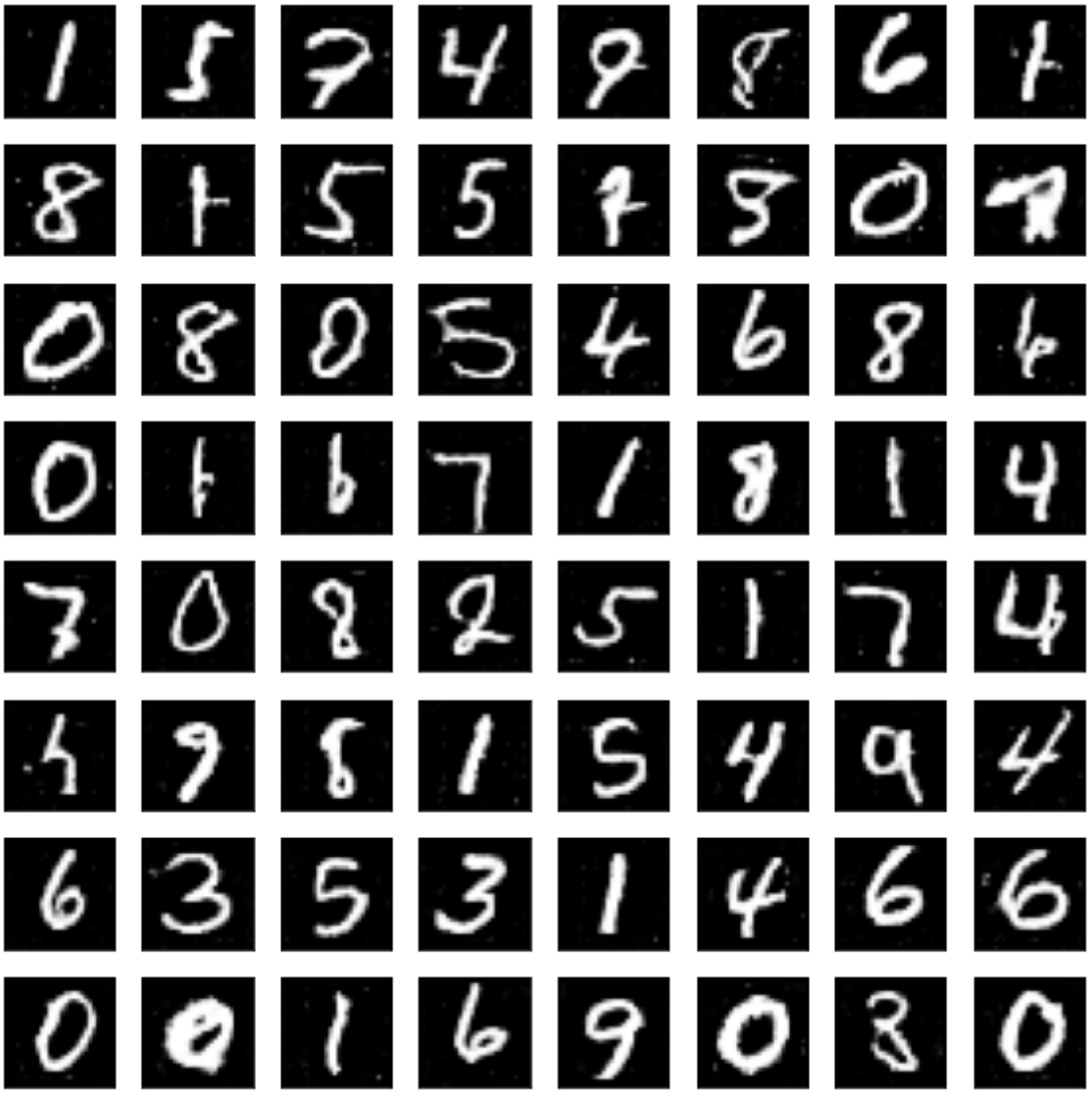
```
This is the 50/468 of epoch 96.
This is the 100/468 of epoch 96.
This is the 150/468 of epoch 96.
This is the 200/468 of epoch 96.
This is the 250/468 of epoch 96.
This is the 300/468 of epoch 96.
This is the 350/468 of epoch 96.
This is the 400/468 of epoch 96.
This is the 450/468 of epoch 96.
Epoch: 96/200 Discriminator Loss: 1.3333  Generator Loss: 1.4719
This is the 50/468 of epoch 97.
This is the 100/468 of epoch 97.
This is the 150/468 of epoch 97.
This is the 200/468 of epoch 97.
This is the 250/468 of epoch 97.
This is the 300/468 of epoch 97.
This is the 350/468 of epoch 97.
This is the 400/468 of epoch 97.
This is the 450/468 of epoch 97.
Epoch: 97/200 Discriminator Loss: 1.7839  Generator Loss: 0.2831
This is the 50/468 of epoch 98.
This is the 100/468 of epoch 98.
This is the 150/468 of epoch 98.
This is the 200/468 of epoch 98.
This is the 250/468 of epoch 98.
This is the 300/468 of epoch 98.
This is the 350/468 of epoch 98.
This is the 400/468 of epoch 98.
This is the 450/468 of epoch 98.
Epoch: 98/200 Discriminator Loss: 3.4745  Generator Loss: 3.6322
This is the 50/468 of epoch 99.
This is the 100/468 of epoch 99.
This is the 150/468 of epoch 99.
This is the 200/468 of epoch 99.
This is the 250/468 of epoch 99.
This is the 300/468 of epoch 99.
This is the 350/468 of epoch 99.
This is the 400/468 of epoch 99.
This is the 450/468 of epoch 99.
Epoch: 99/200 Discriminator Loss: 1.4244  Generator Loss: 1.5133
This is the 50/468 of epoch 100.
This is the 100/468 of epoch 100.
This is the 150/468 of epoch 100.
This is the 200/468 of epoch 100.
This is the 250/468 of epoch 100.
This is the 300/468 of epoch 100.
This is the 350/468 of epoch 100.
This is the 400/468 of epoch 100.
This is the 450/468 of epoch 100.
Epoch: 100/200 Discriminator Loss: 7.3426  Generator Loss: 8.2598
```



```
This is the 50/468 of epoch 146.
This is the 100/468 of epoch 146.
This is the 150/468 of epoch 146.
This is the 200/468 of epoch 146.
This is the 250/468 of epoch 146.
This is the 300/468 of epoch 146.
This is the 350/468 of epoch 146.
This is the 400/468 of epoch 146.
This is the 450/468 of epoch 146.
Epoch: 146/200 Discriminator Loss: 1.9790  Generator Loss: 1.9585
This is the 50/468 of epoch 147.
This is the 100/468 of epoch 147.
This is the 150/468 of epoch 147.
This is the 200/468 of epoch 147.
This is the 250/468 of epoch 147.
This is the 300/468 of epoch 147.
This is the 350/468 of epoch 147.
This is the 400/468 of epoch 147.
This is the 450/468 of epoch 147.
Epoch: 147/200 Discriminator Loss: 1.2184  Generator Loss: 0.7828
This is the 50/468 of epoch 148.
This is the 100/468 of epoch 148.
This is the 150/468 of epoch 148.
This is the 200/468 of epoch 148.
This is the 250/468 of epoch 148.
This is the 300/468 of epoch 148.
This is the 350/468 of epoch 148.
This is the 400/468 of epoch 148.
This is the 450/468 of epoch 148.
Epoch: 148/200 Discriminator Loss: 1.7664  Generator Loss: 1.7874
This is the 50/468 of epoch 149.
This is the 100/468 of epoch 149.
This is the 150/468 of epoch 149.
This is the 200/468 of epoch 149.
This is the 250/468 of epoch 149.
This is the 300/468 of epoch 149.
This is the 350/468 of epoch 149.
This is the 400/468 of epoch 149.
This is the 450/468 of epoch 149.
Epoch: 149/200 Discriminator Loss: 2.0948  Generator Loss: 2.1012
This is the 50/468 of epoch 150.
This is the 100/468 of epoch 150.
This is the 150/468 of epoch 150.
This is the 200/468 of epoch 150.
This is the 250/468 of epoch 150.
This is the 300/468 of epoch 150.
This is the 350/468 of epoch 150.
This is the 400/468 of epoch 150.
This is the 450/468 of epoch 150.
Epoch: 150/200 Discriminator Loss: 1.9737  Generator Loss: 1.8774
```

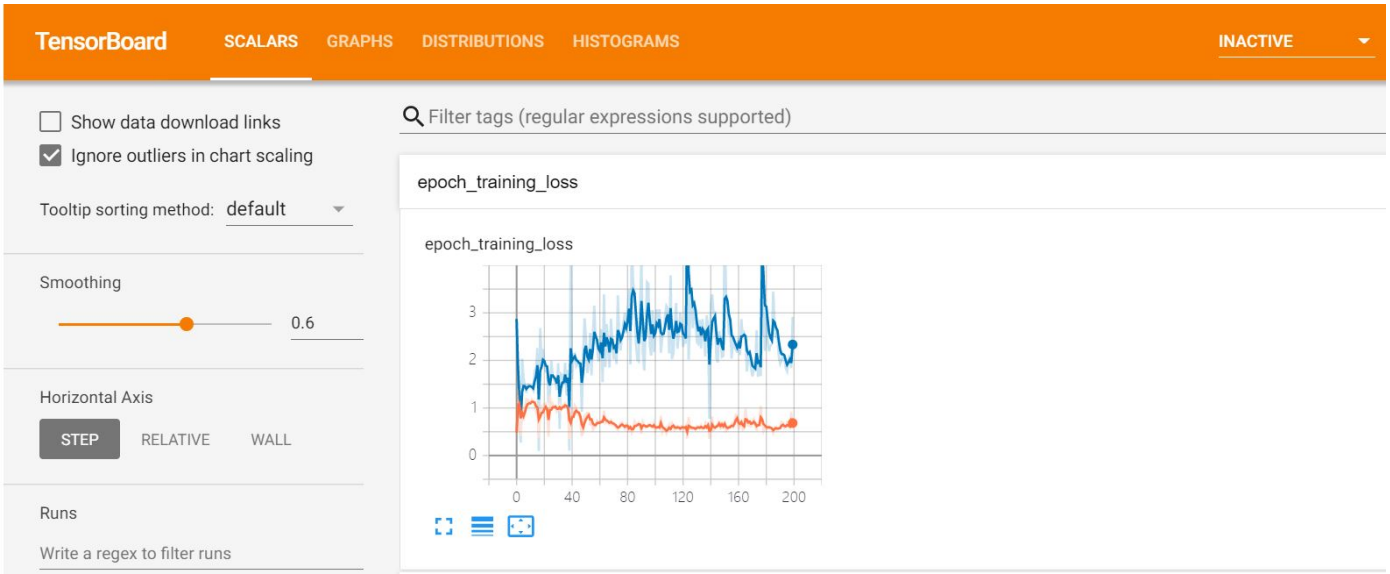


```
This is the 50/468 of epoch 196.
This is the 100/468 of epoch 196.
This is the 150/468 of epoch 196.
This is the 200/468 of epoch 196.
This is the 250/468 of epoch 196.
This is the 300/468 of epoch 196.
This is the 350/468 of epoch 196.
This is the 400/468 of epoch 196.
This is the 450/468 of epoch 196.
Epoch: 196/200 Discriminator Loss: 1.3412  Generator Loss: 1.3508
This is the 50/468 of epoch 197.
This is the 100/468 of epoch 197.
This is the 150/468 of epoch 197.
This is the 200/468 of epoch 197.
This is the 250/468 of epoch 197.
This is the 300/468 of epoch 197.
This is the 350/468 of epoch 197.
This is the 400/468 of epoch 197.
This is the 450/468 of epoch 197.
Epoch: 197/200 Discriminator Loss: 1.3674  Generator Loss: 0.6835
This is the 50/468 of epoch 198.
This is the 100/468 of epoch 198.
This is the 150/468 of epoch 198.
This is the 200/468 of epoch 198.
This is the 250/468 of epoch 198.
This is the 300/468 of epoch 198.
This is the 350/468 of epoch 198.
This is the 400/468 of epoch 198.
This is the 450/468 of epoch 198.
Epoch: 198/200 Discriminator Loss: 1.7052  Generator Loss: 1.5619
This is the 50/468 of epoch 199.
This is the 100/468 of epoch 199.
This is the 150/468 of epoch 199.
This is the 200/468 of epoch 199.
This is the 250/468 of epoch 199.
This is the 300/468 of epoch 199.
This is the 350/468 of epoch 199.
This is the 400/468 of epoch 199.
This is the 450/468 of epoch 199.
Epoch: 199/200 Discriminator Loss: 1.4229  Generator Loss: 0.6955
This is the 50/468 of epoch 200.
This is the 100/468 of epoch 200.
This is the 150/468 of epoch 200.
This is the 200/468 of epoch 200.
This is the 250/468 of epoch 200.
This is the 300/468 of epoch 200.
This is the 350/468 of epoch 200.
This is the 400/468 of epoch 200.
This is the 450/468 of epoch 200.
Epoch: 200/200 Discriminator Loss: 5.2428  Generator Loss: 6.4009
```

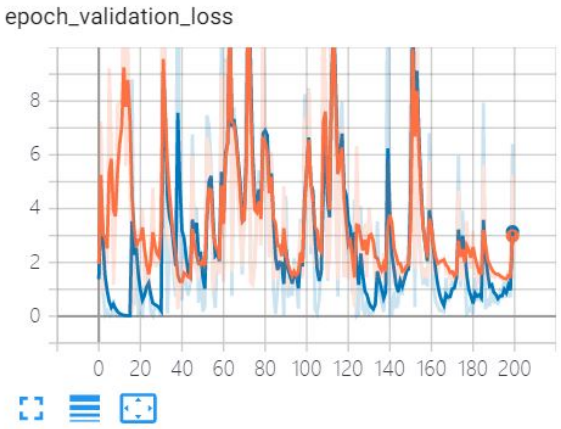


4603.364857912064

In [ ]:
```
%tensorboard --logdir=./logs_mnist
```



epoch_validation_loss



In [ ]: