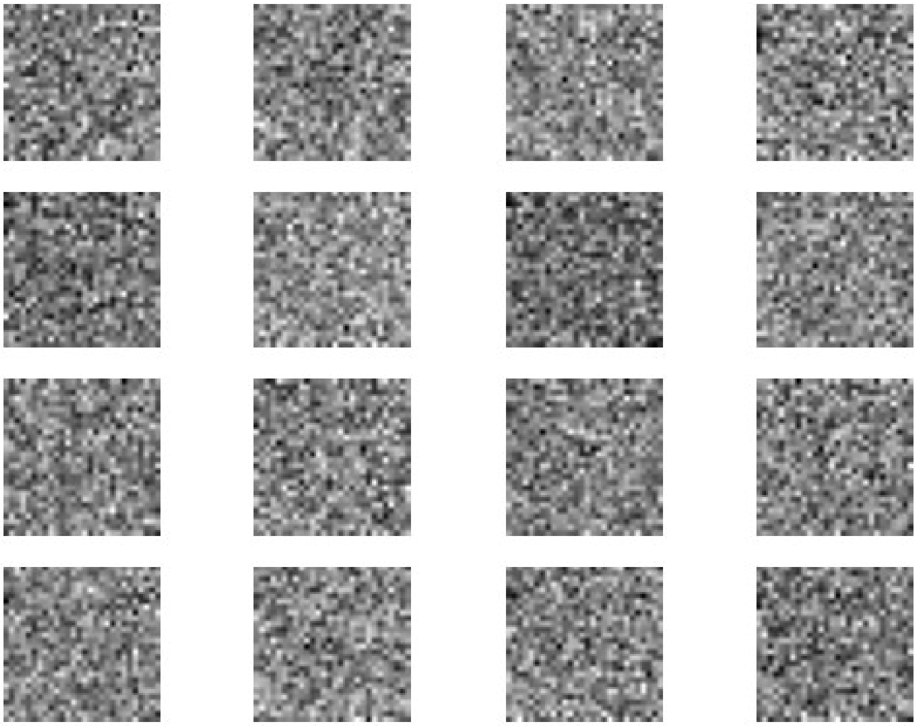
```
import tensorflow as tf
from tensorflow.keras import layers
import numpy as np
import matplotlib.pyplot as plt

# Define the generator model
def build_generator(latent_dim):
    model = tf.keras.Sequential([
        layers.Dense(128, input_dim=latent_dim, activation='relu'),
        layers.Dense(784, activation='sigmoid'),
        layers.Reshape((28, 28, 1))
    ])
    return model

# Define the discriminator model
def build_discriminator():
    model = tf.keras.Sequential([
        layers.Flatten(input_shape=(28, 28, 1)),
        layers.Dense(128, activation='relu'),
        layers.Dense(1, activation='sigmoid')
    ])
    return model

# Define the GAN model
def build_gan(generator, discriminator):
    discriminator.trainable = False
    model = tf.keras.Sequential([
        generator,
        discriminator
    ])
    return model

# Prepare the MNIST dataset
(x_train, _), (_, _) = tf.keras.datasets.mnist.load_data()

x_train = x_train / 255.0
x_train = np.expand_dims(x_train, axis=-1)

# Build the models
latent_dim = 100
generator = build_generator(latent_dim)
discriminator = build_discriminator()
gan = build_gan(generator, discriminator)

# Compile the models
discriminator.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
gan.compile(optimizer='adam', loss='binary_crossentropy')

# Training the GAN
```

```
epochs = 10000
batch_size = 64

for epoch in range(epochs):
    noise = np.random.normal(0, 1, size=(batch_size, latent_dim))
    generated_images = generator.predict(noise)
    idx = np.random.randint(0, x_train.shape[0], batch_size)
    real_images = x_train[idx]
    labels_real = np.ones((batch_size, 1))
    labels_fake = np.zeros((batch_size, 1))
    d_loss_real = discriminator.train_on_batch(real_images, labels_real)
    d_loss_fake = discriminator.train_on_batch(generated_images, labels_fake)
    d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)
    noise = np.random.normal(0, 1, size=(batch_size, latent_dim))
    labels_gan = np.ones((batch_size, 1))
    g_loss = gan.train_on_batch(noise, labels_gan)

    if epoch % 100 == 0:
        print(f"Epoch {epoch}, Discriminator Loss: {d_loss[0]}, Generator Loss: {g_loss}")
        gen_imgs = generator.predict(np.random.normal(0, 1, size=(16, latent_dim)))
        gen_imgs = 0.5 * gen_imgs + 0.5
        fig, axs = plt.subplots(4, 4)
        count = 0
        for i in range(4):
            for j in range(4):
                axs[i, j].imshow(gen_imgs[count, :, :, 0], cmap='gray')
                axs[i, j].axis('off')
                count += 1
        plt.show()
```

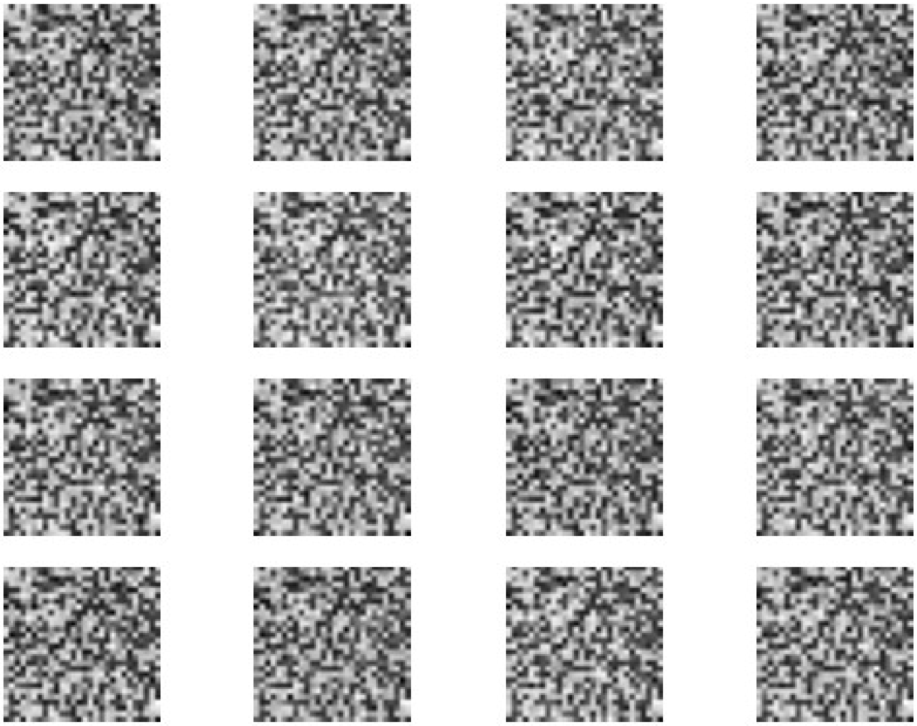
```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
2/2 [==============================] - 1s 23ms/step
Epoch 0, Discriminator Loss: 0.7822366058826447, Generator Loss: 0.44780734181404114
1/1 [==============================] - 0s 158ms/step
```



```
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 13ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 11ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 19ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 18ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 17ms/step
2/2 [==============================] - 0s 22ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 23ms/step
```

```
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 13ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 12ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 12ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 13ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 13ms/step
2/2 [==============================] - 0s 11ms/step
2/2 [==============================] - 0s 14ms/step
2/2 [==============================] - 0s 14ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 11ms/step
2/2 [==============================] - 0s 11ms/step
2/2 [==============================] - 0s 15ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 7ms/step
```

```
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 4ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 4ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 12ms/step
2/2 [==============================] - 0s 12ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 12ms/step
2/2 [==============================] - 0s 11ms/step
Epoch 100, Discriminator Loss: 3.6462402939796448, Generator Loss:
0.0013842870248481631
1/1 [==============================] - 0s 34ms/step
```



```
2/2 [==============================] - 0s 12ms/step
2/2 [==============================] - 0s 11ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 4ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 5ms/step
```

```
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 17ms/step
2/2 [==============================] - 0s 18ms/step
2/2 [==============================] - 0s 18ms/step
2/2 [==============================] - 0s 16ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 5ms/step
2/2 [==============================] - 0s 11ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 10ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 6ms/step
2/2 [==============================] - 0s 9ms/step
2/2 [==============================] - 0s 8ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
2/2 [==============================] - 0s 7ms/step
```