Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Gen AI	**Write a code for VANILLA GAN. USE DATASET MNIST FROM KERAS.**	
Experiment	**Date:**	**Enrolment No:92200133020**

CODE:

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, LeakyReLU, BatchNormalization, Reshape, Flatten, Input
from tensorflow.keras.optimizers import Adam
```

Python

```python
(X_train, _), (_, _) = mnist.load_data()
X_train = (X_train.astype(np.float32) - 127.5) / 127.5
X_train = np.expand_dims(X_train, axis=-1)  # Add channel dimension
```

Python

```python
# 1. Parameters
img_rows, img_cols, channels = 28, 28, 1
img_shape = (img_rows, img_cols, channels)
z_dim = 100
```

Python

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
# 2. Build the Generator
def build_generator(z_dim):
    model = Sequential()
```

```python
    model.add(Dense(256, input_dim=z_dim))
    model.add(LeakyReLU(alpha=0.2))
    model.add(BatchNormalization(momentum=0.8))
    model.add(Dense(512))
    model.add(LeakyReLU(alpha=0.2))
    model.add(BatchNormalization(momentum=0.8))
    model.add(Dense(1024))
    model.add(LeakyReLU(alpha=0.2))
    model.add(BatchNormalization(momentum=0.8))
    model.add(Dense(np.prod(img_shape), activation='tanh'))
    model.add(Reshape(img_shape))
    return model
```

Python

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
# 3. Build the Discriminator
def build_discriminator(img_shape):
    model = Sequential()
    model.add(Flatten(input_shape=img_shape))
    model.add(Dense(512))
    model.add(LeakyReLU(alpha=0.2))
    model.add(Dense(256))
    model.add(LeakyReLU(alpha=0.2))
    model.add(Dense(1, activation='sigmoid'))
    return model
```


Python

```python
# 4. Compile Models
optimizer = Adam(0.0002, 0.5)
```

Python

```python
# 5. Load & Preprocess MNIST Data
(X_train, _), (_, _) = mnist.load_data()
X_train = (X_train.astype('float32') - 127.5) / 127.5
X_train = np.expand_dims(X_train, axis=-1)
```

Python

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
# Create generator and discriminator models
generator = build_generator(z_dim)
discriminator = build_discriminator(img_shape)

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

# Make discriminator non-trainable for the combined model
discriminator.trainable = False

# Build and compile the combined model
z = Input(shape=(z_dim,))
img = generator(z)
valid = discriminator(img)
combined = Model(z, valid)
combined.compile(loss='binary_crossentropy', optimizer=optimizer)
```


GetBot AI: Explain | Find Error | Find Resource Leaks

Click to add a breakpoint

```python
batch_size = 128
save_interval = 1000
half_batch = batch_size // 2

for epoch in range(1, epochs + 1):
    # Train Discriminator
    idx = np.random.randint(0, X_train.shape[0], half_batch)
    real_imgs = X_train[idx]

    noise = np.random.normal(0, 1, (half_batch, z_dim))
    fake_imgs = generator.predict(noise)

    d_loss_real = discriminator.train_on_batch(real_imgs, np.ones((half_batch, 1)))
    d_loss_fake = discriminator.train_on_batch(fake_imgs, np.zeros((half_batch, 1)))
    d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)

    # Train Generator
    noise = np.random.normal(0, 1, (batch_size, z_dim))
    valid_y = np.ones((batch_size, 1))
    g_loss = combined.train_on_batch(noise, valid_y)

    if epoch % save_interval == 0 or epoch == 1:
        print(f"{epoch} [D loss: {d_loss[0]:.4f}, acc.: {100*d_loss[1]:.2f}%] [G loss: {g_loss:.
```


Python

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step
...
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
1/2 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...