EXP No:11 CASE STUDY-GAIN Implements Aim:- The aim of this Case Study is to implement to a Generative Adversarial Network (GAN) to learn and generates Samples froma 2D Gaussian distribution the generator Should Create gynthetic data points. that resenble the real data distribution ,while the discriminator should differentiate blw real and faredata. Algorithm:- 1. Intialize Data Distribution:- • Generate a 2D Gaussian dataset Where each point is Sampled from a normal distribution. 2. Build the Generate Model:- • A neural network that tales random noise (latent Space) as input and outputs a 2D data point. 3. Build the Discriminator :- • Classifies inputas real or Pake using a neural network. 4. Compile the GAN:- • Train the discriminator Seperatelx • Connect generator output to discriminator and train 5. Train the GAN:- • Generate real and Pare data. • Train discriminator or both Here is the extracted text from the image: Program:- import numpy as np import matplotlib.pylot as pit import ten Sor Flow as tf From tensor Flow Keras. layer import Dense, Leaky ReLU def real - data-distribution (n-Samples = 1000): X1 = np.random. normal

axis = 1) def build-generat of input__dim. out-dim), Leaky ReLU (alpha=0.2), Dense (32), Leaky ReLU (alpha= 0.2) Dense. (output-dim)] def buid - discriminator (input-dim): model = Sequential ([Dense (32, input - dim = input-dim), Leaky ReLU (alpha = 0.2), Dense (161, -Leaky Relu (alpha=0.2), Dense (1, activation = 'sigmoid')] return model. def Compile-gan (generator, discrimination, learning__rate=0.002) dis Criminator Compile Coptimizer = IF. Keras.optimizers. Adam (learning-rate), loss = 'binary- Grossentropy', metrics = [accuracy]) discrimination train able = False gan-input = It keras. Input (shape=(generator. input__Shupe) gan-output = discriminator (generator (gan-Input11 gan = IP. Keras. Model |gain-input, gan-output) gan. Compile (optimizer = It. keras optimizers. Adam) return gan. PIE.show() latent-dim=2 generator = buid - generator (latent - dim, 2) dis Criminator = build - discriminator (2) gan = Compile - gan (generator, discriminatorl train- gan generator, driscriminator 1 gan, real-data=128) Result:- In the Conclusion, this implements Succesfully demon strates how a simple GAN Can generate realistic 20 data distributions from a Gaussian dataset. win deep