

這段程式碼實作了一個生成對抗網路（GAN）的基本示範，主要包含以下幾個部分：

1. ****準備環境****：

- 載入必要的 Python 庫，包括`torch`、`torchvision`和`numpy`等。
- 檢查 PyTorch 和相關庫的版本。

2. ****定義生成器和判別器網路****：

- ****生成器（Generator）****：將隨機噪聲轉換為圖像。網路結構包括多層全連接層和激活函數。
- ****判別器（Discriminator）****：判斷輸入圖像是真實的還是由生成器生成的假圖像。網路結構類似，包含多層全連接層和 Dropout 層以防過擬合。

3. ****數據處理****：

- 使用`torchvision`載入 MNIST 數據集，進行標準化處理。
- 定義生成噪聲的函數，以供生成器使用。

4. ****GAN 訓練****：

- 設定訓練過程中的損失函數和優化器。
- 訓練判別器：對真實數據和生成的假數據進行分類，計算損失。
- 訓練生成器：根據判別器的反饋更新生成器的權重，使生成的數據更逼近真實數據。

5. ****結果展示****：

- 訓練過程中，定期儲存生成的圖像樣本。

- 繪製生成器和判別器的損失曲線，顯示模型訓練的進展。
- 顯示不同訓練周期的生成圖像，觀察生成效果的變化。

6. **模型評估**：

- 使用 MMD（Maximum Mean Discrepancy）來評估生成圖像與真實圖像之間的距離，衡量生成模型的表現。

這段程式碼展示了如何從頭開始實作 GAN，包括網路架構的定義、訓練過程的設計以及結果的可視化和評估。