Report: Synthetic Data Generation for Fashion Items with GANs

Objective

The goal of this assignment is to use a Generative Adversarial Network (GAN) to create synthetic images of fashion items. By training the GAN on the Fashion-MNIST dataset, the objective is to generate realistic-looking fashion item images that resemble the real samples in the dataset. GANs consist of two neural networks, a generator and a discriminator, which compete against each other in a process that improves the generator's ability to create realistic synthetic images.

Dataset

The Fashion-MNIST dataset contains 60,000 grayscale images of 28×28 pixels, representing 10 different fashion categories, including t-shirts, pants, dresses, shoes, and bags. Each image is labeled with its corresponding category, and the dataset provides a diverse set of fashion items that are ideal for training the GAN.

- **Total Images**: 60,000 training images, 10,000 test images.
- **Classes**: 10 classes including t-shirts, trousers, pullover, dresses, etc.

Model Architecture

Generator

The generator's job is to take a random noise vector (latent vector) and transform it into an image that resembles a fashion item. The generator consists of several layers that upsample the noise vector into a higher-dimensional space. This network uses a combination of dense layers and transposed convolutions to increase the spatial resolution of the generated image.

- **Input**: A random vector (latent space) of size 100.
- **Layers**:
    - Dense Layer (e.g., 128 units, activation: ReLU).
    - Reshape Layer to convert the vector into a feature map.
    - Transposed Convolutions (2D) to upsample the image progressively.
    - Final output layer (28×28 pixels, 1 channel, sigmoid activation).

Discriminator

The discriminator is a binary classifier that distinguishes between real and synthetic images. It takes an image as input and outputs a probability indicating whether the image is real (from the Fashion-MNIST dataset) or fake (generated by the generator).

- **Input**: 28×28 grayscale image.
- **Layers**:
    - Convolutional Layers (e.g., 64, 128 filters, kernel size 5×5, activation: LeakyReLU).
    - Flattening Layer.
    - Dense Layer for binary classification (real or fake).
    - Output: A single probability value (sigmoid activation).

GAN Architecture

The GAN combines the generator and discriminator:

- The **generator** creates synthetic fashion images, while the **discriminator** evaluates whether the images are real or fake.
- Both networks are trained in alternating steps: the discriminator is updated with real and fake images, and the generator is updated based on the discriminator's feedback.

Training Procedure

Training Process

1. **Initialize Generator and Discriminator**: The generator creates images from random noise, and the discriminator evaluates them alongside real images.
2. **Alternating Training**:
   - **Step 1**: Train the discriminator to distinguish real and fake images. It is fed both real Fashion-MNIST images and fake images generated by the generator.
   - **Step 2**: Train the generator to fool the discriminator. The generator is updated based on how well it can trick the discriminator into classifying fake images as real.
3. **Loss Functions**:
   - **Discriminator Loss**: Binary cross-entropy between predicted labels (real/fake) and actual labels.
   - **Generator Loss**: Binary cross-entropy between predicted labels (real/fake) and the target label (real).

Hyperparameters:

- **Epochs**: 20 (Adjusted for limited computing power)
- **Batch Size**: 24 (Reduced for better efficiency)
- **Learning Rate**: 0.0001 (for both generator and discriminator, reduced for stability)
- **Optimizer**: Adam optimizer ($\beta 1 = 0.001$ for smoother updates)

Evaluation

Synthetic Image Generation

During training, synthetic images are generated by the generator at various stages. These images are displayed to visually inspect the progress of the generator in creating more realistic images over time.

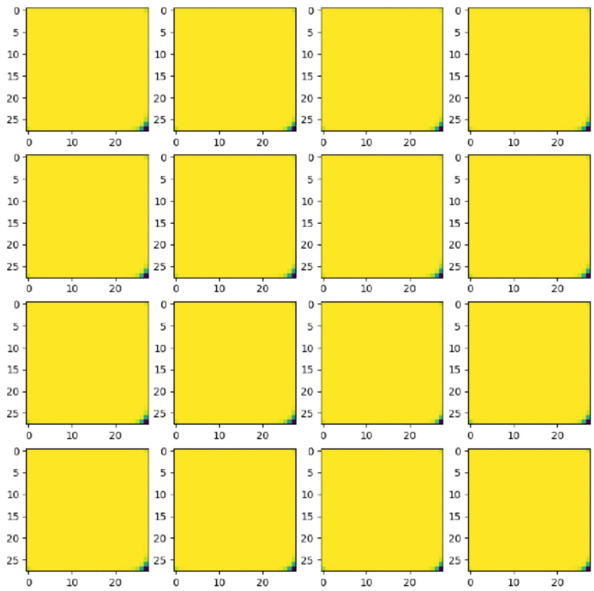
- **Early Stage**: The generated images are usually blurry or have random patterns, as the generator is still learning.
- **Mid Stage**: The generated images begin to show some recognizable features of fashion items but may still lack fine details.

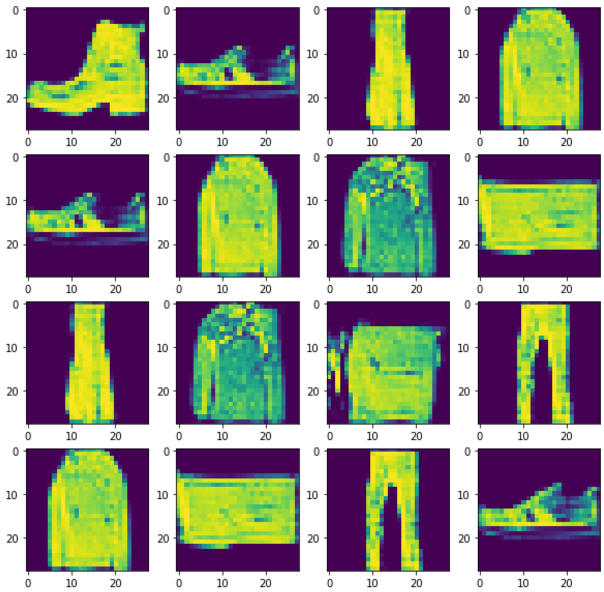
- **Late Stage**: The generated images become more refined and start resembling real Fashion-MNIST images.

Comparison with Real Fashion-MNIST Samples

At different stages of training, the synthetic images generated by the GAN are compared with real Fashion-MNIST images to assess the quality of the generated samples.

- **Early Training**: The generated images are far from realistic, with distorted features and lack of clear structure.
- **Mid Training**: The generator starts producing coherent patterns, but some items may still be poorly generated.





Early Stage (epoch 5) Mid Stage (Epoch 20)

Results

1. **Training Progress**: At the beginning of the training, the generated images are random noise. Over the epochs, the generator improves its ability to generate more realistic images. The generator's performance becomes noticeably better as it learns from the feedback provided by the discriminator.
2. **Sample Images**:
   - **Epoch 1**: Sample images are blurred and abstract.

- **Epoch 10**: Some recognizable fashion items are generated, but with low resolution and artifacts.
- **Epoch 20**: The generated fashion items are realistic, with clear structures and details.

3. **Quality Evaluation**:
    - **Precision**: The generator effectively learns to create fashion items that resemble real ones.
    - **Diversity**: The generated images show a wide variety of fashion items, demonstrating the model's ability to generalize across the different Fashion-MNIST classes.

Conclusion

The GAN successfully generates synthetic images of fashion items after training on the Fashion-MNIST dataset. At the later stages of training, the generator produces visually appealing and realistic images, demonstrating the power of GANs for synthetic data generation. This model can be extended to generate more complex datasets and used for augmenting real-world image datasets for fashion-related applications.