Fashion GANerator

Dharathi Venkatesh

Content

1. Introduction

The Fashion GANerator project delves into creating computer-generated images of ankle boots. Using a Generative Adversarial Network (GAN), the computer learns to make images resembling real ankle boots. We carefully select ankle boot images from the Fashion MNIST dataset and prepare them for the computer.

A Generative Adversarial Network (GAN) is a computational model that uses a dynamic interplay between a discriminator and a generator to generate realistic data, such as images or other content. The Discriminator acts as a discerning judge, learning to distinguish between real and computer-generated images. On the other hand, the Generator is the creative force, transforming random inputs into images. The challenge for the Generator is to consistently create images realistic enough to fool the Discriminator. Through a continuous learning process, the Generator improves its ability to craft lifelike images, while the Discriminator enhances its discernment skills. This interplay allows the computer system to generate images that closely mimic reality, exemplified in the Fashion GANerator project, particularly in recreating ankle boot images. This collaboration between discernment and creativity makes GANs invaluable in the world of computer-generated imagery.

2. Objective

The Fashion GANerator project aimed to explore and apply Generative Adversarial Networks (GANs) to create realistic images of ankle boots. The computer was trained to distinguish between real and computer-generated images, refining the GAN's abilities over iterations. Various GAN implementations were compared, focusing on efficiency and performance. The entire journey was documented on GitHub for a thorough understanding and future reference.

3. Dataset

The dataset for the Fashion GANerator project is sourced from Fashion MNIST, with a specific focus on ankle boots among various clothing items. To enhance efficiency, the dataset size was reduced, and each image was standardised to 28x28 pixels. Pixel values were adjusted for computer-friendly processing. This refined dataset serves as the foundation for training the Generative Adversarial Network (GAN) to generate lifelike images of ankle boots.

4. Implementation

The step-by-step implementation of the Fashion GANerator project:

1. Dataset Selection:

- Utilised the Fashion MNIST dataset, focusing specifically on ankle boots among various clothing items.

2. Data Preprocessing:

- Reduced the dataset size for computational efficiency.
- Standardised each image to 28x28 pixels.
- Adjusted pixel values to ensure compatibility with computer processing.

```
(x_train,y_train),(x_test,y_test)=fashion_mnist.load_data()
x_train=x_train[np.isin(y_train,[9])] # Number 9 is "ankle boot"

# modification: reducing dataset to save time
print("original dataset size: ", len(x_train), len(x_test))
x_train = x_train[:100]
x_test = x_test[:100]

x_train = (x_train.astype('float32')/255.0)*2.0-1.0 # [-1,1] Section
x_test = (x_test.astype('float32')/255.0)*2.0-1.0
x_train = np.reshape(x_train, (len(x_train), 28, 28, 1))
x_test = np.reshape(x_test, (len(x_test), 28, 28, 1))

print("reduced dataset size: ", len(x_train), len(x_test))
```

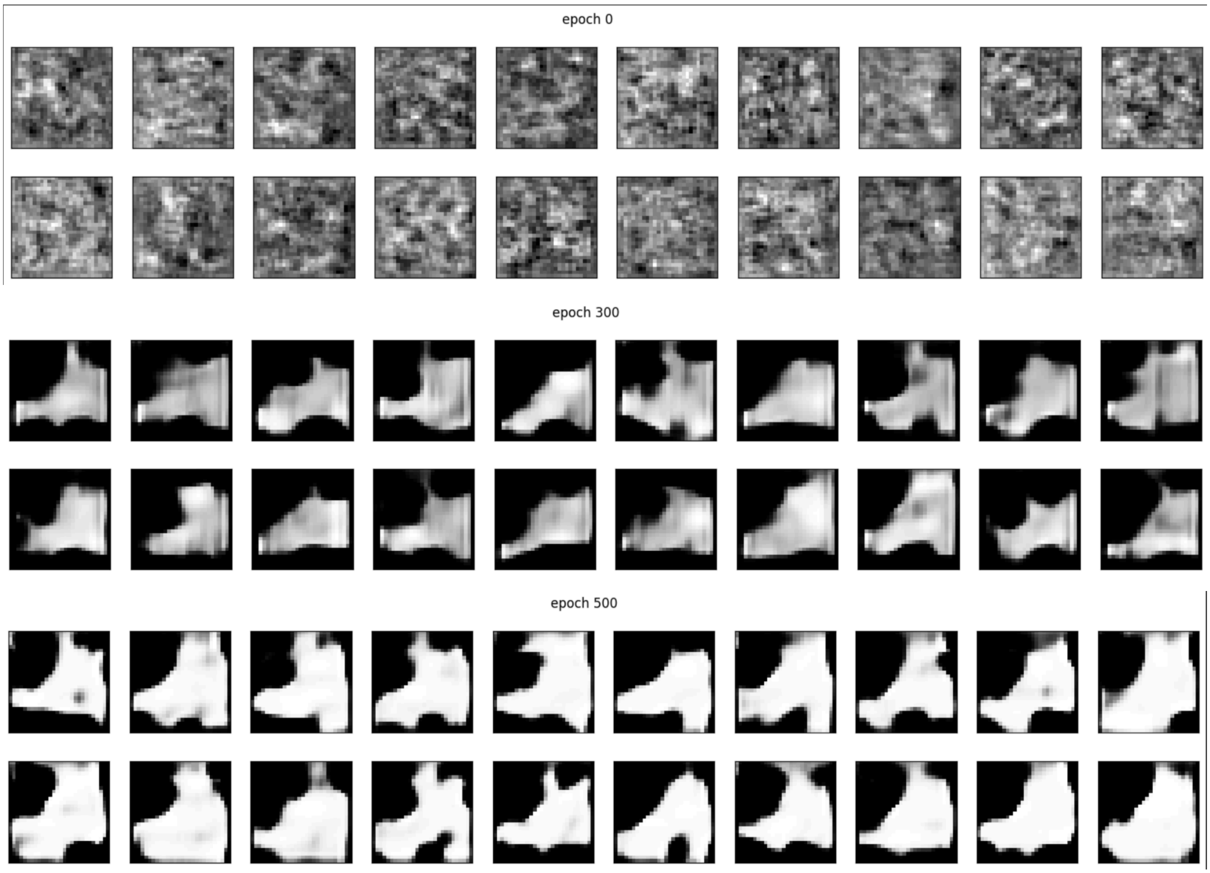
```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [==============================] - 0s 0us/step
original dataset size:  6000 10000
reduced dataset size:  100 100
```

3. GAN Architecture Setup:

- Implemented a Generative Adversarial Network (GAN) consisting of a Discriminator and a Generator.

4. Discriminator & Generator Training:

- Trained the Discriminator to distinguish between real and generated images.
- Trained the Generator to create realistic images that could potentially fool the Discriminator.
- Iteratively refined its abilities through exposure to examples.
- The adversarial learning process continued iteratively.



5. Alternative GAN Exploration:

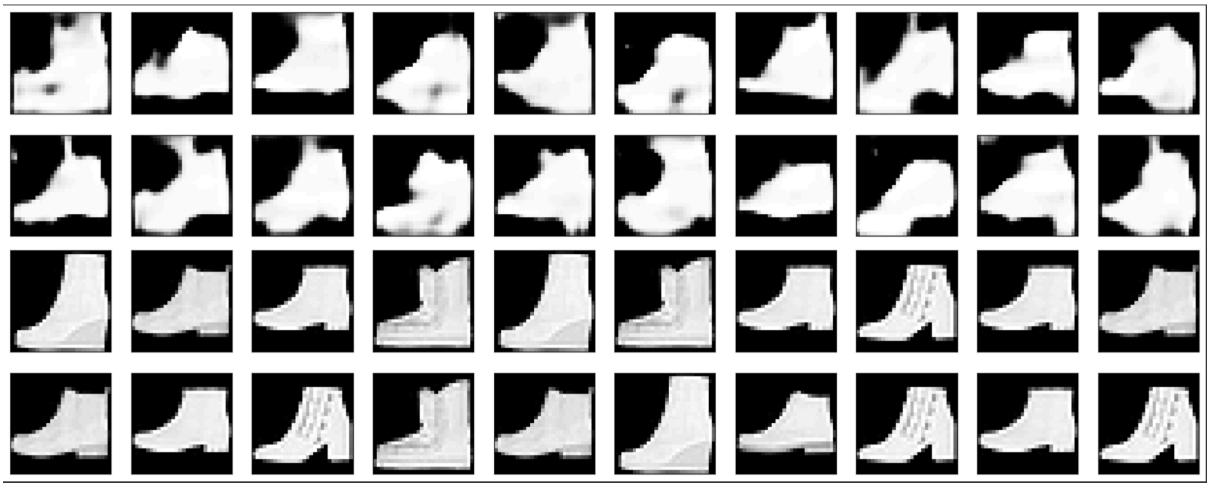
- Explored an alternative GAN implementation, experimenting with a smaller Discriminator for increased efficiency.

- Changes made to the new model (GAN1: original model; GAN2: The new and better model):
    - Discriminator architecture in GAN2 uses fewer filters in Conv2D layers compared to GAN1.
    - Generator architecture in GAN2 has a modified Dense layer with fewer channels than GAN1.
    - Batch normalisation in the generator of GAN2 operates with a reduced number of channels.
    - Conv2D layers in both GAN1 and GAN2 utilise the same (2,2) stride, with adjustments in the number of filters.
    - Both GAN1 and GAN2 include dropout layers with a specified dropout rate for regularisation during training.
    - Trainable parameters in the discriminator and generator networks are adjusted based on architectural changes.
    - The summary of discriminator, generator, and GAN networks provides insights into the overall network modifications from GAN1 to GAN2.

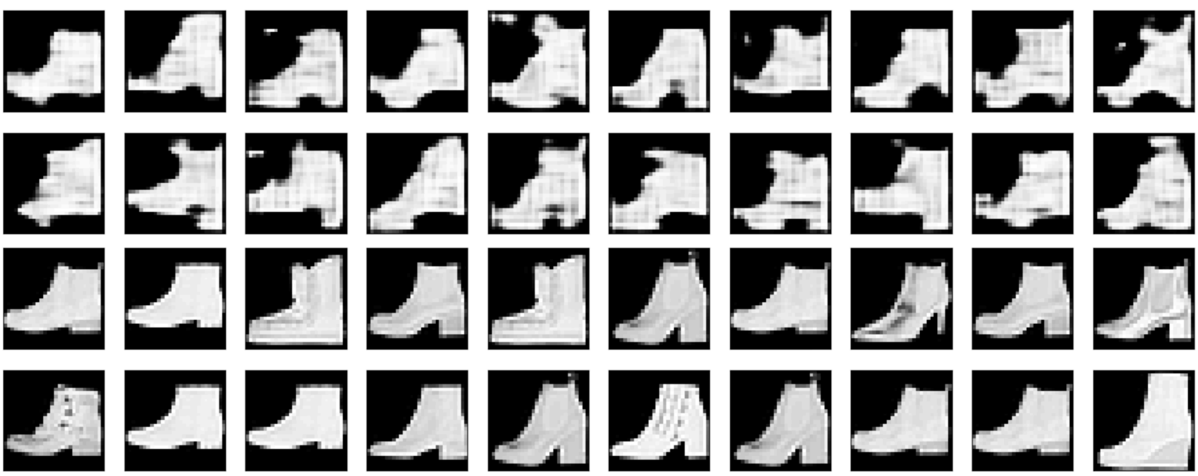
5. Result and Evaluation

The effectiveness of the Fashion GANerator project is assessed through a visual examination of the generated images. The code produces 20 synthetic images using the trained Generator, showcasing its ability to autonomously create images resembling real ankle boots. These generated images are then compared with their closest counterparts from the original training set. The evaluation includes a display of both the synthetic and real images, emphasising the GAN's proficiency in generating realistic content. Additionally, the differences between the generated and closest real images are quantified and analysed, offering insights into the GAN's accuracy.

Evaluation of the first model:



Evaluation of the second (better) model:



Displaying the improvement in the new and better GAN:

```
Succesful in reducing discriminator's size
Succesful in reducing GAN's size
Discriminator Size Ratio: 25.6%
GAN Size Ratio: 41.0%
    Difference: 1.1492, Time: 77.71
Model: "model_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_6 (InputLayer)        [(None, 100)]             0

 model_4 (Functional)        (None, 28, 28, 1)         106641

 model_3 (Functional)        (None, 1)                 29089

=================================================================
Total params: 135,730
Trainable params: 104,881
Non-trainable params: 30,849
```

5. Conclusion & Future Work

In conclusion, this project explored the generation of images using Generative Adversarial Networks (GANs). By modifying the architecture of the GAN, we experimented with different configurations to improve the generation of realistic images. The results, showcased through image samples, demonstrate the impact of architectural adjustments on the generated outputs. Moving forward, there is ample scope for further enhancements, including fine-tuning hyperparameters, experimenting with additional GAN architectures, and exploring advanced techniques for image generation. This project serves as a foundation for future endeavours in the exciting field of generative modelling.