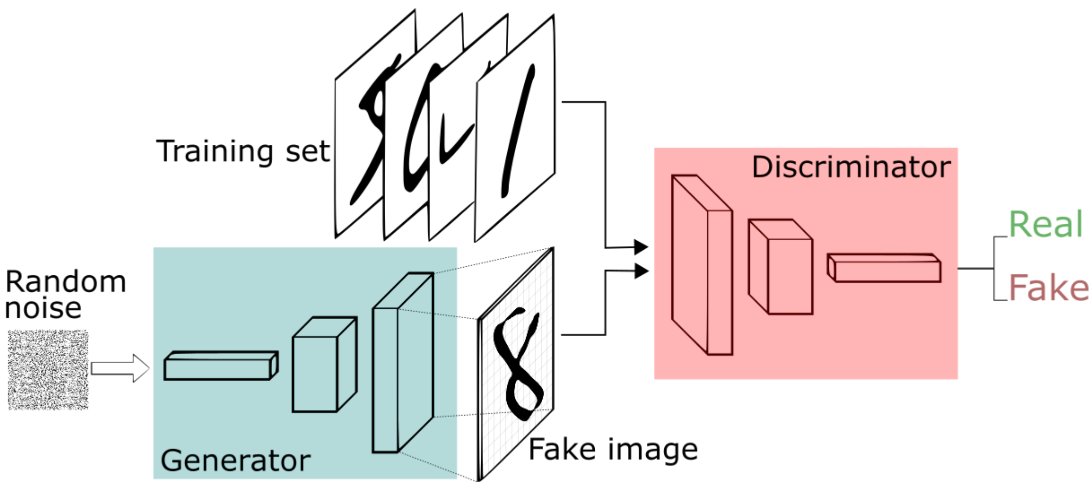
Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks Implementation



by
Arsalan Sepahpour A16556049
Justin Volheim A16582348

What are GANs?

GANs, or Generative Adversarial Networks, are a type of deep learning model that consists of two neural networks: a generator and a discriminator. The generator network is trained to generate synthetic data that resembles real data, while the discriminator network is trained to distinguish between real and synthetic data.

During training, the generator generates synthetic data, and the discriminator classifies the data as real or synthetic. The generator is then updated based on the feedback from the discriminator to improve its ability to generate more realistic data. This process continues until the generator produces synthetic data that is indistinguishable from real data.

Initial Proposal:

Initially, we had the idea of using a GAN to generate audio data to see if it sounded like English. However, the audio dataset we had was too small to produce meaningful results. We then stumbled upon a music dataset that consisted of MIDI files, which allowed us to separate each instrument. We thought about generating an image of each MIDI instrument and passing it through the GAN. Unfortunately, this approach required extensive data processing, which would have taken a long time. Therefore, we decided to pivot and instead use the Fashion MNIST dataset, a classic image dataset in deep learning.

MNIST Fashion Dataset:

The MNIST Fashion Dataset is a widely used benchmark dataset in the field of computer vision and machine learning. It consists of 70,000 grayscale images of 28x28 pixels, which represent different clothing items such as shirts, dresses, shoes, and bags. The dataset is divided into 60,000 training images and 10,000 testing images, each labeled with a corresponding class from 10 different categories. The categories include T-shirts/tops, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots. The MNIST Fashion Dataset was created as a successor to the original MNIST dataset, which contains handwritten digits. The dataset is widely used for tasks such as image classification, object detection, and generative models such as GANs.

Discriminator and Generator:

A GAN (Generative Adversarial Network) consists of two neural networks, the generator and the discriminator, which work together to generate realistic synthetic data. The generator is responsible for creating the synthetic data by mapping a random noise vector to the output data space. It generates synthetic data by learning the patterns and structures of the real data through training. The generator tries to produce data that is as realistic as possible and can trick the discriminator into thinking that the synthetic data is real.

The discriminator, on the other hand, is responsible for distinguishing between the real data and the synthetic data produced by the generator. It receives both real and synthetic data as input and assigns a probability to each sample, indicating the likelihood that the sample is real. The discriminator is trained to correctly identify the real data and reject the synthetic data generated by the generator.

During training, the generator and discriminator are trained simultaneously in an adversarial manner. The generator tries to create synthetic data that is increasingly difficult for the discriminator to distinguish from the real data, while the discriminator becomes better at distinguishing between the two types of data. As a result, the generator learns to produce more realistic data that can pass the discriminator's evaluation. This process continues until the generator produces data that is indistinguishable from the real data, resulting in a successful GAN.

We have utilized object-oriented programming to implement both the discriminator and generator models. These models are implemented in a manner that replicates the approach described in the paper "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks".

Discriminator and Generator Implementation:

```
class Generator(nn.Module):

    def __init__(self):

    super(Generator,self).__init__()

    self.model = nn.Sequential(
        nn.ConvTranspose2d(64, 512, 4, 1, 0, bias = False),
        nn.BatchNorm2d(512),
        nn.ReLU(True),

        nn.ConvTranspose2d(512, 256, 4, 2, 1, bias=False),
        nn.BatchNorm2d(256),
        nn.ReLU(True),

        nn.ConvTranspose2d(256, 128, 4, 2, 1, bias=False),
        nn.BatchNorm2d(128),
        nn.ReLU(True),

        nn.ConvTranspose2d(128, 64, 4, 2, 1, bias=False),
        nn.BatchNorm2d(64),
        nn.ReLU(True),

        nn.ConvTranspose2d(64, number_of_channels, 4, 2, 1,bias=False),
        nn.Tanh()
        )

    def forward(self,x):
    return self.model(x)
```

```
class Discriminator(nn.Module):

    def __init__(self):

    super(Discriminator,self).__init__()

    self.model = nn.Sequential(
        nn.Conv2d(number_of_channels, 64, 4, 2, 1, bias=False),
        nn.LeakyReLU(0.2, inplace=True),

        nn.Conv2d(64, 128, 4, 2, 1, bias=False),
        nn.LeakyReLU(0.2, inplace=True),

        nn.Conv2d(128, 256, 4, 2, 1, bias=False),
        nn.LeakyReLU(0.2, inplace=True),

        nn.Conv2d(256, 512, 4, 2, 1, bias=False),
        nn.LeakyReLU(0.2, inplace=True),

        nn.Conv2d(512, 1, 4, 1, 0, bias=False),
        nn.Sigmoid()
    )

    def forward(self,x):
    return self.model(x)
```

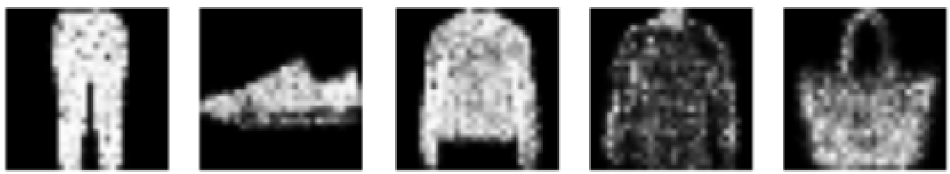
Generating Images with GANs:

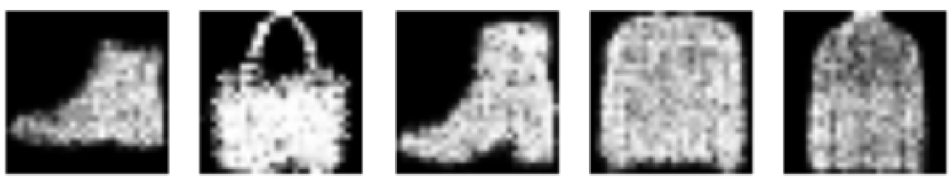
In order to utilize a GAN to create images, it is necessary to provide the discriminator with a substantial dataset to effectively function alongside the Generator. To accomplish this, we selected the Mnist Fashion dataset, which is a large standardized dataset that was not used in the original paper. The Mnist Fashion dataset comprises various fashion items, ranging from shoes to jackets, and a few random samples from this dataset are presented below.

Results:

Training the GAN on the Fashion MNIST dataset, consisting of over 60,000 images for 100 epochs, with a batch size of 128. The Adam optimizer was used with a learning rate of 0.0002 and a β of 0.5.

The generated images with these parameters showed a high degree of resemblance to the original images, with clear patterns and details that were consistent with the fashion items represented in the dataset. Some generated images can be seen here.





These results demonstrate the effectiveness of the GAN algorithm in generating fashion images that are visually appealing and consistent with the original dataset.

Summary

Overall, the results of this implementation of GANs has further indicated that the GAN algorithm is successful in generating high-quality fashion images that closely resemble the original Fashion MNIST dataset and therefore would be an excellent method for generating other forms of image shaped data from faces to music.

References

https://medium.com/neuronio/audio-generation-with-gans-428bc2de5a89

https://www.kaggle.com/datasets/zalando-research/fashionmnist

https://blog.paperspace.com/getting-started-with-dcgans/

https://www.kaggle.com/code/psycheshaman/pytorch-human-speech-classification/notebook