Generative Adversarial Network

Bruno Bonaiuto Bolivar
Deep Learning 2022

Università degli Studi di Trieste

Introduction to GANs

- A new framework for estimating generative models via an Adversarial process, proposed by Ian Goodfellow, A model G that captures the distribution and A model D that estimates the probability that a sample came from the training data
- The objective is to generate realistic data
- Generative models refers to any model that takes a training set and learns to represent an estimate of that distribution.
- GANs are focus primarily on the sample generation.

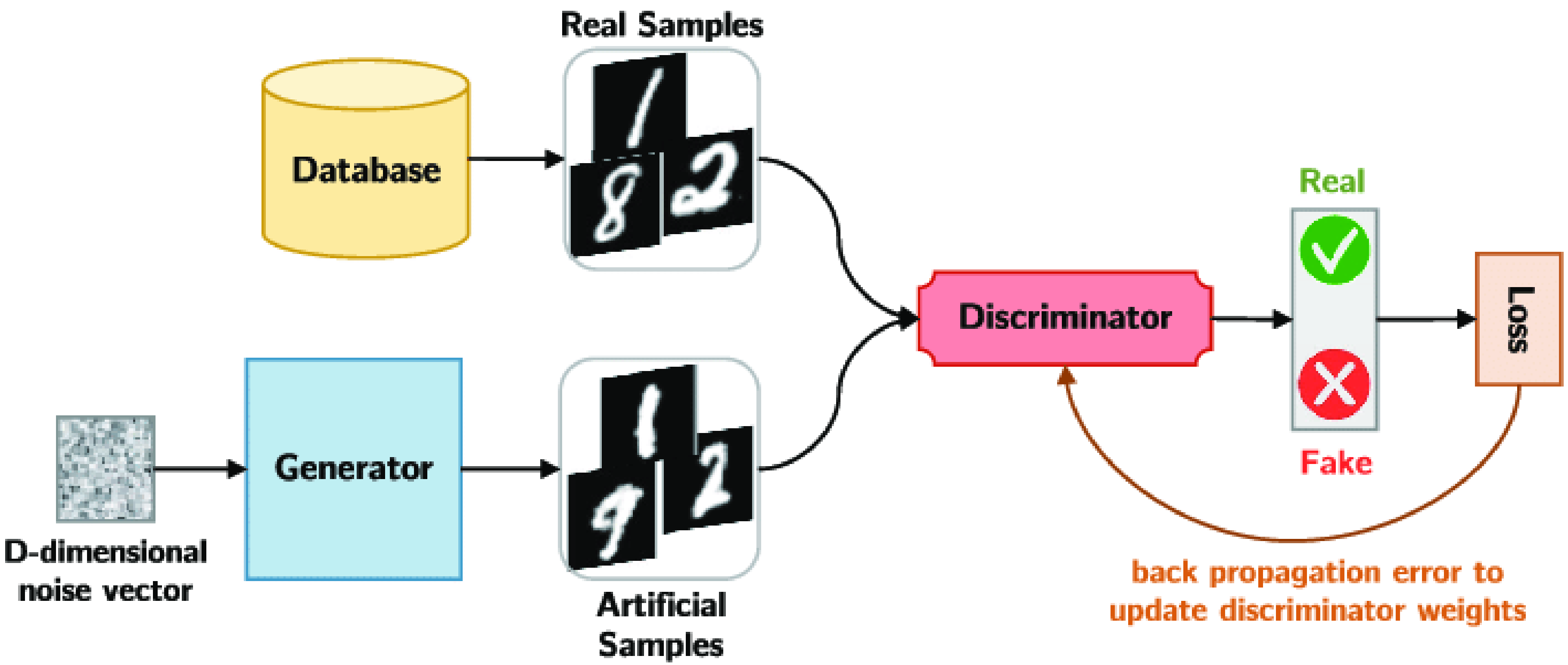
Introduction to GANs

- The generative model is pitted against an adversary and can be thought by trying to produce fake currency and use it without detection
- The discriminative model that learns to determine whether a sample is from the model distribution or the data distribution
- A competition game for both teams

Motivations

- The Training and Sampling for high-dimensional probability distributions
- The use in reinforcement Learning
- The use in Semi-supervised learning
- GANs work with multi-model outputs
- Single image super-resolution

The Architecture



The Framework

- The basic idea of GANs is to set up a game between the generator and the discriminator
- The solution is a local minimum
- Each player is a differentiable function
- The discriminator is trained using supervised learning
- The generator is trained using Gradient Descent algorithm

The Training

- The training process Consists of simultaneous SGD.
- It is possible to use the gradient-based optimization algorithm of your choice
- The protocol that works the best in practice is simultaneous gradient, with one step for each player
- Phase one:
  - Training the discriminator, at each step providing a batch of real images and fake images generated by the generator
  - Optimization BCE

The Training

- The phase two:
  - Training the generator, at each step providing a random noise compose batch to generate images
  - Define the output with a set of labels as real
  - Optimize BCE

Cost functions

- The generator cost

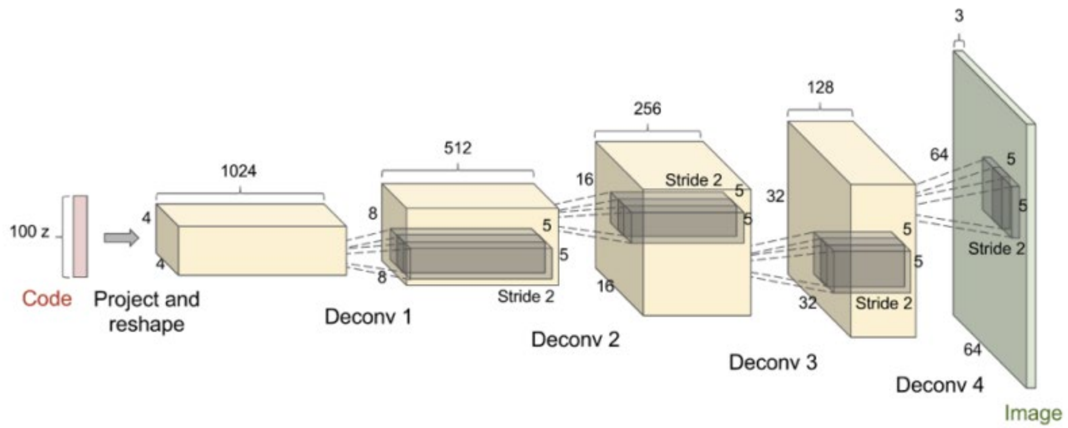
$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D(G(\boldsymbol{z}))$$

- The discriminator cost is:

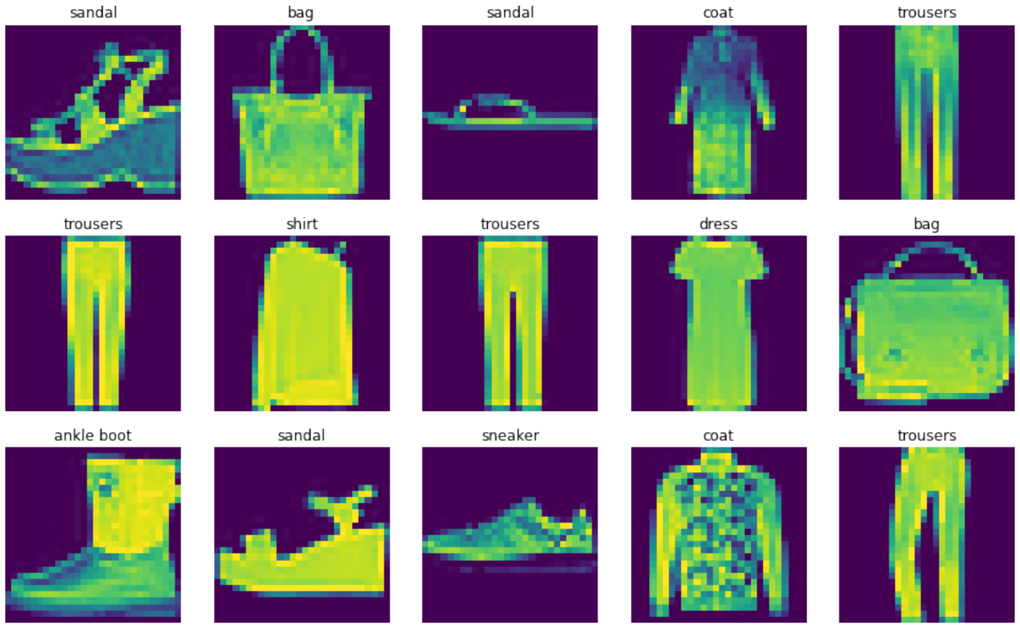
$$J^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

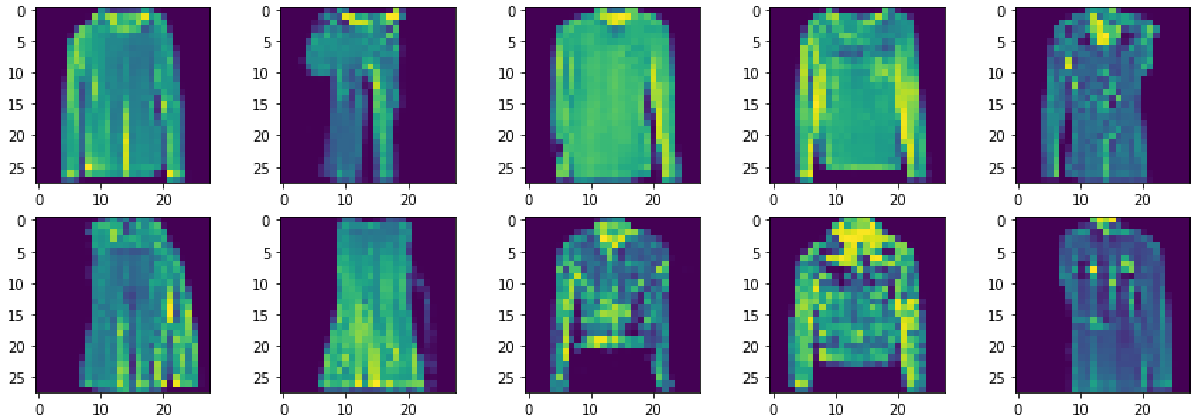
DCGAN

- Deep, convolution GAN
- Using batch normalization
- Neither Pooling nor unpooling layer
- Adam optimizer
- Labels
- The virtual batch normalization
- Can be balance G and D

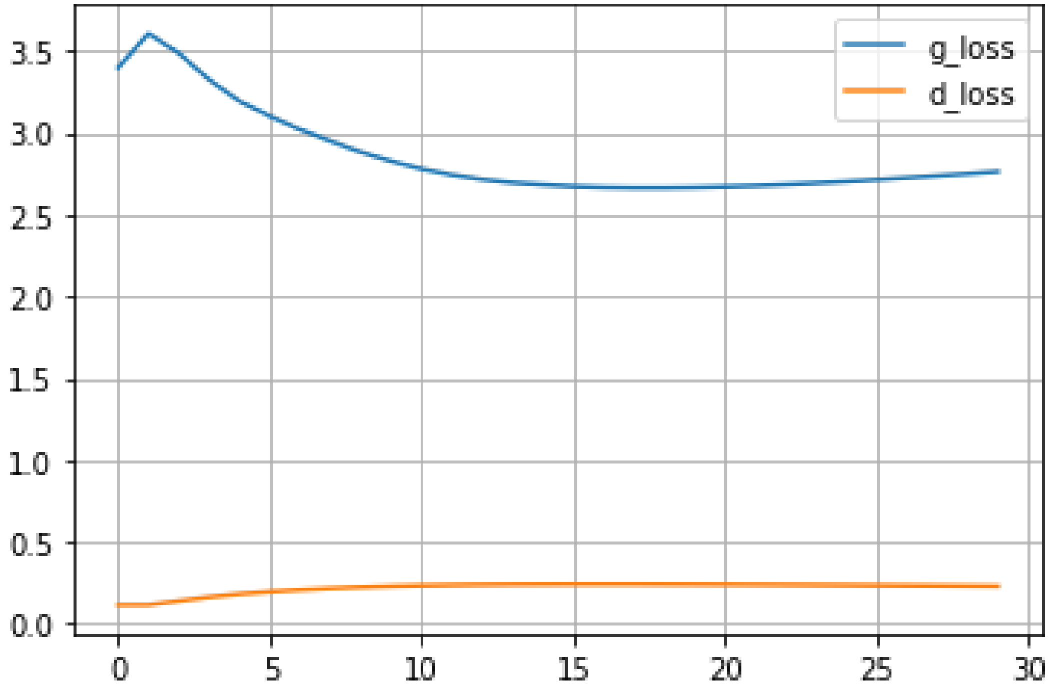


The Model


sandal
bag
sandal
coat
trousers
trousers
shirt
trousers
dress
bag
ankle boot
sandal
sneaker
coat
trousers



The Model



References

Goodfellow, Ian. ‘NIPS 2016 Tutorial: Generative Adversarial Networks’. 2017. Web.

Loyola-González, Octavio. ‘Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View’. *IEEE Access* 7 (2019): 154096–154113. Web.