DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Department of Information Engineering

Digital Forensics Course

Project 1: Generative Adversarial Networks

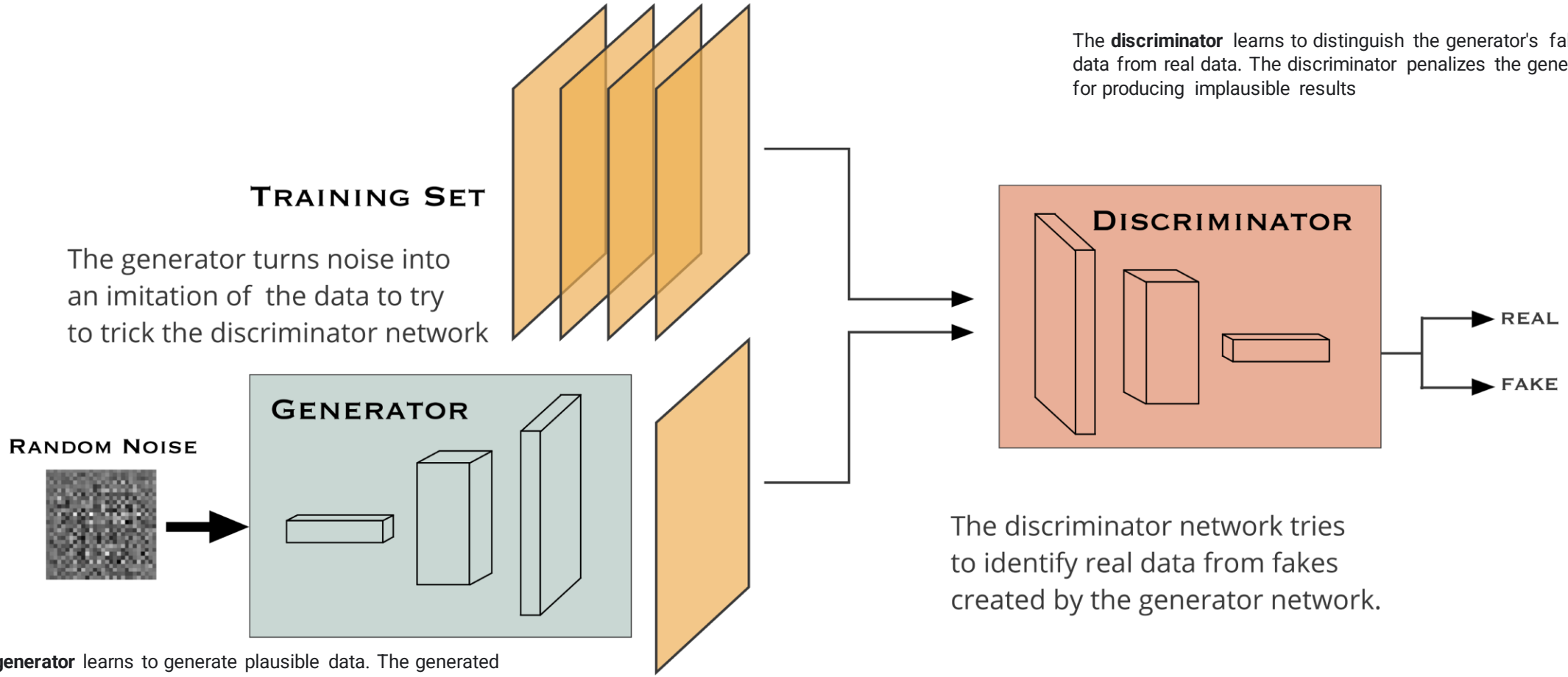
Supervisor: prof. Simone Milani

Team Members

Esraa Alnajjar 2009462

Marian Ashaq 2041388

A generative adversarial network (GAN) is **a machine learning (ML) model in which two neural networks compete with each other to become more accurate in their predictions**

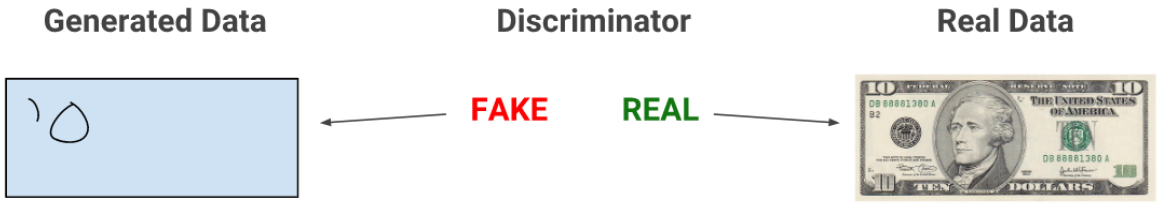


The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results

The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.

Procedure :

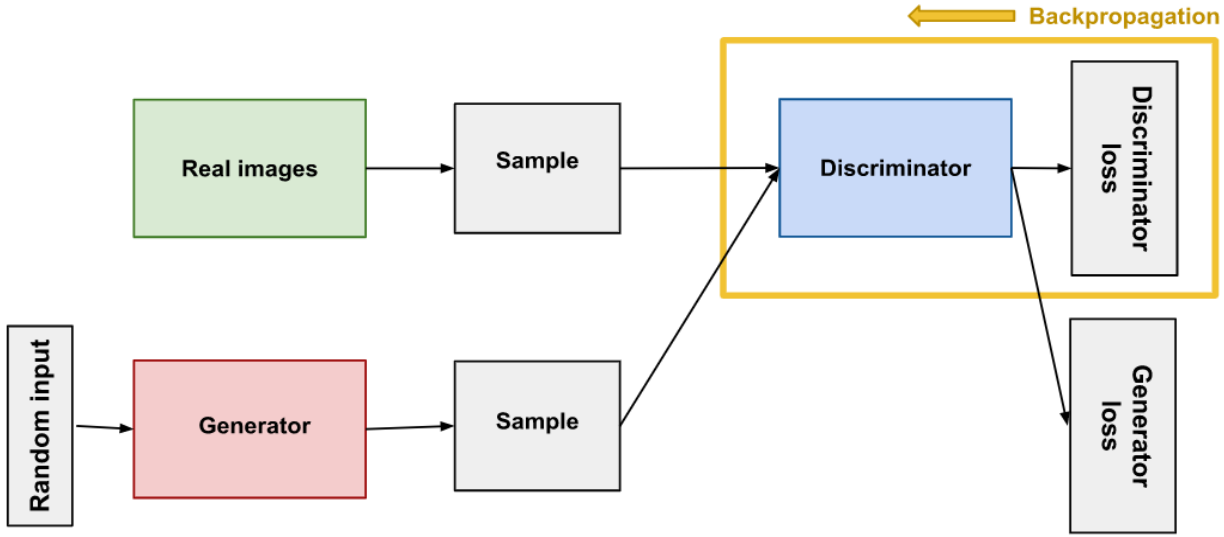
When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake:



As training progresses, the generator gets closer to producing output that can fool the discriminator:

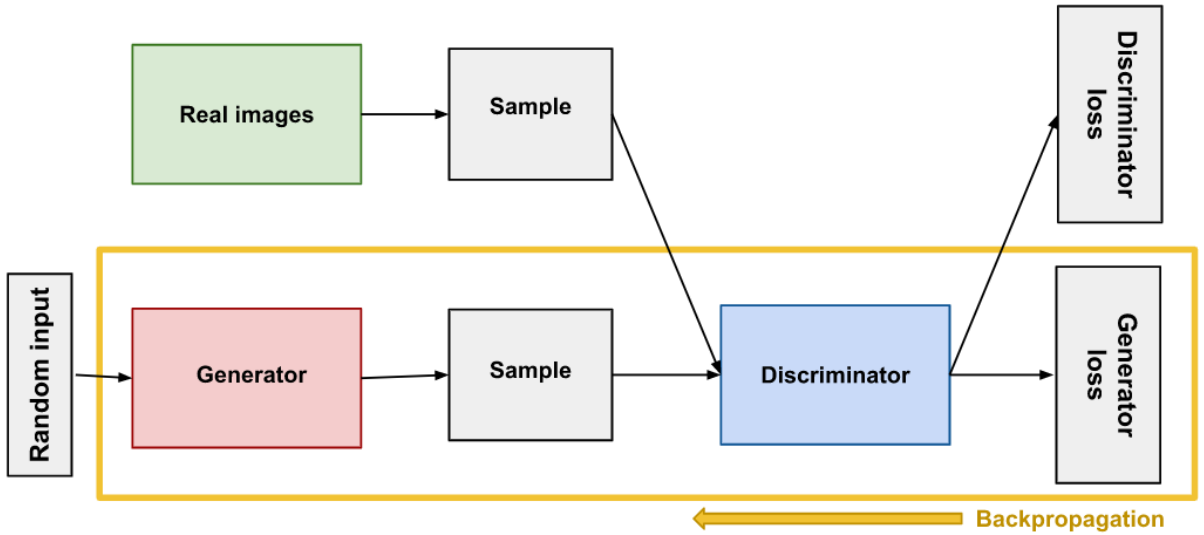
Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real, and its accuracy decreases:

The Discriminator :



Click to add text

The Generator :



Differences between GAN and DCGAN

Simple GAN

The generator in a simple GAN is a fully connected network

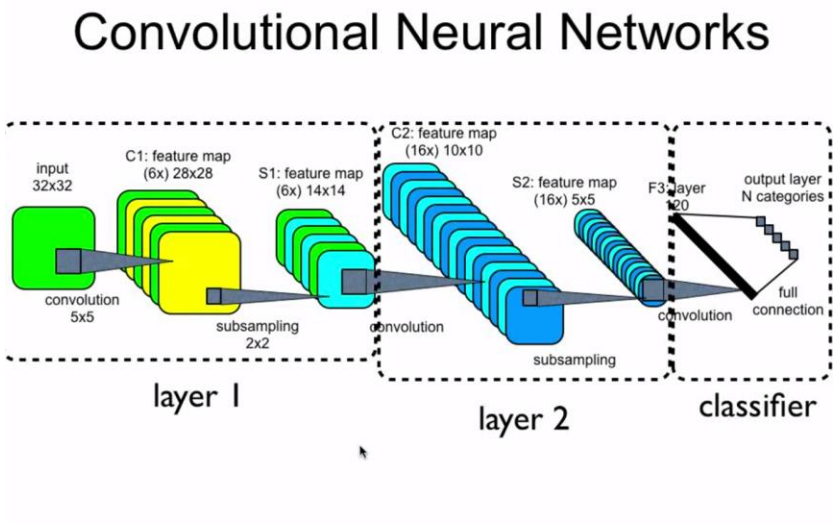
The DCGAN (Deep Convolutional Generative adversarial Network)

It is composed mainly of convolutional layers and strides for up sampling and down sampling of 2D image without Max pooling or fully connected layers

up sampling : increases the sampling rate ,double image dimensions, increases the block resolution

Down sampling : decreases the sampling rate and decreases dimensionality features which may lead to the loss of some information

DCGAN procedure in points



1-replace all max pooling with convolutional strides

2- use transposed convolution for up sampling

3-eliminate the fully connected layers

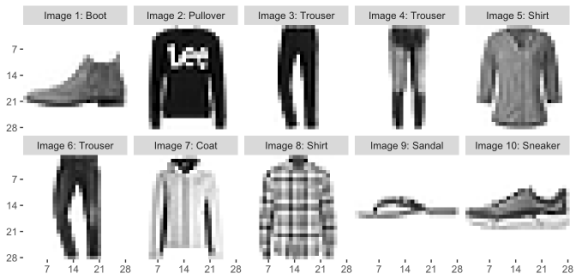
4-Then use Batch normalization except the output layer for the generator and the input layer of the discriminator

5-Use ReLU in the generator for activation except for the output which uses tanh

6-use LeakyReLU in the discriminator as activation function

We used here the Fashion MNIST dataset

Training images= **60000**



Test images=**10000**

Why we use keras

open source runs on top of **tensorflow** user friendly

why we use

TensorFlow

open-source deep learning framework

provides **multiple abstraction levels** for building and training models.

Now we can go and explain the model from the code

GANs Quantitative Evalutation Metrics (1 / 2)

The Inception Score

One of the **quantitative metrics** for evaluating the quality of generated images

The higher the inception score the better the quality of the generated images

Inception score problems

favors memory GAN

fails to detect some bad trapping model,

favors detectable objects rather than realistic,

favors generator, asymmetric,

affected by image resolution.

GANs Quantitative Evaluation Metrics (2 / 2)

The Frechet Inception Distance(FID)

One of the **quantitative metrics** for evaluating the quality of generated images

improvement of inception score model

The distances between two distributions(real and fake images) summarized as multivariate Gaussian then calculated by FID

The lower FID score indicates more realistic images

FID Limits

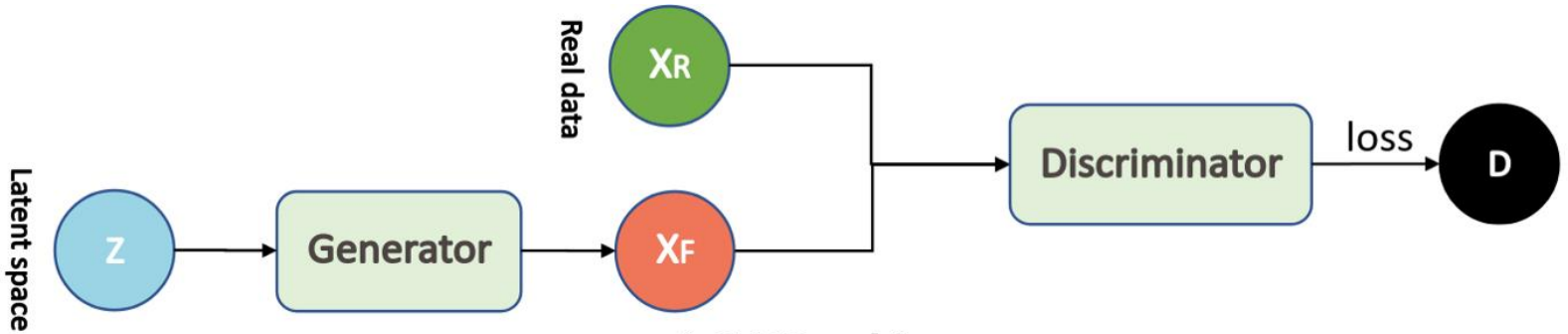
Consistent with human judgement,

more robust to noise than IS,

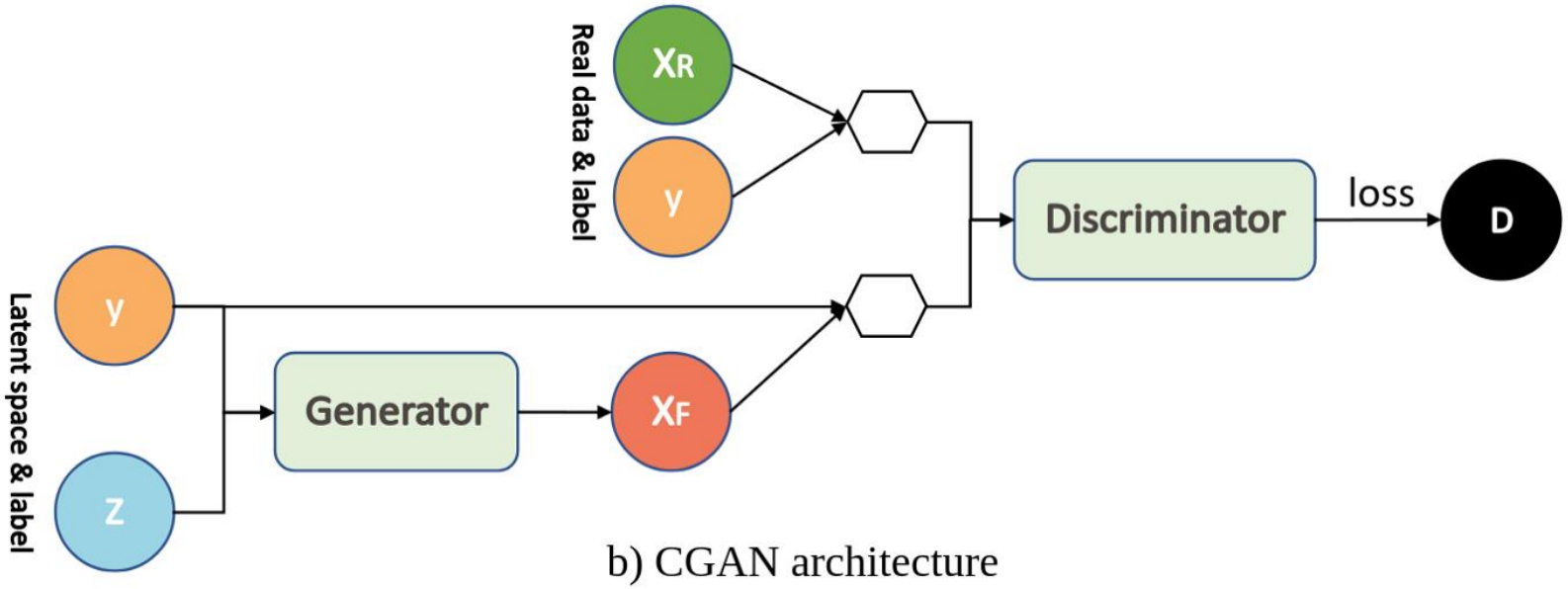
FID detect mode dropping,

sensitive to some artifact

Conditional GAN :



a) GAN architecture



b) CGAN architecture