INTRODUCCIÓN A LAS REDES GENERATIVAS ADVERSARIAS (GANS)

Rosa Karina Torres Calderon

17 de agosto del 2023

Aplicaciones

FaceApp

Aplicaciones

DEEP FAKE

Videos de DeepFake

Tom Cruise is Iron Man [DeepFake]
Robert Pattison as Batman [DeepFake]
John Krasinski as Star-Lord [DeepFake]

Generative Adversarial Networks

Las redes generativas adversarias (GANs), fueron introducidas en 2014 por Ian Goodfellow, et al.

source : imgflip.com

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio‡
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

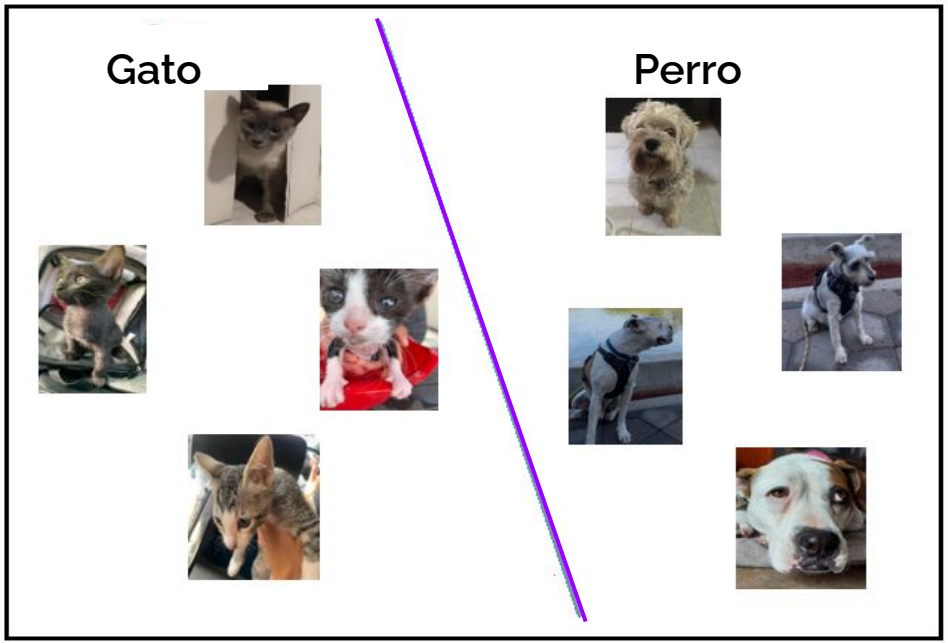
1 Introduction

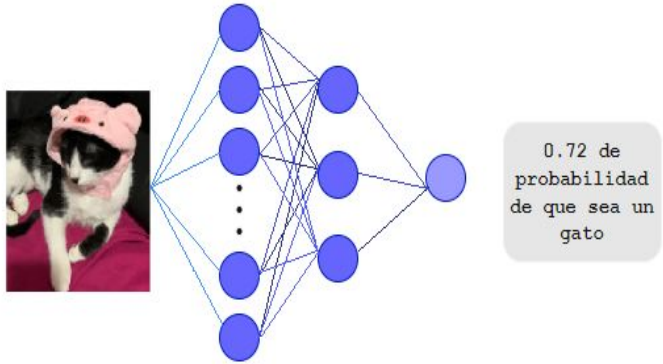
The promise of deep learning is to discover rich, hierarchical models [2] that represent probability distributions over the kinds of data encountered in artificial intelligence applications, such as natural images, audio waveforms containing speech, and symbols in natural language corpora. So far, the most striking successes in deep learning have involved discriminative models, usually those that map a high-dimensional, rich sensory input to a class label [14, 22]. These striking successes have primarily been based on the backpropagation and dropout algorithms, using piecewise linear units [19, 9, 10] which have a particularly well-behaved gradient . Deep *generative* models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. We propose a new generative model estimation procedure that sidesteps these difficulties. [1]

In the proposed *adversarial nets* framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

*Jean Pouget-Abadie is visiting Université de Montréal from Ecole Polytechnique.
†Sherjil Ozair is visiting Université de Montréal from Indian Institute of Technology Delhi
‡Yoshua Bengio is a CIFAR Senior Fellow.
[1] All code and hyperparameters available at `http://www.github.com/goodfeli/adversarial`

¿Qué es un modelo discriminativo?





Modelo discriminativo estima $p(y|x)$

¿Qué es un modelo generativo?

Modelo generativo estima $p(x)$

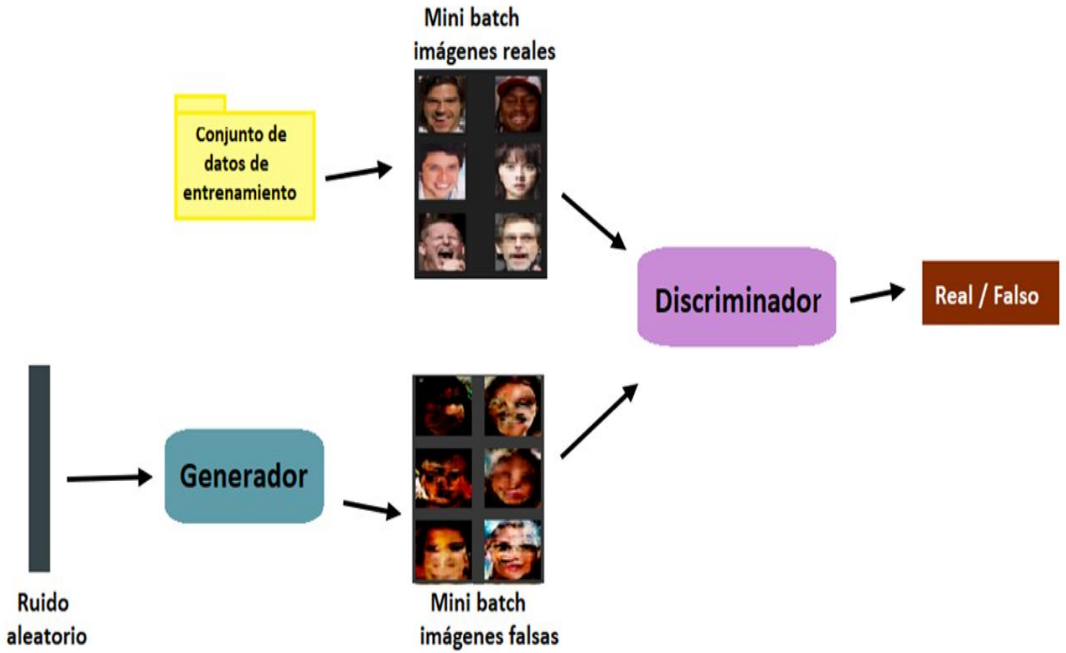
Si el conjunto de datos está etiquetado podemos construir un modelo generativo que estime la distribución $p(x|y)$

- Tenemos una conjunto de observaciones X
- Suponemos que las observaciones se han generado de acuerdo con alguna distribución desconocida p_{data}
- El modelo generativo p_{model} trata de imitar p_{data}. Si logramos el objetivo, podemos tomar una muestra de p_{model} para generar nuevas observaciones que parecieran ser obtenidas de p_{data}.
- p_{model} se considera exitoso si:
    1. Puede generar observaciones que parezca que provienen de p_{data}.
    2. Puede generar observaciones que son diferentes de X.

¿Qué es una GAN?

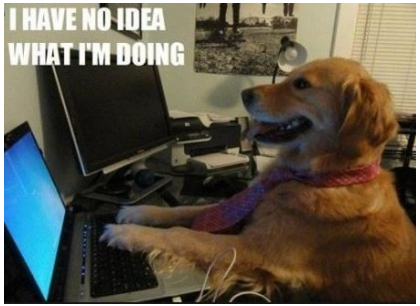
Se compone de:

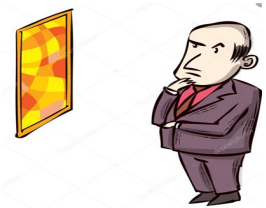
- Generador
- Discriminador



Intuición

Generador





Discriminador

Arquitectura de las GANs

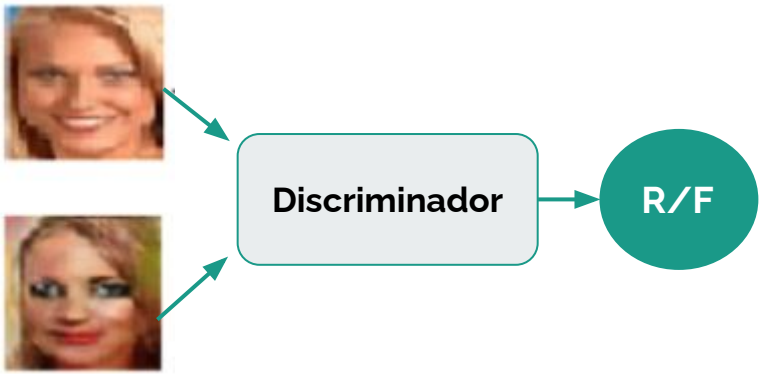
Discriminador

El discriminador es un clasificador binario, cuyo objetivo es clasificar entre datos reales y datos creados por el generador

Entrada del Discriminador
Los datos de entrenamiento del discriminador provienen de dos fuentes:

- ***Datos reales***: Datos reales del conjunto de entrenamiento (+)
- ***Datos falsos***: Instancias creadas por el generador(-)

Salida del Discriminador
Valor de probabilidad de que una imagen sea real



Arquitectura de las GANs

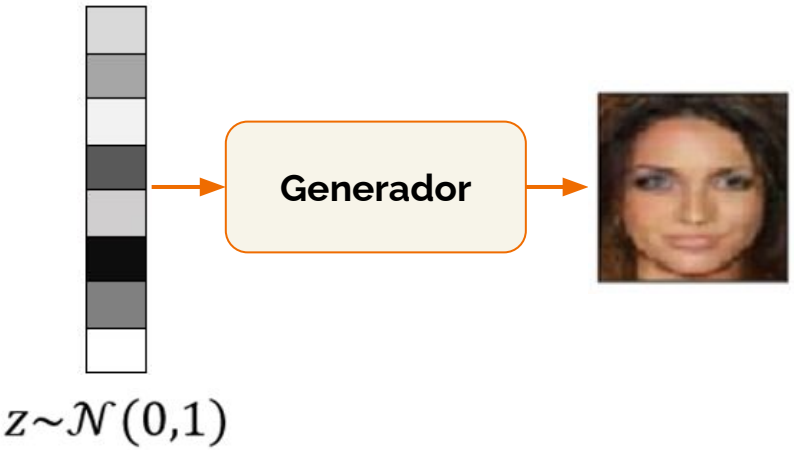
Generador

El propósito del generador es generar datos sintéticos que sean indistinguibles de los datos reales para el discriminador.

La parte generadora de la GAN aprende a crear los datos incorporando la retroalimentación del discriminador.

Entrada del Generador Vector de ruido proveniente de una distribución normal

Salida del Generador Imagen del mismo tamaño que las imágenes provenientes del conjunto de datos de entrenamiento



Matriz de confusión del Discriminador

	Salida del discriminador	
Entrada	**Real (1)**	**Falso (0)**
Imágenes reales (*X*)	Verdaderos positivos	Falsos negativos
Imágenes falsas (*X'*)	Falsos positivos	Verdaderos negativos

Discriminador
El Discriminador busca minimizar FP y FN

	Salida del discriminador	
Entrada	**Real (1)**	**Falso (0)**
Imágenes reales (*X*)	Verdaderos positivos	Falsos negativos
Imágenes falsas (*X'*)	Falsos positivos	Verdaderos negativos

Generador
El Generador busca maximizar FP

Función de costo

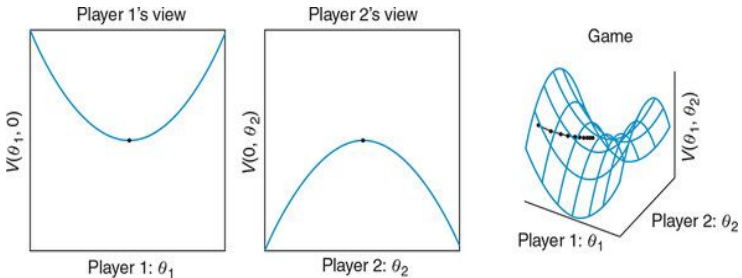
- Función de costo para el Generador

$$J^{(G)}(\theta^{(G)}, \theta^{(D)})$$

- Función de costo para el Discriminador

$$J^{(D)}(\theta^{(G)}, \theta^{(D)})$$

En una GAN cada red solo tiene permitido actualizar sus propios parámetros.



¿Cuándo sabemos que la GAN ha terminado el proceso de entrenamiento?

- Equilibrio de Nash (game zero-sum).

En GANS el equilibrio de Nash se alcanza en las siguientes condiciones:

- El Generador produce ejemplos falsos indistinguibles de los reales
- El Discriminador adivina (50/50) para determinar si un dato es real o falso

Función de costo del discriminador

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}}\log\big(D(x)\big) - \frac{1}{2}\mathbb{E}_{z}\log\Big(1 - D\big(G(z)\big)\Big)$$

Muestras reales

$$D(x) \to 1$$

Etiqueta 1

D(x) = 0.96 *error pequeño*

D(x) = 0.13 *error grande*

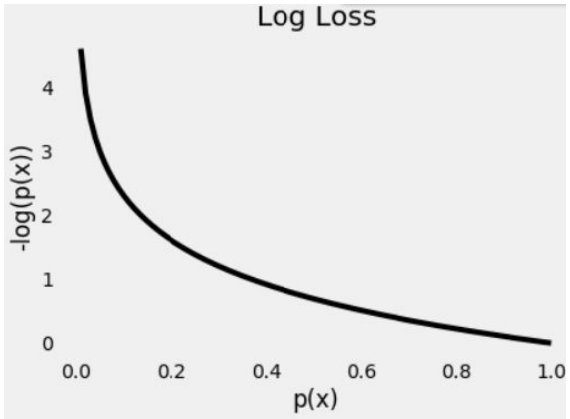
Muestras falsas

$$D(G(z)) \to 0$$

Etiqueta 0

D(G(z)) = 0.13 *error pequeño*

D(G(z)) = 0.96 *error grande*

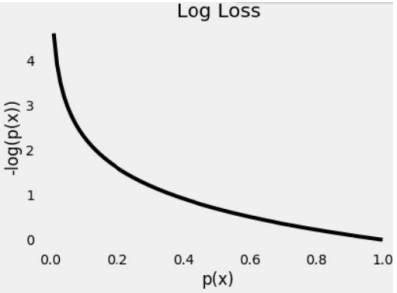


Función de costo del generador

- Función de costo del Generador

$$J^G = -J^D$$

- Configuración del juego: NON-Saturating GAN



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

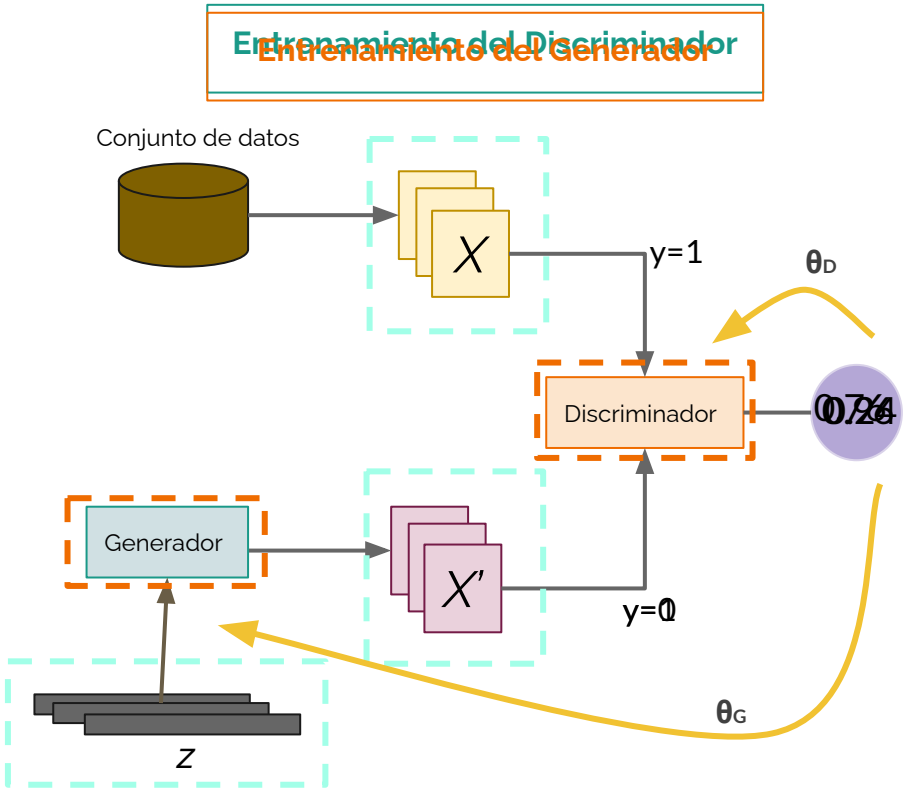
$$D(G(z)) \to 1$$

D(G(z)) = 0.96 *error pequeño*

Entrenamiento

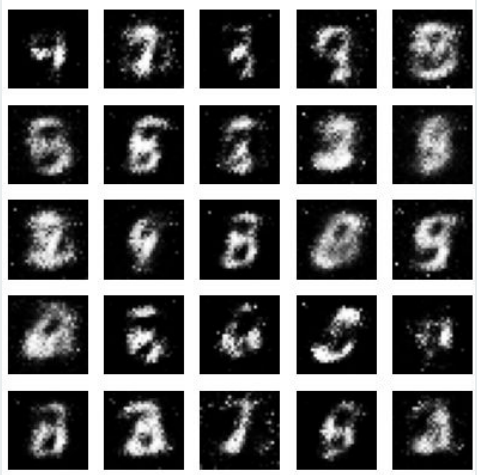
Para cada iteración se *hace*:

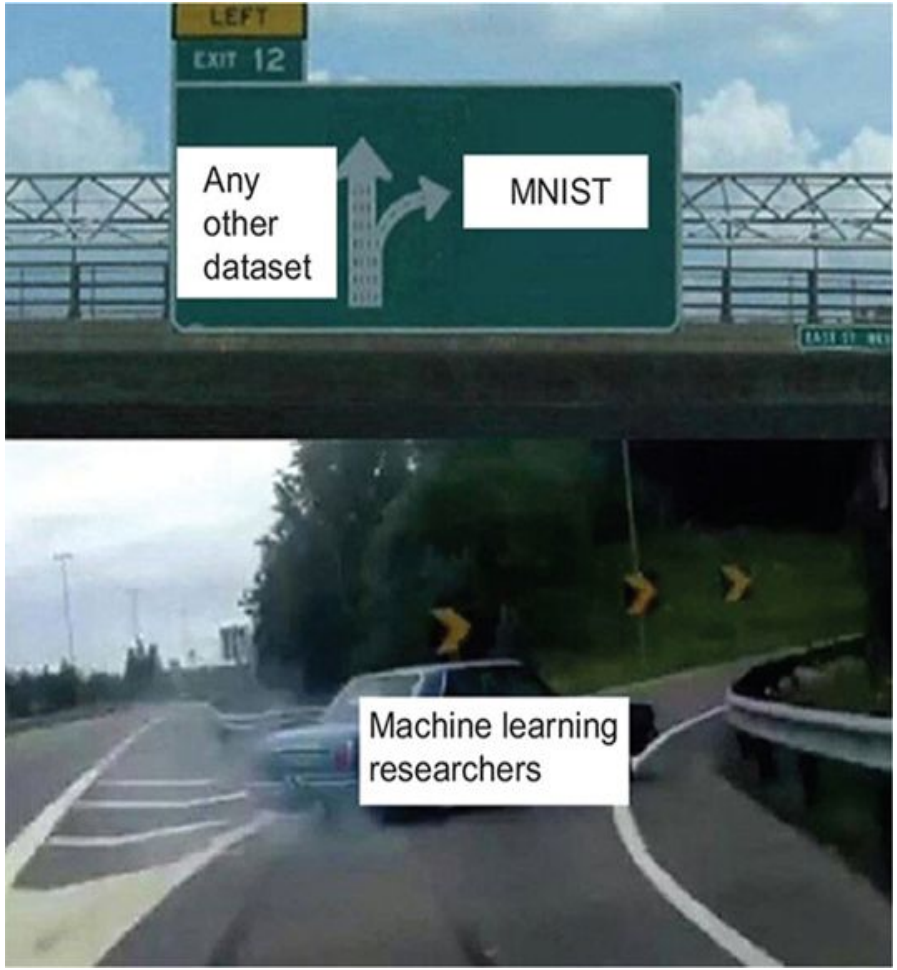
- Entrenamiento del Discriminador
  1. Mini lote de ejemplos reales *X*
  2. Mini lote de vectores de ruido *z*, que generan un mini lote de ejemplos falsos *G(z) = X'*
  3. Se asigna la etiqueta 1 a los ejemplos de entrada reales y 0 a los falsos
  4. Se calcula el error para *D(X)* y *D(X')*. La función de costo penaliza los errores al clasificar instancias reales como falsas y viceversa
  5. El error se propaga hacia atrás para actualizar los parámetros del discriminador
- Entrenamiento del Generador
  1. Mini lote de vectores de ruido *z*, que genera un mini lote de ejemplos falsos *G(z) = X'*
  2. .Se asigna la etiqueta de 1 a los ejemplos de falsos
  3. Se calcula el error *D(X')*
  4. El error se propaga hacia atrás para actualizar los parámetros del generador



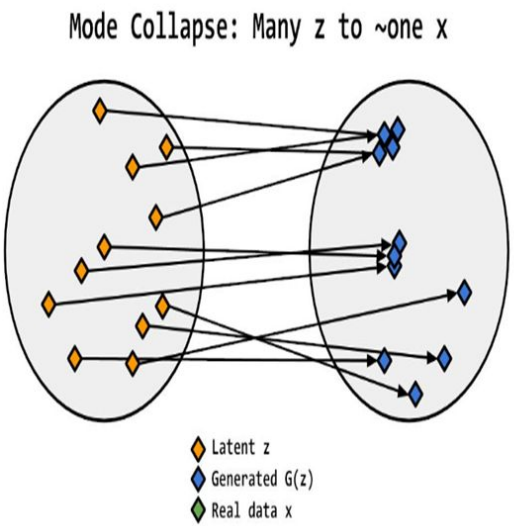
Práctica 1

Vainilla GAN MNIST





Retos del entrenamiento




Mode Collapse on CelebA (Source: Geometric GAN)

§**No convergencia.** El generador y el discriminador no logran alcanzar un equilibrio.

§**Desvanecimiento de los gradientes** Si el discriminador es demasiado bueno, el entrenamiento del generador puede fallar debido a que el discriminador no proporciona suficiente información (los gradientes se desvanecen) para que el generador aprenda

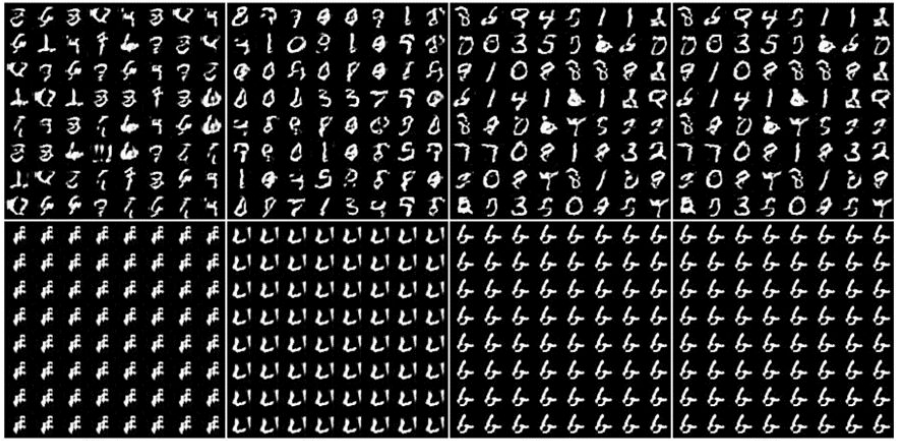
§**Colapso de moda** El colapso de moda es un problema que ocurre cuando el generador aprende a mapear diferentes valores de entradas z al mismo punto de salida.

Retos del entrenamiento

§**Sensibilidad a los Hiperparámetros.** Las GAN son muy sensibles a los cambios en los hiperparámetros, incluso en las arquitecturas más simples.

Encontrar el conjunto de parámetros que funcionen a menudo es un caso de prueba y error.

§**Sobregeneralizacion.** Se dice que una GAN sobregeneralizó cuando la red aprende elementos que no deberían de existir basados en los datos reales, por ejemplo: que el generador genera la imagen de una vaca con múltiples cuerpos pero un sola cabeza.



DCGANs

Las redes generativas adversarias convolucionales o DCGAN por sus siglas en inglés (Deep Convolutional Generative Adversarial Networks) son una extensión directa de las redes descritas anteriormente.

A diferencia de sus antecesoras, las DCGANs contienen en su arquitectura capas convolucionales en el discriminador y capas de convolucionales transpuestas en el generador

UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

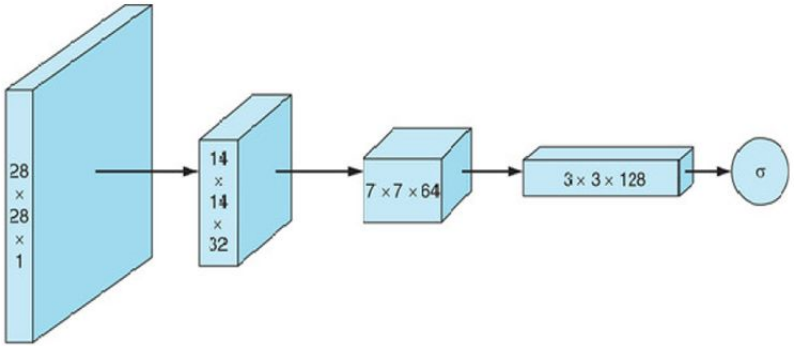
Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

ABSTRACT

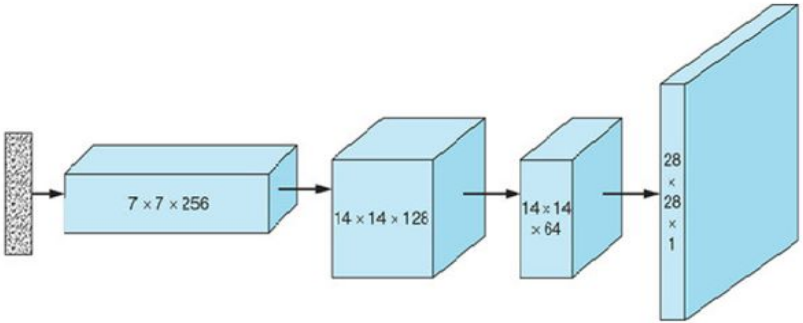
In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability as general image representations.

Arquitectura de la DCGAN

Discriminador



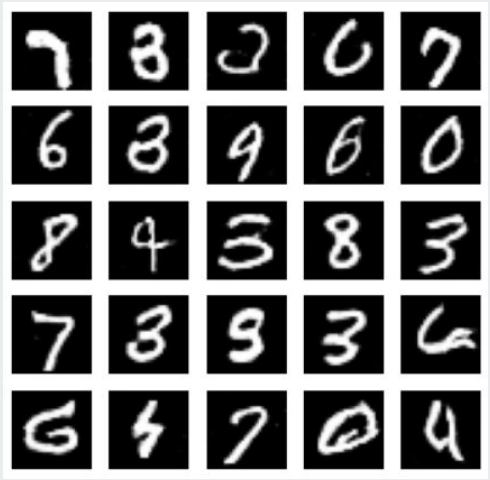
Generador

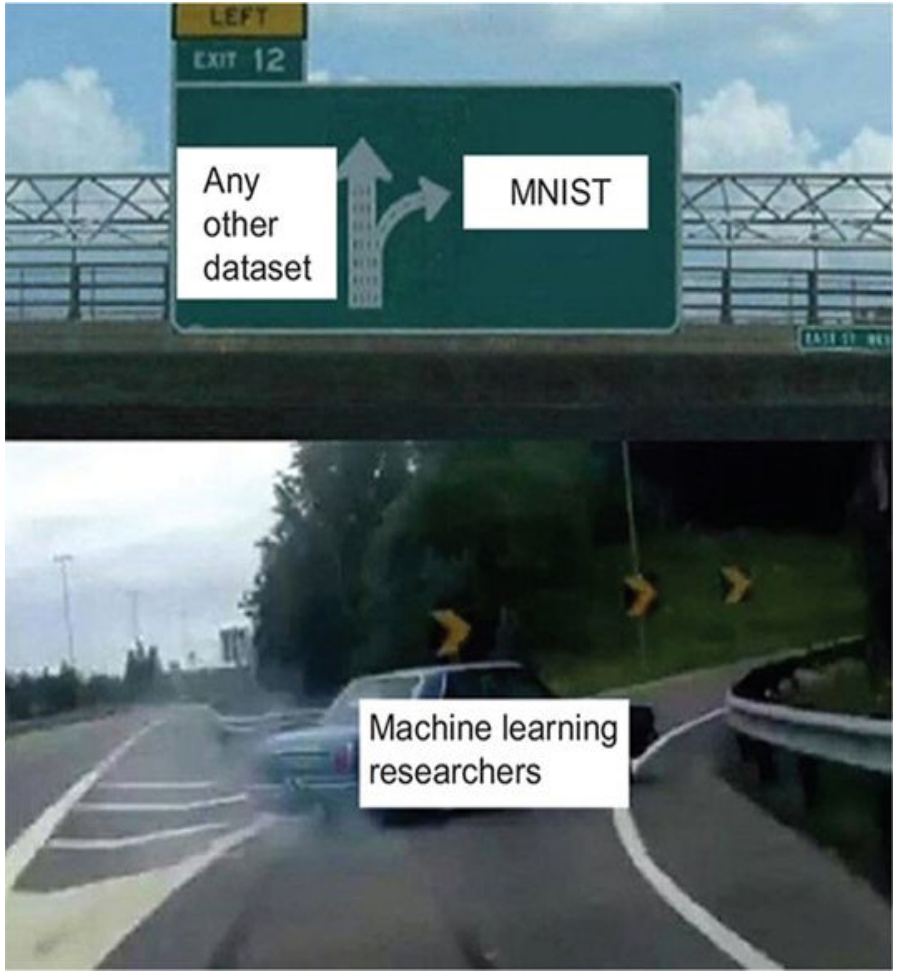


A Comprehensive Guide to Convolutional Neural Networks
What is Transposed Convolutional Layer?

Práctica 2

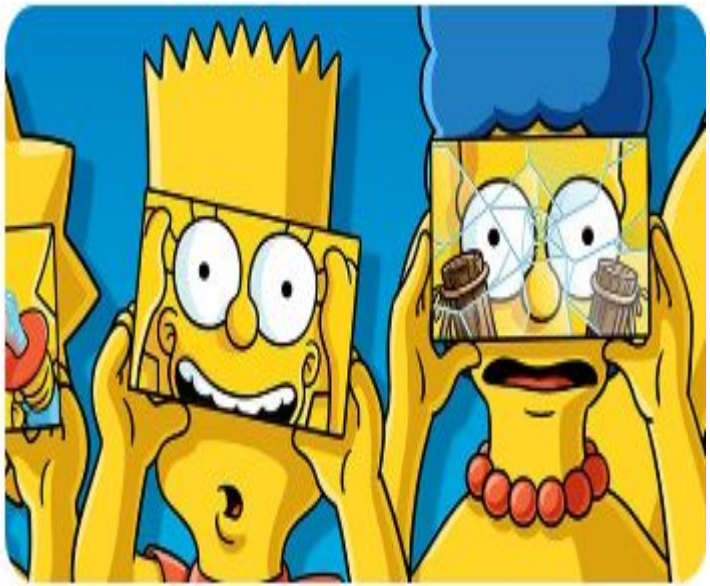
DCGAN MNIST





Ejercicio 1

DCGAN SIMPSON



¿Cómo resolver los problemas del entrenamiento de una GAN?

Utilizar otras funciones de costo

- No-Saturation
- Wasserstein

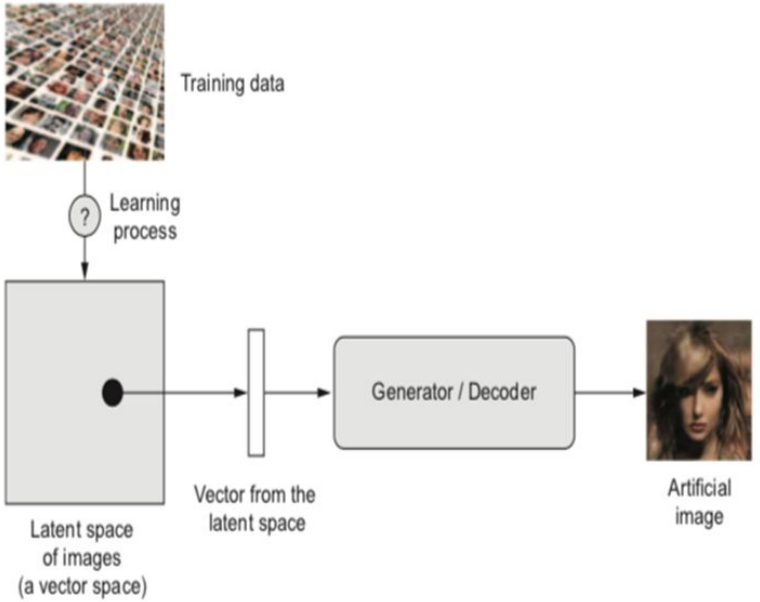
Agregar profundidad a la red

Trucos al entrenar GANs

- Darle más entrenamiento al Discriminador
- Cambiar etiquetas de suaves a ruidosas

Espacio Latente

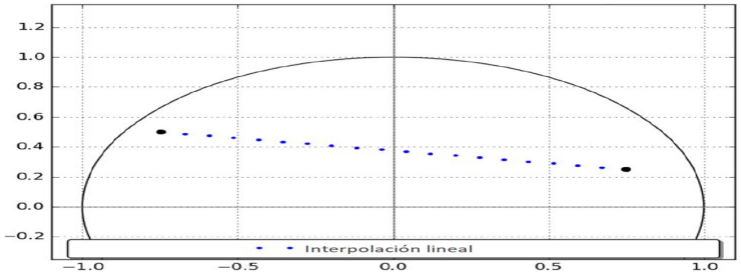
La idea clave de la generación de imágenes es desarrollar un espacio latente de representaciones de baja dimensión (las características más importantes que describen una observación)
El modelo capaz de realizar este mapeo, es el generador



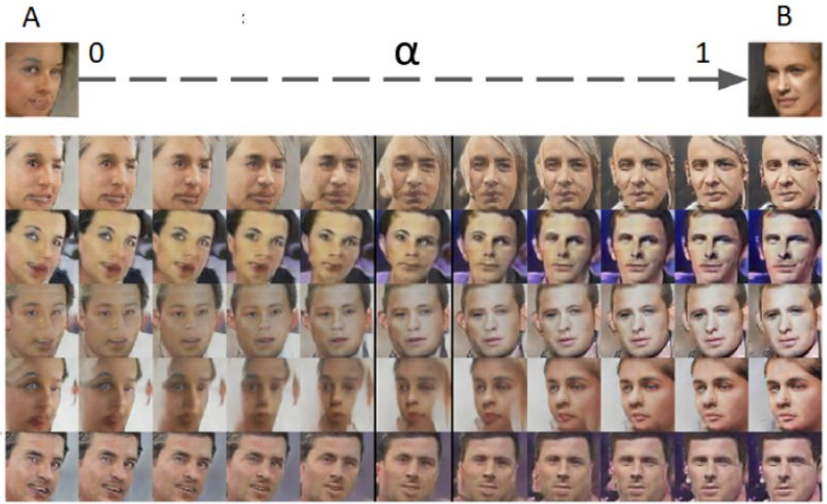
Interpolación

Se puede crear una ruta lineal entre dos puntos en el espacio latente.

Estos puntos se pueden utilizar para generar una serie de imágenes que muestran una transición entre las dos imágenes generadas.

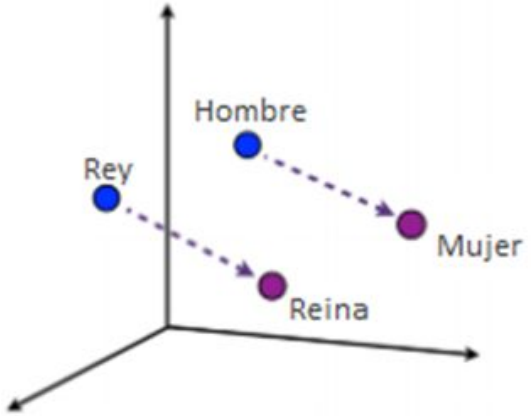


$$z_{new} = z_A(1-\alpha) + z_B\alpha$$

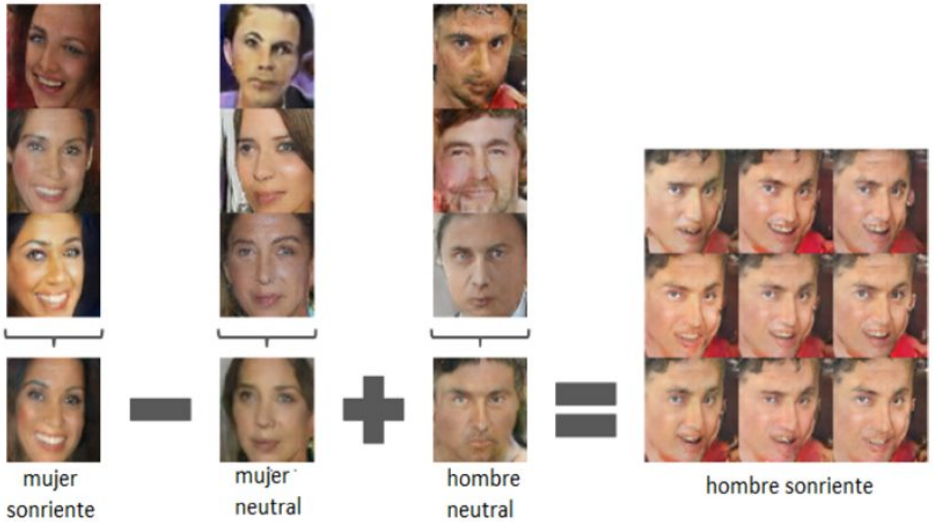


Aritmética en el espacio latente

El generador es capaz de realizar operaciones aritméticas en el espacio latente.

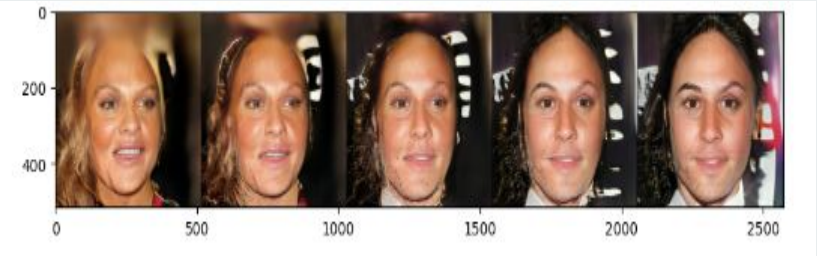


$$mujer_{sonriente} - mujer_{neutral} + hombre_{neutral} = \ ?$$



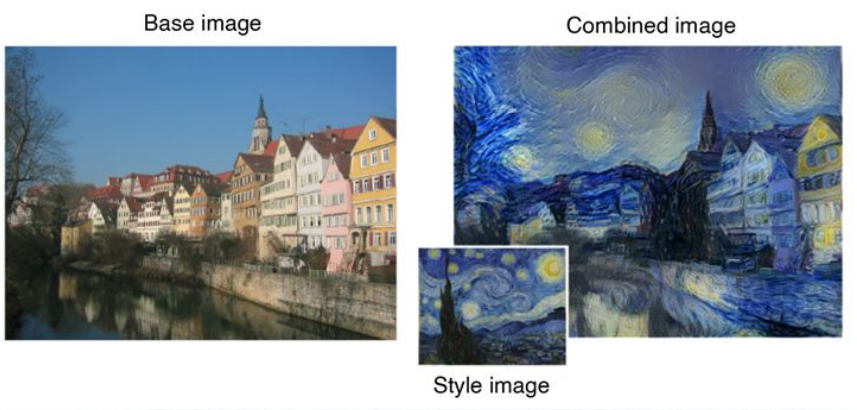
Práctica 4

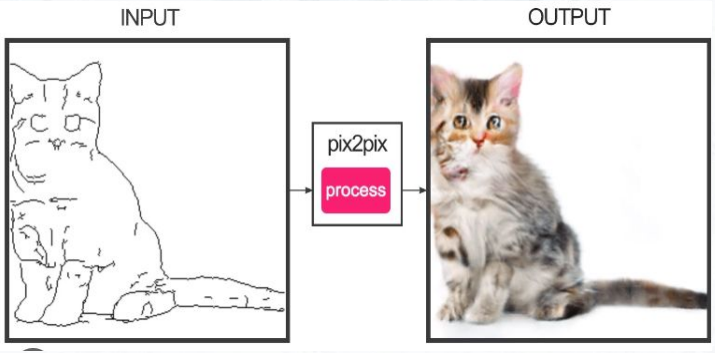
Operaciones en el espacio de latencia



Imágenes generadas por StyleGAN (*thispersondoesnotexist.com*)

Aplicaciones





¡Gracias por su atención!

Referencias

Fighting Deepfake by exposing the convolutional traces on images
Deep FaceLab: A simple, flexible and extensible face swapping framework
Generative Adversarial Nets
Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks
NIPS 2016: Tutorial Generative Adversarial Networks
How to explore the GAN latent space when generating faces