Carrera de Especialización en Inteligencia Artificial

# APRENDIZAJE PROFUNDO

## CLASE 8

## TRANSFER LEARNING

## GENERATIVE ADVERSARIAL NETWORKS

Docente: Dr. Ing. Marcos Uriel Maillot

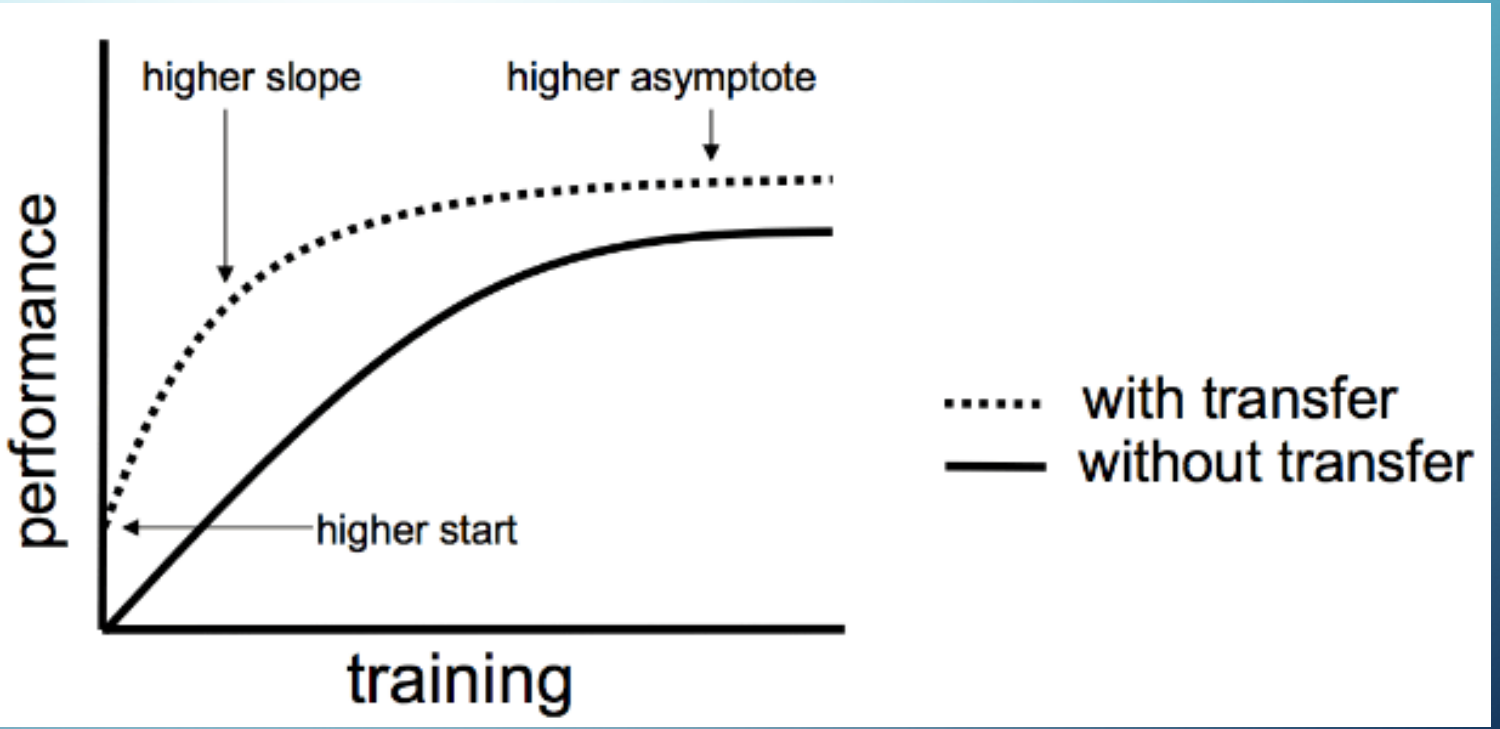
# Transfer Learning

- No se suele diseñar y entrenar un modelo desde **CERO**
- Se emplean **modelos existentes y probados** con sus parámetros ya entrenados.
- Normalmente, los modelos que se toma de “base” cumplen una **tarea genérica.**
- Al modelo “base” se le hacen los ajuste necesarios para  la nueva **tarea específica** que deben cumplir.

# Transfer Learning

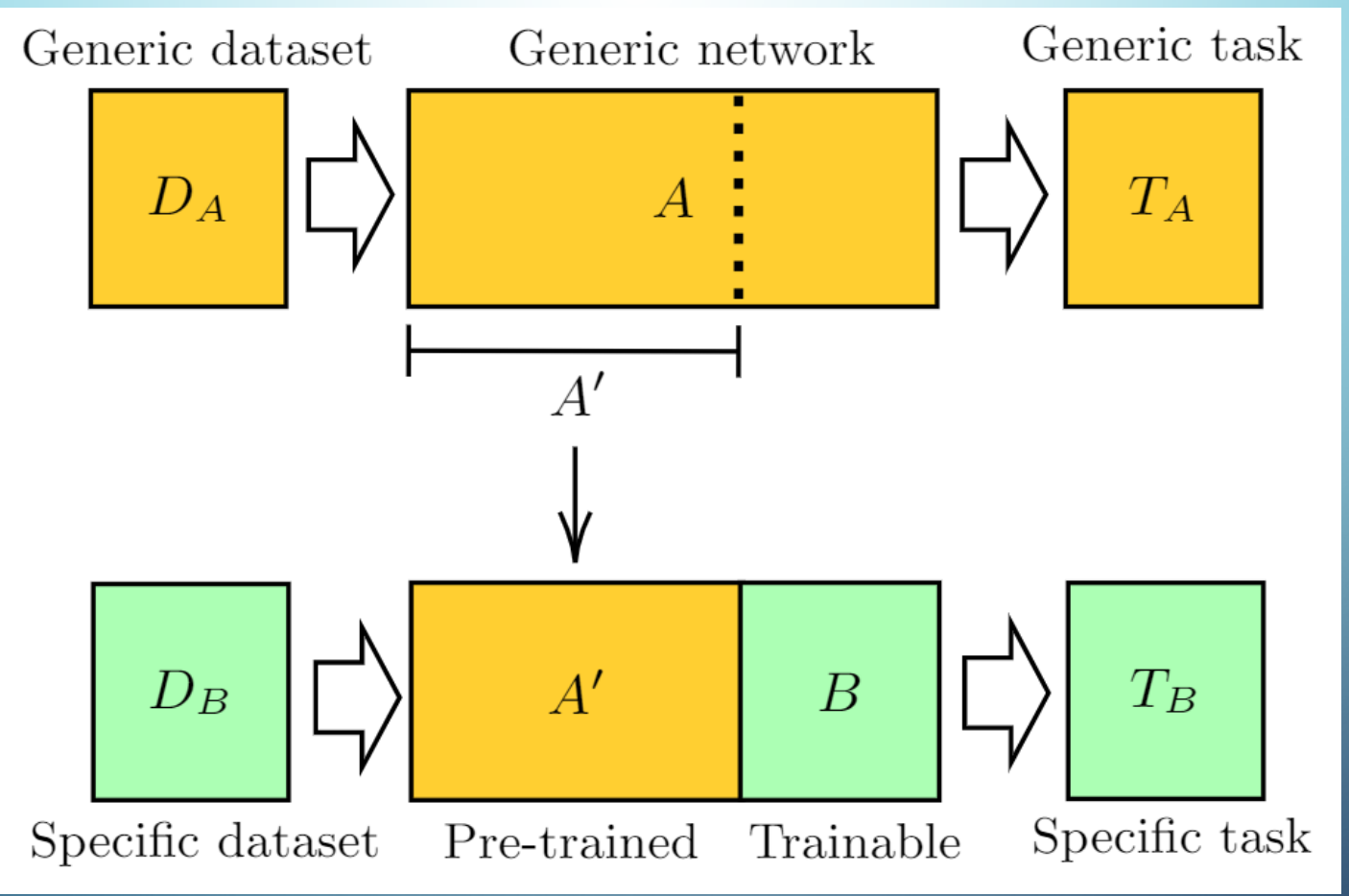
Ventajas:

- pocos datos
- pre-trained models
- pre-trained embeddings
- Simulations
- Cambio de dominio



# Transfer Learning

Adaptación de modelo base para cumplir la tarea específica




Se reentrena la nueva arquitectura con el dataset específico bajo la tarea específica a cumplir

# Transfer Learning - ¿Qué estrategia usar?

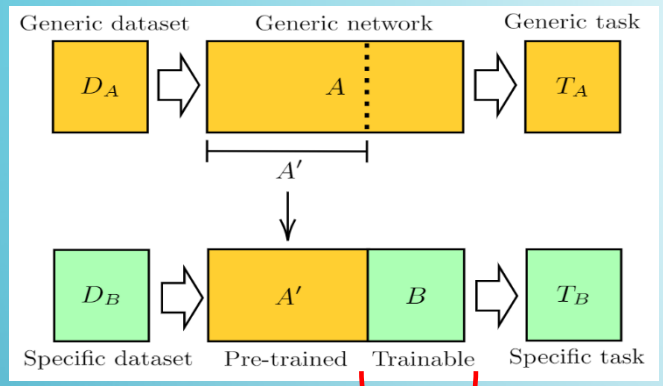
Dataset semejante al generic

Dataset distinto al generic

Dataset pequeño

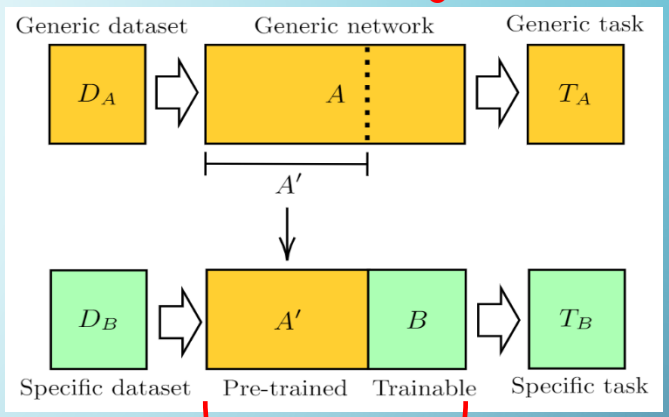
Dataset grande

feature extractor

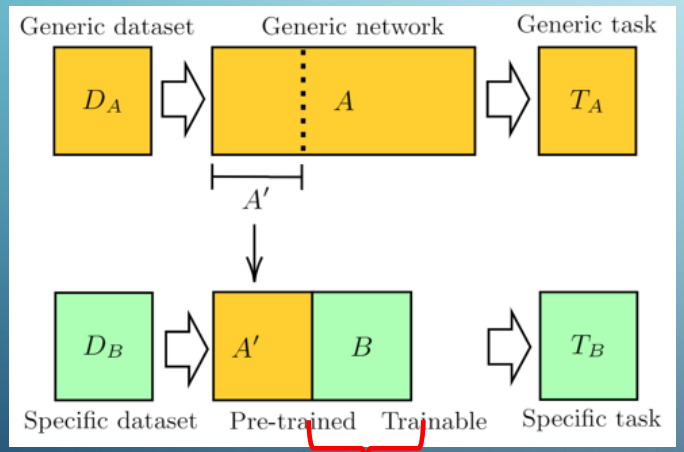



entrenar

fine tunning

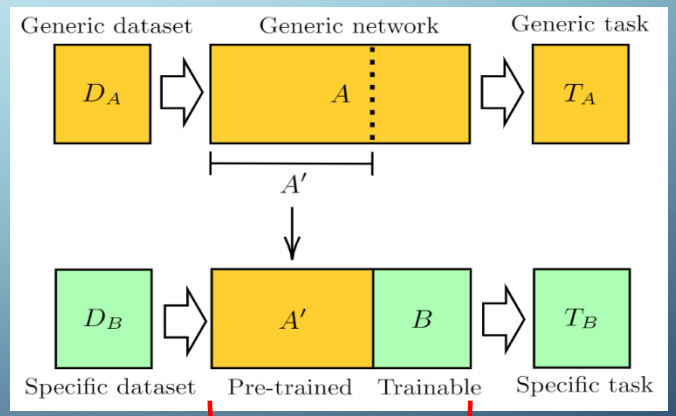



reentrenar

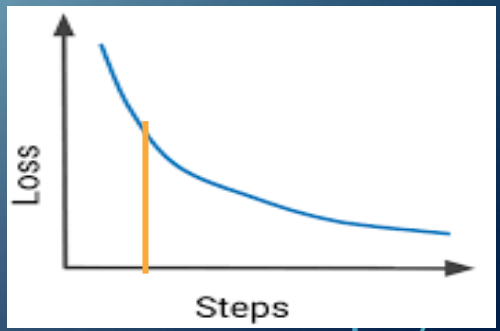



entrenar

feature extractor desde menor profundidad




reentrenar

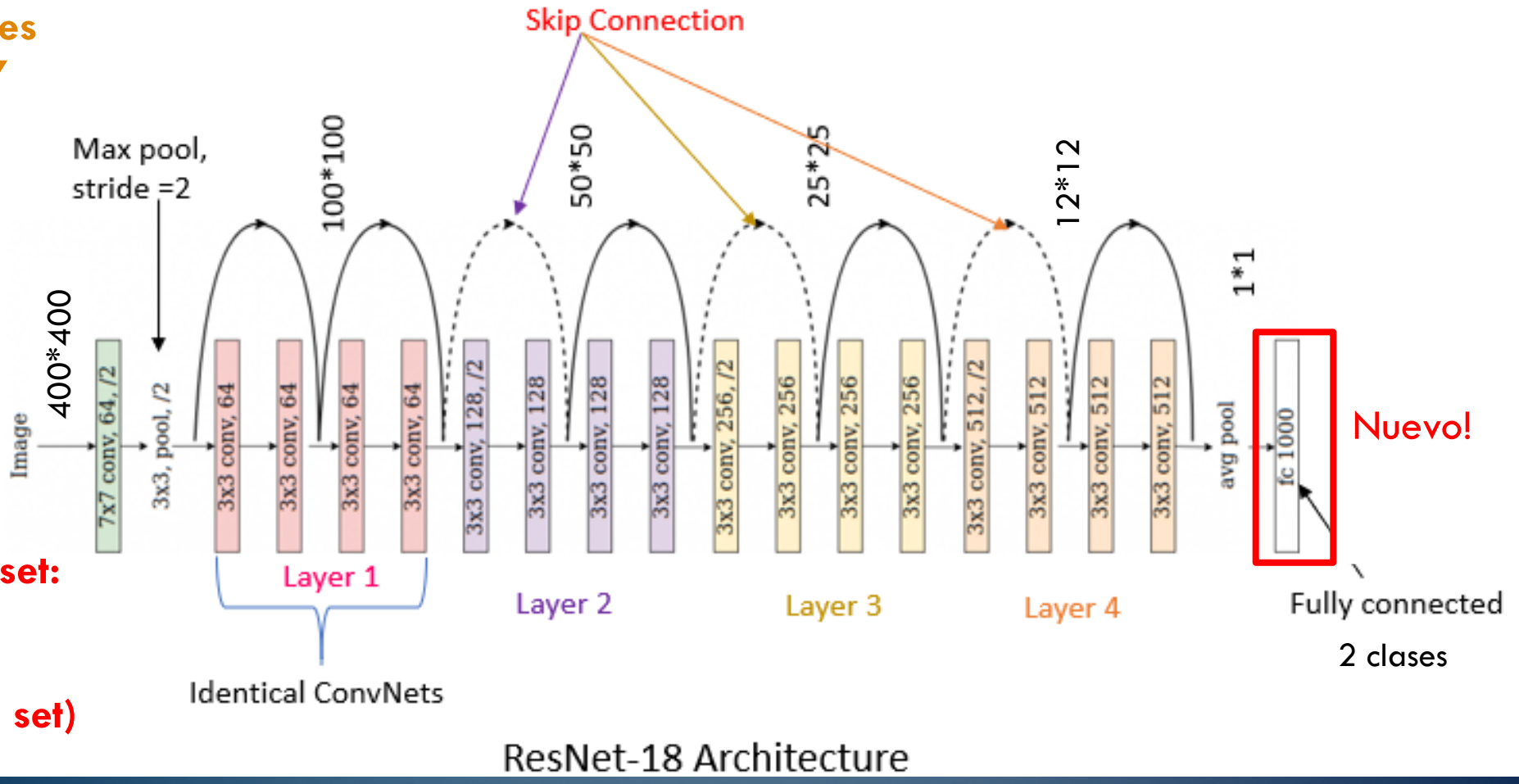



fine tunning (from pre-trained models)

# Transfer Learning

## Adaptación de modelo base para cumplir la tarea específica

**Generic Dataset:**
**ImageNet 1K**
- **1000 clases**
- **1.281.167 (train set)**
- **RGB**

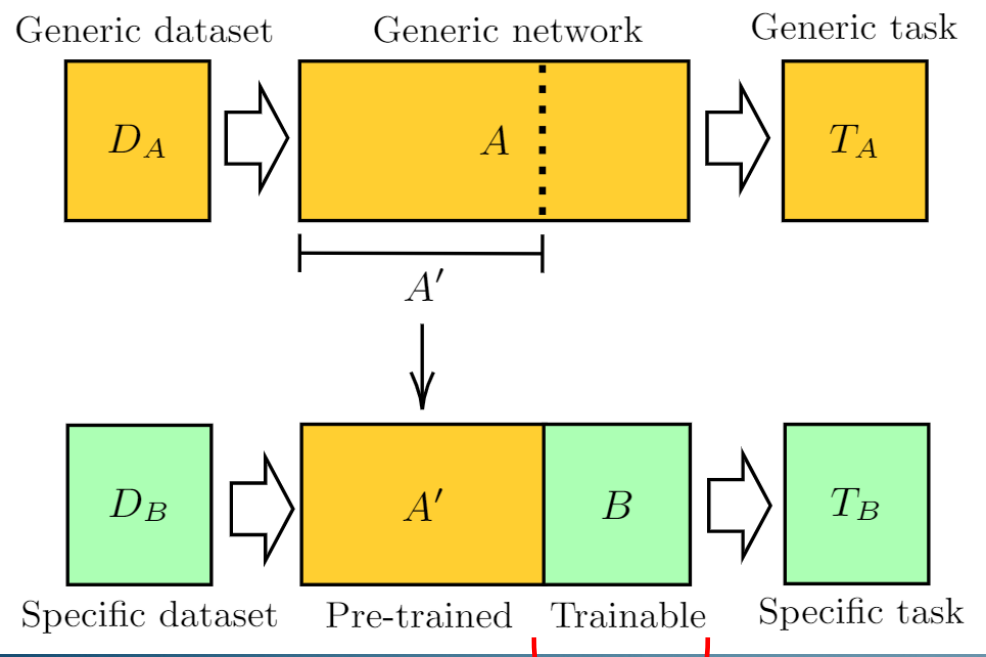



**Specific Dataset:**
**Bees & ants**
- **2 clases**
- **120 (train set)**
- **RGB**

# Transfer Learning

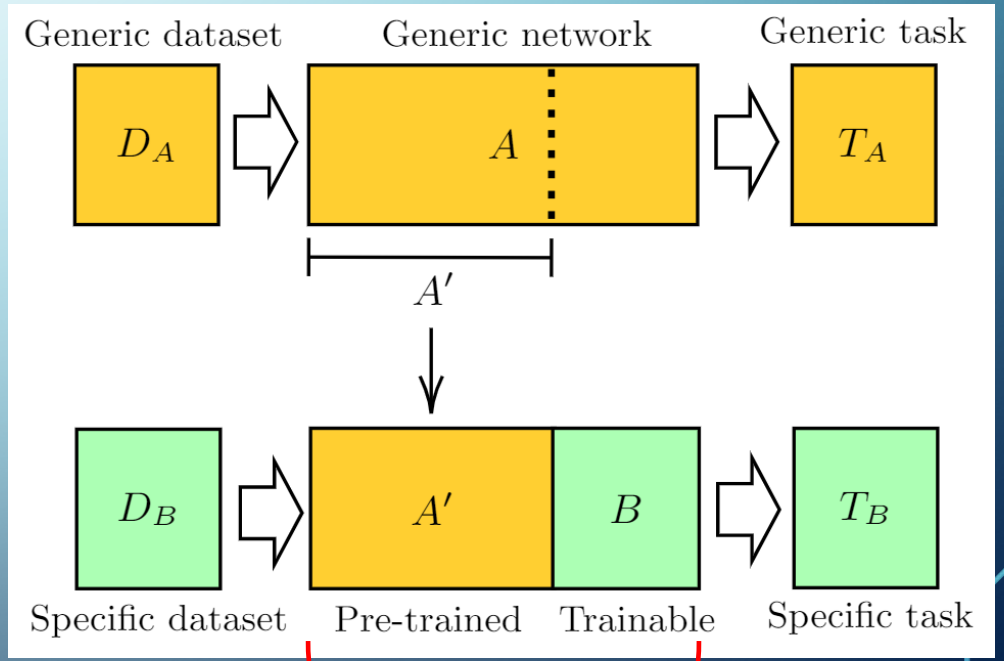
Ver Colab

## Feature extractor




entrenar

## Fine tuning




reentrenar

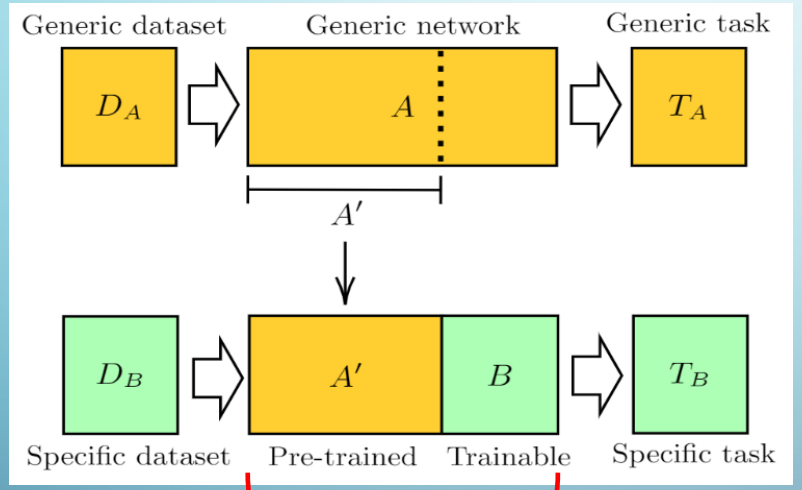
# Transfer Learning - ¿Qué estrategia usar?

Dataset semejante al generic

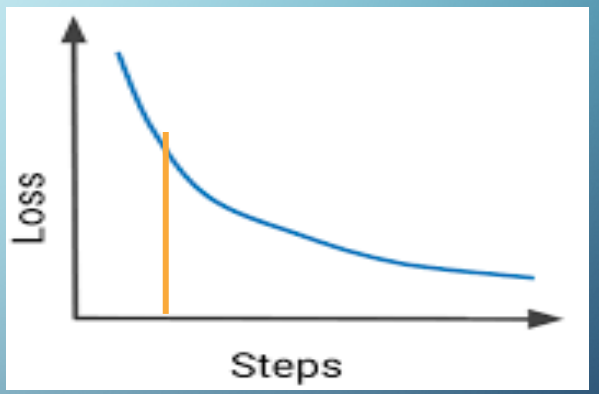
Dataset pequeño

Dataset grande

Dataset distinto al generic




entrenar




fine tunning (from pre-trained models)

https://pytorch.org/blog/introducing-torchvision-new-multi-weight-support-api/

# Transfer Learning - ¿Qué estrategia usar?

feature extractor

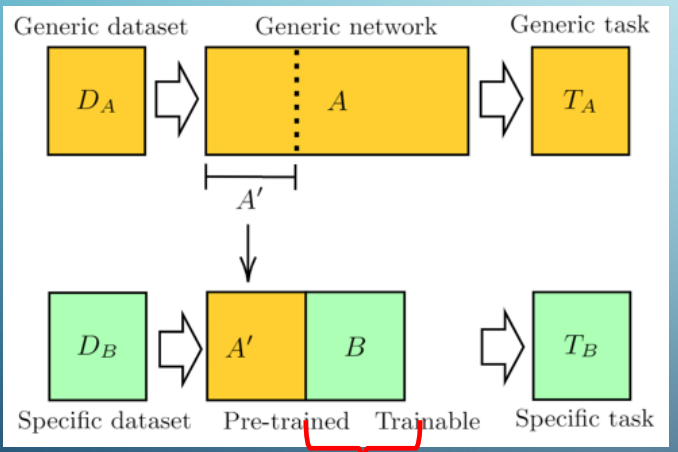
Dataset semejante al generic

fine tunning

Dataset pequeño

Dataset grande

https://discuss.pytorch.org/t/how-can-i-replace-the-forward-method-of-a-predefined-torchvision-model-with-my-customized-forward-function/54224/8




https://github.com/mortezamg63/Accessing-and-modifying-different-layers-of-a-pretrained-model-in-pytorch/blob/master/README.md

fine tunning (from pre-trained models)

feature extractor desde menor profundidad

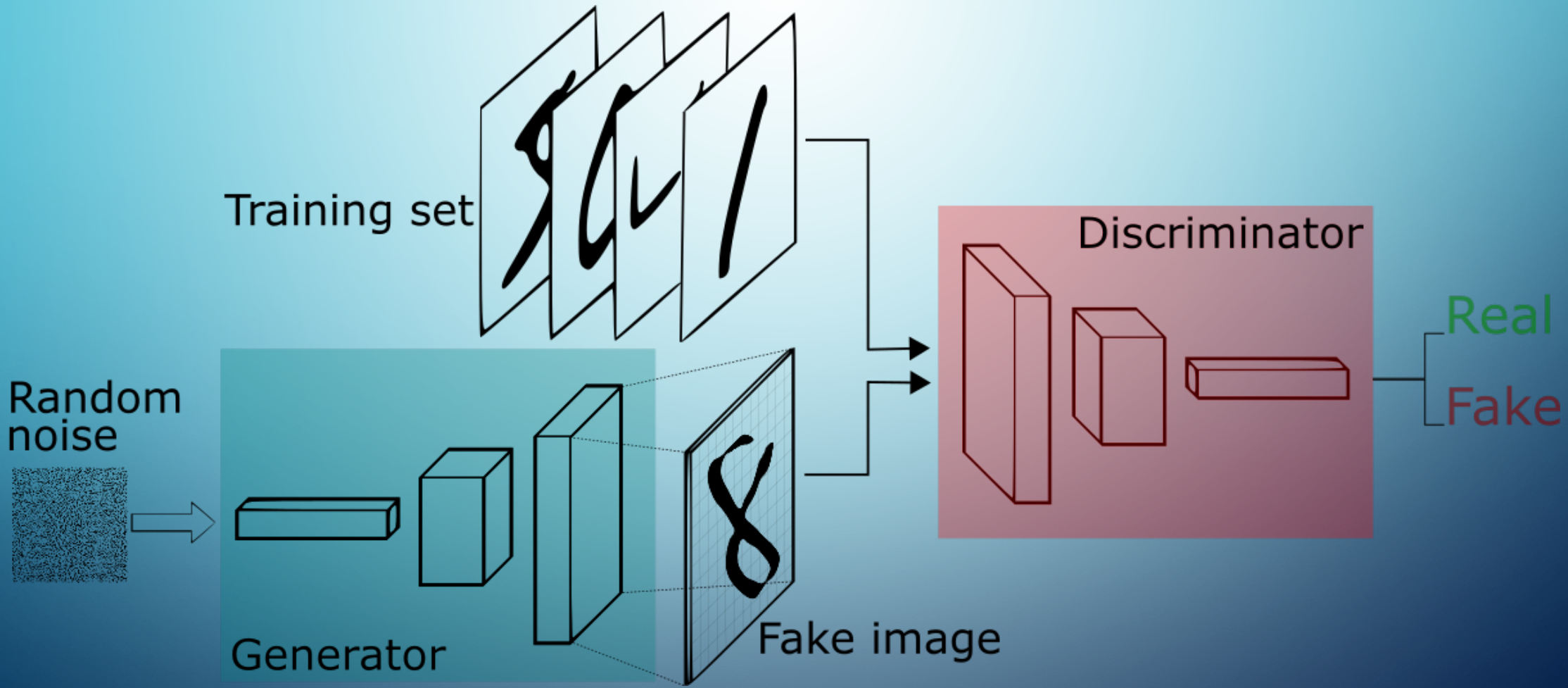
entrenar

Dataset distinto al generic

# ¡Un merecido descanso!

# Generative Adversarial Network (GAN)

2 redes neuronales enfrentadas: **G**enerador - **D**iscriminador



# Generative Adversarial Network (GAN)

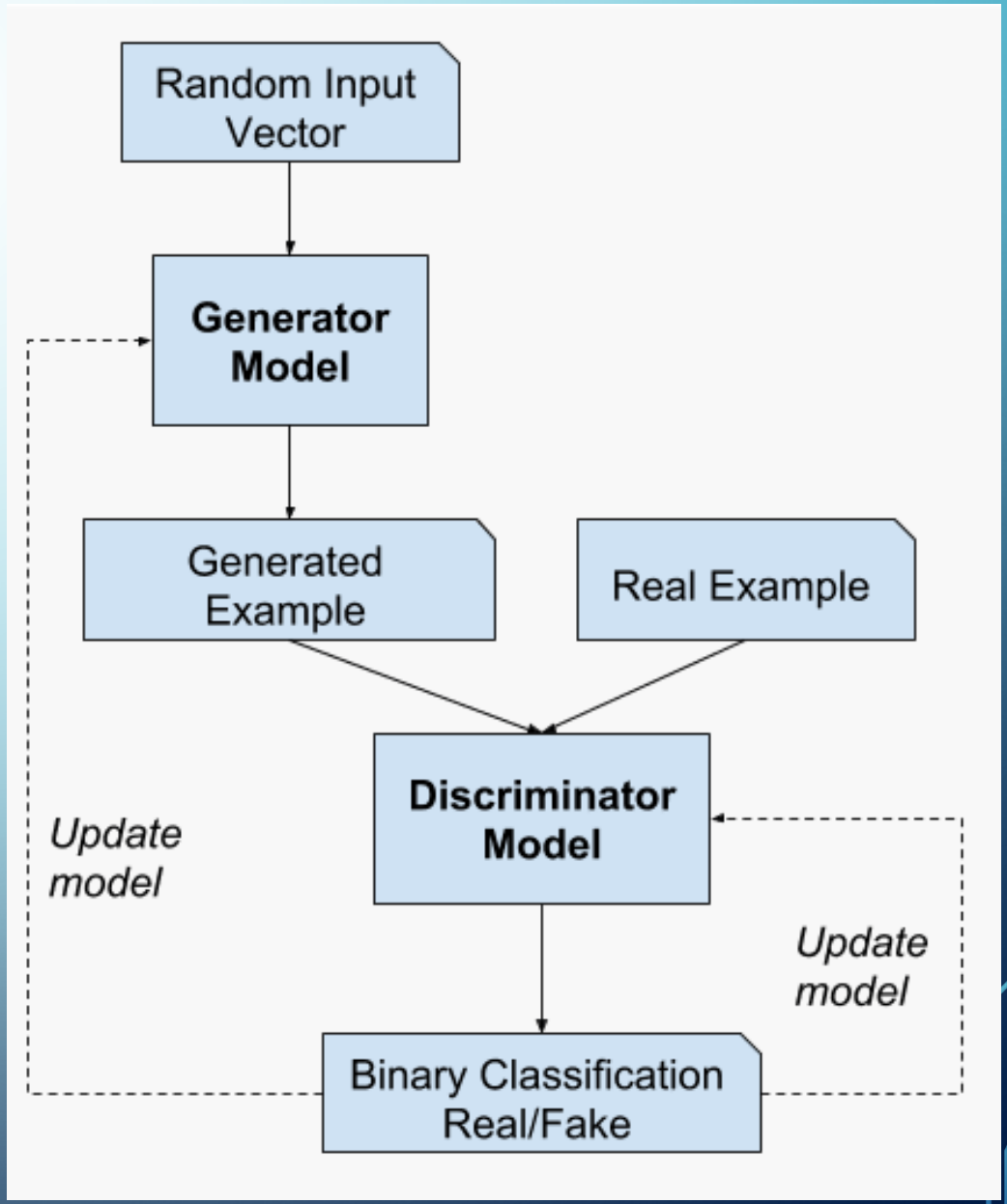
2 redes neuronales enfrentadas:

**G**enerador - **D**iscriminador

**G** → debe aprender una func de prob sobre los objetos que deseamos crear a partir de un RND (vector)

Quiero generar un perro→ ¿cual es la func de prob para que tome un valor de ella y saque un perro?

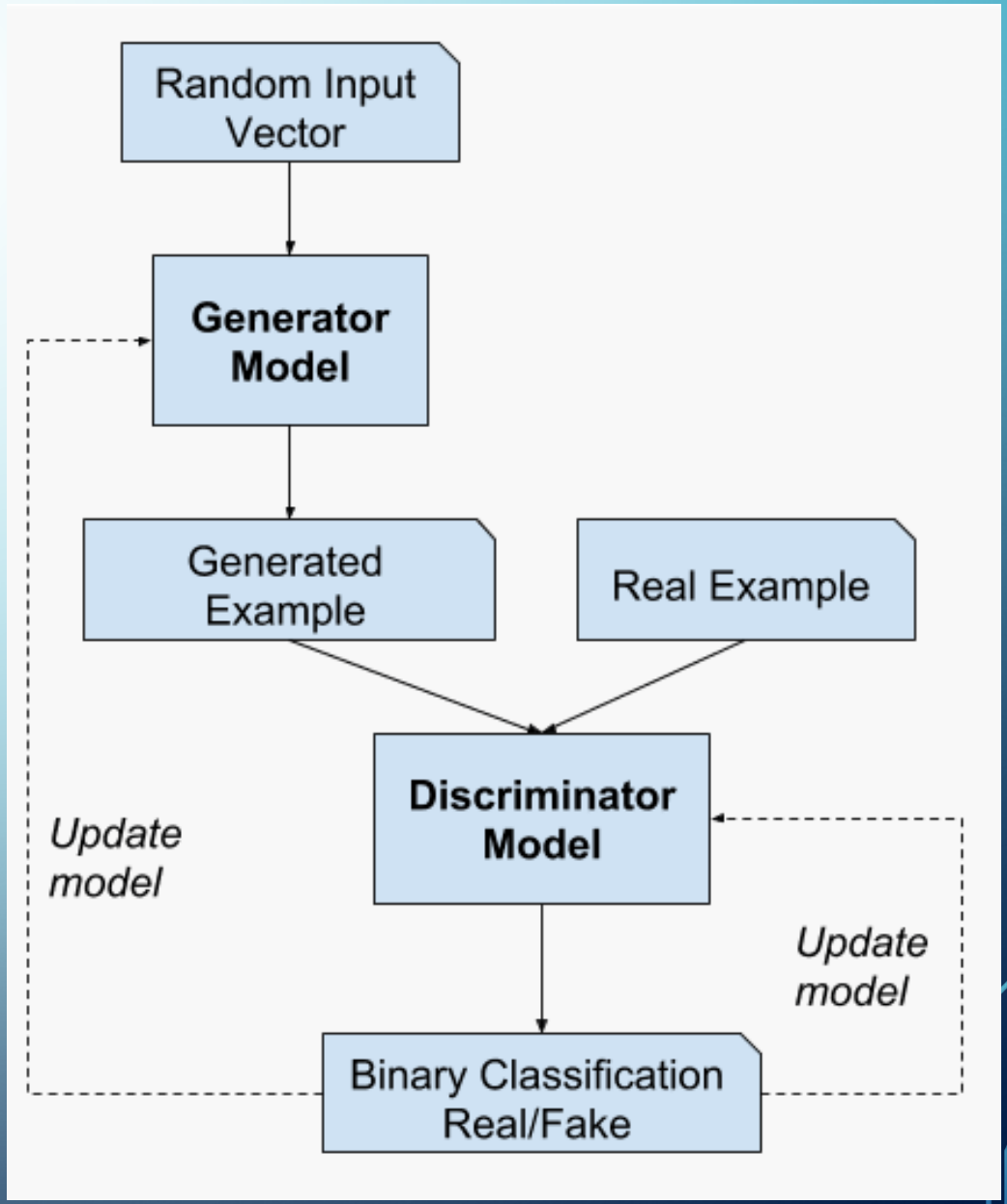
**generative**



# Generative Adversarial Network (GAN)

**G** → se entrena de forma indirecta ("supervisada"… pero de distinta forma)

**D** → determina si una muestra es real **1** o falsa **0**




**entrenamiento**

**G** → se entrena para que **D** falle

**D** → se entrena para no fallar

**adversarial**

# Generative Adversarial Network (GAN) – Función de costo

$z$ – vector aleatorio

$x$ – vector muestra (real)

$pz$ – func prob $z$

$pg$ – func prob $x$ generated

$pr$ – func prob $x$ real

**G** – generator (NN)

**D** – discriminator (NN)

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$
$$= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1-D(x)]$$

**D(x)=1 si x es real**

**D(x)=0 si x es falsa**

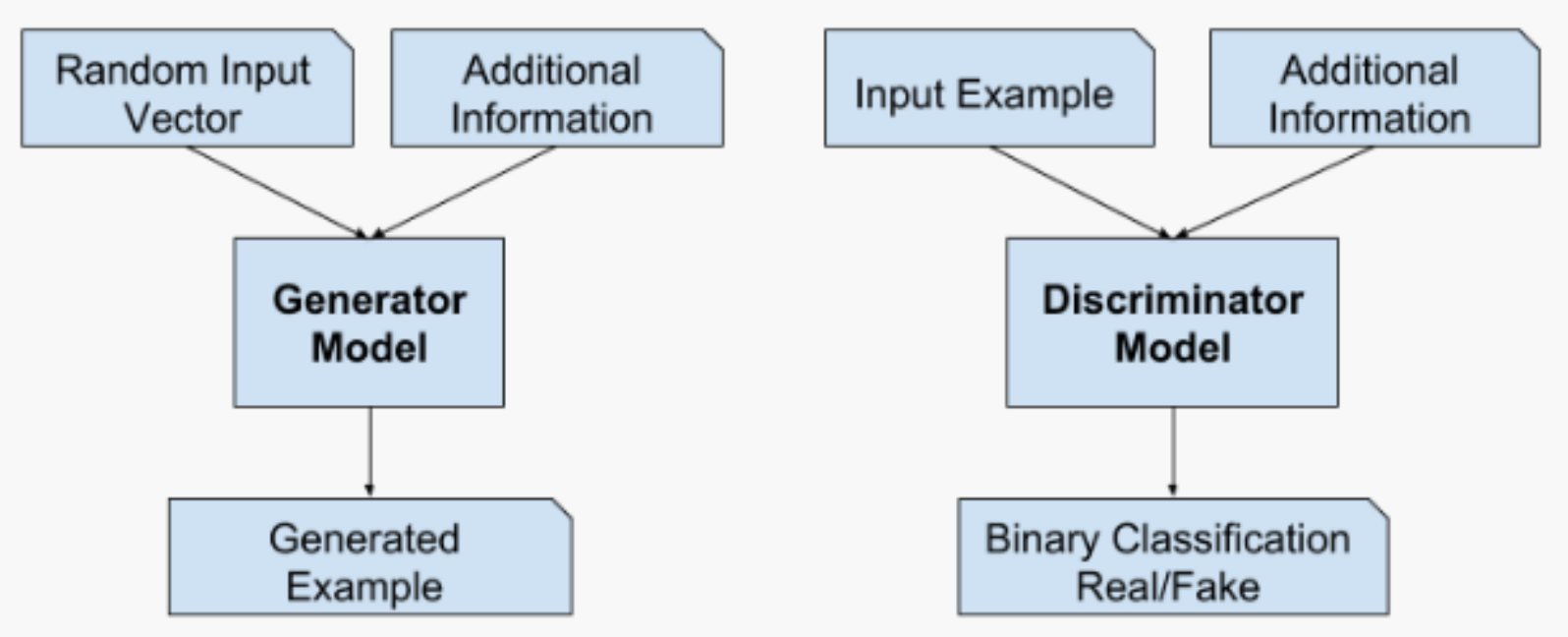
$$L(G,D) = \int_x \Big( p_r(x)\log(D(x)) + p_g(x)\log(1-D(x)) \Big) dx$$

https://neptune.ai/blog/gan-loss-functions

# Generative Adversarial Network (GAN)

## Conditionals GANs

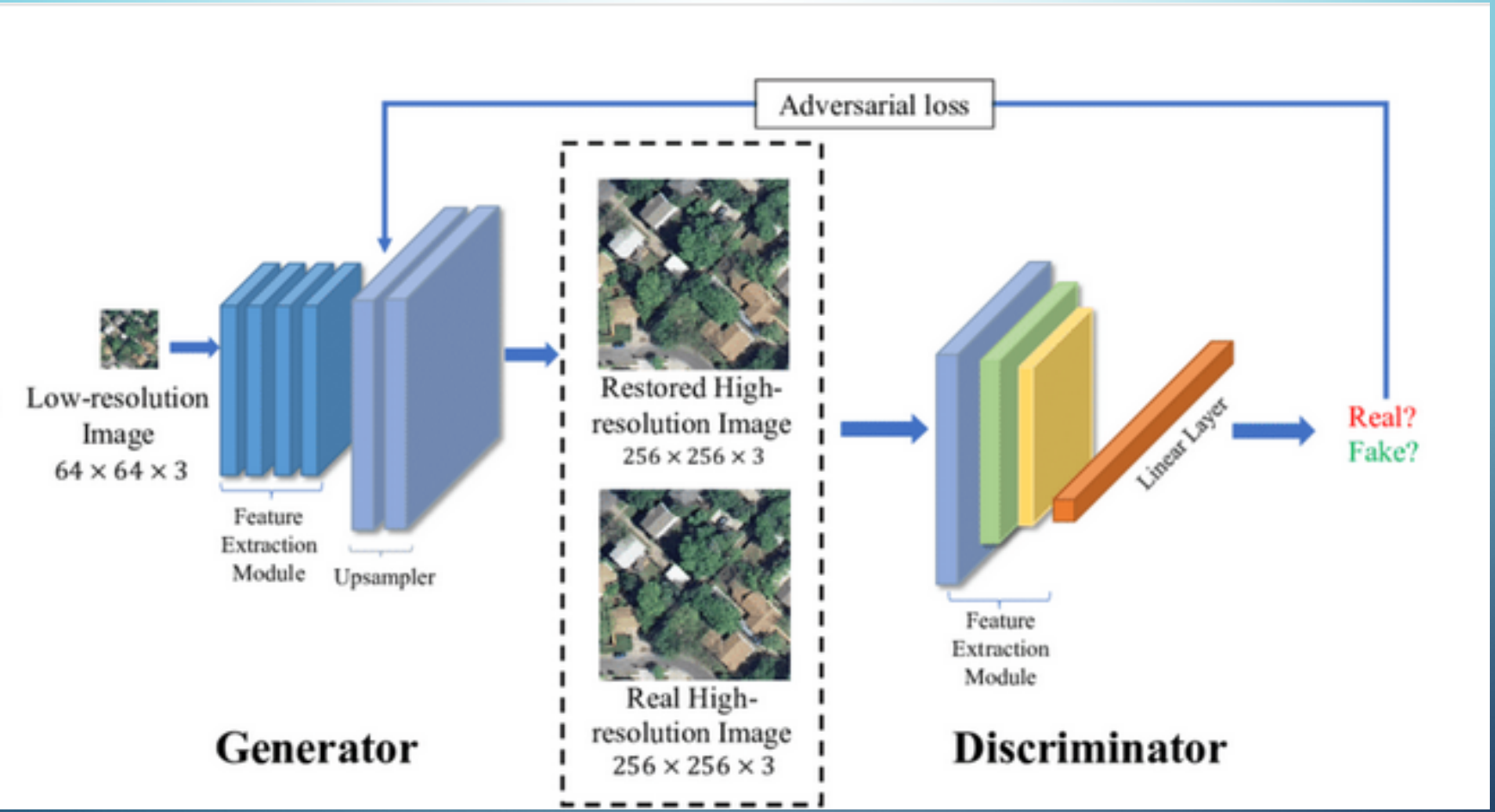
Se le pasa un ‘label’ para que genere algo bajo ese ‘label’




https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

# Generative Adversarial Network (GAN)

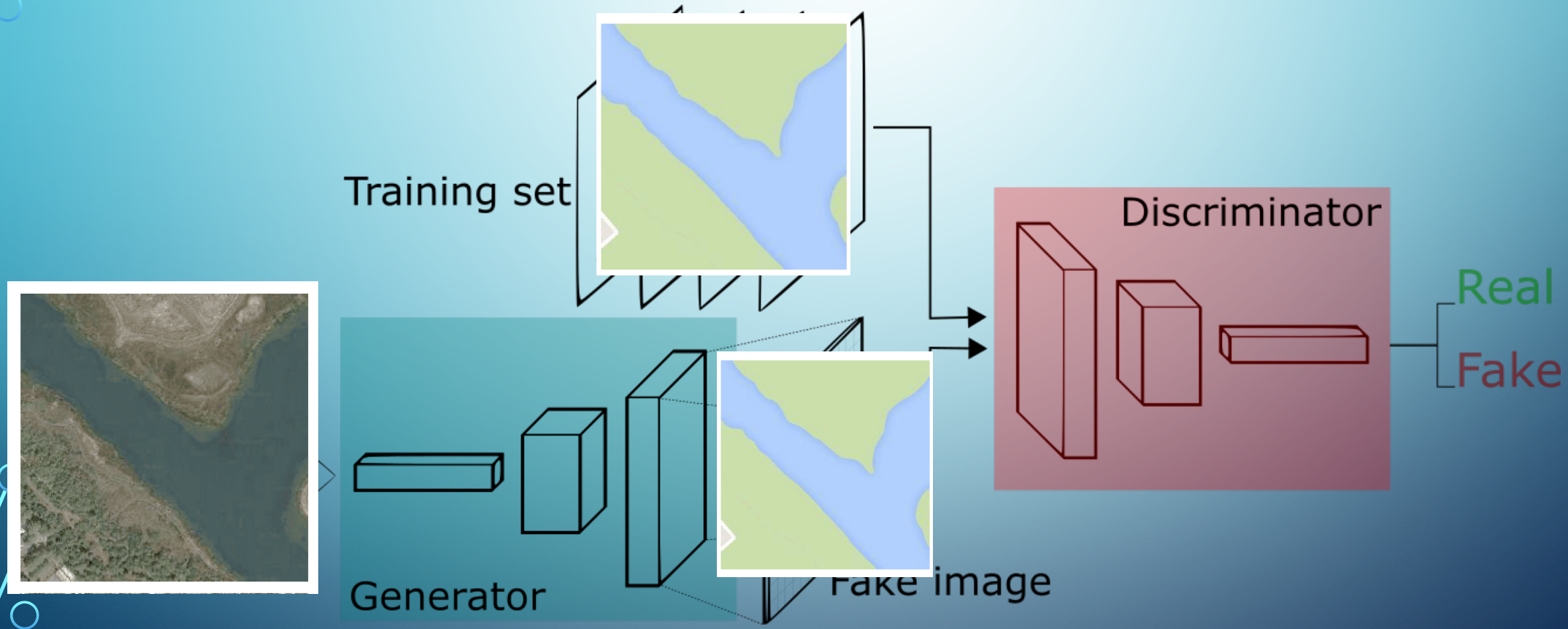
## Super resolution GANs




https://www.researchgate.net/publication/361868139_SWCGAN_Generative_Adversarial_Network_Combining_Swin_Transformer_and_CNN_for_Remote_Sensing_Image_Super-Resolution

# Generative Adversarial Network (GAN)

## Remote Sensing Image to Map Translation GANs




https://www.mdpi.com/2072-4292/13/10/1936

# Generative Adversarial Network (GAN)

GANs

- ver colab

https://github.com/Yangyangii/GAN-Tutorial

https://github.com/hindupuravinash/the-gan-zoo