

Aprendizaje Profundo

Facultad de Ingeniería
Universidad de Buenos Aires



Profesores:

Marcos Maillot
Antonio Zarauz
Gerardo Vilcamiza

TRANSFER LEARNING

GENERATIVE ADVERSARIAL NETWORKS

. Transfer learning

- Ventajas
- Estrategias
- Ejemplo en colab

. Generative adversarial networks

- Introducción
- Usos
- Ejemplo en colab



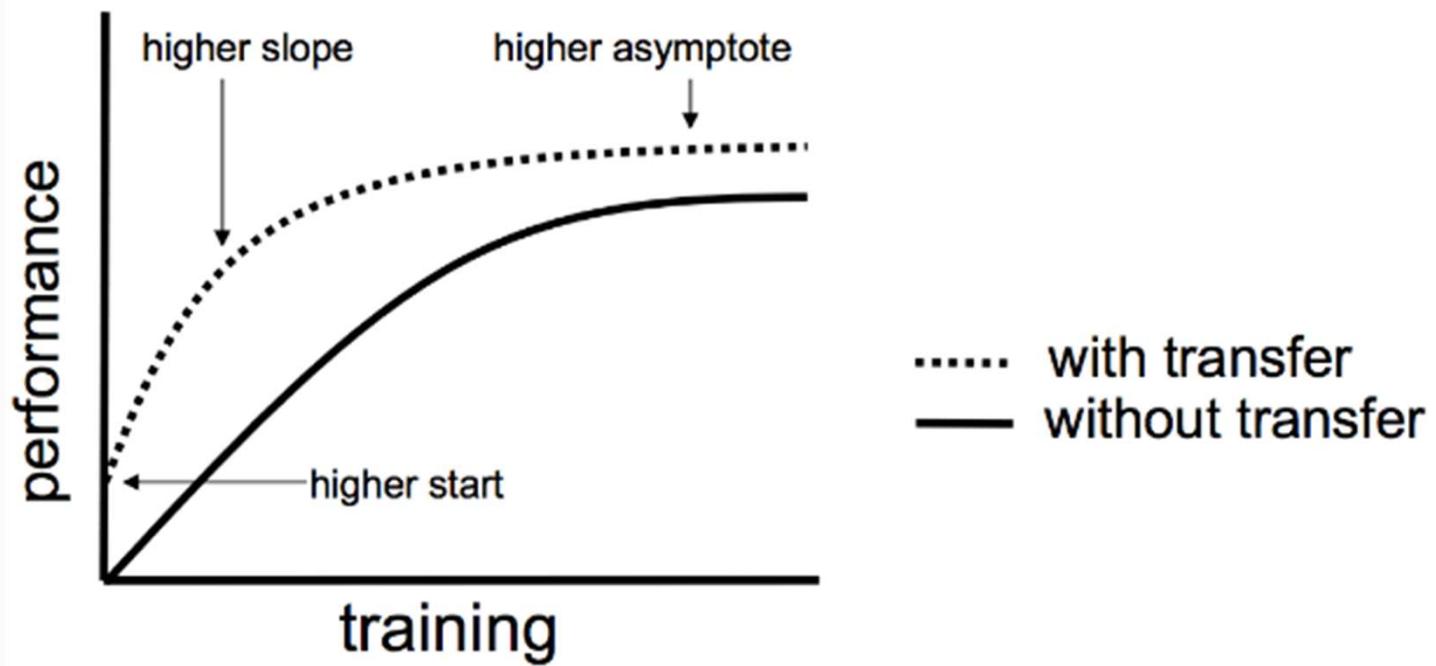
Transfer Learning

- 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

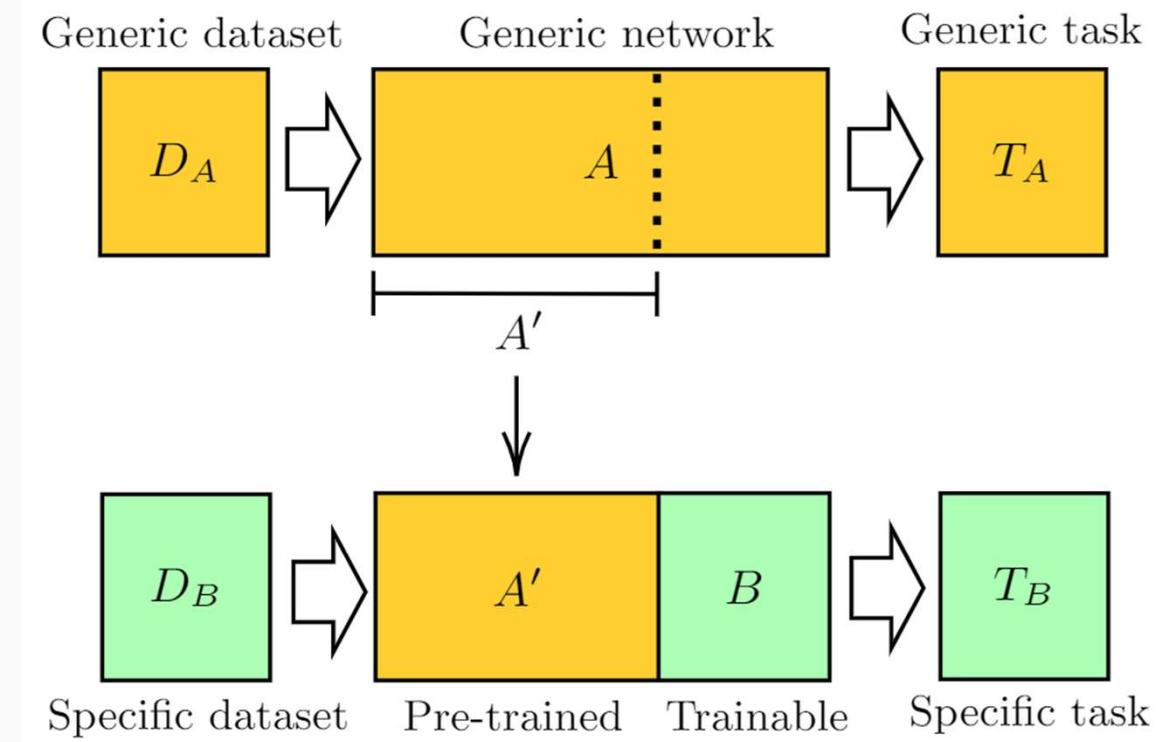
Ventajas:

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



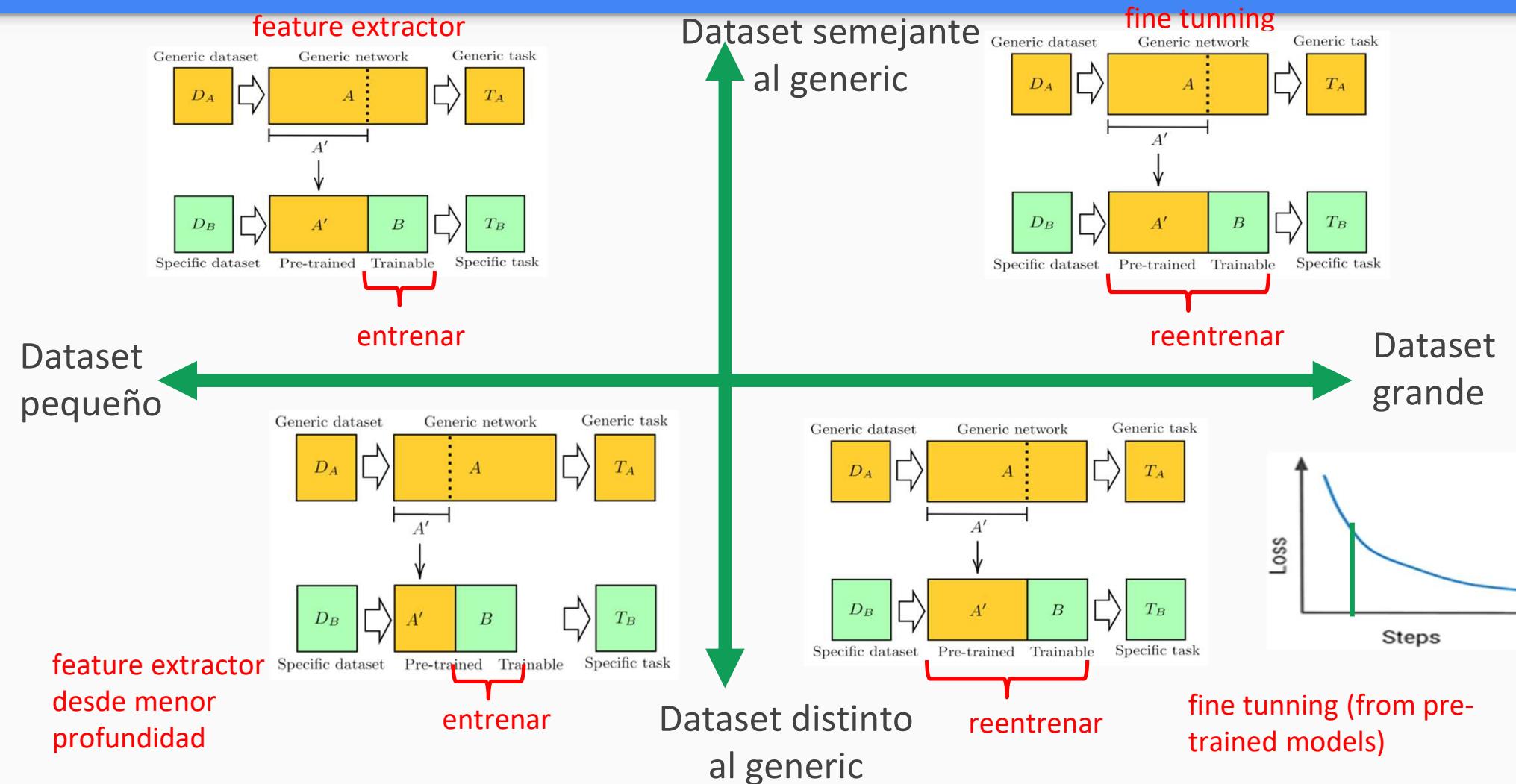
Transfer Learning

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?



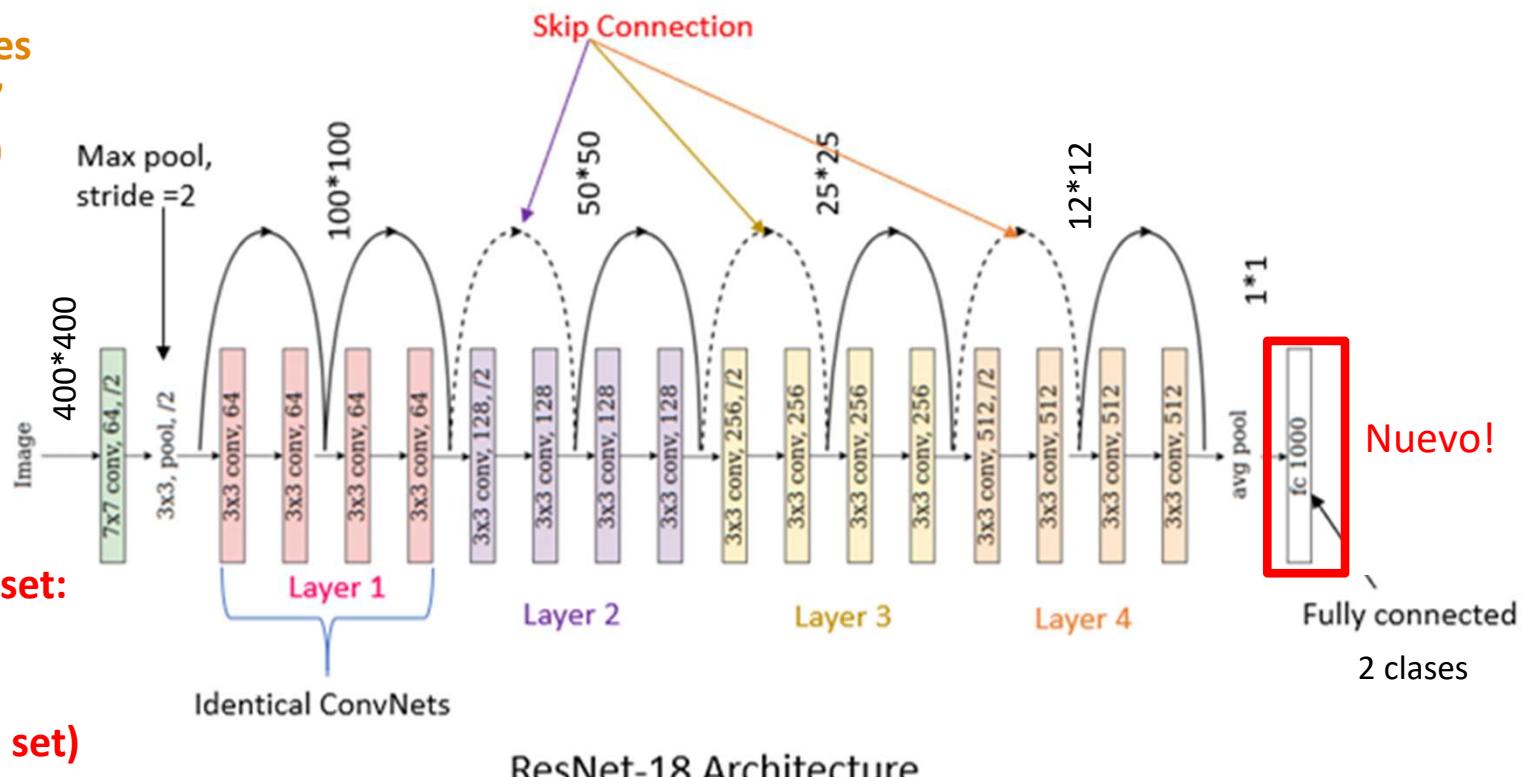
Transfer Learning - ejemplo colab

Generic Dataset:

ImageNet 1K

- 1000 clases
- 1.281.167 (train set)
- RGB

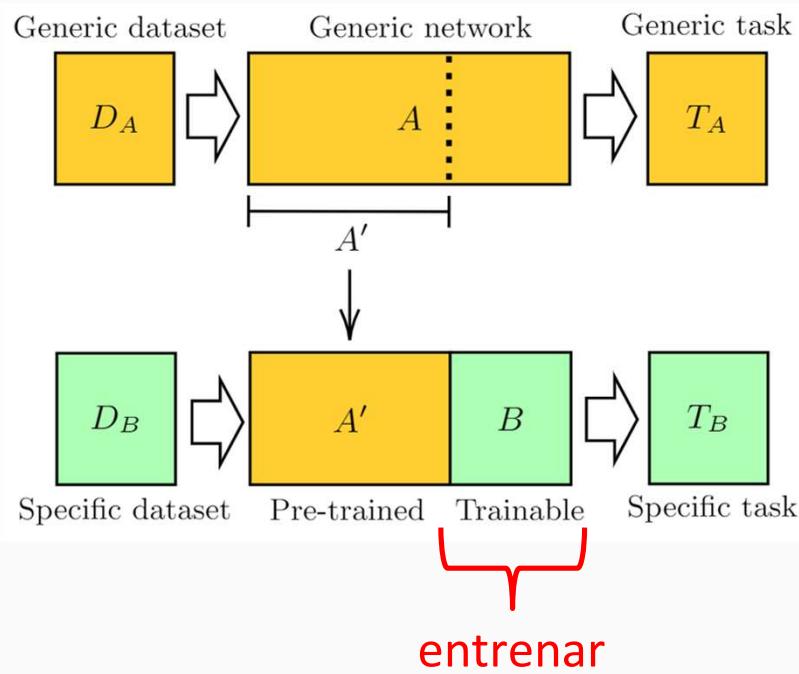
Adaptación de modelo base para cumplir la tarea específica



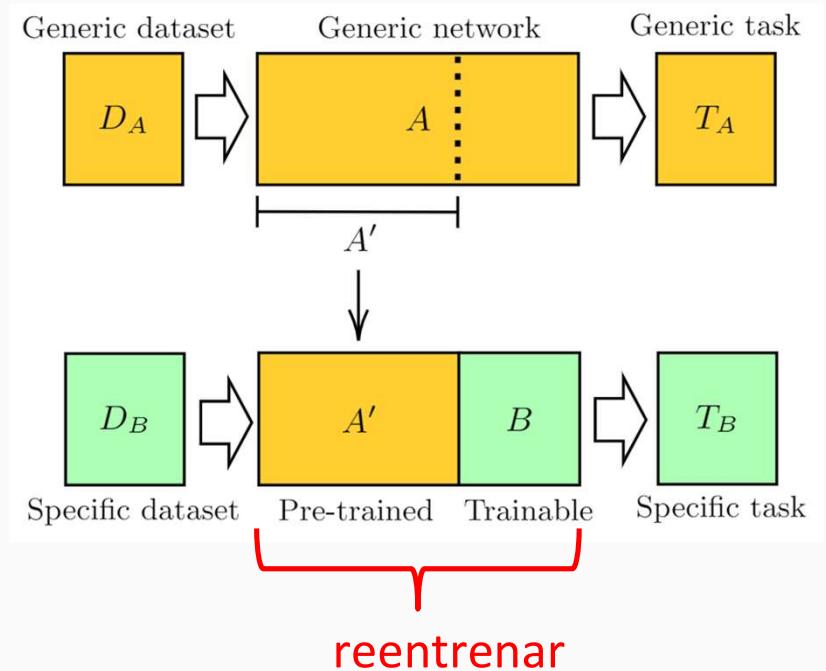
Transfer Learning - ejemplo colab

[Ver Colab](#)

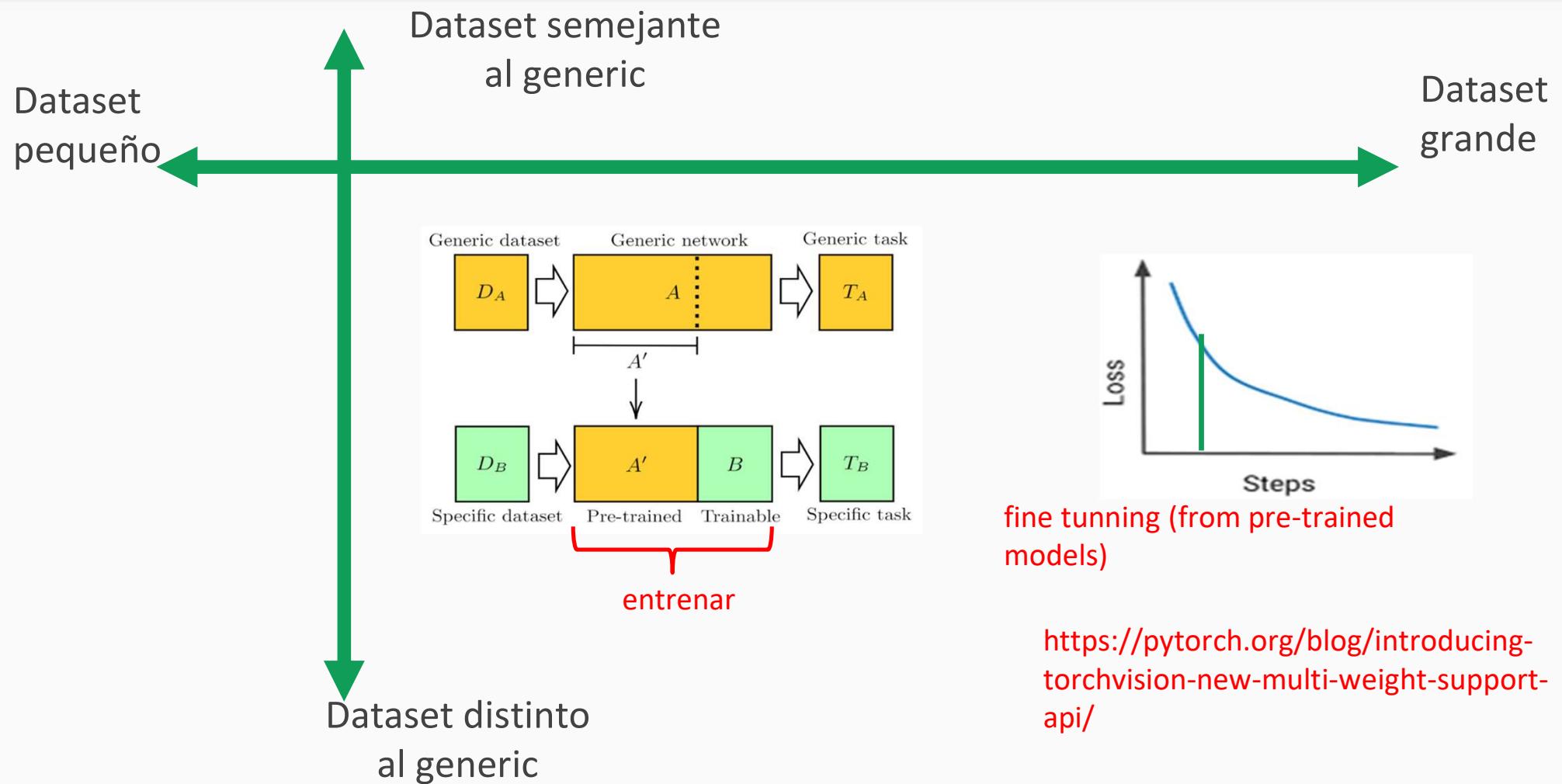
Feature extractor



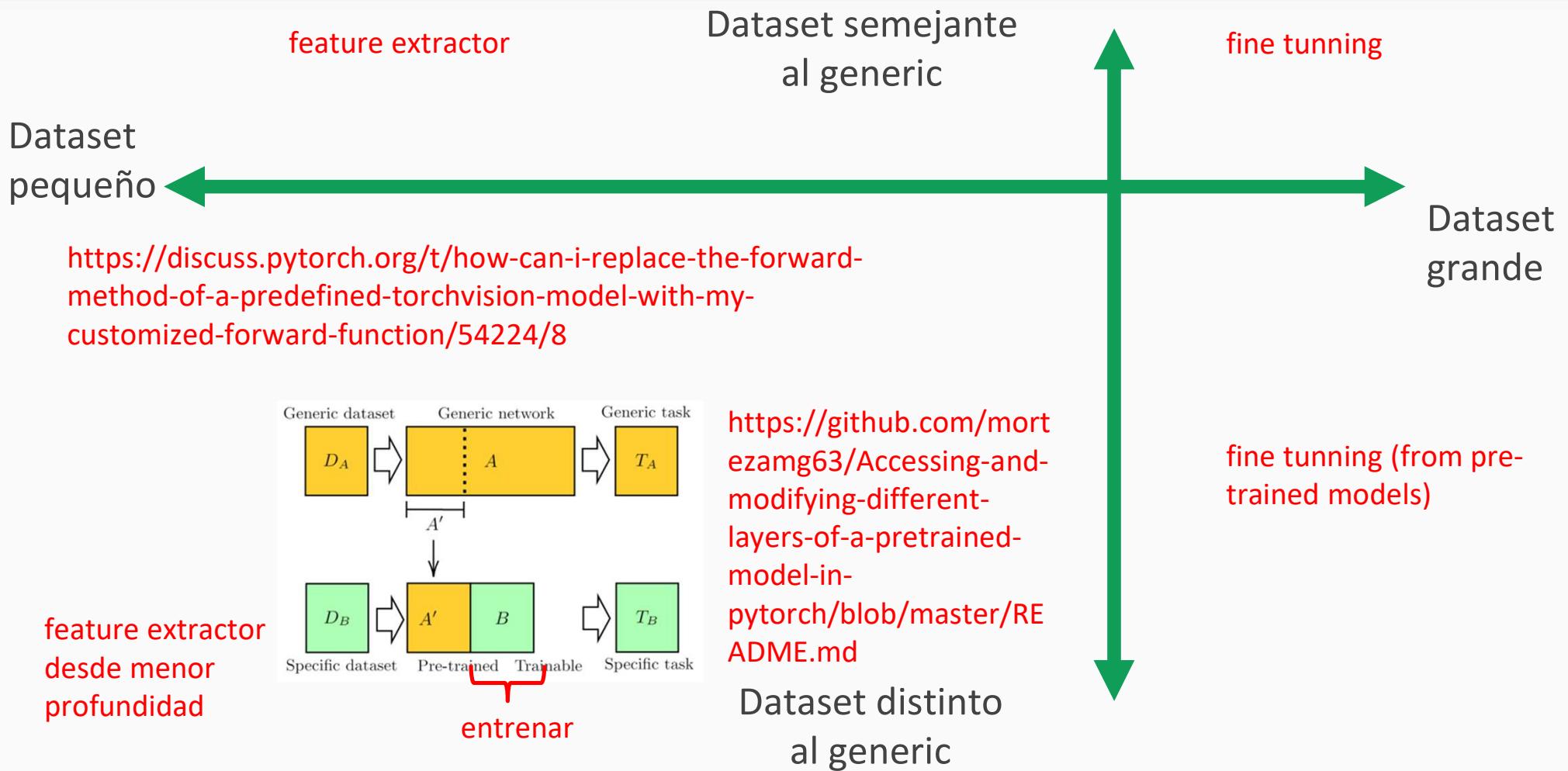
Fine tuning



Transfer Learning - ¿Qué estrategia usar?



Transfer Learning - ¿Qué estrategia usar?



¡Un merecido descanso!



Generative Adversarial Network (GAN)

Configuración de DL que busca **aprender la distribución de prob de los datos de entrenamiento** para poder **generar nuevos datos a partir de esa distribución**.

Se logra entrenando 2 modelos compitiendo:

G→ generador que busca generar datos “falsos”

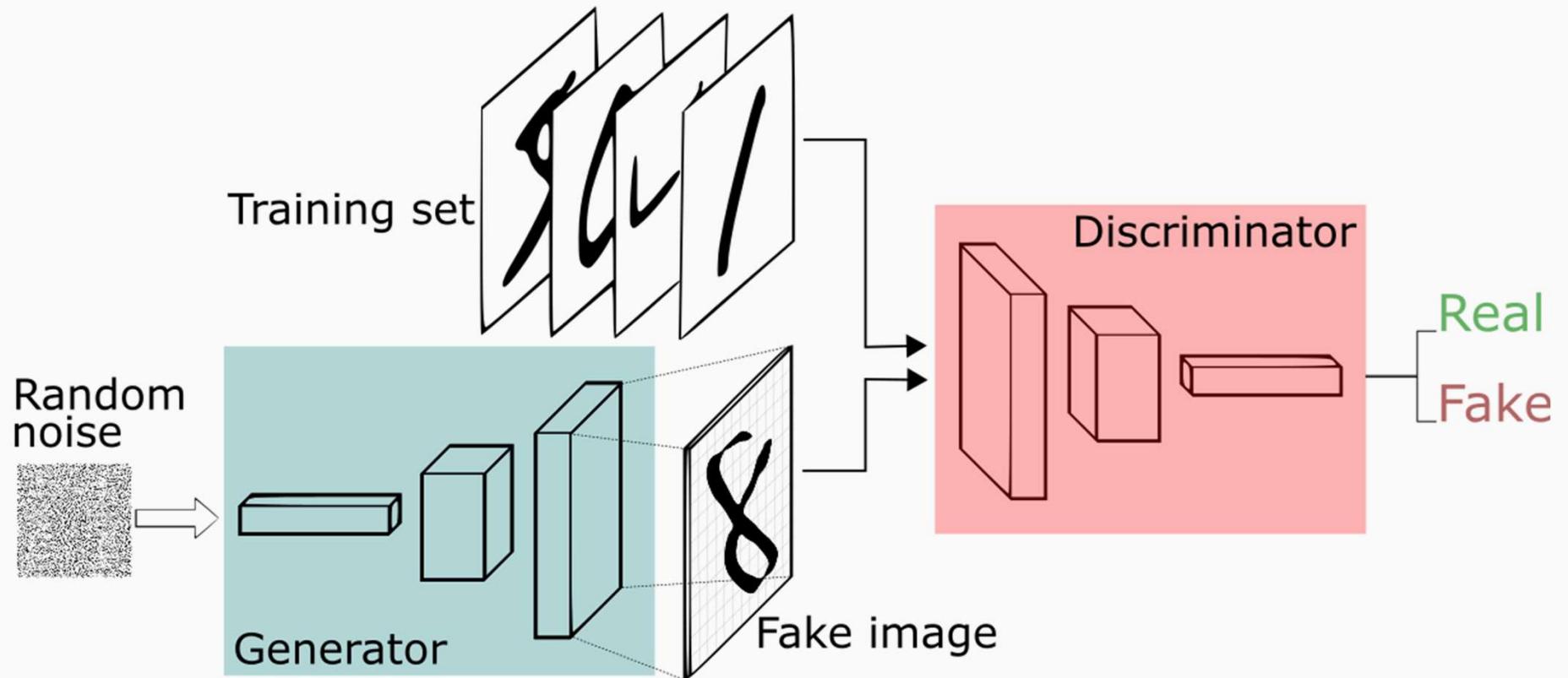
D→ discriminador que busca identificar datos “verdaderos” o “falsos”

Cuando el sistema “converge” el **D** no es capaz de distinguir datos reales de falsos (50% de real y 50% de falso)

Ese punto de convergencia es teórico y aún no se ha podido alcanzar.

Generative Adversarial Network (GAN)

2 redes neuronales enfrentadas: **Generador - Discriminador**



Generative Adversarial Network (GAN)

2 redes neuronales enfrentadas:

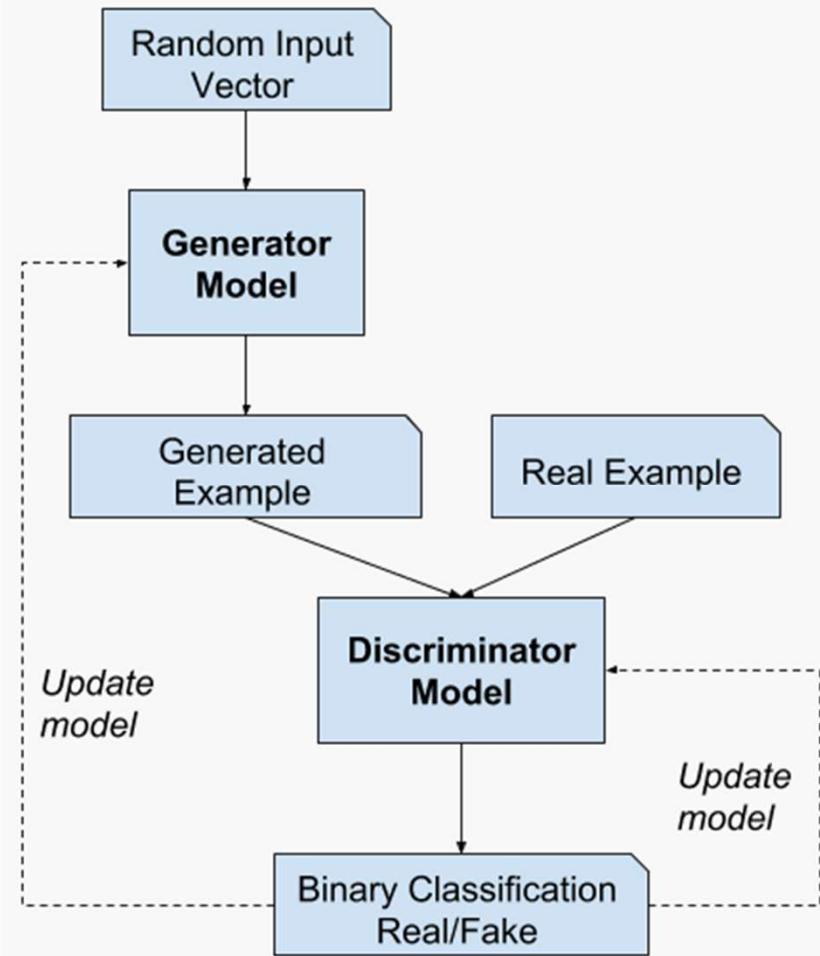
G - **Discriminador**

G → se entrena para que **D** falle

D → se entrena para no fallar

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

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



Generative Adversarial Network (GAN) – Función de costo

z – vector aleatorio

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

p_z – func prob z $= \mathbb{E}_{x \sim p_r(x)}[\log D(x)] + \mathbb{E}_{x \sim p_g(x)}[\log(1 - D(x))]$

p_g – func prob x generated

p_r – func prob x real

G – generator (NN)

D(x)=1 si x es real

D – discriminator (NN)

D(x)=0 si x es falsa

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

Generative Adversarial Network (GAN)

Entrena GANs es actualmente una especie de “trabajo de artesano”

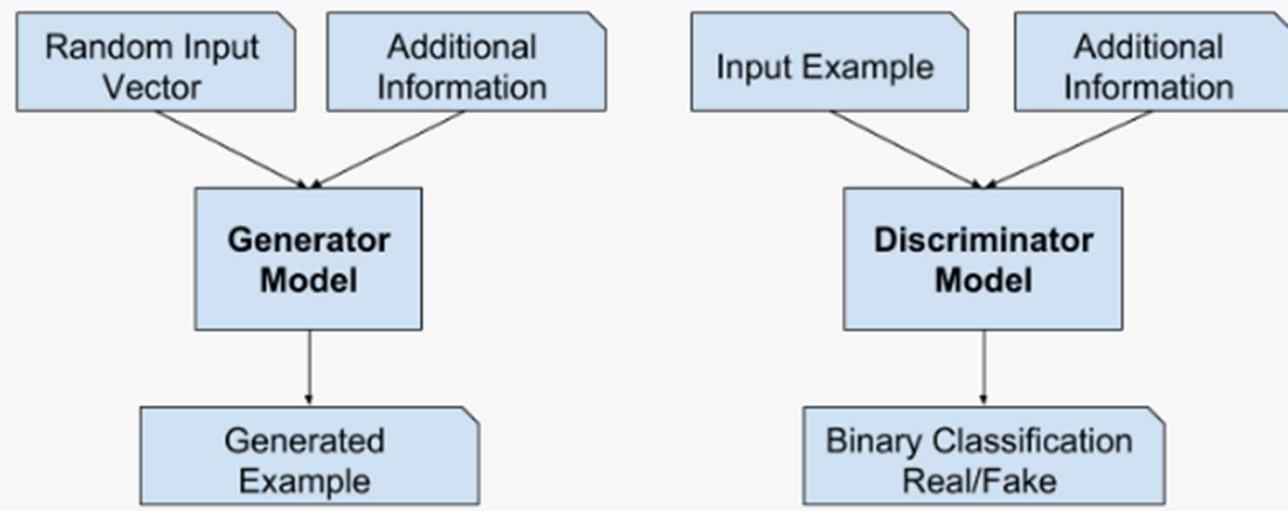
Algunos trucos son:

- TanH como output del generador.
- Poner etiquetas reales a muestras falsas para entrenar generador.
- Usar batch normalization y minibatch.
- Evitar Maxpool, usar average pool o hacer downsampling con el stride de conv.
- Evitar usar ReLu, usar LeakyReLu.
- Usar Adam como optimizador, beta_1 = 0,5, lr=0,0002.
- Usar label smoothing (en lugar de 1 – 0; usar 0,9 – 0,1 para las clases real-fake).

Generative Adversarial Network (GAN)

Conditionals GANs

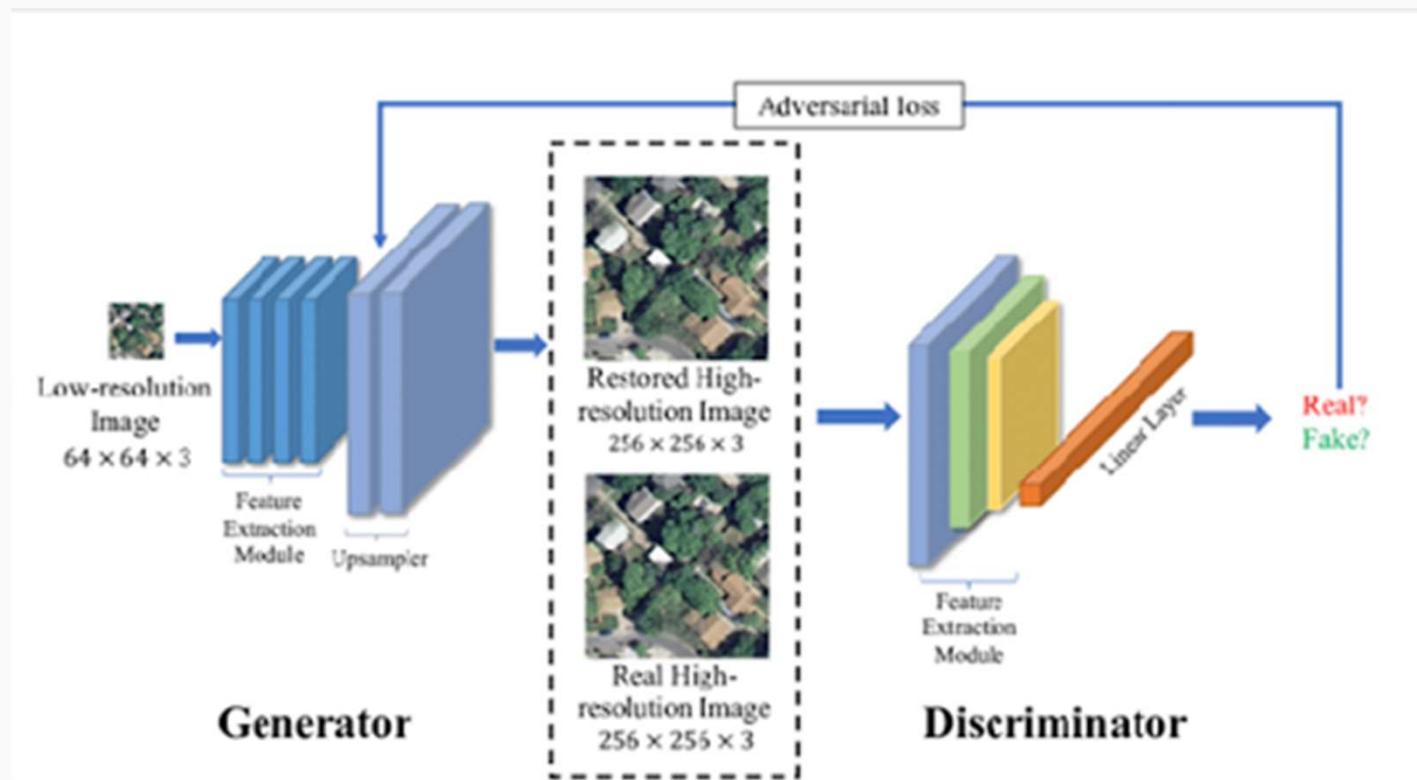
Se le pasa un ‘label’ para que genere algo bajo ese ‘label’



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

Generative Adversarial Network (GAN)

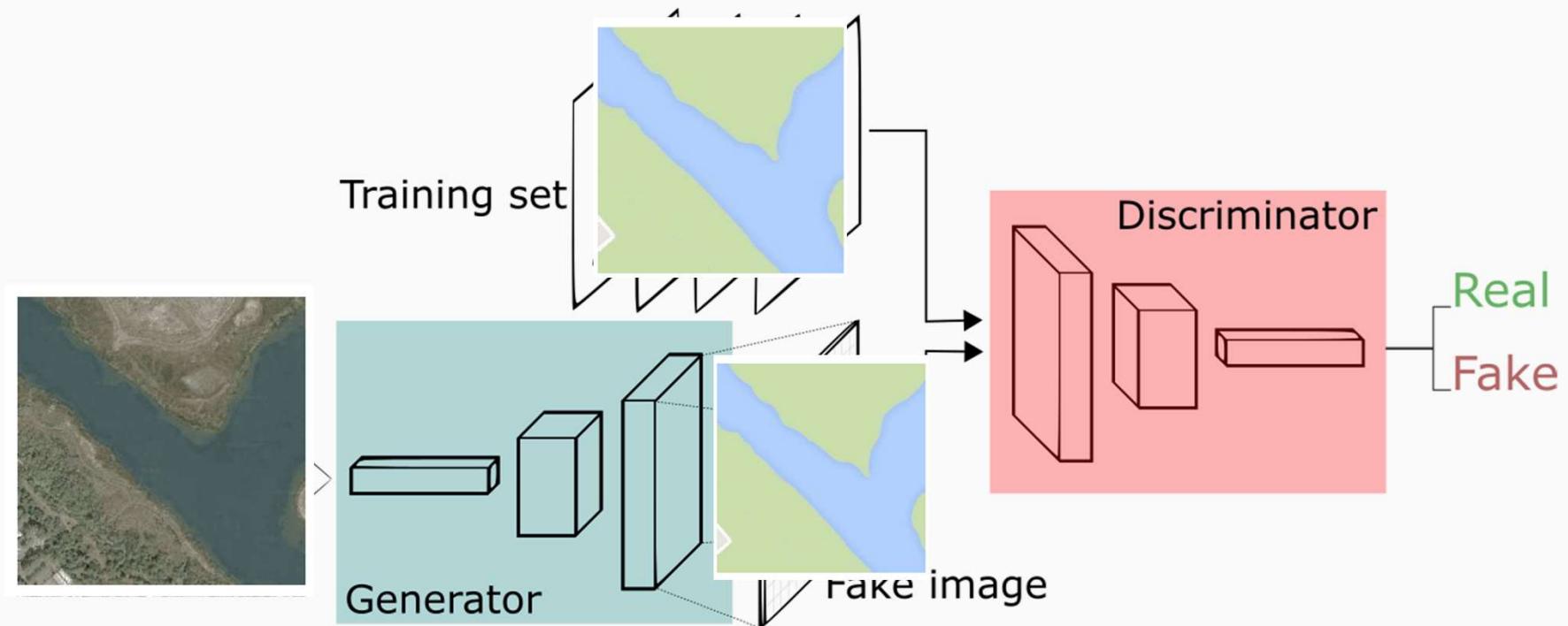
Super resolution GANs



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>

How to train GANs:

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

<https://arxiv.org/abs/1606.03498>

<https://github.com/soumith/ganhacks>

<https://dropsofai.com/best-practices-for-training-stable-gans/>

Mas sobre GANs:

https://docs.pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

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

<https://github.com/nashory/gans-awesome-applications>