



Módulo 7 – Fundamentos de Deep Learning

Transfer Learning

Especialización en Ciencia de Datos

Contenido



1. Transfer Learning.
2. Implementación Keras – Tensorflow.

Transfer Learning



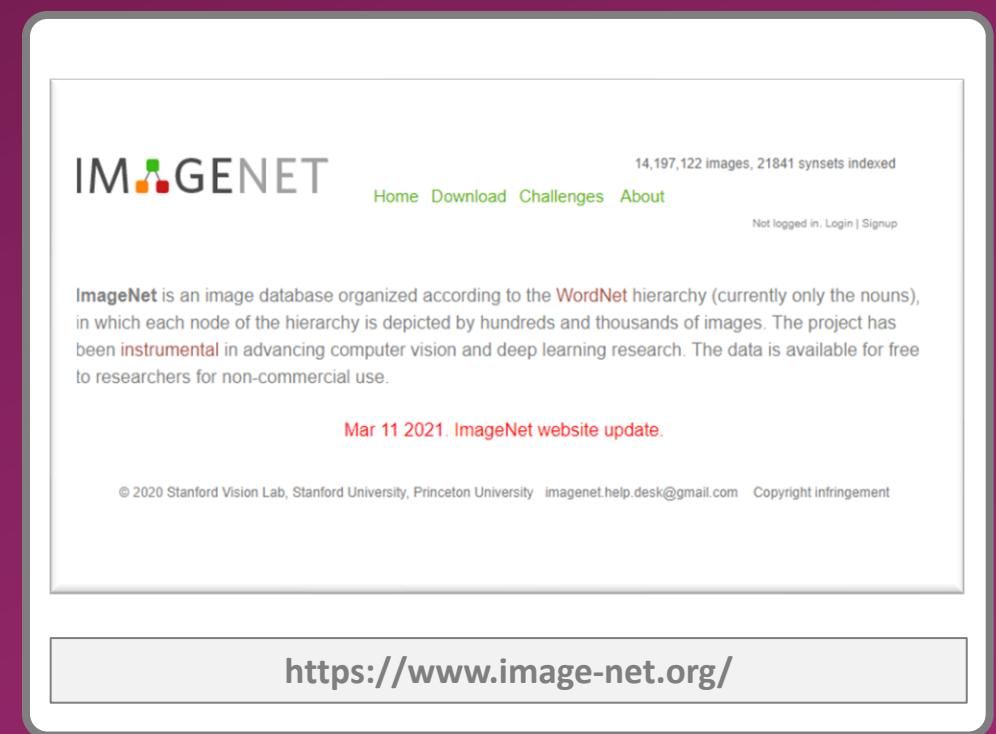
KIBERNUM

Evolución de Modelos CNN

Durante los últimos años ha habido esfuerzos múltiples de diferentes equipos en el avance de arquitecturas de redes neuronales cada vez más eficientes. El proyecto **ImageNet** es una gran base de datos visual diseñada para su uso en la investigación de software de reconocimiento de objetos visuales.

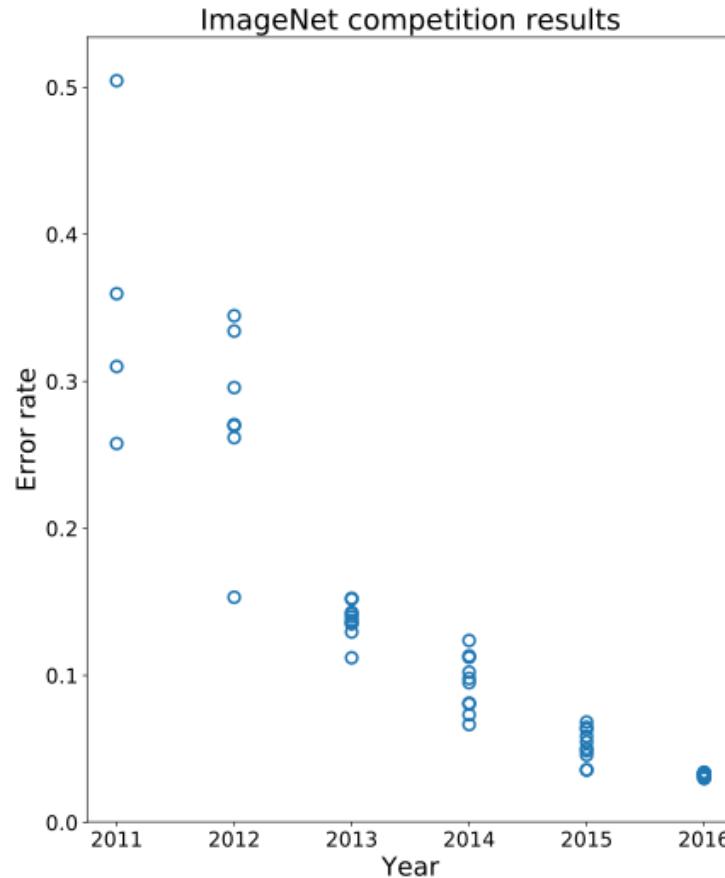
El proyecto ImageNet organiza un concurso de software anual, ImageNet Large Scale Visual Recognition Challenge (ILSVRC), donde los programas de software compiten para clasificar y detectar correctamente objetos y escenas.

El proyecto ha anotado a mano más de 14 millones de imágenes para indicar qué objetos se representan, y en al menos un millón de imágenes, también se proporcionan cuadros delimitadores. ImageNet contiene más de 20.000 categorías.

A screenshot of the ImageNet website homepage. The header features the "IMAGENET" logo with a colorful "I". To the right of the logo are links for "Home", "Download", "Challenges", and "About", along with a "Not logged in. Login | Signup" link. A small note above the "About" link states "14,197,122 images, 21841 synsets indexed". The main content area contains a brief description of ImageNet as an image database organized according to the WordNet hierarchy, noting its instrumental role in advancing computer vision and deep learning research. It mentions that the data is available for free to researchers for non-commercial use. Below this is a red text update: "Mar 11 2021. ImageNet website update." At the bottom of the page, there is a copyright notice: "© 2020 Stanford Vision Lab, Stanford University, Princeton University imagenet.help.desk@gmail.com Copyright infringement".

<https://www.image-net.org/>

Evolución de Modelos CNN



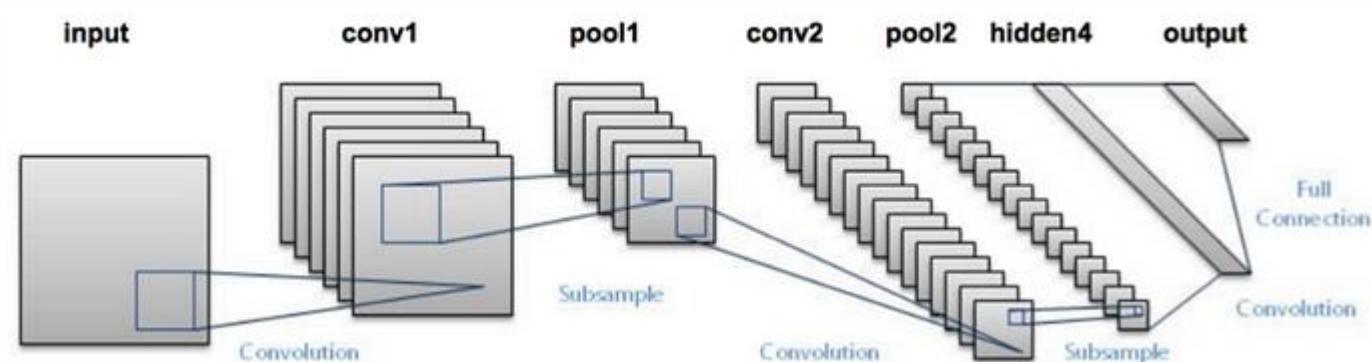
➤ Esta es parte de la evolución que han tenido los competidores de ImageNet. Nótese como ha ido disminuyendo la tasa de errores en sus entrenamientos.

➤ A continuación, listaremos los competidores que mejores avances han logrado hasta el momento con sus arquitecturas.

Evolución de Modelos CNN

LeNet-5 (1998)

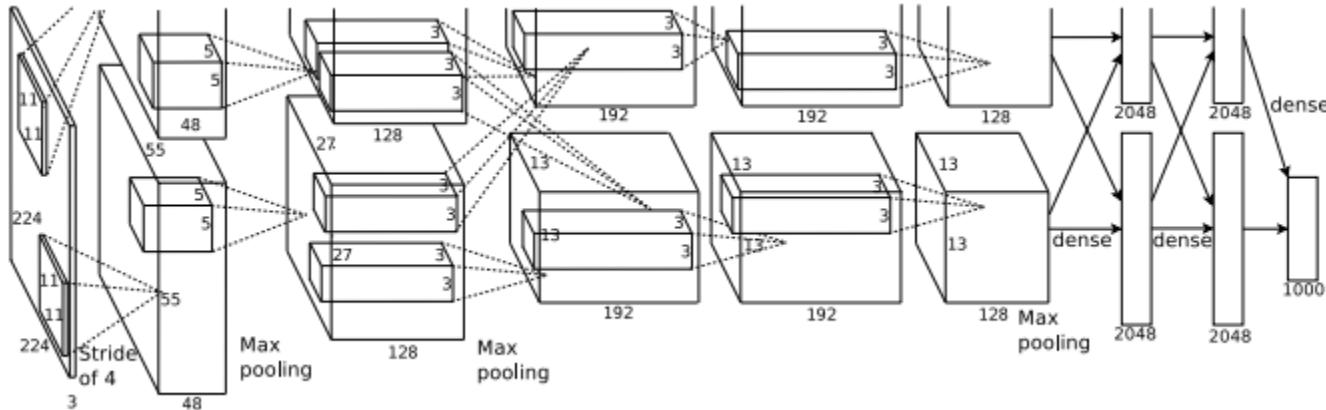
Varios bancos aplicaron LeNet-5, una red convolucional pionera de 7 niveles de LeCun et al en 1998, que clasifica dígitos, para reconocer números escritos a mano en cheques digitalizados en imágenes de entrada en escala de grises de 32x32 píxeles. La capacidad de procesar imágenes de mayor resolución requiere capas más grandes y convolucionales, por lo que esta técnica estaba limitada por la disponibilidad de recursos informáticos de esa época.



Evolución de Modelos CNN

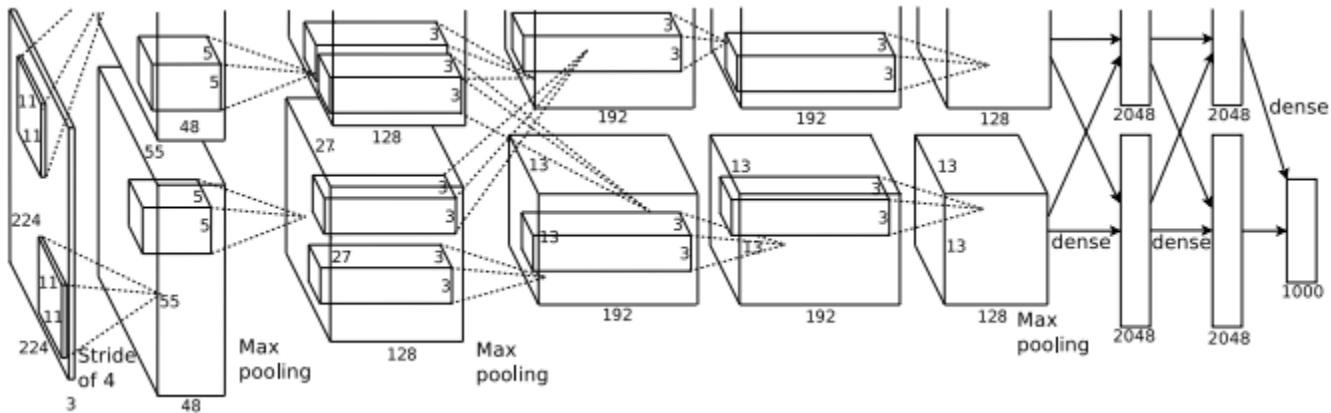
AlexNet-5 (2012)

En 2012, AlexNet superó significativamente a todos los competidores anteriores y ganó el desafío al reducir el error de los cinco primeros del 26 % al 15,3 %. La tasa de error del segundo lugar entre los 5 primeros, que no fue una variación de CNN, fue de alrededor del 26,2%.



Evolución de Modelos CNN

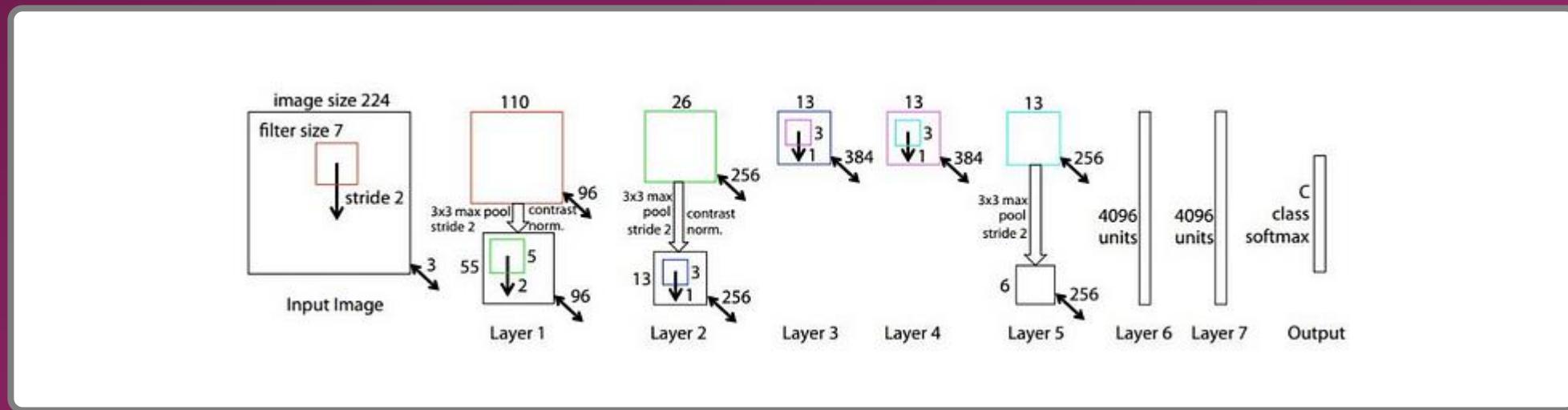
La red tenía una arquitectura muy similar a LeNet de Yann LeCun et al, pero era más profunda, con más filtros por capa y con capas convolucionales apiladas. Consistía en 11x11, 5x5, 3x3, circunvoluciones, agrupación máxima, abandono, aumento de datos, activaciones de ReLU, SGD con impulso. Adjuntó activaciones de ReLU después de cada capa convolucional y completamente conectada. AlexNet se entrenó durante 6 días simultáneamente en dos GPU Nvidia Geforce GTX 580, razón por la cual su red se divide en dos canales. AlexNet fue diseñado por el grupo SuperVision, formado por Alex Krizhevsky, Geoffrey Hinton e Ilya Sutskever.



Evolución de Modelos CNN

ZFNet-5 (2013)

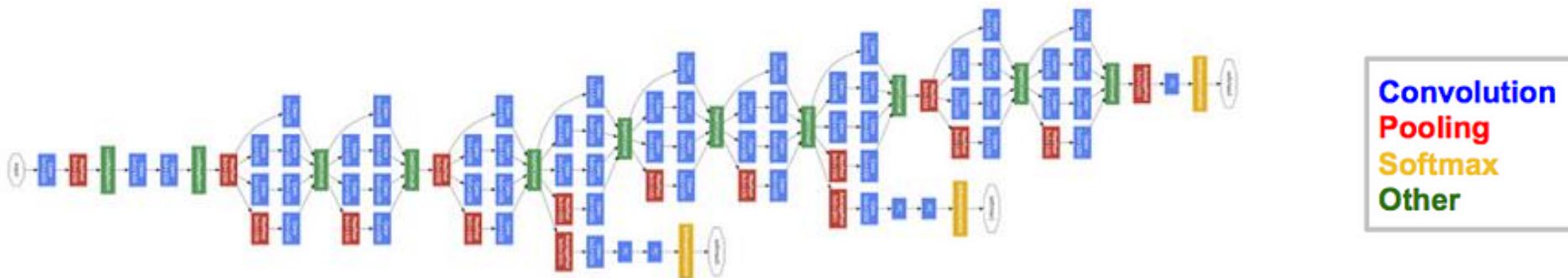
Como era de esperar, el ganador de ILSVRC 2013 también fue una CNN que se conoció como ZFNet. Logró una tasa de error entre los 5 primeros del 14,8%, que ahora ya es la mitad de la tasa de error no neuronal mencionada anteriormente. Fue principalmente un logro al ajustar los hiperparámetros de AlexNet mientras se mantenía la misma estructura con elementos adicionales de aprendizaje profundo como se discutió anteriormente en este ensayo.



Evolución de Modelos CNN

GoogLeNet/Inception(2014)

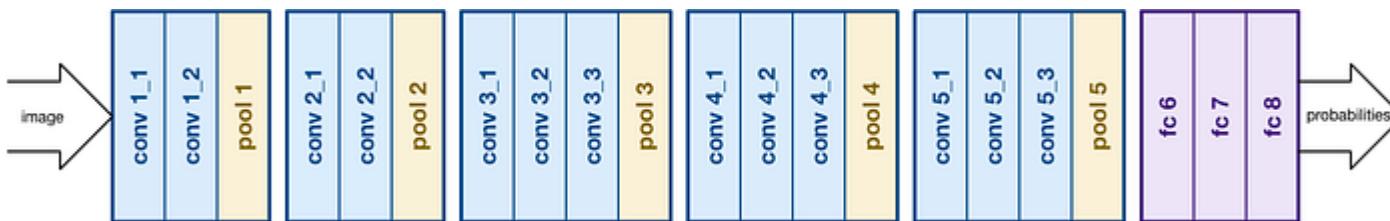
El ganador de la competencia ILSVRC 2014 fue GoogLeNet (también conocido como Inception V1) de Google. ¡Logró una tasa de error entre los 5 primeros del 6,67%! Esto estuvo muy cerca del nivel de desempeño humano que los organizadores del desafío ahora se vieron obligados a evaluar. Resulta que esto fue bastante difícil de hacer y requirió algo de entrenamiento humano para superar la precisión de GoogLeNets. Después de unos días de capacitación, el experto humano (Andrej Karpathy) pudo lograr una tasa de error entre los cinco primeros del 5,1% (modelo único) y del 3,6% (conjunto). La red usó una CNN inspirada en LeNet pero implementó un elemento novedoso que se denomina módulo de inicio. Usó normalización por lotes, distorsiones de imagen y RMSprop. Este módulo se basa en varias convoluciones muy pequeñas para reducir drásticamente el número de parámetros. Su arquitectura consistía en una CNN de 22 capas de profundidad, pero redujo la cantidad de parámetros de 60 millones (AlexNet) a 4 millones.



Evolución de Modelos CNN

VGGNet (2014)

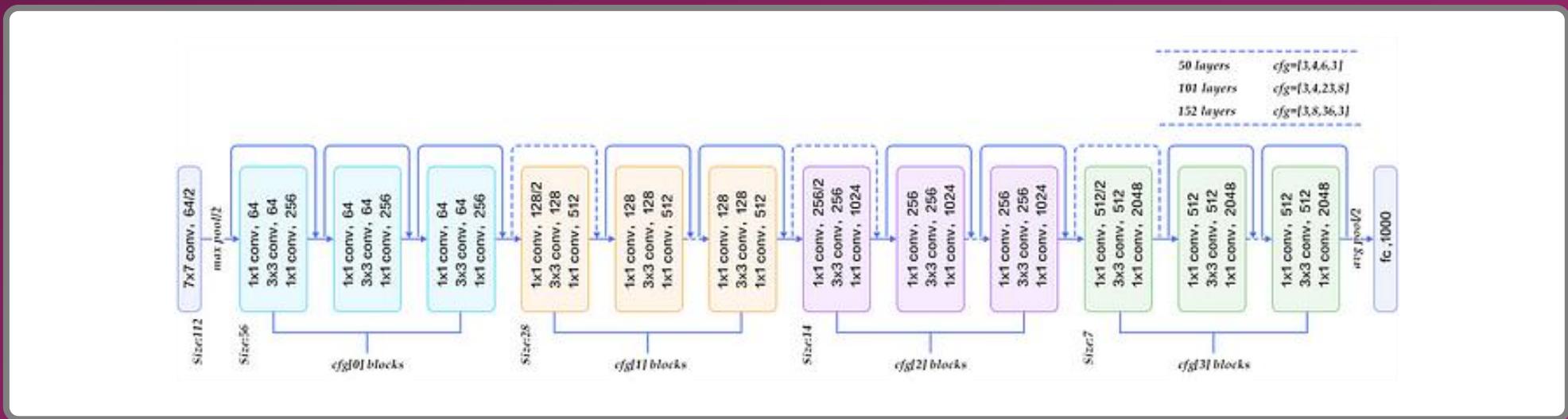
El subcampeón de la competencia ILSVRC 2014 es apodado VGGNet por la comunidad y fue desarrollado por Simonyan y Zisserman. VGGNet consta de 16 capas convolucionales y es muy atractivo debido a su arquitectura muy uniforme. Similar a AlexNet, solo convoluciones de 3x3, pero muchos filtros. **Entrenado en 4 GPU durante 2-3 semanas. Actualmente es la opción preferida en la comunidad para extraer características de las imágenes.** La configuración de peso de VGGNet está disponible públicamente y se ha utilizado en muchas otras aplicaciones y desafíos como un extractor de características de referencia. Sin embargo, VGGNet consta de 138 millones de parámetros, lo que puede ser un poco difícil de manejar.



Evolución de Modelos CNN

ResNet (2015)

Por fin, en ILSVRC 2015, la llamada Red Neural Residual (ResNet) de Kaiming He et al. presentó una arquitectura novedosa con "conexiones saltadas" y características de normalización de lotes pesados. Estas conexiones de salto también se conocen como unidades cerradas o unidades recurrentes cerradas y tienen una gran similitud con los elementos exitosos recientes aplicados en RNN. Gracias a esta técnica, pudieron entrenar una NN con 152 capas y aún así tener una complejidad menor que VGGNet. **Logra una tasa de error entre los 5 primeros del 3,57 %, lo que supera el rendimiento a nivel humano en este conjunto de datos.**



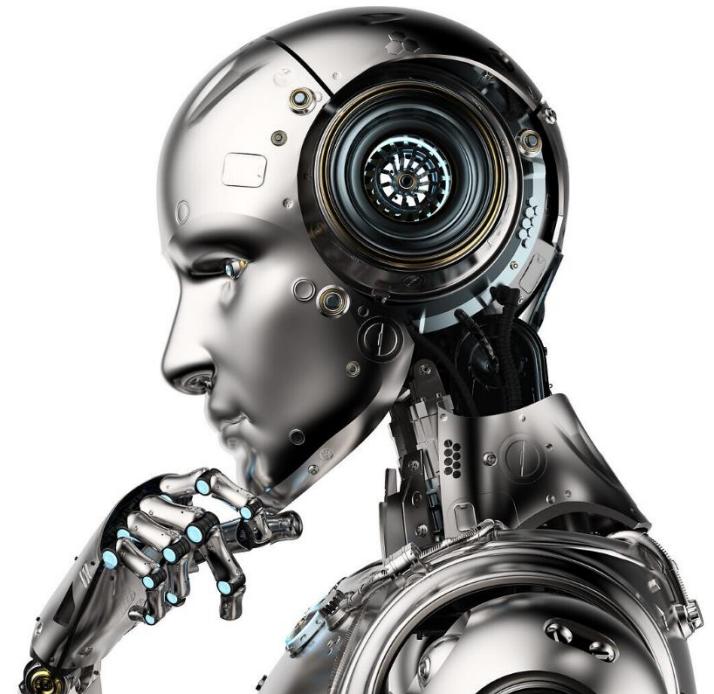
Evolución de Modelos CNN

Tabla de Resumen.

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

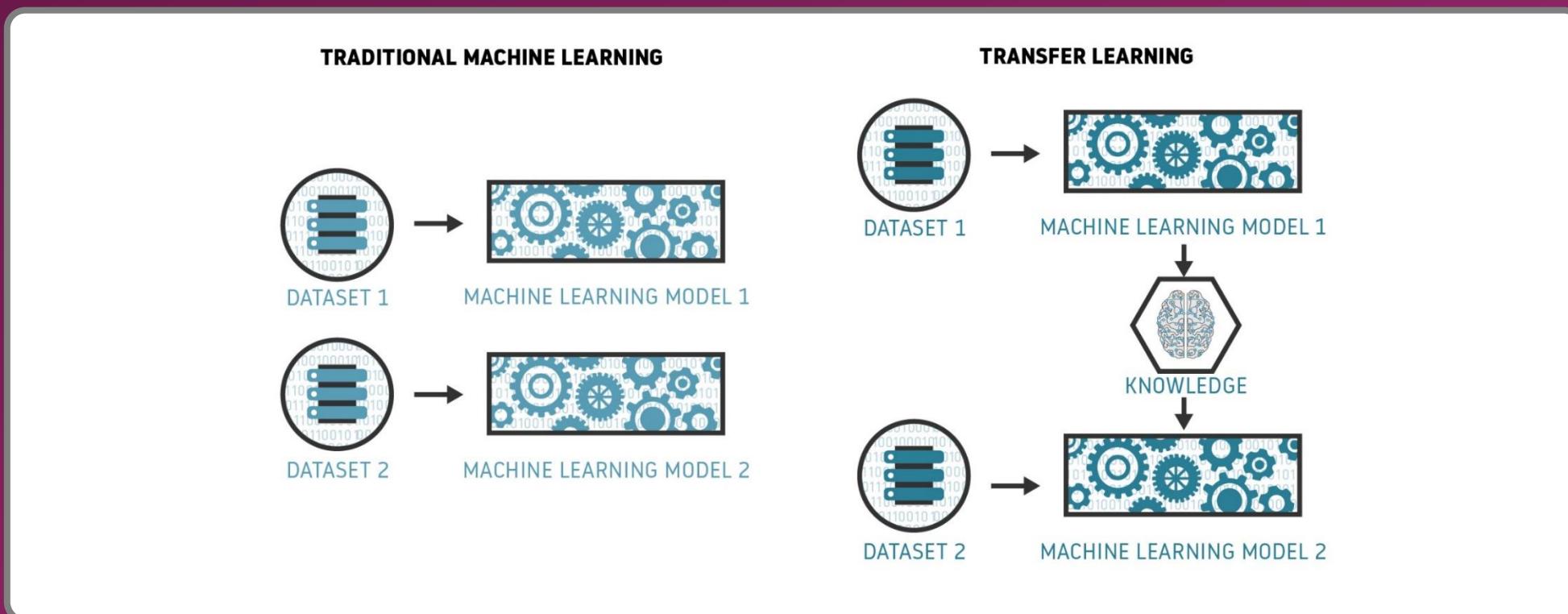
Desafíos de Deep Learning

- Algunos desafíos que enfrentamos cuando entrenamos un modelo Deep Learning:
- Requieren MUCHOS, MUCHOS, MUCHOS datos, lo que, si su modelo también está supervisado, significa que necesita muchos datos etiquetados también.
- Etiquetar muestras de datos es muy tedioso y requiere mucho tiempo.
- El proceso de entrenamiento consume muchos recursos computacionales, dada la complejidad de los algoritmos y la gran cantidad de datos requeridos.
- A pesar de la alta precisión de los algoritmos de última generación, se basan en conjuntos de datos muy específicos y sufren grandes pérdidas de rendimiento cuando se introducen nuevos patrones y casos en escenarios de operaciones reales.



Transfer Learning

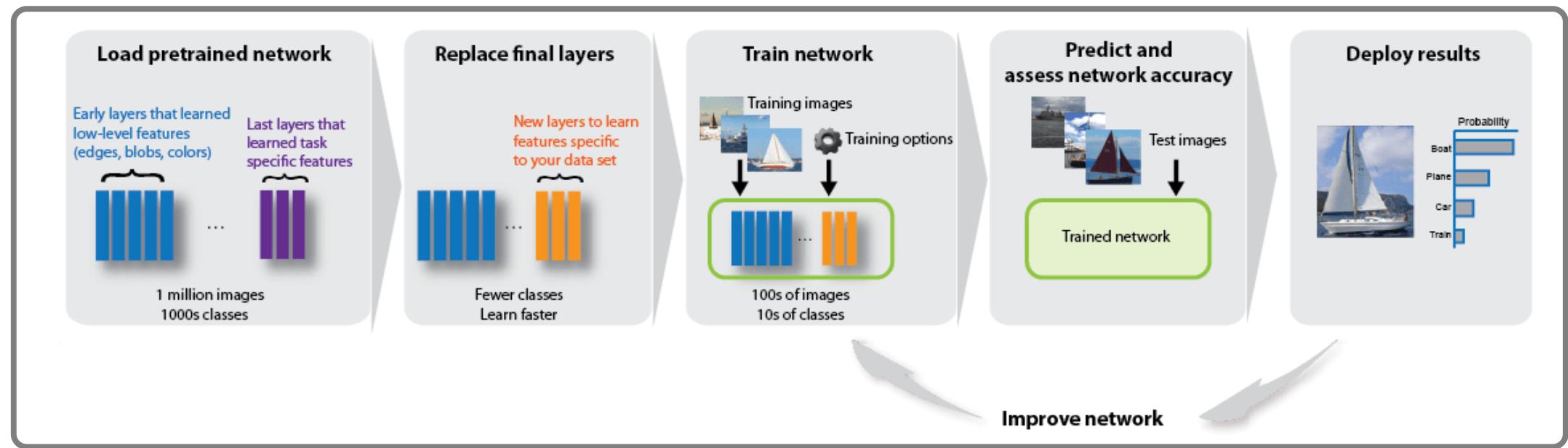
El aprendizaje de transferencia es un concepto emocionante que tiene como objetivo volver a desarrollar la idea tradicional de que los modelos de aprendizaje automático deben reconstruirse desde cero, incluso cuando el nuevo caso de estudio es solo un cambio de distribución de características (variables de dominio). Con el aprendizaje por transferencia, **en lugar de tener que entrenar su modelo una y otra vez, podría usar el conocimiento adquirido para una tarea para resolver otras relacionadas.**



Transfer Learning

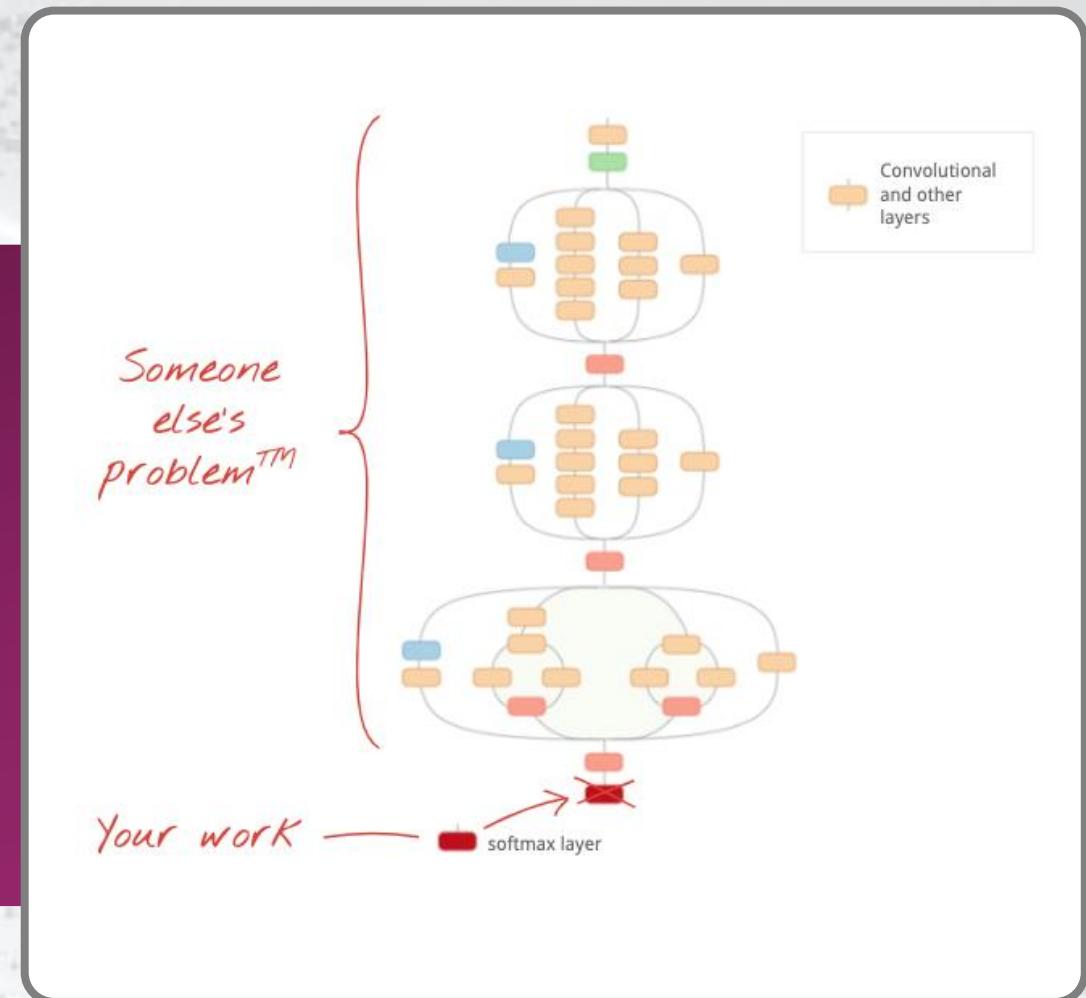
Por ejemplo, **GoogLeNet** ha sido entrenado en más de un millón de imágenes y puede clasificar imágenes en 1000 categorías de objetos (como teclado, taza de café, lápiz y muchos animales). La red ha aprendido ricas representaciones de características para una amplia gama de imágenes. La red toma una imagen como entrada y genera una etiqueta para el objeto en la imagen junto con las probabilidades para cada una de las categorías de objetos.

El **aprendizaje por transferencia** se usa comúnmente en aplicaciones de aprendizaje profundo. Puede tomar una red previamente entrenada y usarla como punto de partida para aprender una nueva tarea. Ajustar una red con transferencia de aprendizaje suele ser mucho más rápido y más fácil que entrenar una red con pesos inicializados aleatoriamente desde cero. Puede transferir rápidamente las funciones aprendidas a una nueva tarea utilizando una cantidad menor de imágenes de entrenamiento.



Transfer Learning

- Con el **aprendizaje por transferencia**, se beneficia tanto de las arquitecturas de redes neuronales convolucionales avanzadas desarrolladas por los mejores investigadores como del entrenamiento previo en un enorme conjunto de datos de imágenes.
- Usando una red neuronal convolucional compleja, ya entrenada, como una caja negra, reentrenando solo el cabezal de clasificación. Esto es aprendizaje por transferencia. Más adelante veremos cómo funcionan estos complicados arreglos de capas convolucionales. Por ahora es problema de otro.



TensorFlow Hub

The screenshot shows the TensorFlow Hub homepage. At the top, there's a navigation bar with links for 'Instalación', 'Aprende', 'API', 'Recursos' (which is underlined), 'Comunidad', 'Por qué TensorFlow', a search bar, a language switcher ('Español - A...'), a 'GitHub' link, and a 'Acceder' button.

The main content area has a header 'Hub' and a navigation menu with tabs: 'Descripción general' (which is selected), 'Guía', 'Instructivos', 'API', and 'Modelos'. Below this, there's a banner with the text '¡Reserva! TensorFlow regresa a Google I/O el 10 de mayo' and a 'Regístrate ahora' button.

The central text on the page reads: 'TensorFlow Hub es un repositorio de modelos de aprendizaje automático entrenados.'

To the left, there are three sections with icons and links:

- Ver la guía** (with a gear icon): Obtén información sobre cómo usar TensorFlow Hub y cómo funciona.
- Ver los instructivos** (with a stack of books icon): Los instructivos muestran ejemplos de extremo a extremo que usan TensorFlow Hub.
- Ver los modelos** (with a document icon): Encuentra modelos entrenados de TF, TFLite y TF.js para tu caso de uso.

To the right, there's a code snippet demonstrating how to use TensorFlow Hub:

```
!pip install --upgrade tensorflow_hub  
  
import tensorflow_hub as hub  
  
model = hub.KerasLayer("https://tfhub.dev/googlennlm-en-dim128/2")  
embeddings = model(["The rain in Spain.", "falls",  
                   "mainly", "In the plain!"])  
  
print(embeddings.shape) # (4, 128)
```

At the bottom of the page is a large URL box containing the address <https://www.tensorflow.org/hub>.

TensorFlow Hub

TensorFlow

Instalación Aprende ▾ API ▾ Recursos ▾ Comunidad ▾ Por qué TensorFlow ▾ Buscar

Español – A... GitHub Acceder

Descripción general Guía Instructivos API Modelos ▾

Busca modelos entrenados de la comunidad de TensorFlow en [TFHub.dev](#)

Class Label

BERT

$E_{[CLS]}$ E_1 E_2 ... E_N

$I_{[CLS]}$ Tok 1 Tok 2 ... Tok N

Single Sentence

BERT

Utiliza BERT para tareas de PLN, incluidas la clasificación de texto y la respuesta a preguntas.

[Consulta el modelo ↗](#)

Detección de objetos

Utiliza el modelo Faster R-CNN Inception ResNet V2 640x640 para detectar objetos en imágenes.

[Consulta el modelo ↗](#)

Transferencia de estilo

Transfiere el estilo de una imagen a otra con el modelo de transferencia de estilo de imagen.

[Consulta el modelo ↗](#)

Food V1.1

Type	Score
Sachertorte	0.821
Black Forest gateau	0.028
Devil's food cake	0.023
Chocolate brownie	0.014

Clasificador de alimentos en el dispositivo

Usa este modelo de TFLite para clasificar fotos de comida en un dispositivo móvil.

[Consulta el modelo ↗](#)

<https://www.tensorflow.org/hub>

Faster R-CNN with Inception Resnet v2





Dudas y consultas



Fin de la Presentación



KIBERNUM