ResNet-50

Deep Residual Learning for Image Recognition

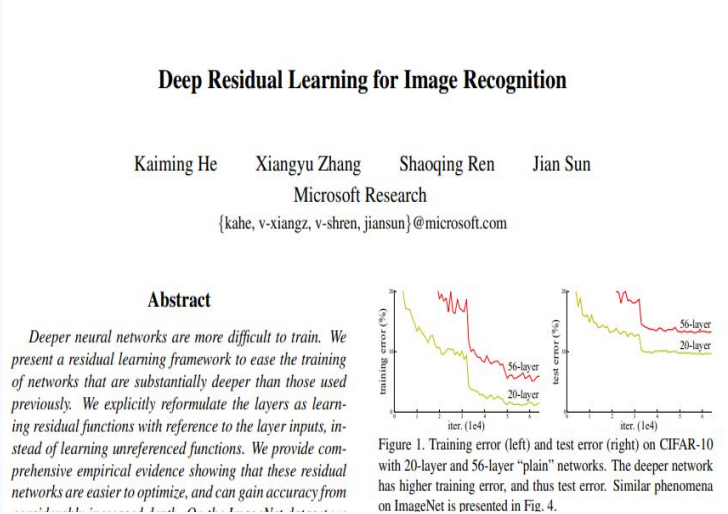
Table of contents

01

Introducción

Introducción

- Resnet50 se refiere a Residual Network que utiliza 50 capas convolucionales.
- Introducido en 2015 por He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian en el paper “Deep Residual Learning for Image Recognition”.
- Demostrar que estas redes residuales son más fáciles de optimizar y pueden ganar precisión con una profundidad considerablemente mayor.



Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

¿Aprender mejores redes es tan fácil como apilar más capas?

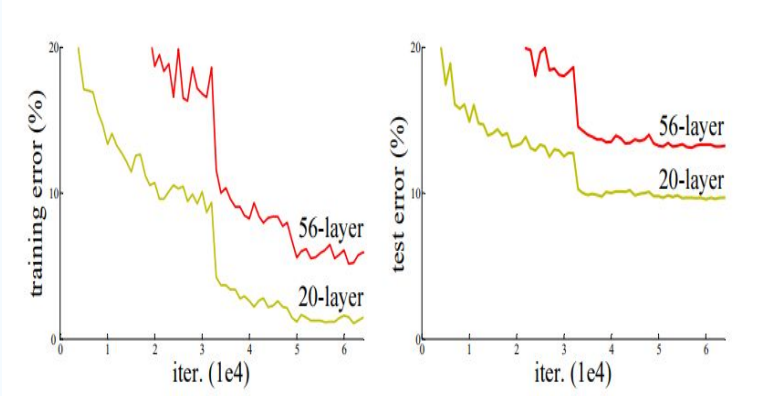
- Agregar más capas introduce algunos problemas durante el entrenamiento como:

Problema de los gradientes desvanecidos/explosivos

- Dificulta la convergencia desde el principio.
- Solución:
  - Inicialización normalizada
  - Capas de normalización intermedias

Problema de degradación

- Las redes más profundas son capaces de empezar a converger.
- Añadir más capas al modelo conduce un mayor error de entrenamiento.
- No es causada por el overfitting.



Dataset: CIFAR - 10

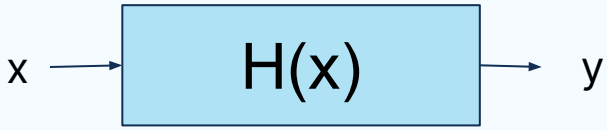
02

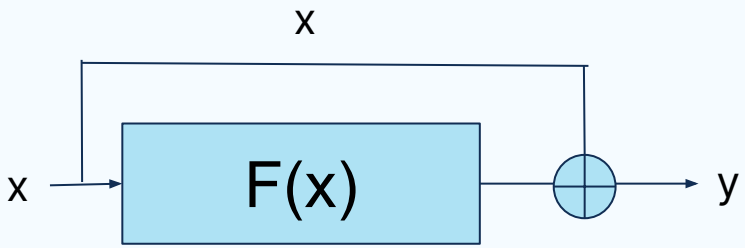
Arquitectura

Residual Learning

- H(x): mapeo subyacente que debe ser ajustada por unas cuantas capas apiladas
- x: Entradas a la primera de estas capas
- En lugar de esperar que las capas apiladas aproximen H(x), dejamos explícitamente que estas capas aproximen una función residual:

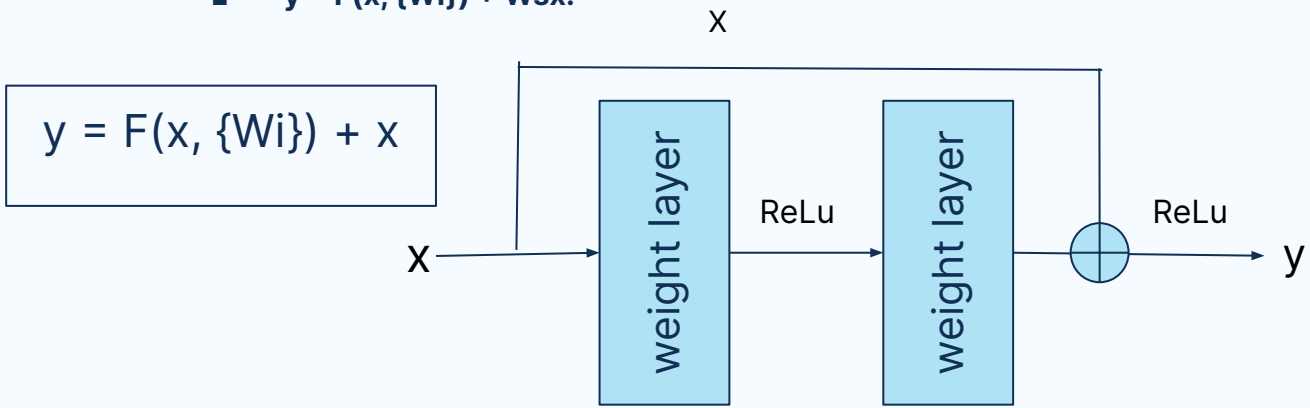
$$F(x) := H(x) - x$$





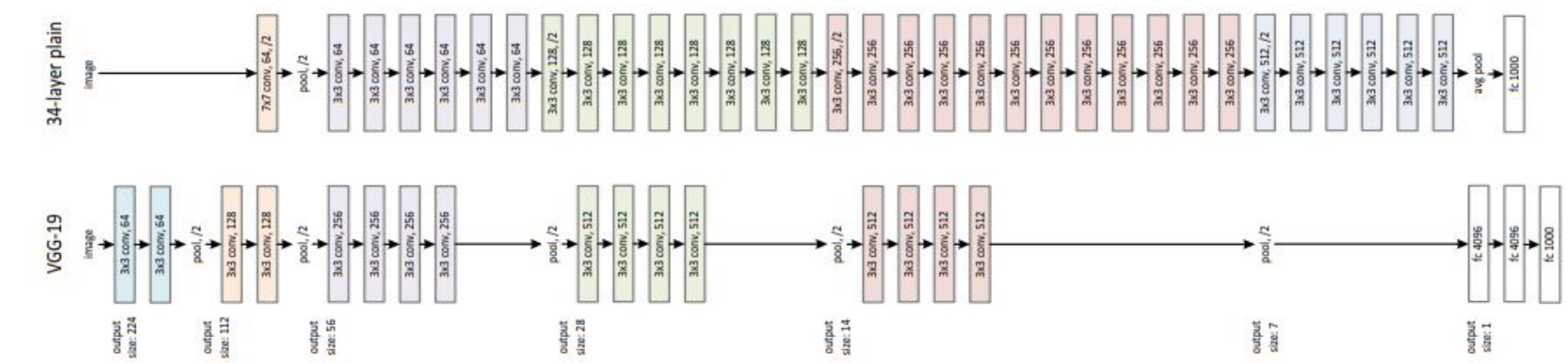
Residual Block

- **F(x, {Wi})**: Representa el mapeo residual que debe aprenderse.
- La operación F + x se realiza mediante una conexión abreviada y una suma de elementos.
- Dimensiones diferentes (entrada/salida):
  - Realizar una proyección lineal Ws por las conexiones de acceso directo para igualar las dimensiones:
    - **y = F(x, {Wi}) + Wsx.**



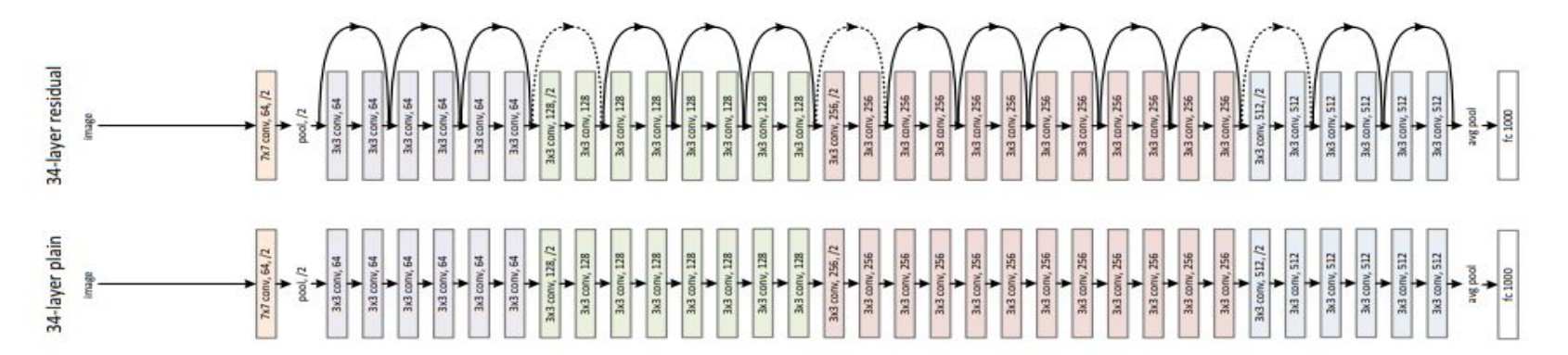
Arquitectura

- **PLAIN NETWORK:**
- Las capas convolucionales tienen en su mayoría filtros de 3×3
  - Para el mismo tamaño del mapa de características de salida, las capas tienen el mismo número de filtros.
  - Si el tamaño del mapa de características se reduce a la mitad, el número de filtros se duplica para preservar la complejidad temporal por capa.
- El modelo básico de 34 capas tiene 3.600 millones de FLOPs, en cambio, VGG-19 tiene 19.600 millones de FLOPs.



Arquitectura

- **RESIDUAL NETWORK**:
- Inserta conexiones de atajo que convierten la red en su versión residual equivalente.
- Cuando hay diferencia en las dimensiones, existen dos opciones:
    - El atajo sigue realizando el mapeo de identidad, rellena con entradas cero adicionales para aumentar la dimensionalidad.
    - Se utiliza la ecuación del atajo de proyección, igualando las dimensiones (mediante convoluciones 1×1).

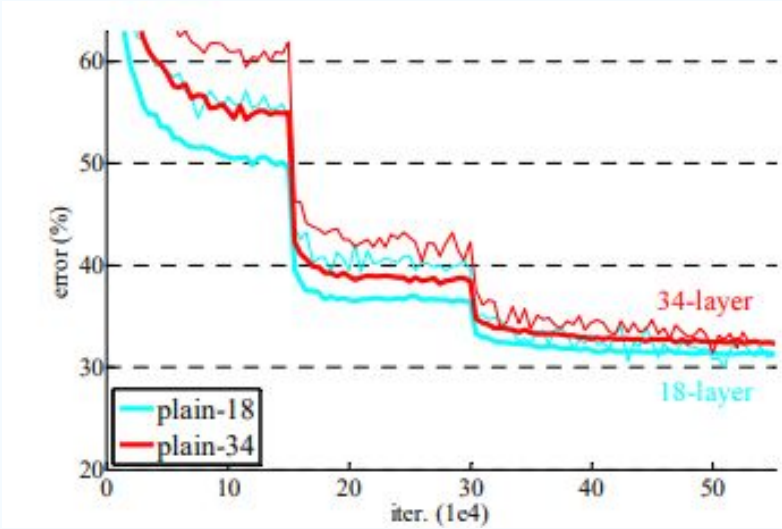


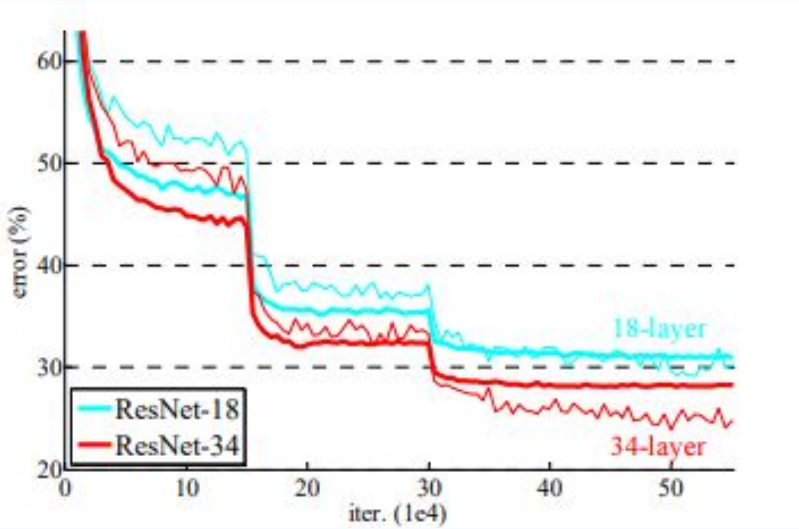
Implementación

La arquitectura ResNet incluye los siguientes elementos:

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Resultados


error (%)
iter. (1e4)
plain-18
plain-34
34-layer
18-layer


error (%)
iter. (1e4)
ResNet-18
ResNet-34
18-layer
34-layer

03 DATASET

IMAGENET

14,197,122 images, 21841 synsets indexed

Home Download Challenges About

Not logged in. Login | Signup

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Competition

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

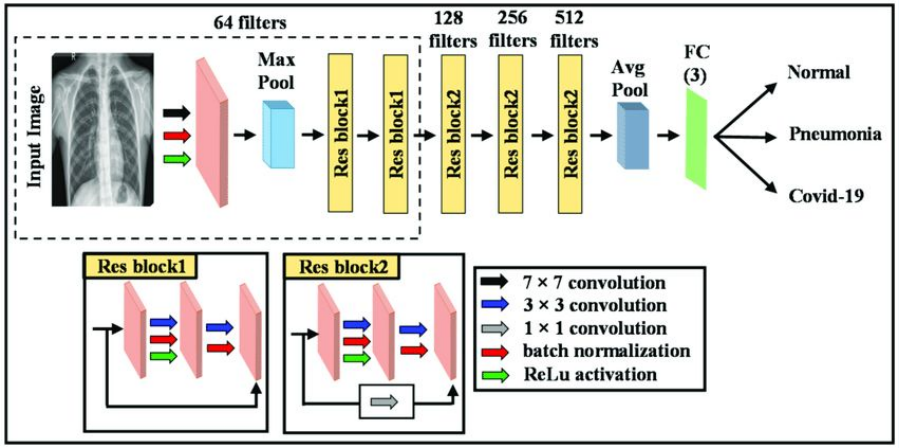
For details about each challenge please refer to the corresponding page.

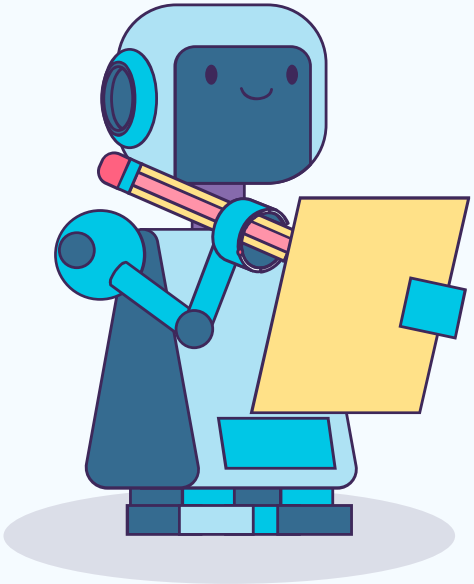
- ILSVRC 2017
- ILSVRC 2016
- ILSVRC 2015
- ILSVRC 2014
- ILSVRC 2013
- ILSVRC 2012
- ILSVRC 2011
- ILSVRC 2010

04

Áreas de aplicación

- Clasificación de imágenes
- Detección de objetos
- Segmentación semántica
- Reconocimiento facial
- Procesamiento de imágenes médicas
- Análisis de imágenes en tiempo real





Thanks!