

Redes Convolucionales aplicadas a imágenes

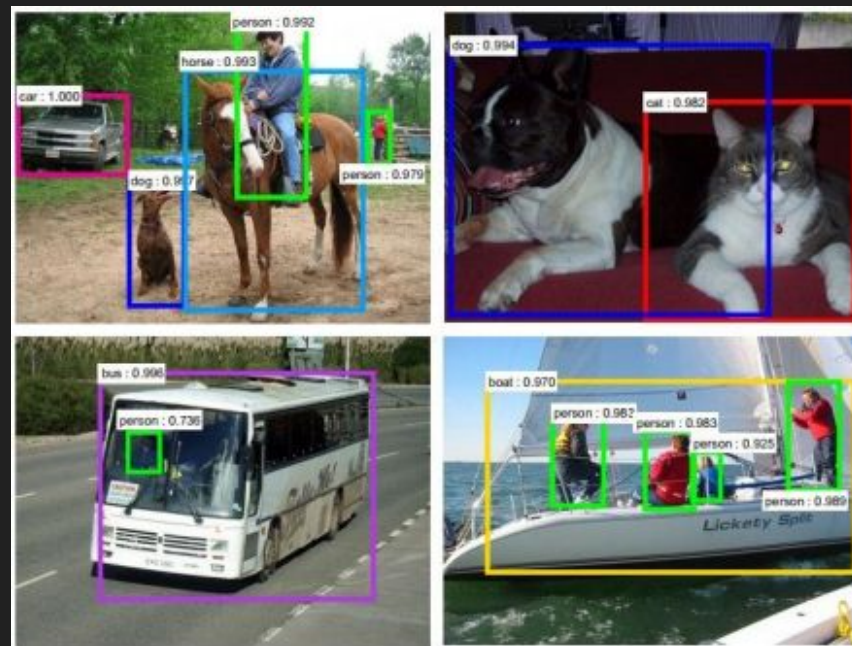
Nicolas Peretti
Pablo Pastore

Casos de uso

Retrieval



Detección



Casos de uso

Segmentación



Vehículos autónomos



Casos de uso

Descripción de imagen



Reconocimiento facial



Casos de uso

Clasificación de taxonomía de galaxias

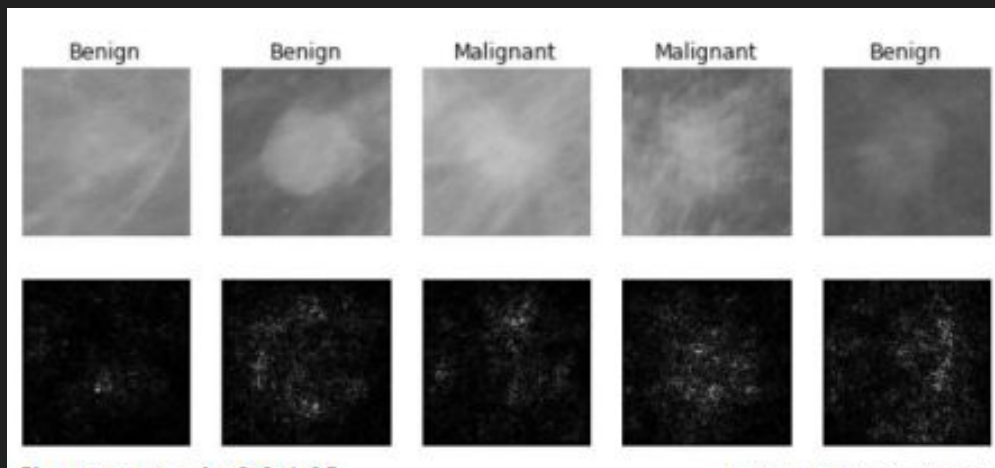


Detección de caminos



Casos de uso

Análisis de mamografía

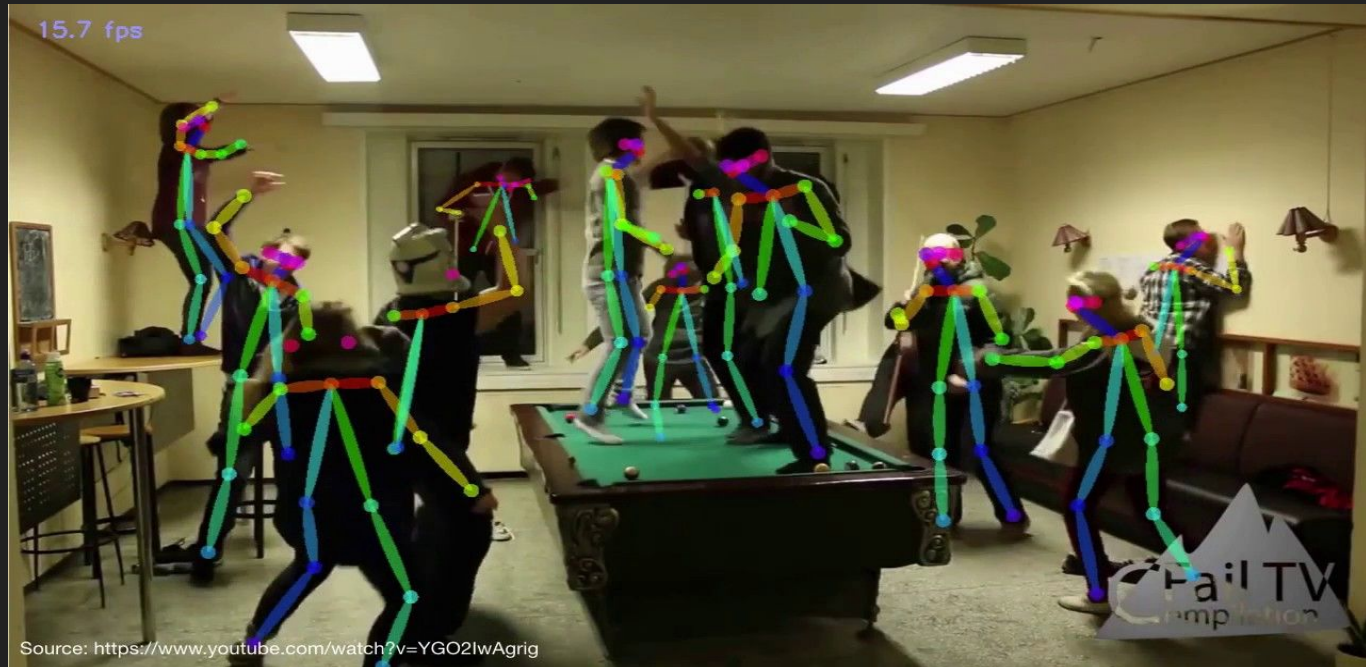


Clasificación de señales de tránsito



Casos de uso

Estimación de pose

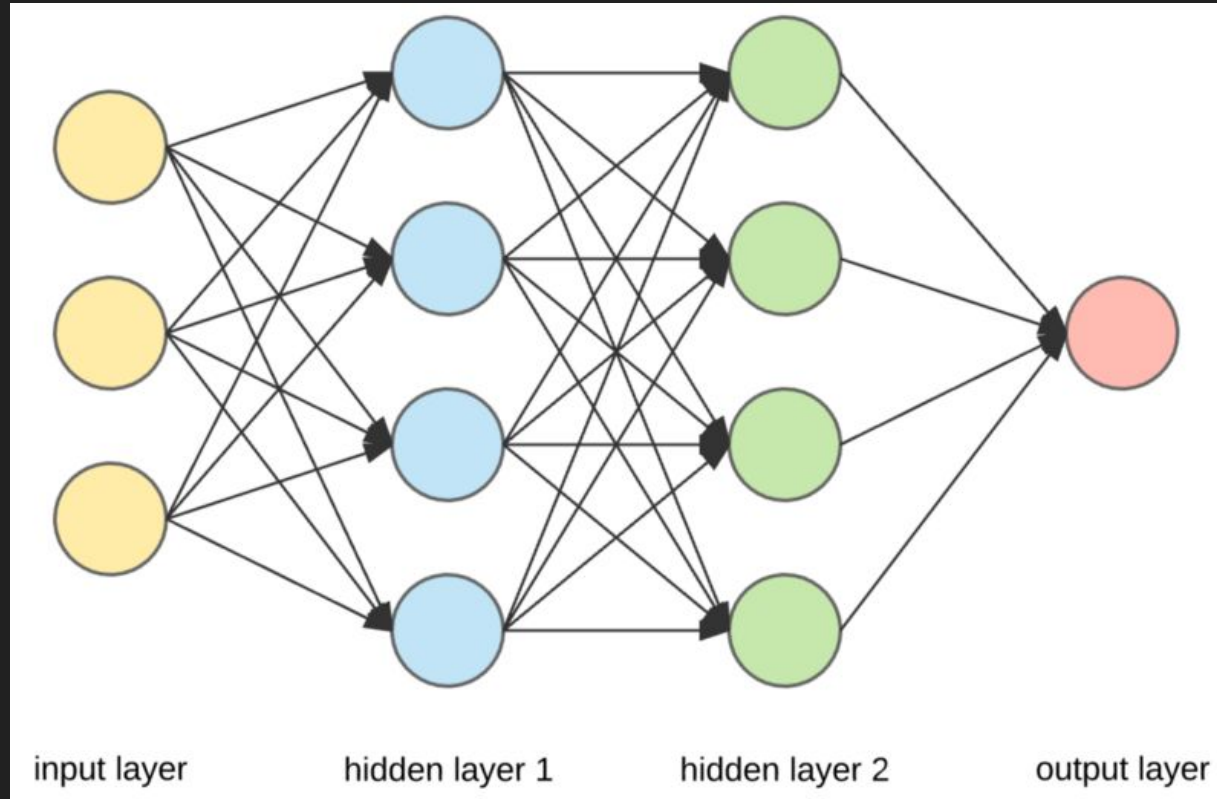


Casos de uso

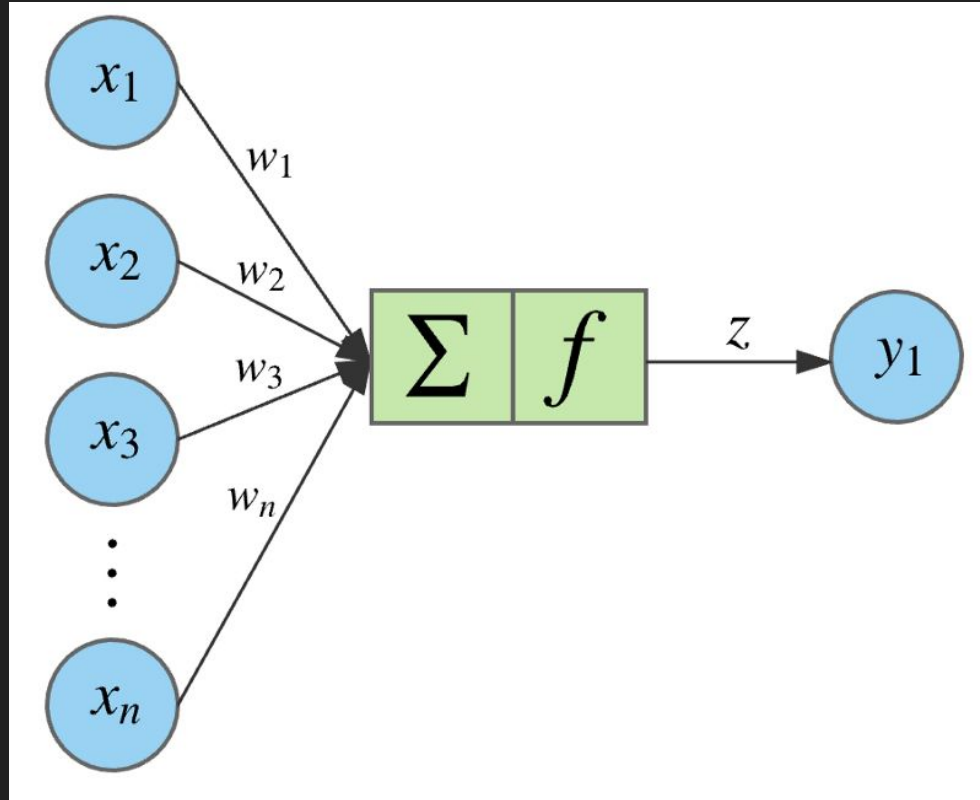
GANs



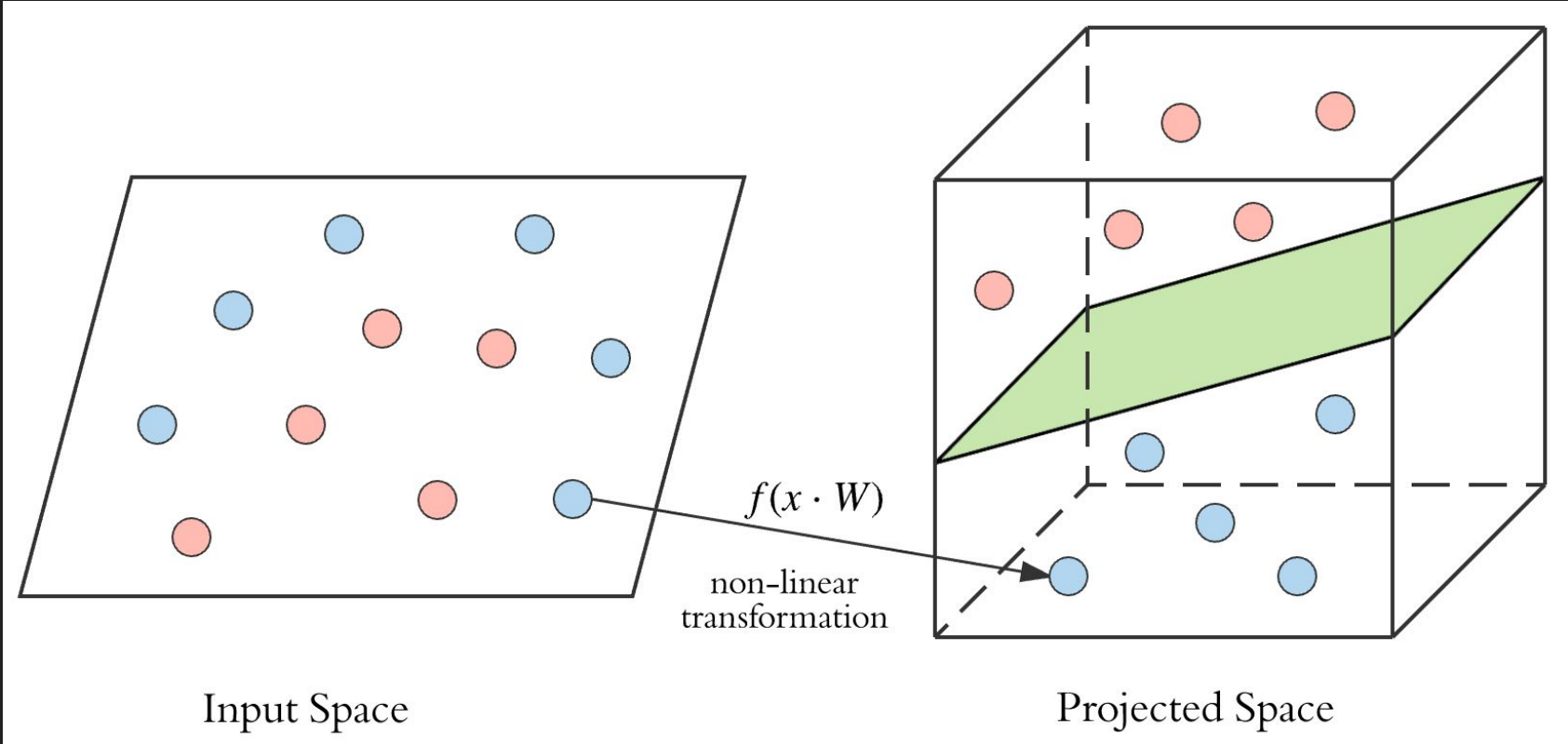
Redes Neuronales



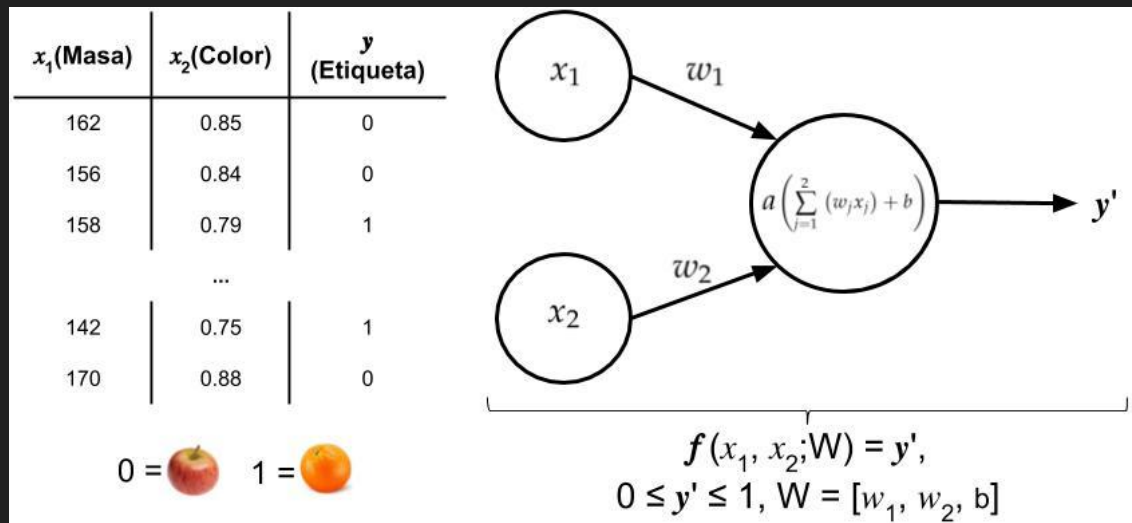
Redes Neuronales



Redes Neuronales



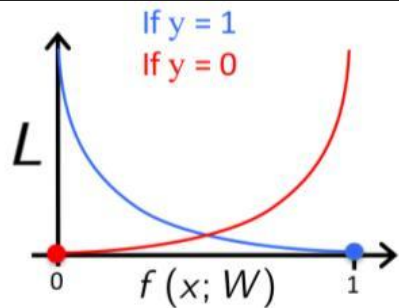
Función de costo



→ Supongamos que deseamos entrenar un clasificador de frutas

→ Debemos evaluar la exactitud del modelo: **Función de Costo**

$$L(x, y; W) = \begin{cases} -\log(f(x; W)) & y = 1 \\ -\log(1 - f(x; W)) & y = 0 \end{cases}$$

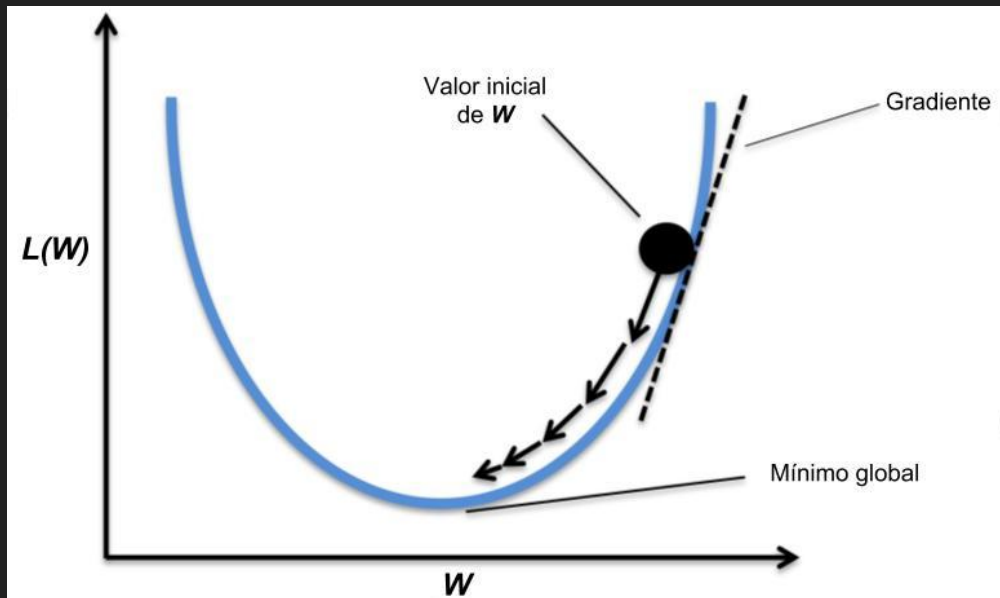


Aprendizaje por descenso del gradiente



Aprendizaje por descenso del gradiente

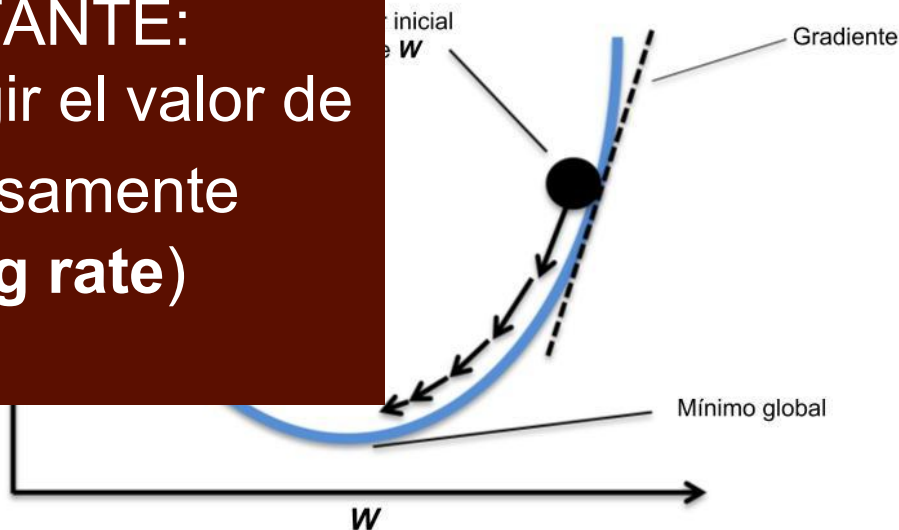
$$W_{t+1} = W_t - \delta \nabla_{W_t} L(W_t)$$



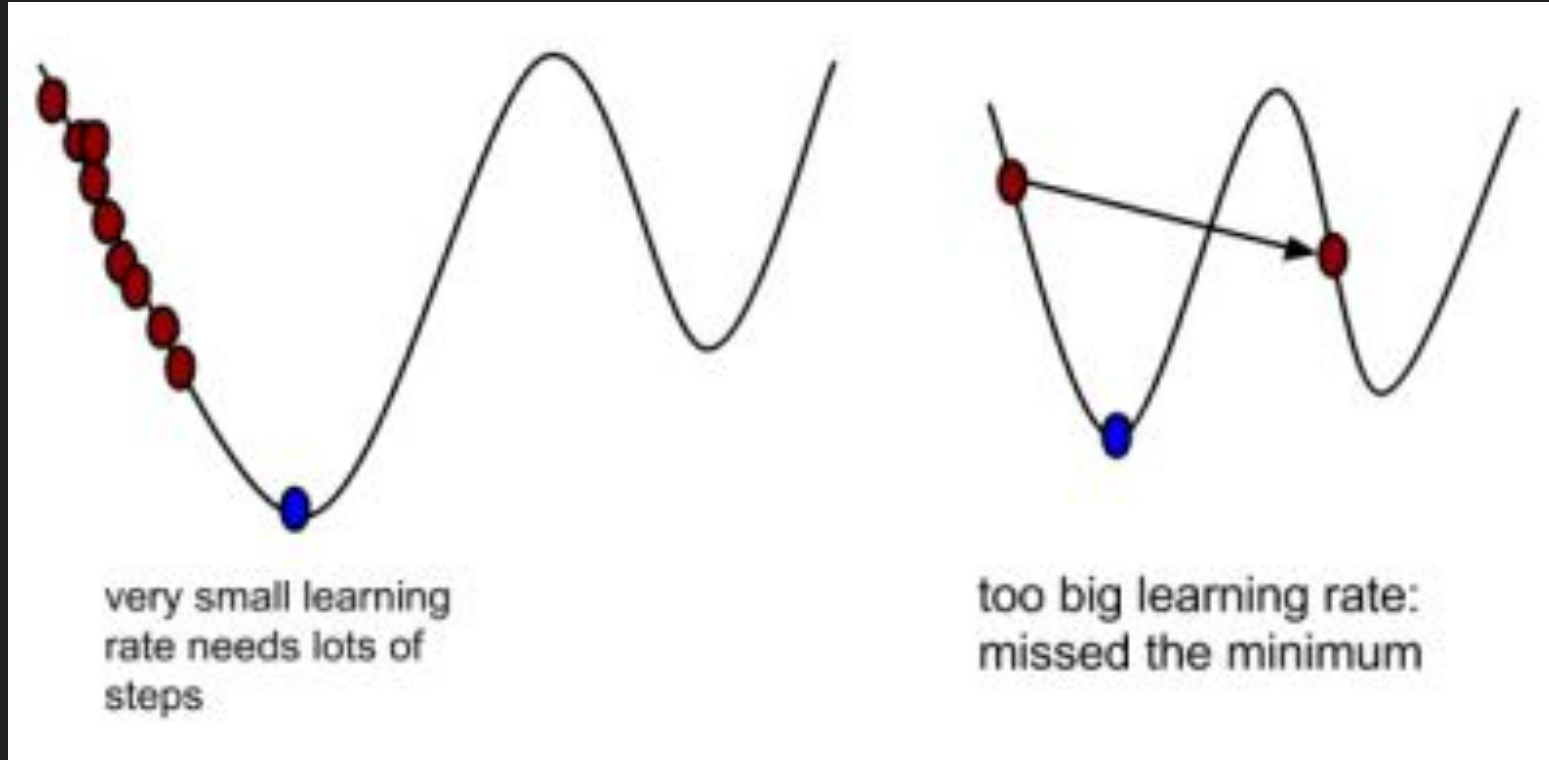
Aprendizaje por descenso del gradiente

$$W_{t+1} = W_t - \delta \nabla_{W_t} L(W_t)$$

IMPORTANTE:
Debemos elegir el valor de δ cuidadosamente
(learning rate)



Aprendizaje por descenso del gradiente



(mini batch) SGD

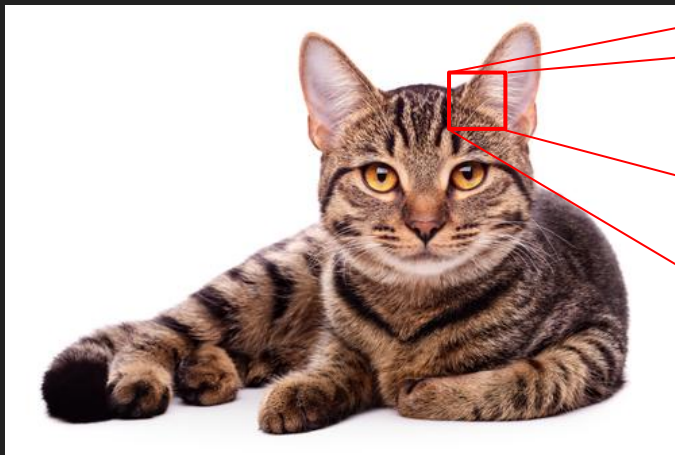
→ Problemas con el algoritmo previo:

- ◆ Costoso en grandes volúmenes de datos.
- ◆ Imposible de ejecutar cuando los datos no entran en memoria.

→ Solución:

- ◆ 1) Seleccionar un subconjunto de nuestro conjunto de entrenamiento.
- ◆ 2) Computar función de costo, computar gradiente y actualizar pesos.
- ◆ 3) Volver al paso (1)

Aprendizaje (automático) de representaciones



54	58	255	8	0	
45	0	78	51	100	74
85	47	34	185	207	21
22	20	148	52	24	147
52	36	250	74	214	278
	158	0	78	51	247
		72	74	136	251
				74	

Lo que la computadora ve

Aprendizaje (automático) de representaciones

Algunos desafíos:



variación de
clases



iluminación



deformación

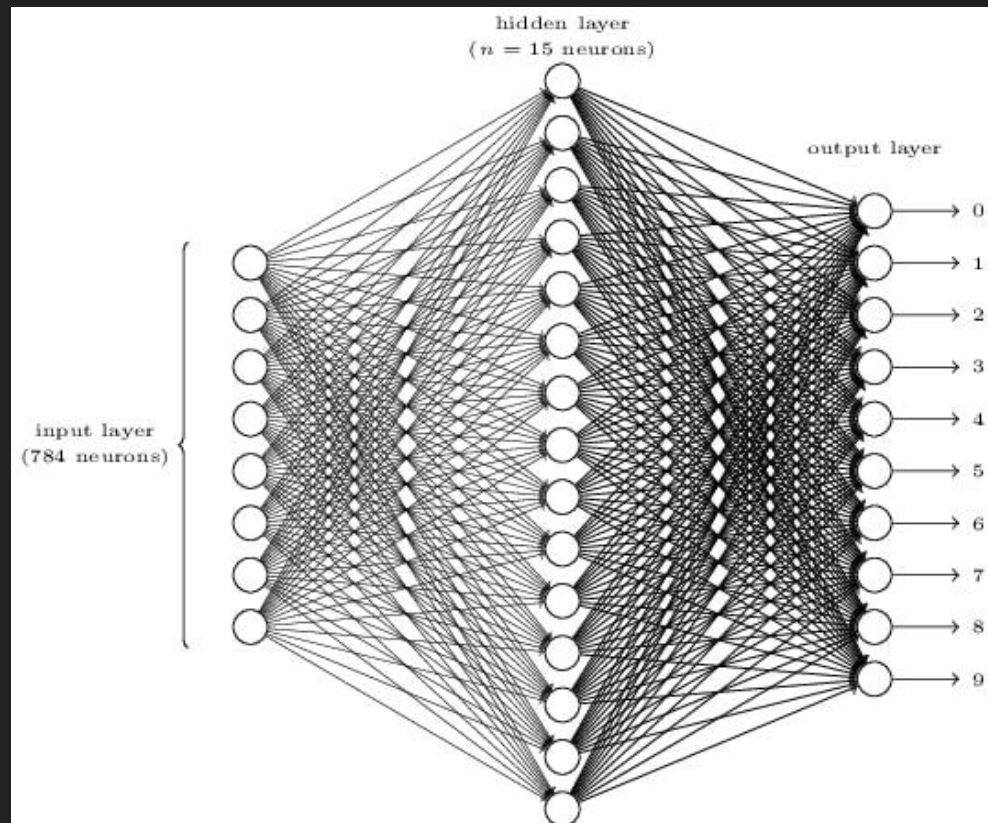


oclusión

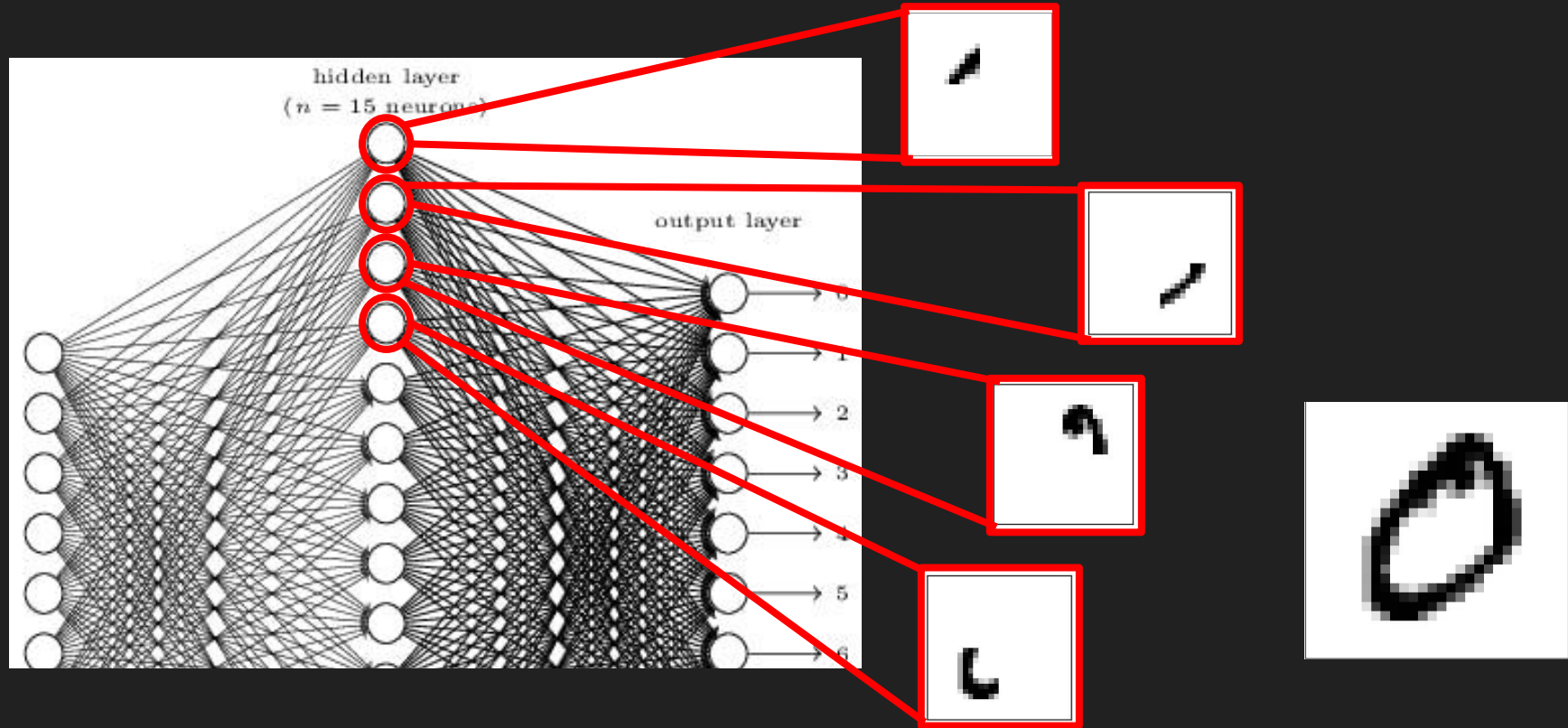
Red neuronal como clasificador de dígitos



- Imágenes de dígitos de 28 x 28 píxeles
- $28 \times 28 = 784$ píxeles
- 784 valores de entrada a nuestra red

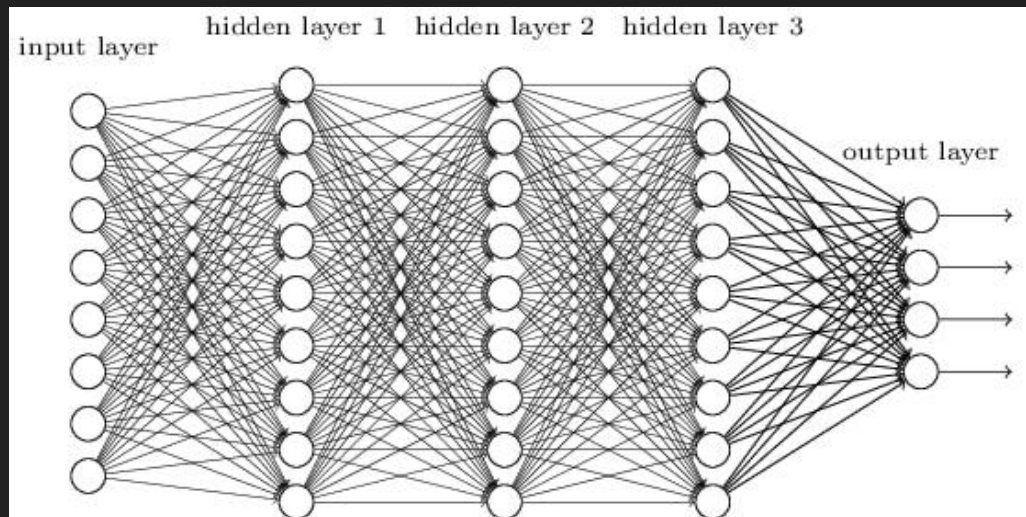


Red neuronal como clasificador de dígitos



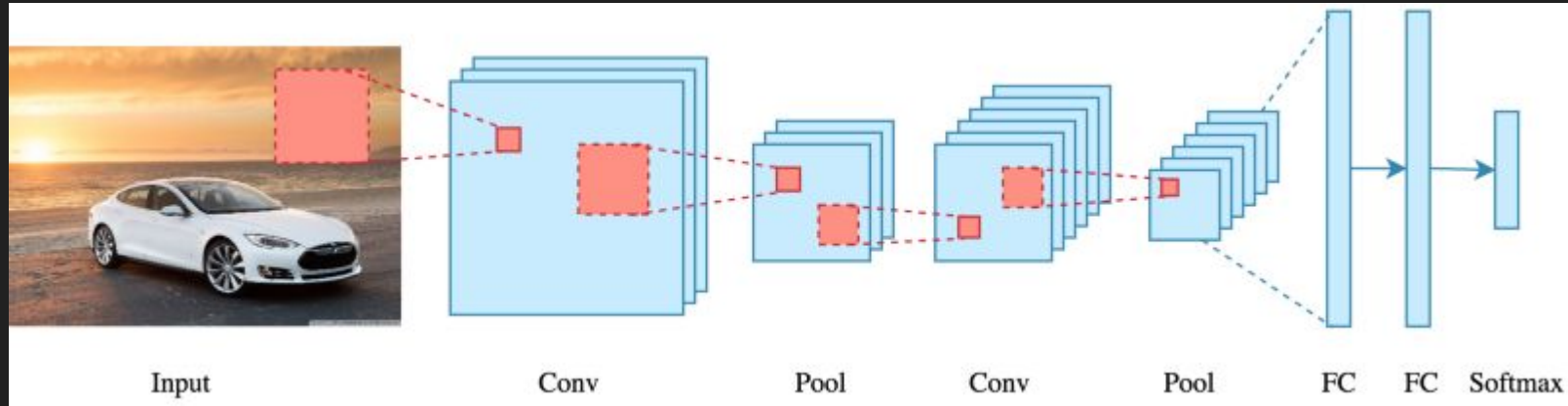
Red neuronal en imágenes

- No tiene en cuenta estructura espacial de la imagen.
- Deseamos construir redes más profundas
 - ◆ Crece la cantidad de parámetros
 - ◆ En la práctica: 3 capas mejor que 2 capas. 4, 5, etc capas no siempre ayudan

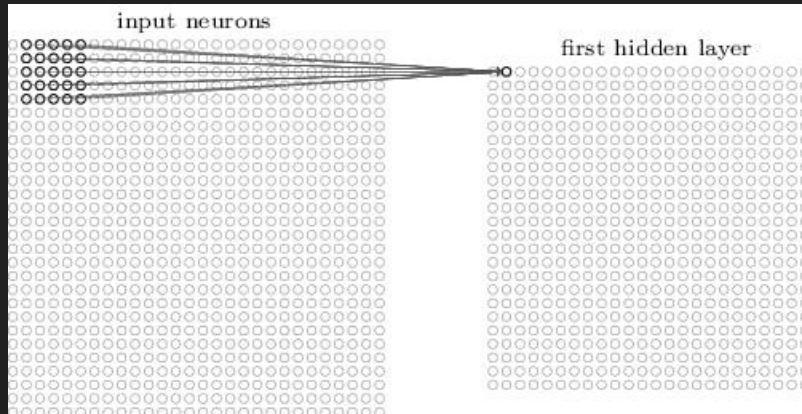
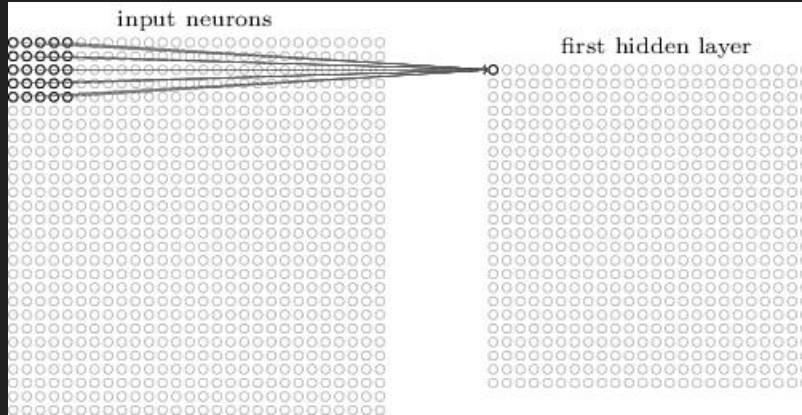


Redes convolucionales (CNN)

- Approach red neuronal: trabajar con toda la imagen a la vez
- Approach convolucional: inspeccionar la imagen de a pequeñas partes



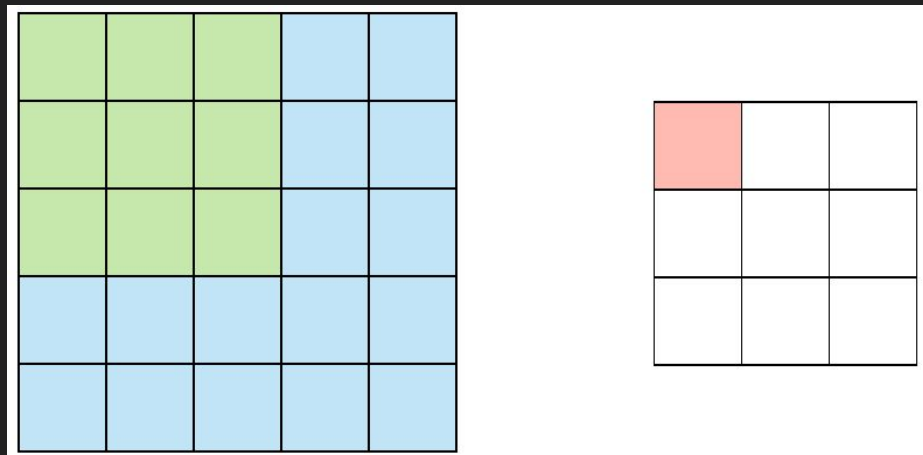
CNN: Convolución



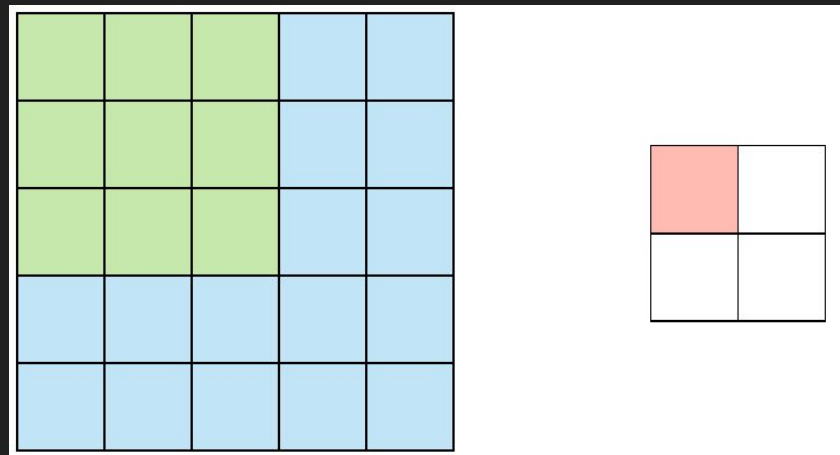
- Campo receptivo local (5x5 pixeles)
- Pesos Compartidos
- Stride = 1

CNN: Convolución

Resultados con diferentes valores de **stride**:

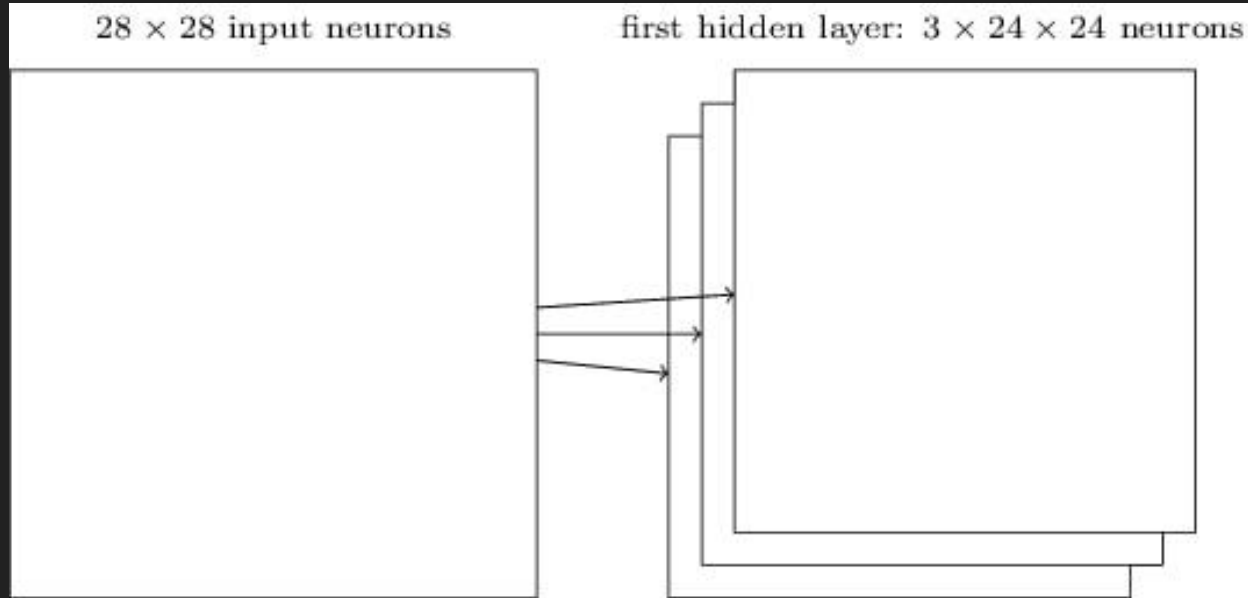


Stride = 1



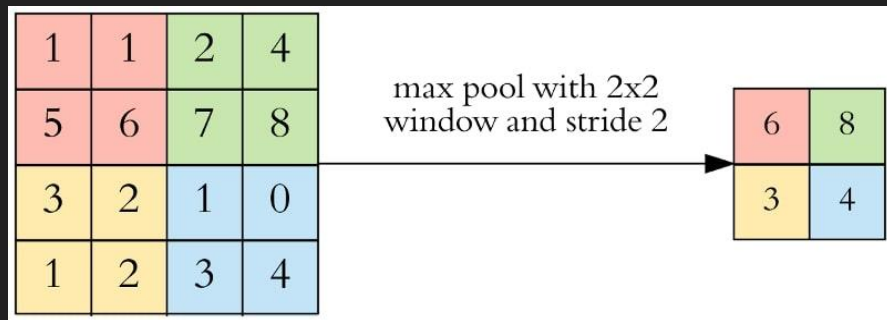
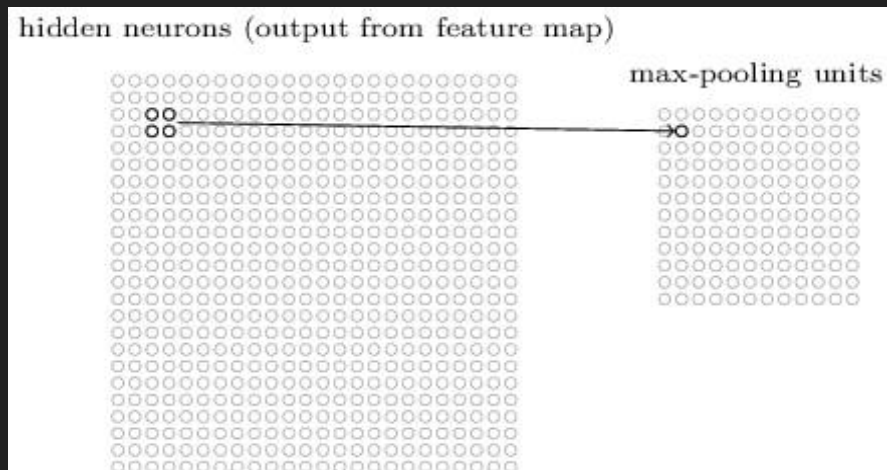
Stride = 2

CNN: Convolución

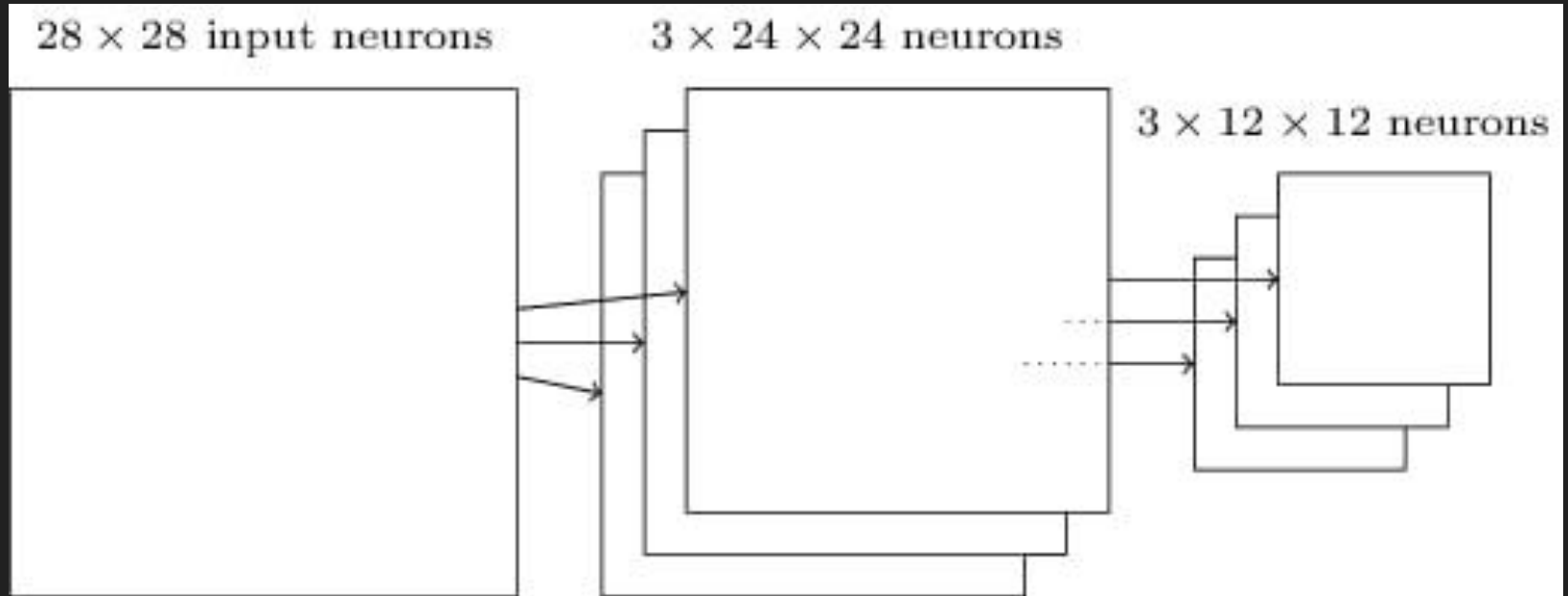


Convolución con tres filtros (o kernels)

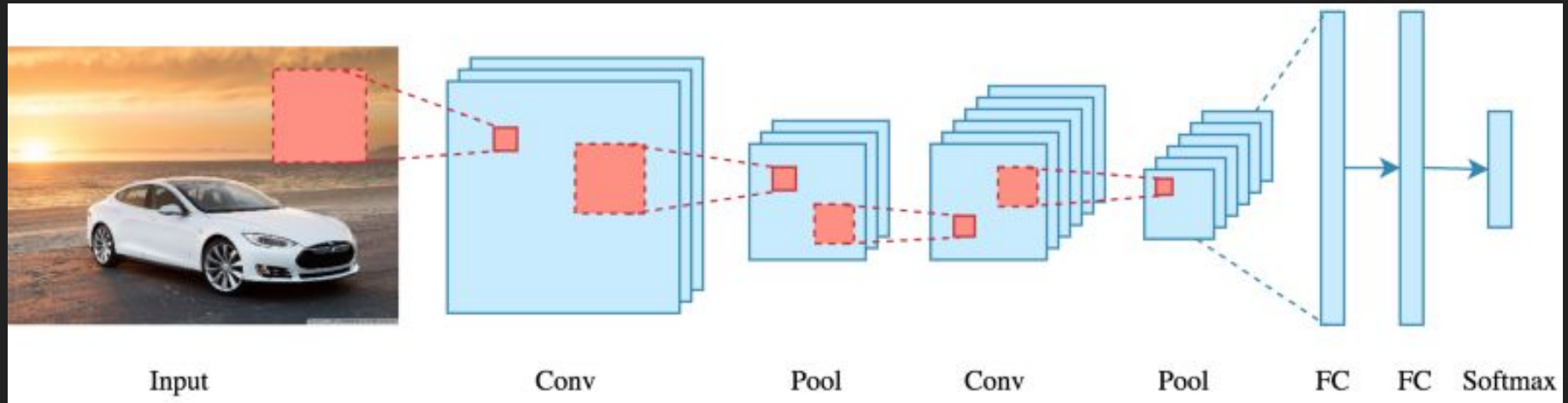
CNN: Pooling



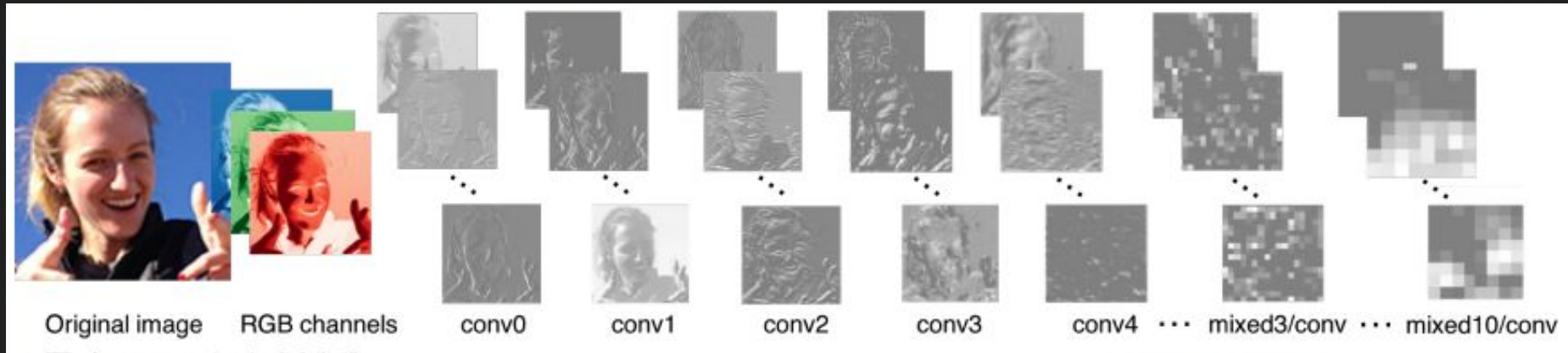
CNN: Conv + Pool



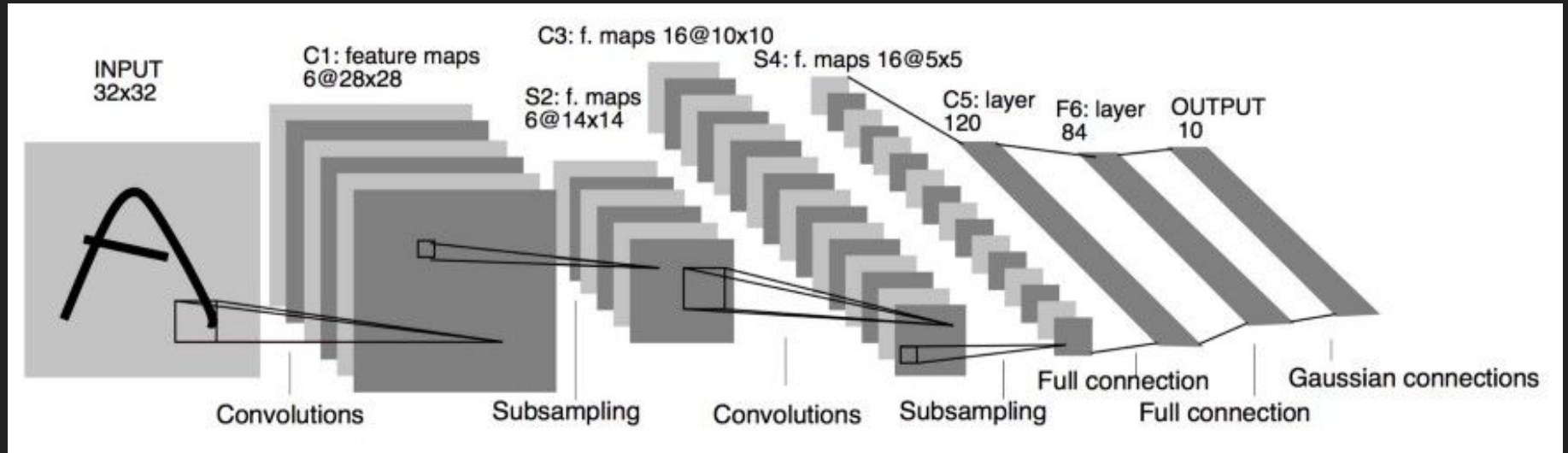
CNN: Arquitectura completa



CNN: Visualización

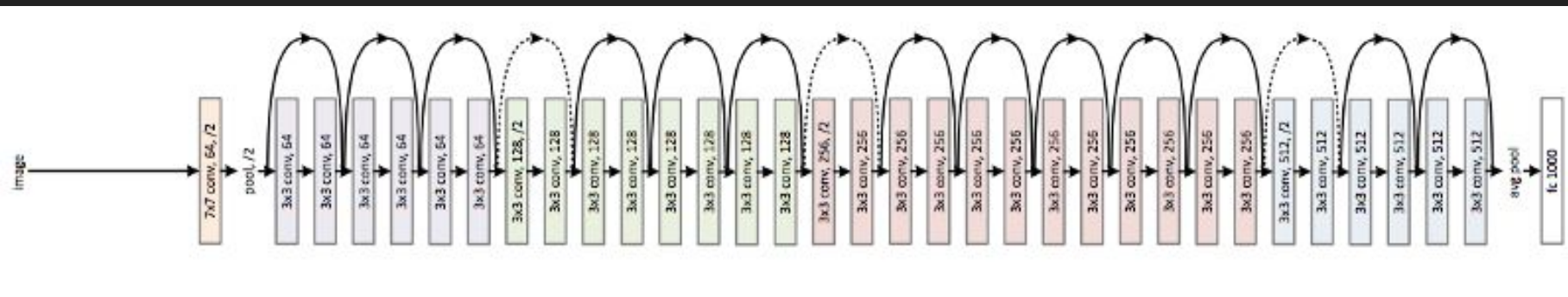


CNN: LeNet



CNN: ResNet

- 152 capas (“ultra-deep”)
- “mantener” el input original
- 8 GPUs (2-→3 semanas)



CNN: ResNet

- 152 capas (“ultra-deep”)
- “mantener” el input original
- 8 GPUs (2-→3 semanas)

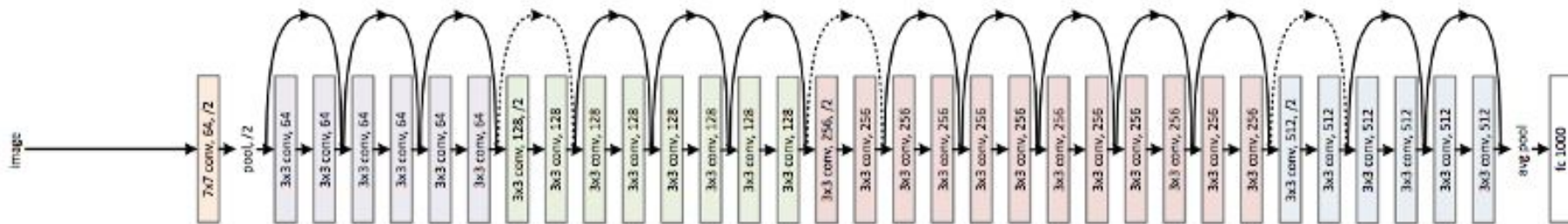
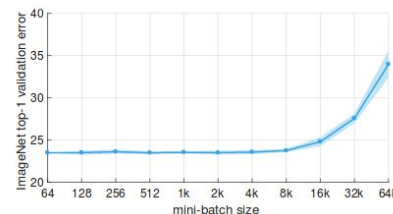
Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour

Priya Goyal Piotr Dollár Ross Girshick Pieter Noordhuis
Lukasz Wesolowski Aapo Kyrola Andrew Tulloch Yangqing Jia Kaiming He

Facebook

Abstract

Deep learning thrives with large neural networks and large datasets. However, larger networks and larger datasets result in longer training times that impede research and development progress. Distributed synchronous SGD offers a potential solution to this problem by dividing SGD minibatches over a pool of parallel workers. Yet to make this scheme efficient, the per-worker workload must be large, which implies nontrivial growth in the SGD mini-batch size. In this paper, we empirically show that on the

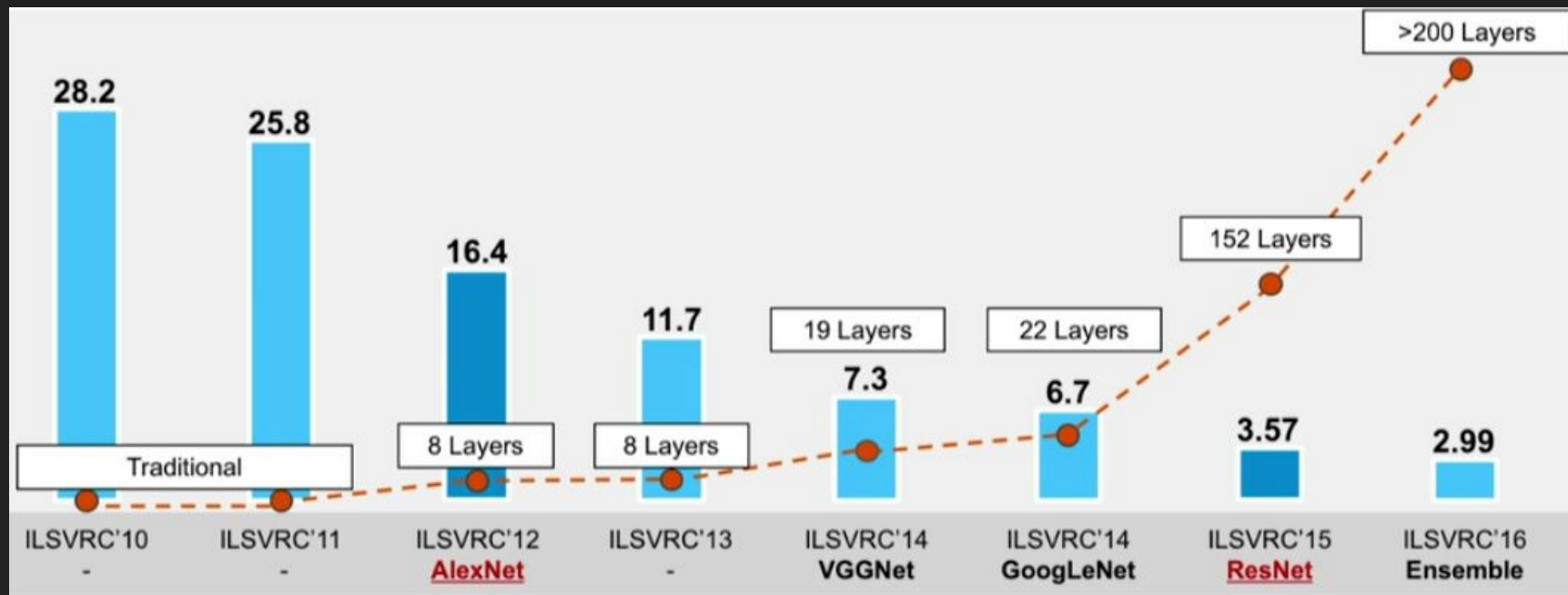




ImageNet Large Scale Visual Recognition Challenges



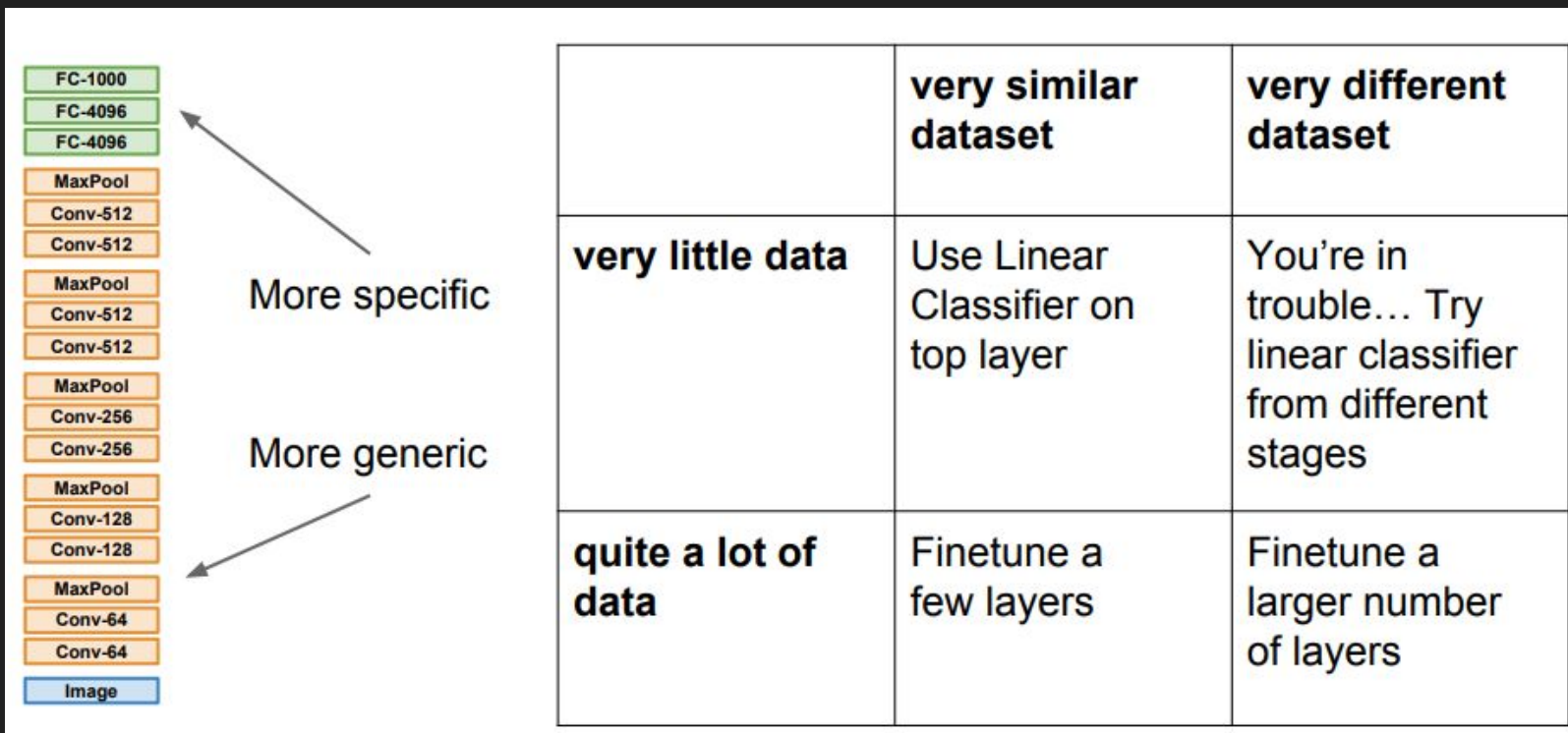
CNN: Imagenet



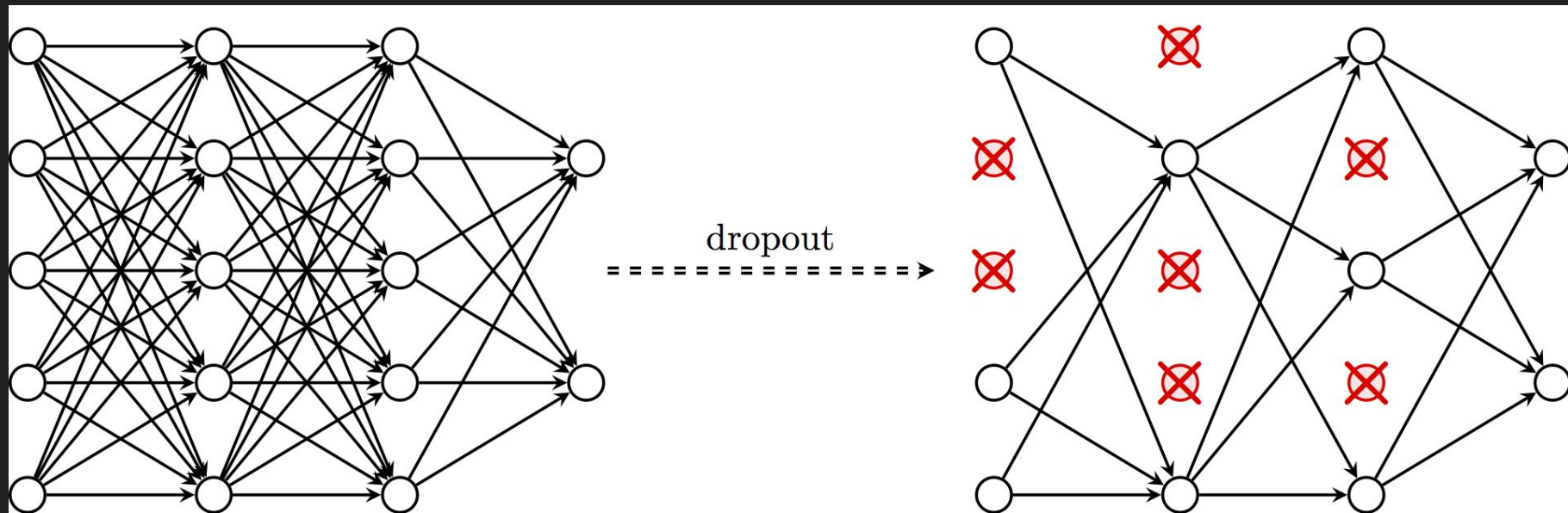
Imagenet: Error rate top 5

CNN: Transfer learning

~~“You need a lot of data if you want to train/use CNNs”~~



Entrenamiento: Tips/Tricks



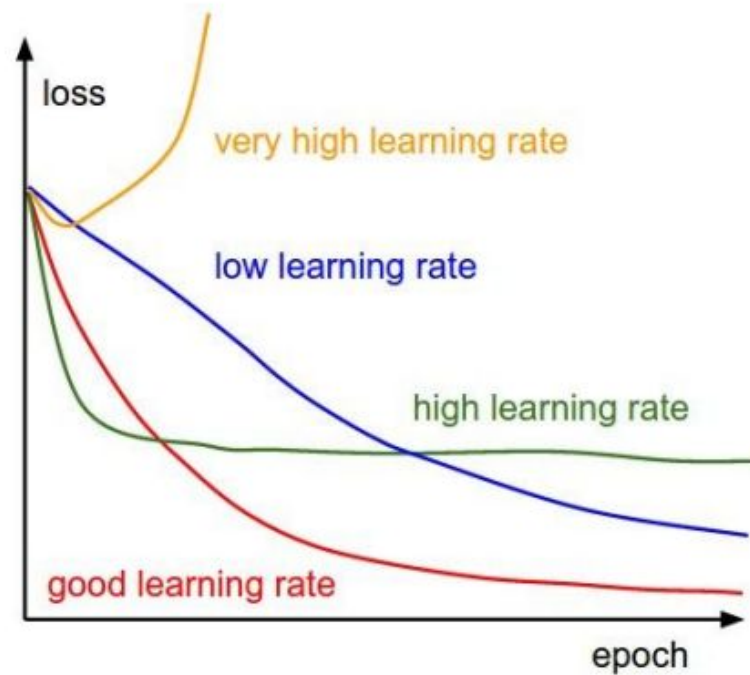
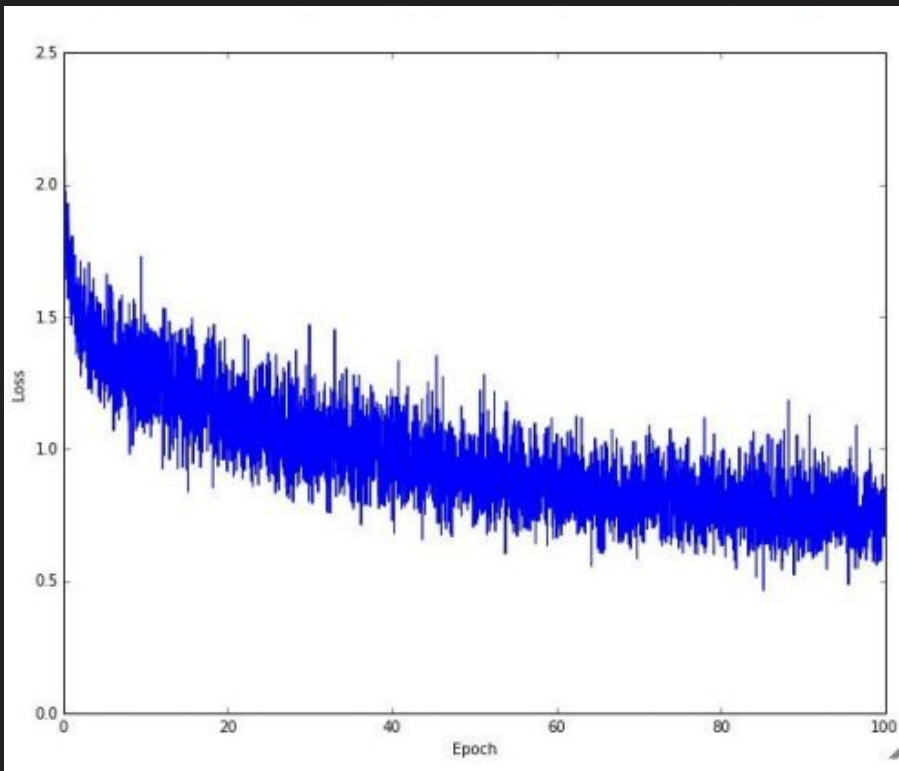
Entrenamiento: Tips/Tricks



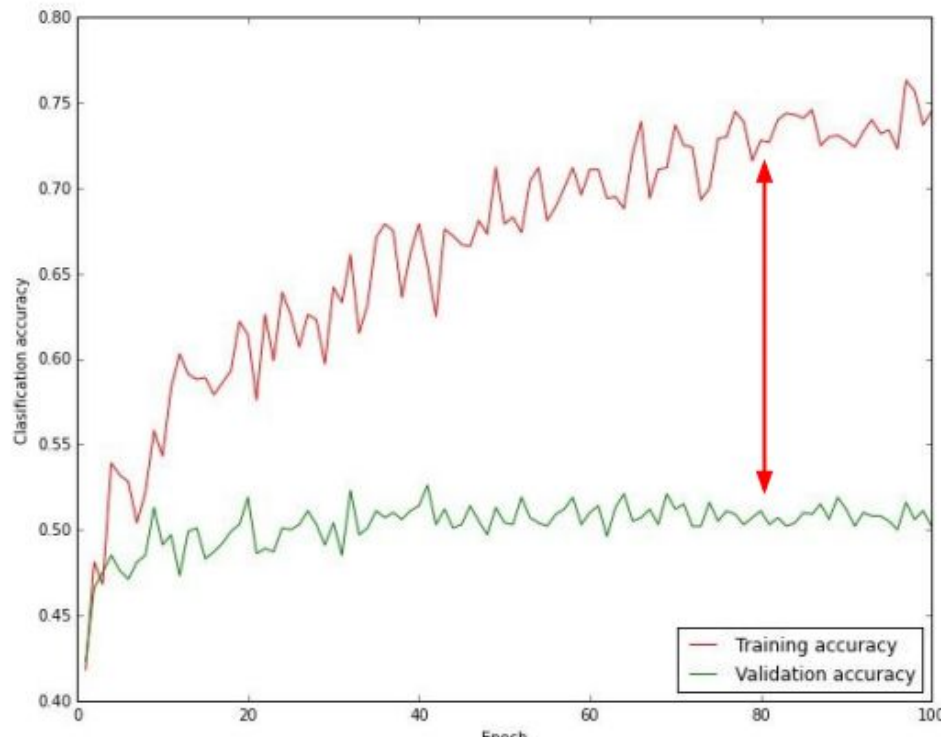
Ajuste de hiperparametros

- Arquitectura de red neuronal.
- Learning rate inicial, actualización.
- Regularización (ej. dropout).
- Más:
<https://goo.gl/p5Voeh>

Entrenamiento: Tips/Tricks



Entrenamiento: Tips/Tricks



big gap = overfitting
=> increase regularization strength?

no gap
=> increase model capacity?

Más info:

→ Intro a CNNs:

<http://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>

→ CS231n

◆ Course: <http://cs231n.stanford.edu/>

◆ Videos: https://www.youtube.com/playlist?list=PL16j5WbGpaM0_Tj8CRmurZ8Kk1gEBc7fg

→ Michael Nielsen's Book: <http://neuralnetworksanddeeplearning.com/>

→ Deep Learning Book: <https://www.deeplearningbook.org/>

→ Andrej Karpathy's Blog: <http://karpathy.github.io/>

¿Preguntas?