



Deep Learning con Pytorch

Juan Pablo Morales
@juanpamf

Ciclo de trabajo en Deep Learning

La importancia de entender cómo aprende una red

- **“No free lunch” theorems**
 - No existe un algoritmo de aprendizaje que sea una bala de plata
- **Leaky abstractions**
 - No se puede esperar una API mágica que haga todo el trabajo por nosotros (A. Karpathy)
- **Trabajo duro**
 - Modelo complejo (ej. Tiempo de cómputo)
 - Entendimiento de generalización
 - Visualización

Algoritmos de aprendizaje

- Espacio de modelos H
- Una función de pérdida L
- Un método de optimización

El entrenamiento de la red corresponde a encontrar el modelo dentro de H cuya pérdida es mínima sobre el dataset.

Algoritmos de aprendizaje

- Predicción del modelo: $m(x)$
- Función de pérdida: $L(m(x), y)$

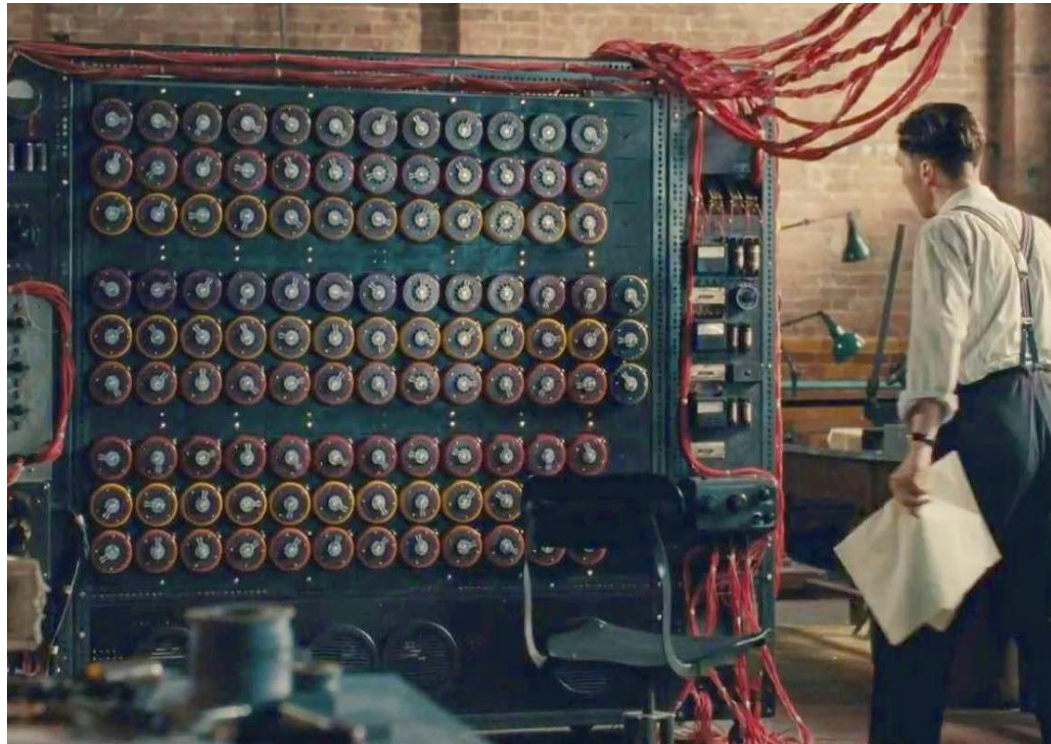
- Función de pérdida total:

$$L(m, D) = \frac{1}{N} \sum_{x \in D} L(m(x), y)$$

- Problema de minimización:

$$\hat{m} = \min_{m \in H} L(m, D)$$

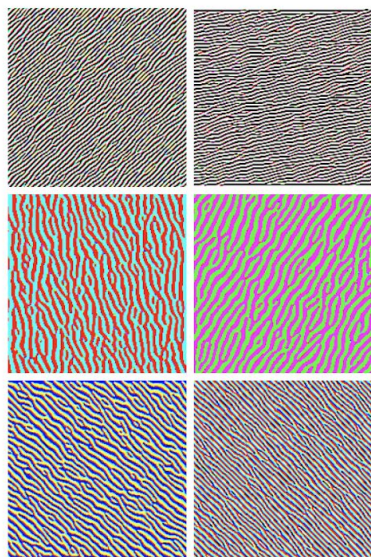
Espacio de modelos parametrizado



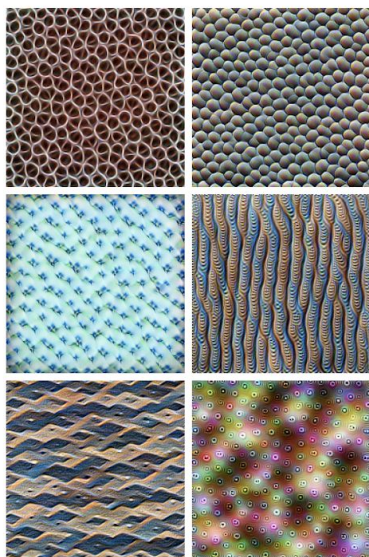
Un flujo con similitudes al ML tradicional

- Objetivo
- Preparación de los datos
 - Features
- Algoritmo de aprendizaje
 - Fija los parámetros (settings internos)
- Tuning
 - Búsqueda de hiperparámetros (settings externos)

Diferencias: Preparación de los datos



Edges (layer conv2d0)



Textures (layer mixed3a)



Patterns (layer mixed4a)

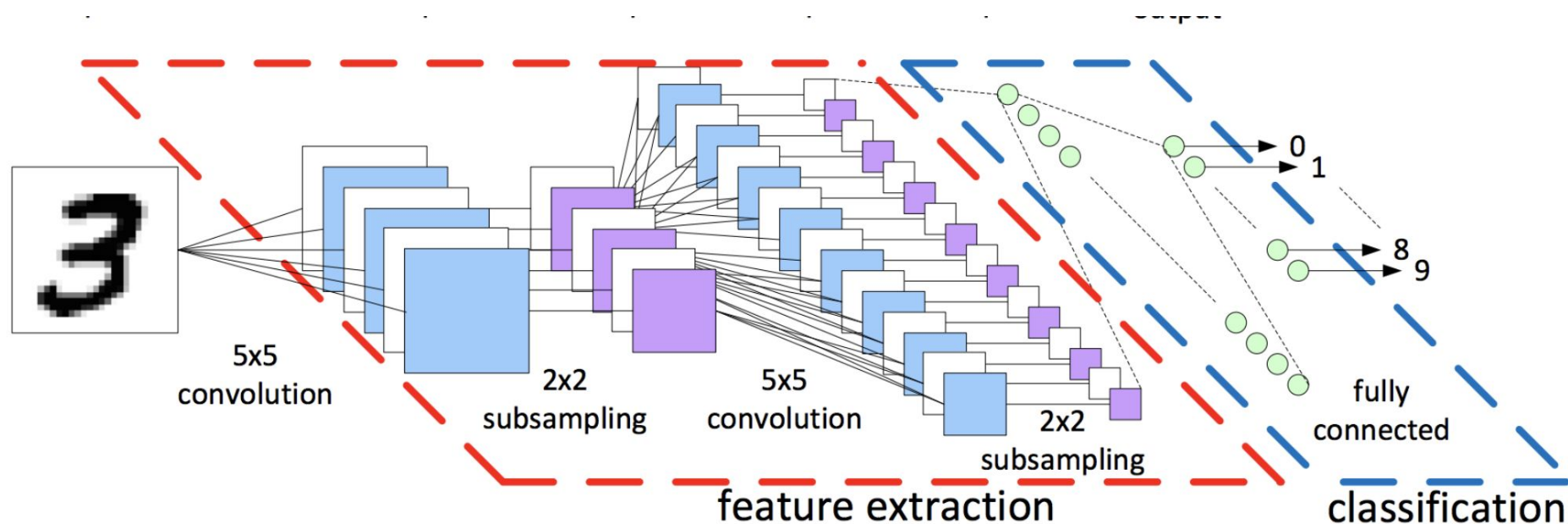


Parts (layers mixed4b & mixed4c)



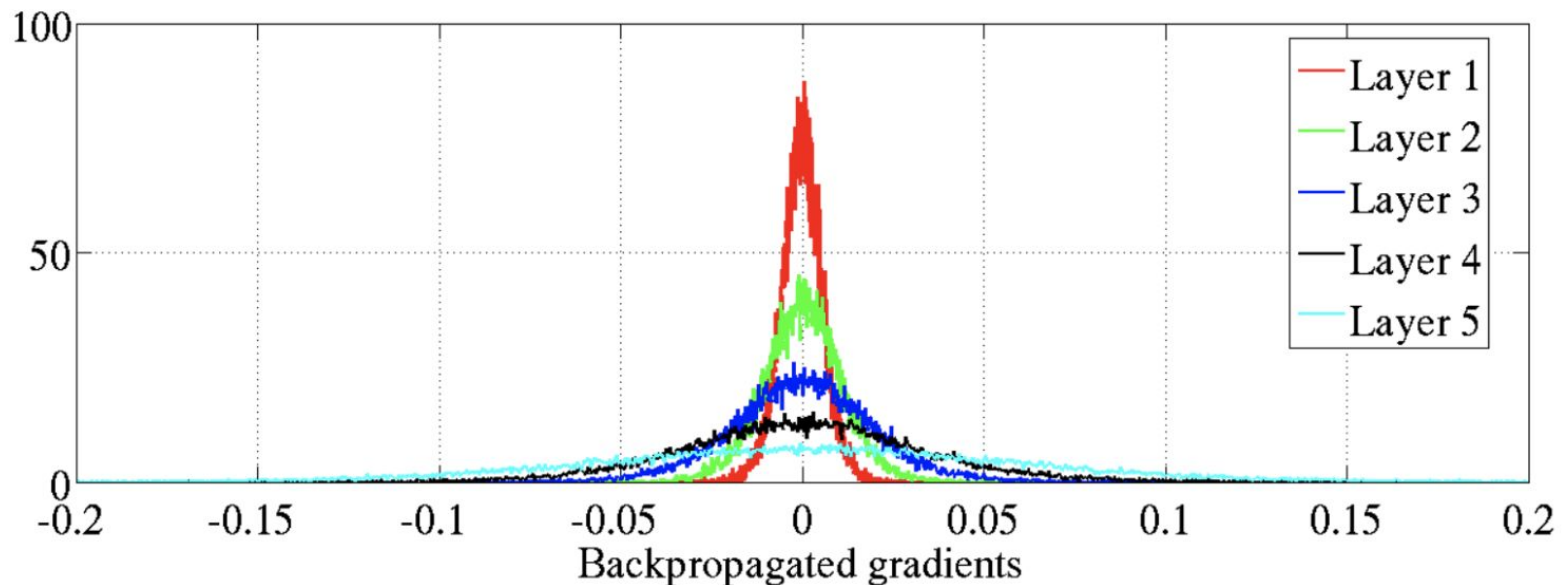
Objects (layers mixed4d & mixed4e)

Diferencias: Algoritmo de aprendizaje



Architecture

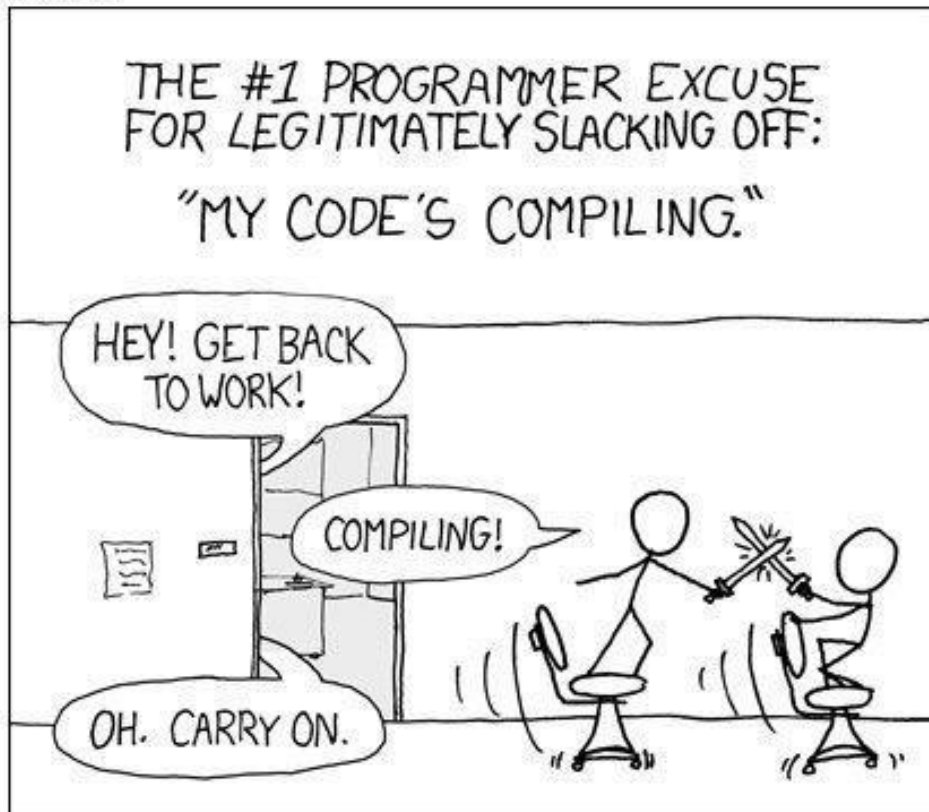
Diferencias: Algoritmo de aprendizaje



*Backpropagated gradients normalized histograms (Glorot and Bengio, 2010).
Gradients for layers far from the output vanish to zero.*

Diferencias: Tuning

2007



2017



original comic xkcd.com/30.
(CC) 2007 Randall Munroe

Flujo del deep learning

- Definición del problema
 - $X \rightarrow Y$
 - Error
- Preparación de los datos
- Diseño del algoritmo de aprendizaje
- Tuning