

Tema 3.1

Frameworks para Deep Learning

Deep Learning

Máster Oficial en Ingeniería Informática

Universidad de Sevilla

Contenido

- Niveles de programación
- Ecosistema actual

Niveles de programación

- La mejor forma de trabajar con redes neuronales es simular los modelos en un ordenador.
- Programación a **nivel 0**:
 - Podemos elegir el lenguaje de nuestra elección e implementar los conceptos a mano.
 - ¿Qué pasa si quiero cambiar la arquitectura de la red? Si no lo he hecho bien, tendría que re-programarla desde cero, sobre todo para ajustar la propagación del gradiente.

Niveles de programación

- Programación a **nivel 1**:
 - APIs que proveen de bloques que podemos usar.
 - Tenemos que encajar las piezas para construir nuestro modelo.
 - Librerías de **diferenciación automática**.
- Programación a **nivel 2**:
 - APIs que proveen una capa de abstracción sobre los detalles de modelo.
 - Encajamos bloques que representan partes de la arquitectura que queremos montar.



Lenguajes de programación



Lisp

El primer
lenguaje
diseñado para
IA



R

Lenguaje para
estadística,
análisis de
datos y
bioinformática



Python

Lenguaje
universal por
su facilidad de
uso y gran
ecosistema de
librerías



C++

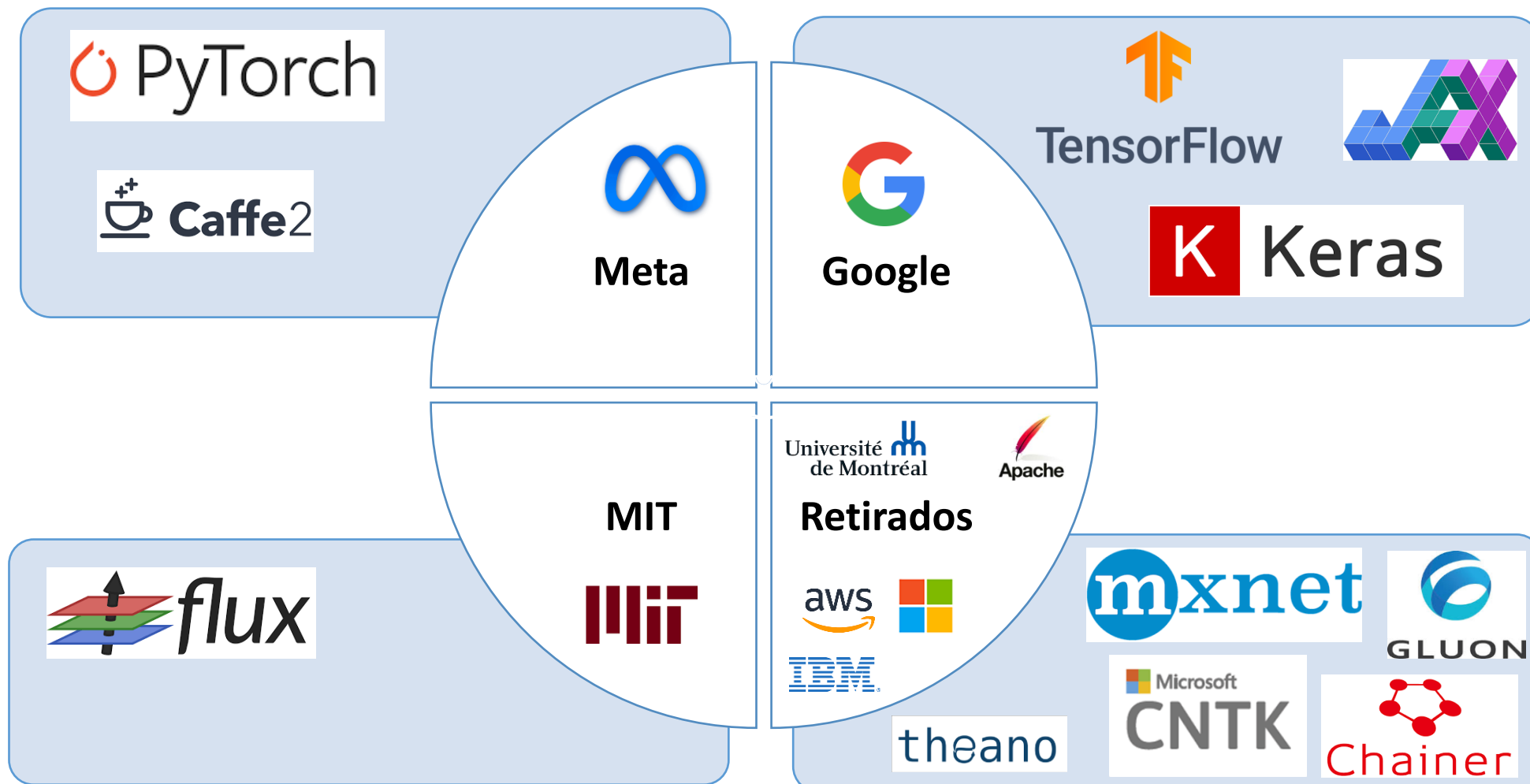
Usado por su
alto
rendimiento.
Detrás de
librerías de
Python y R



Julia

Nuevo
lenguaje para
IA y
computación
científica,
eficiencia y
flexibilidad

Ecosistema actual en Deep Learning



Ecosistema actual

- **PyTorch:**

- Creado por **Facebook (Meta)** en 2017.
- Ahora parte de la Linux Foundation
- Un porte de Torch (Lua) a **Python** (backend en C++)
- Permite usar depuradores de código Python estándar.
- Programación a nivel 1 y 2.
- Integra Caffe2 para un nivel 2 puro.
- PyTorch compite con TensorFlow directamente

P Y T  R C H

Ecosistema actual

- **Keras:**

- Creado por F. Chollet (Google) en 2015.
- Programación a nivel 2: prototipado rápido de redes en **Python**.
- Requiere de un **backend**. Desde la versión 3 se permite:
 - Tensorflow
 - PyTorch
 - JAX.
- Es el mejor framework para comenzar con conocimientos básicos.
- **Potente** para modelos sencillos, limitado para complejos.
- API muy madurada, totalmente integrada en Tensorflow 2.



Ecosistema actual

- **TensorFlow:**

- Creado por **Google** en 2015.
- Lenguaje principal: **Python**, con backend en C++.
- En clusters de ordenadores y en plataformas móviles (iOS, Android).
- Empleado sobre todo para despliegue de modelos escalable.
- Programación a nivel 1: curva de aprendizaje alta.
- Programación a nivel 2 a través de Keras.
- ¡¡Versión 2 es incompatible con versión 1!!



Ecosistema actual

- **JAX:**

- Creado por **Google** en 2018.
- Lenguaje principal: **Python**, con backend en C++ y XLA (compiladores).
- Enfocado en cálculo numérico de alto rendimiento y escalable.
- Basado en programación funcional (funciones puras).
 - PyTorch y Tensorflow se basan en la programación orientada a objetos.
- Programación a nivel 1: programación tensorial ligado a Numpy
- Extensiones para nivel 2 (Flax, Haiku...)
- Utilizado por xAI para entrenar a Grok.



Ecosistema actual

- **Flux.jl:**

- Creado por el **MIT** en 2017.
- Lenguaje principal: **Julia**, con backend en Julia (todo integrado y eficiente)
- Programación funcional, permite declarar modelos con base matemática y conexión con computación científica
- Programación a nivel 1: programación tensorial ligado a Array de Julia
- Utilizado en investigación y ciencia de datos



Ecosistema actual

- Orientación:
 - **Keras**: para prototipado rápido de redes y para aprender conceptos básicos.
 - **Pytorch**: para investigación y exploración de arquitecturas en estado del arte
 - **Tensorflow**: para integración con entornos de despliegue (cloud, móvil).
 - **JAX**: alto rendimiento, escalabilidad e investigación avanzada.
 - **Flux**: enfoque científico y lenguaje Julia.