The Keras framework

Go with the (tensor)flow

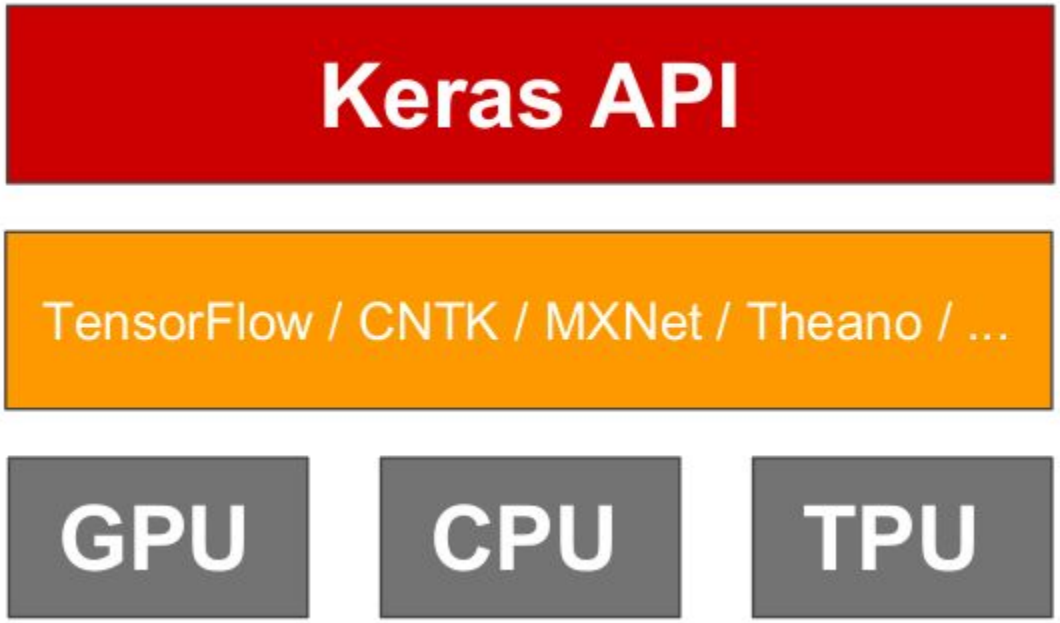
Filippo Biscarini
Senior Scientist
CNR, Milan (Italy)

Nelson Nazzicari
Senior Scientist
CREA, Lodi (Italy)

Why Keras?

- Deep learning on CPU/GPU/TPU
- Many available alternatives (PyTorch, Lasagne, fast.ai, Theano, TensorFlow, Caffe, Mxnet …)
- Keras was born as a “library agnostic” layer supporting mainly three backends: Tensorflow, Theano and Microsoft CNTK …
- … but then Tensorflow won the race and “ate” Keras

What is Keras?



From François Chollet

What is Keras?

- Python library (modules and submodules):
  - `import keras.utils`
  - `from keras.models import Sequential`
  - `from keras.layers import Dense, Dropout, Activation, Flatten`
- Clean, consistent API
- “User friendly” (as much as these things go)
- Define models via building blocks
- User defined forward-propagation → back-propagation is automatic

Who is behind Keras?

633 contributors

Microsoft

Google

NVIDIA

aws

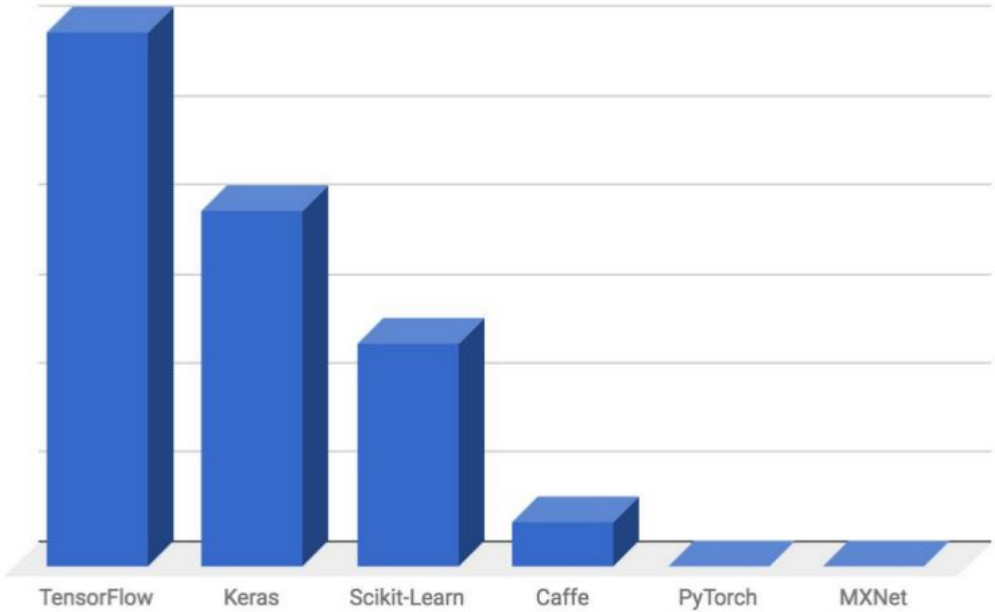
From François Chollet

Keras pros

- Large community of users
- Multi-backend, multi-platform
- Easy and quick development and deployment of deep learning models

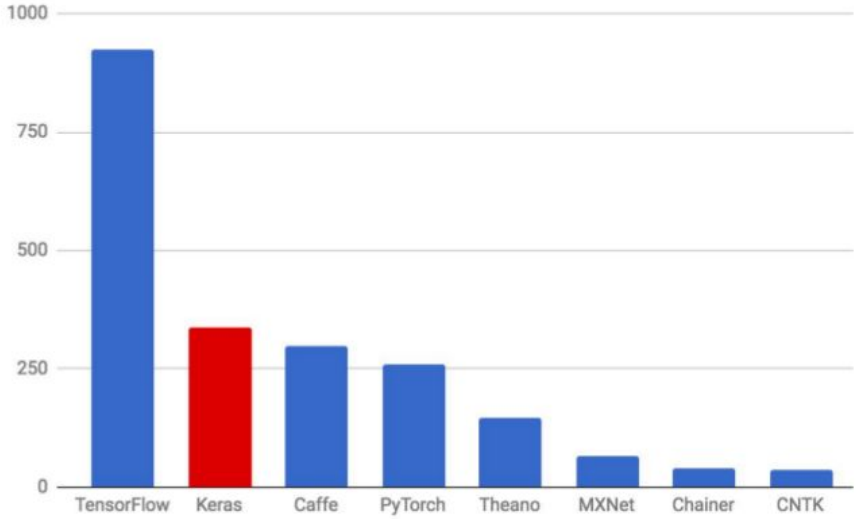
Keras - large community

Hacker News jobs board mentions - out of 964 job postings

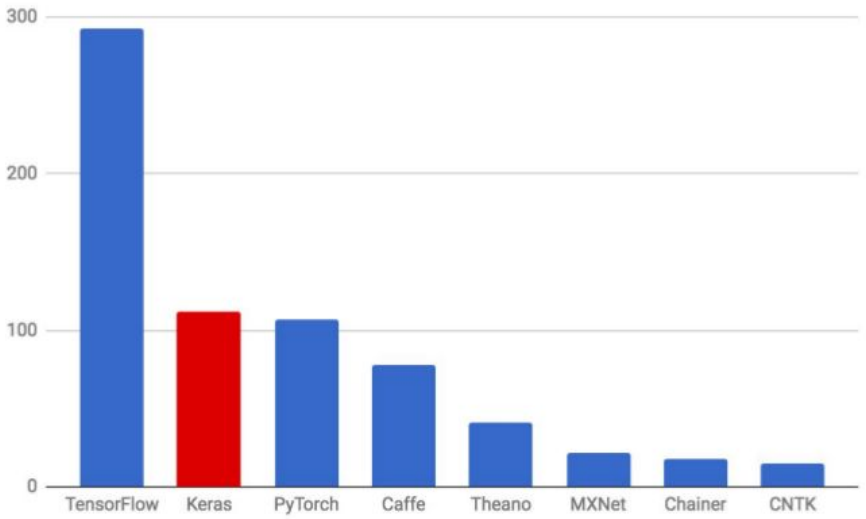


From François Chollet

Keras - arXiv mentions



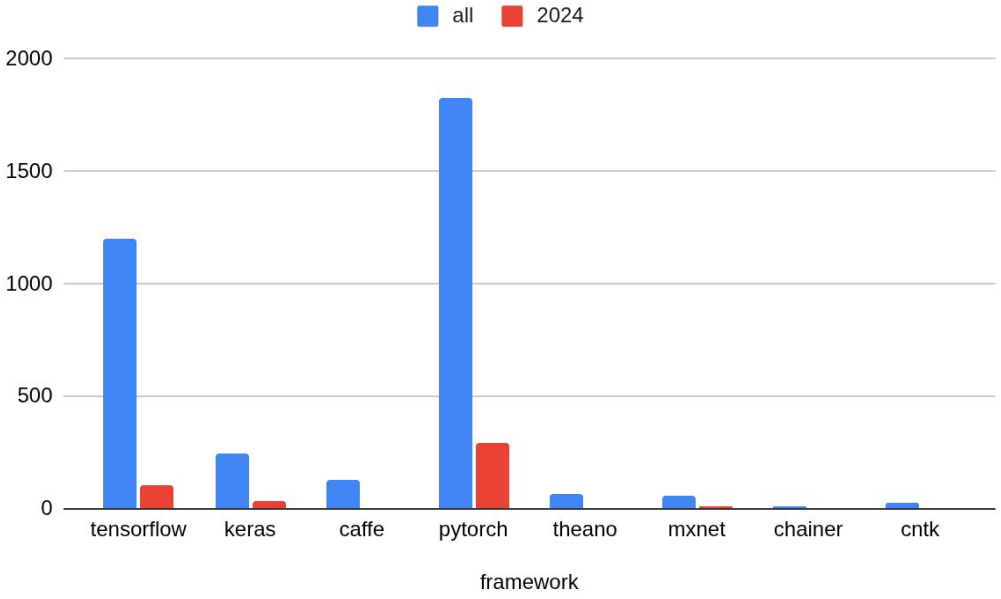
arXiv mentions as of 2018/03/07 (past 3 months)

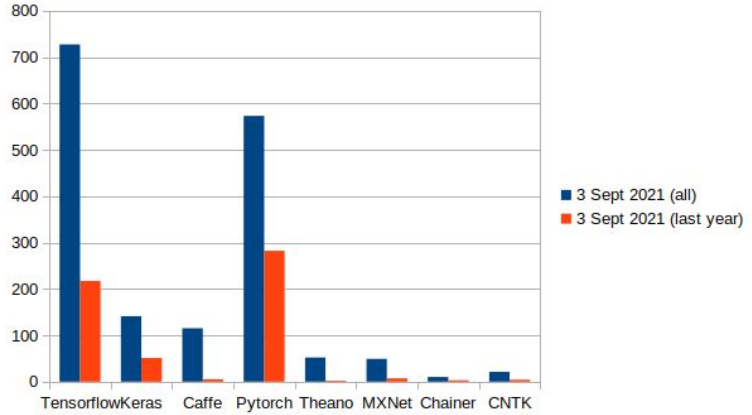


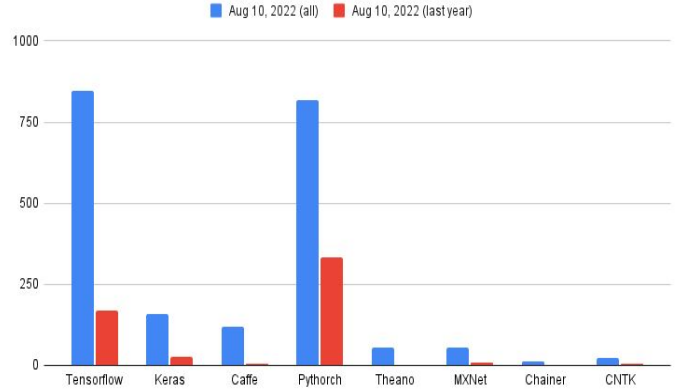
arXiv mentions as of 2018/03/07 (past 1 month)

From François Chollet

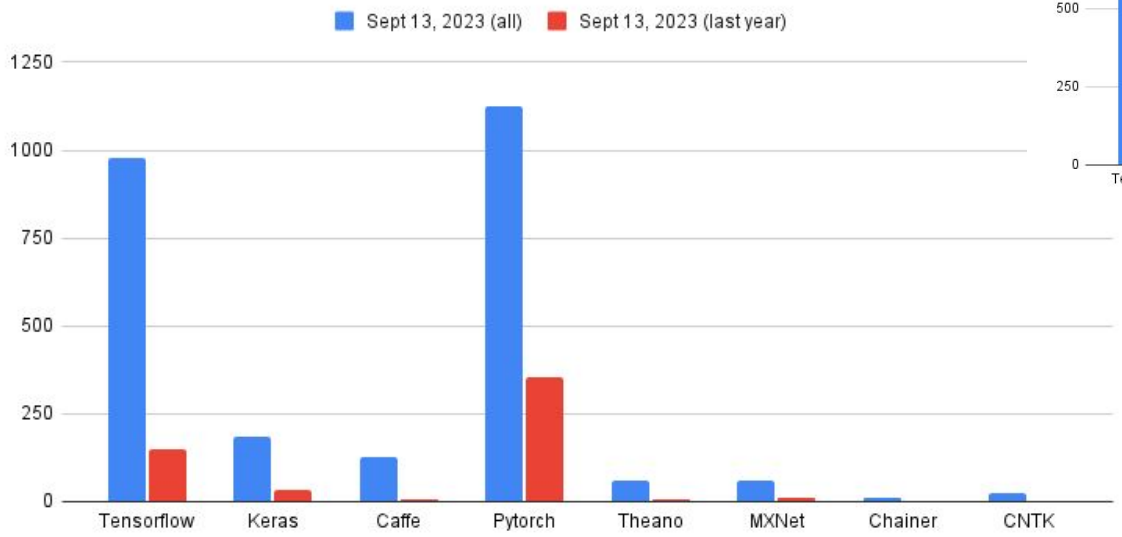
Keras - arXiv mentions

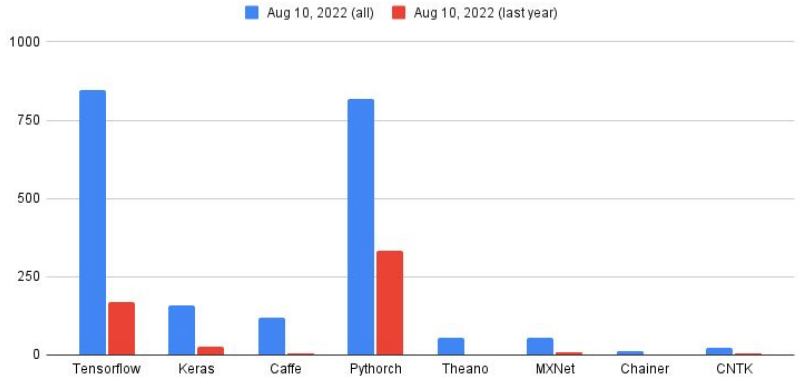






Keras - arXiv mentions





Keras/Tensorflow vs Pytorch

[on google trends]

Keras/Tf - multi-backend, multi-platform

- develop in **Python**, **R** (maybe)
- on **Unix/Linux**, **MacOS**, **Windows**
- high-level wrapper for **TensorFlow** (but also Theano, CNTK etc.)
  - In practical terms it's now part of TensorFlow
- **CPU**, **GPU** (Nvidia, Amd), **TPU**

Keras/Tf - easy and quick

- designed for humans, not machines: consistent and simple APIs, clear code, clear error feedback
- easy to learn and use: > productivity, > freedom to explore ideas
- easy, yet flexible: lower level APIs (e.g. TensorFlow) allow to implement anything you need

A keras workflow

1. **Prepare and split data**
   a. `NumPy arrays`
   b. `mydata = tf.keras.utils.image_dataset_from_directory(...)`
   c. … but also sequences and text
2. **Define the model** (`from keras.layers import Dense, Dropout, Activation, Flatten ...`)
3. **Compile the model** (`model.compile(...)`)
   a. Choose loss, optimizer
4. **Fit the model** (`model.fit(...)`)
5. **Predict result for unknown value** (`model.evaluate(...)`)
6. **Modify until satisfied**
7. **Save for future use** (`model.save(...)`)

[REF] Keras tutorials and docs

- https://www.tutorialspoint.com/keras/keras_introduction.htm [very good]
- https://keras.io/getting_started/intro_to_keras_for_engineers/
- https://keras.io/api/ the official doc
- https://playground.tensorflow.org to see a neural network live

Keras

- It’s time for exercises!
- see notebook “keras_basics”