

Practice session 7. Image classification using Keras



TensorFlow



PyTorch

“The more ~~languages~~ frameworks you know the easier it is to learn new ones”*

What is important when selecting the framework:

- Purpose of project (industry vs research, deployment to mobile device?)
- Experience (some frameworks more intuitive for python programmers)
- Community (it is important to know where to look for help)
- Model availability (OpenAI's GPT-3 has over 175 billion parameters, and GPT-4 will have over 100 trillion parameters)

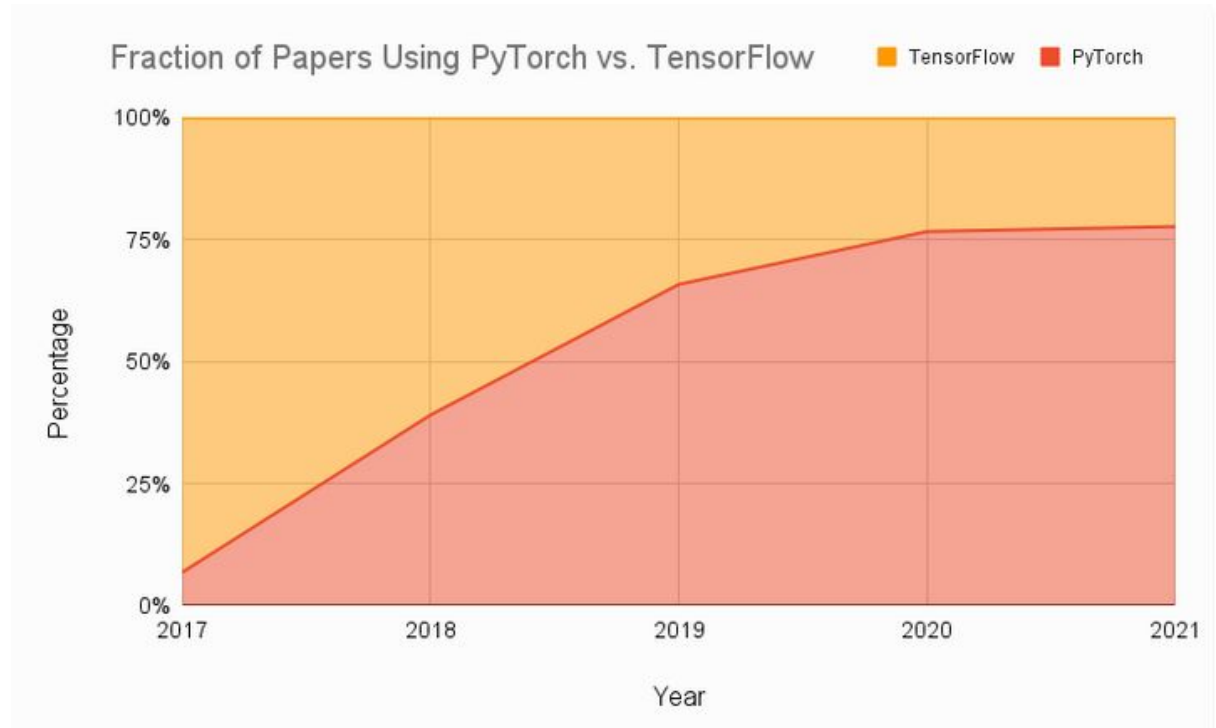
* All arguments in the presentation are subjective opinion and can be criticized

Research trends

<https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/>

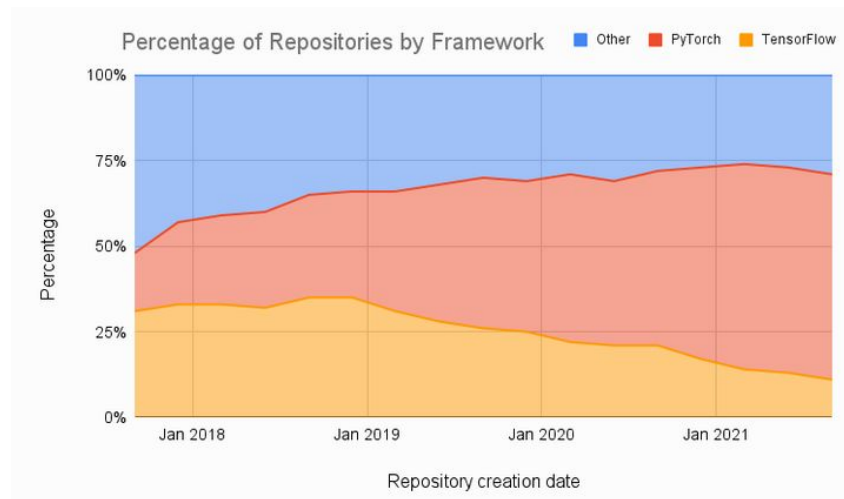
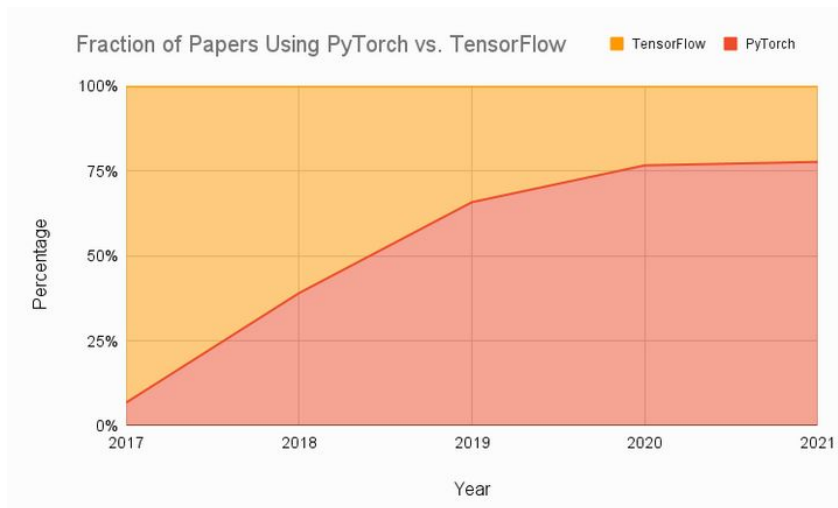
Imperative Programming
vs Symbolic Programming

Dynamic Computation
Graph vs Static
Computation Graph



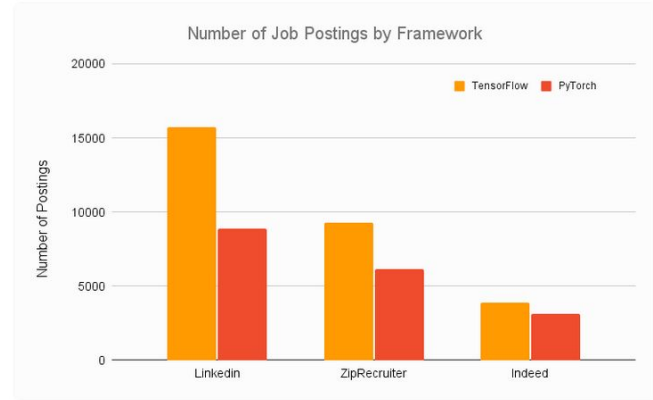
Papers with code

<https://paperswithcode.com/task/image-classification>



Why do we still need Tensorflow?

- Model deployment
- Serialization
- Device management
- Visualizer
- Open-source
- Documentation (actually both are good)



LinkedIn and ZipRecruiter data correspond to all jobs posted in the U.S. while Indeed data corresponds to jobs posted within 100 miles of 11 major cities in the U.S.

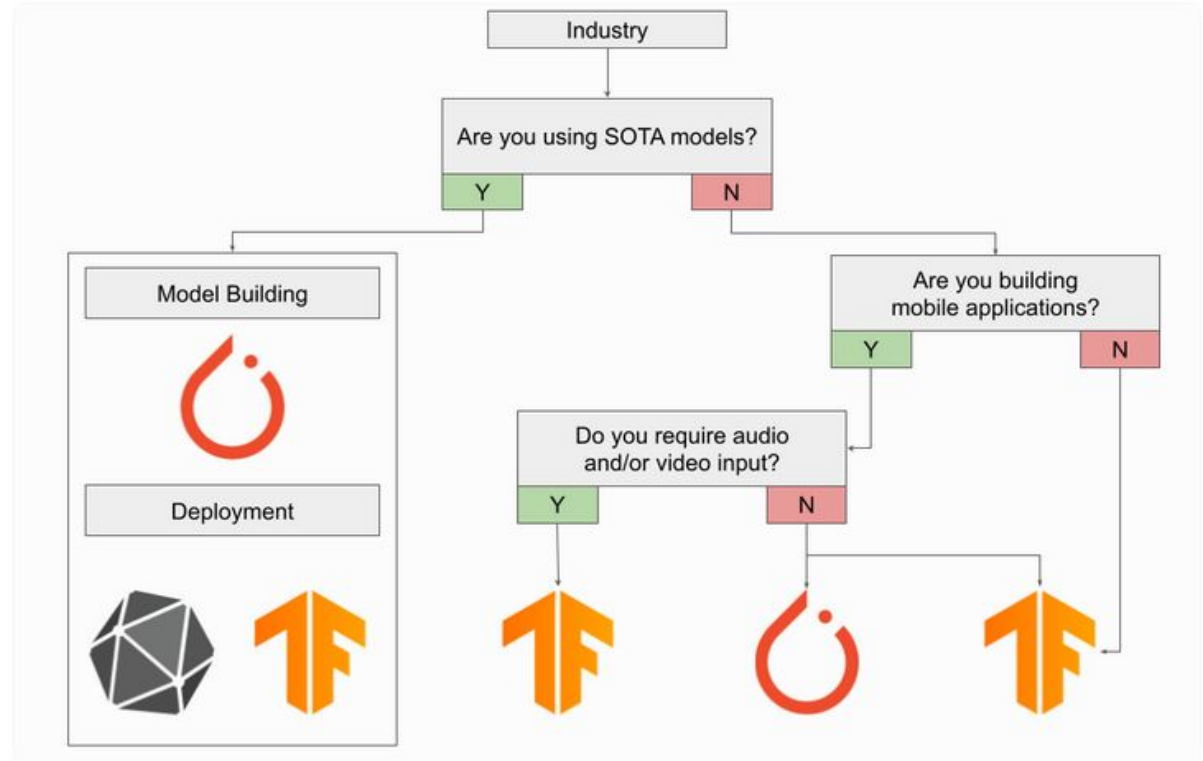
Pytorch is better for rapid prototyping in Research. Good start for small-scale projects. Better utilization and optimization for use of GPUs.

Tensorflow is better for large-scale deployments. Good for cross-platform and embedded deployment. Bigger community base and many users.

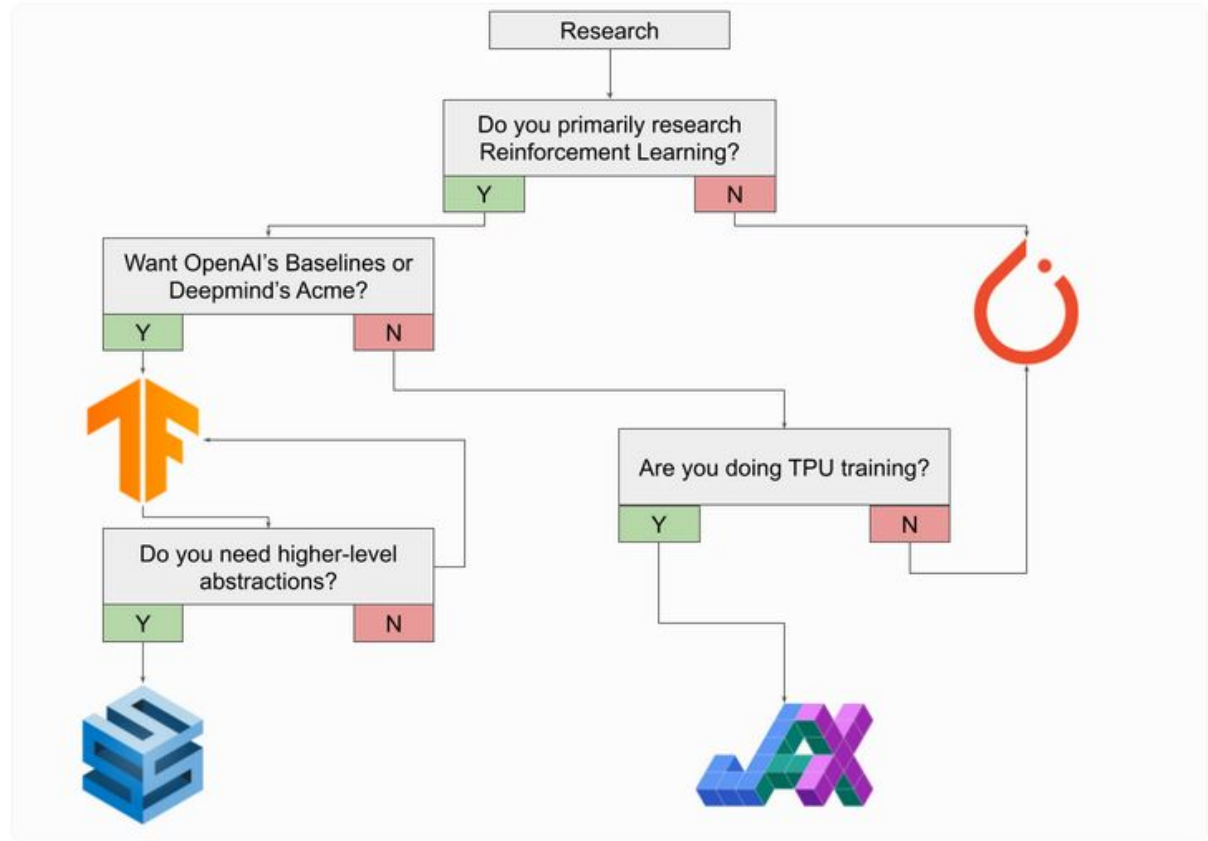
Google AI: primary publish research with TensorFlow

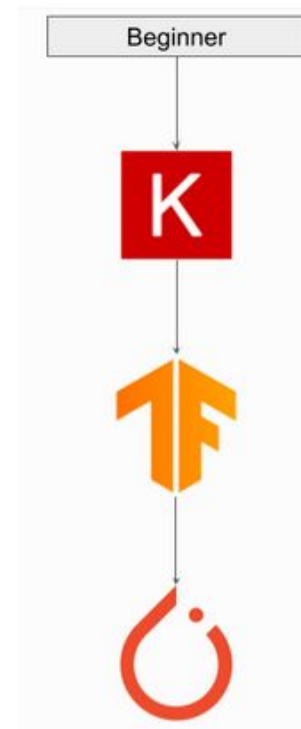
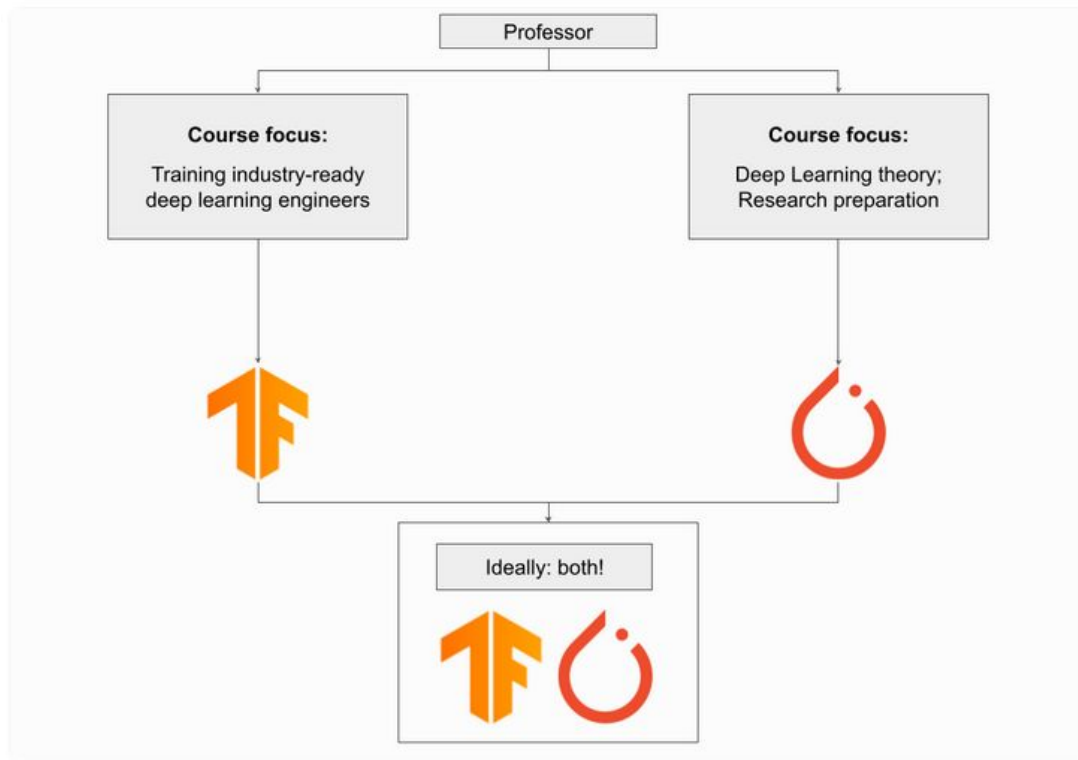
DeepMind: standardized the use of Tensorflow in 2016, although in 2020 announced use JAX to accelerate their research

OpenAI: standardized the use of PyTorch in 2020, however, older baselines in Tensorflow. **Baselines provides high-quality implementation of Reinforcement Learning algorithms, so TensorFlow may be the best choice for Reinforcement Learning practitioners.**



JAX: another library developed by DeepMind. With its updated version of Autograd, JAX can automatically differentiate native Python and NumPy code. JAX is able to differentiate through all sorts of python and NumPy functions, including loops, branches, recursions, and more. This is **incredibly useful for Deep Learning apps as we can run backpropagation pretty much effortlessly**.





*<https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/>

Example

Pytorch:

<https://colab.research.google.com/drive/1KBTkiz1bDLavswWsv71gKBoHC8KEI04X?usp=sharing>

Tensorflow:

<https://colab.research.google.com/drive/1Rt6ROvjlc2Ash7Nusrwsc5eFmYzMn45C?usp=sharing>