Pose Detection Demo

- How is this done?
- Is it easy to do?
- Is it cloud-service or what, how much does it cost?

Pose Detection Demo

- How is this done?
- Is it easy to do?
- Is it cloud-service or what, how much does it cost?
  - This is running locally in browser (also in mobile!), costs nothing, it's open source too!

Pose Detection Demo

- How is this done?
- Is it easy to do?
- Is it cloud-service or what, how much does it cost?
  - This is running locally in browser (also in mobile!), costs nothing, it's open source too!
- How and is it easy: Let's see it and more in this presentation!

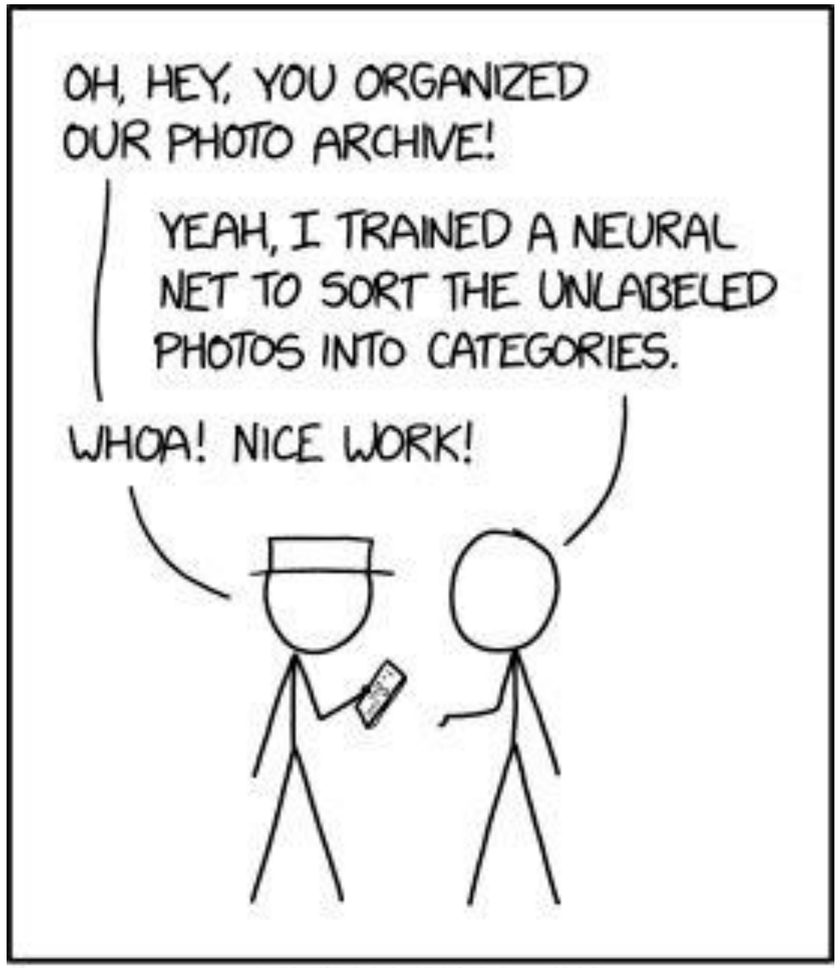
Tensorflow.js and Huggingface Transformers.js: Machine learning in JavaScript

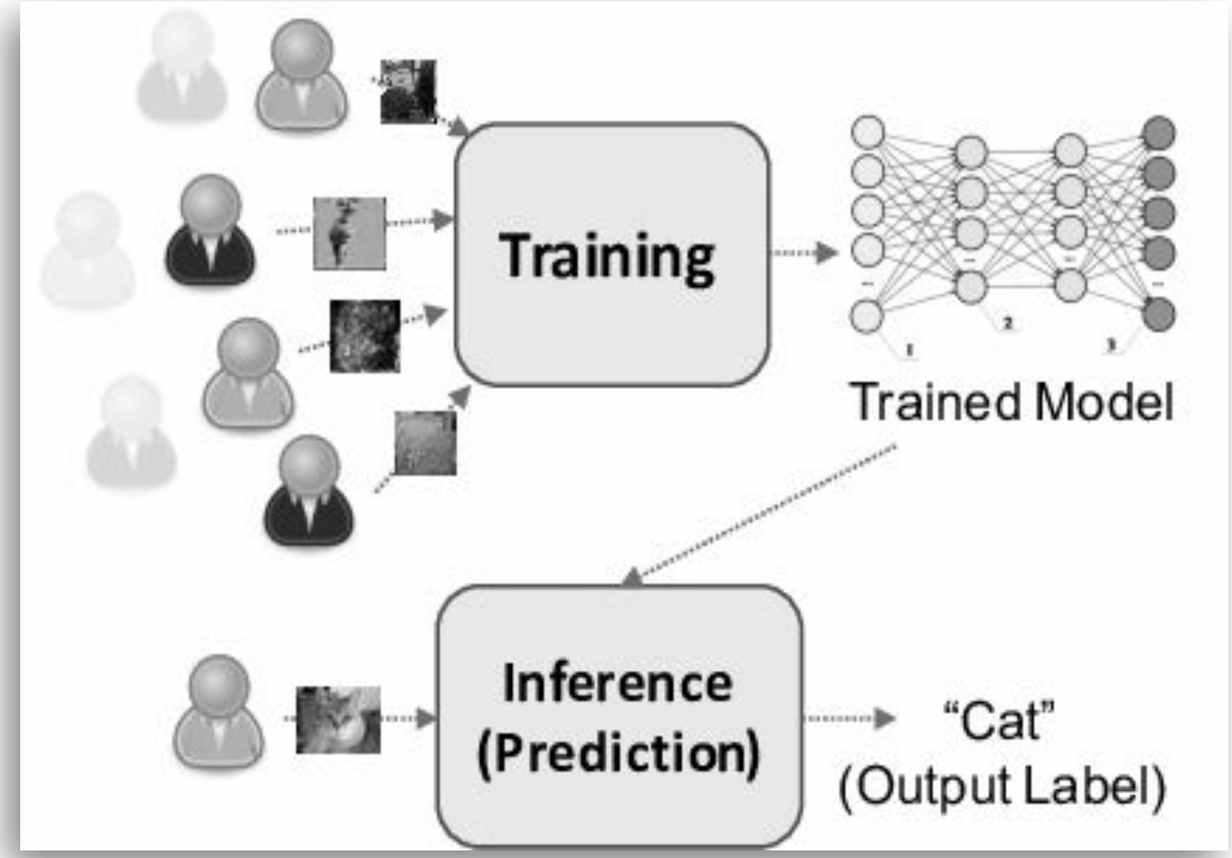
Reaktor

Deck template

Machine learning - What? Why the fuss?

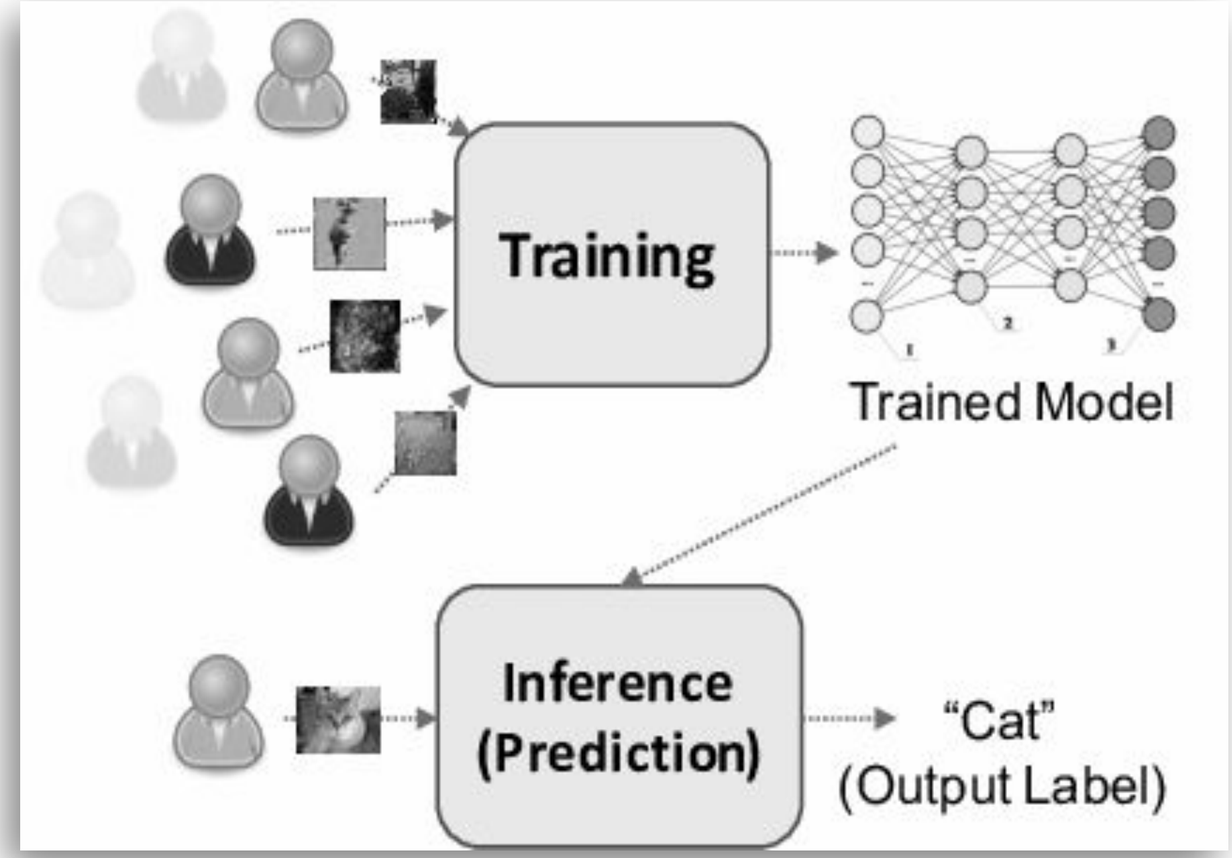
- Wikipedia: "*Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can* ***learn*** *from data and* ***generalize to unseen data****, and thus perform tasks without explicit instructions.*"
- "*Recently, artificial neural networks (part of ML) have been* ***able to surpass many previous approaches in performance****.*"





Learning and Inferring

- In machine learning there's typically two phases: Learning and Inferring (=prediction)
- In "Supervised learning" and "Unsupervised learning" the learning phase is done just during developing of the system, and the result (trained model), that system has learned, is used for inferring in the final system

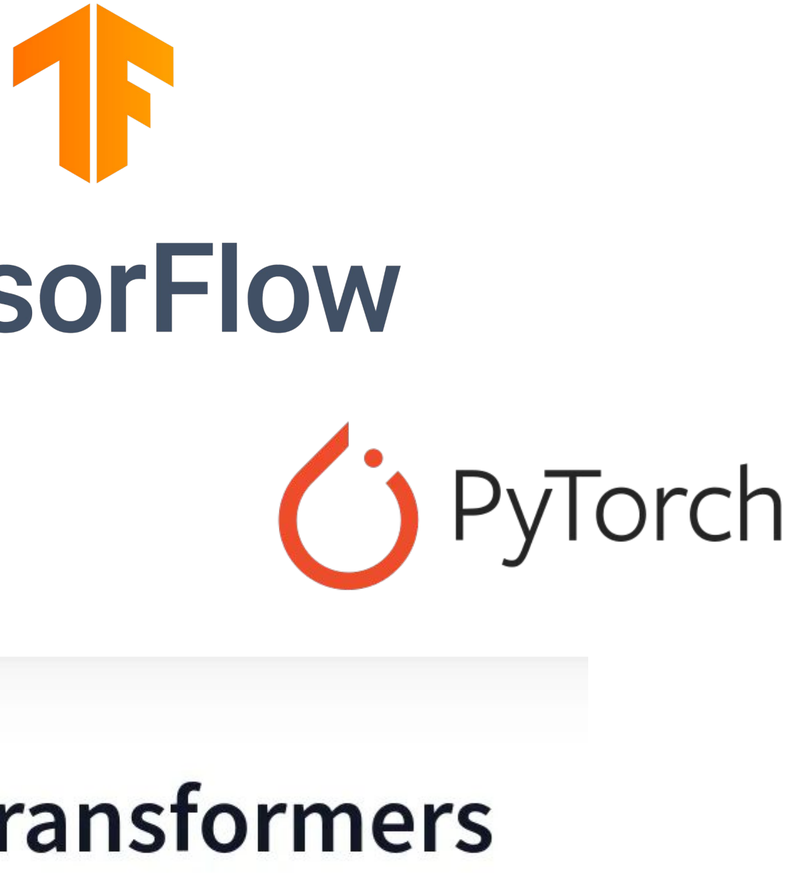


Learning and Inferring

- In "Reinforcement learning" the learning is done also in the final system so that system learns during the execution, from its dynamic environment, by interacting with the end users, etc.
- There is also machine learning that is something outside of the above, or something that combines the above, or is in between some of the above
- It's very very wild-wild west! :D

Ruling techs in the world of Machine Learning at the beginning of 2024

- Python rulez (what JavaScript not?)
  - especially in the "learning" side, in making new innovation
- Three “big players" at 2024:
  Tensorflow, **PyTorch**
  and **Hugging Face Transformers**
  - They implement the learning and inferring of the models and dictate in which format models are stored
  - Their have practices how to share also the pre-trained models ("hubs", "model zoos")
  - Huge ecosystems



Tensorflow

- A free and open-source software library for machine learning and artificial intelligence
- Can be used across a range of tasks but has a particular focus on training and inference of deep neural networks
- Originally developed by the Google Brain team for Google's internal use in research and production
- Can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java, facilitating its use in a range of applications in many sectors

https://www.tensorflow.org/

Sharing models:
https://www.tensorflow.org/hub
Nowadays in
https://www.kaggle.com/models

PyTorch

- Python machine learning library based on the Torch library (a Lua based lib)
- Used for applications such as computer vision and natural language processing
- Originally developed by Meta AI and now part of the Linux Foundation umbrella
- Python interface is more polished and the primary focus of development, PyTorch also has a C++ interface

https://pytorch.org/

Sharing models:
https://pytorch.org/hub/

Hugging Face

https://huggingface.co/

Sharing models, datasets and “spaces” (apps made by the community):
https://huggingface.co/models
https://huggingface.co/datasets
https://huggingface.co/spaces

- The **Transformers** library is a Python package that contains open-source implementations of transformer models for text, image, and audio tasks
- Compatible with the **PyTorch**, **TensorFlow** and JAX deep learning libraries
- Includes implementations of notable models like BERT and **GPT-2**
- Hugging Face Hub
- Provides also execution as cloud-service (like OpenAI)

Demo?

- So, come on! How it was done?
- Show something practical!

Easy to use, batteries-included, libs and tools?

- Something for just regular software developers
  - doing applications for real (not in lab),
  - knowing software development practices well,
  - but not having knowledge, time and desire to know all from bottom to top?
- How and what we can apply right now to our software easily?

Low-hanging fruits

- Let's skip the laborious "learning" part and go to the "inferring" :D

Low-hanging fruits

- Let's skip the laborious "learning" part and go to the "inferring" :D

- Luckily there are nice **JavaScript** alternatives from earlier mentioned big players:
  - Tensorflow.js
  - Hugging Face Transformers.js

https://www.tensorflow.org/js

“Models”:
https://www.tensorflow.org/js/models

Tensorflow.js

- A JavaScript library for training and deploying machine learning models in the browser and on Node.js
- In web browsers (also in mobile!), utilizes **WebGL and WebGPU** for hardware-accelerated graphics
- On Node.js, accelerated by the TensorFlow C binary, runs tensor operations on the GPU with CUDA
- Has also higher level libs that they confusingly call "models"
  - not just to the underlying mathematical model, but to the entire package that includes the pre-trained weights, the architecture, and the high-level functions for processing inputs and interpreting outputs

Transformers.js

Transformers.js (from Hugging Face)

https://huggingface.co/docs/transformers.js

Models for Transformers.js: https://huggingface.co/models?library=transformers.js

- Transformers.js uses ONNX Runtime to run models in the browser
- => "you can easily convert your pretrained **PyTorch**, **TensorFlow**, or JAX models to ONNX"
- Benefits from huge model and dataset and example/documentation collection of the Hugging Face
- The WebGPU support coming in version 3 (in alpha now)
  https://cloudblogs.microsoft.com/opensource/2024/02/29/onnx-runtime-web-unleashes-generative-ai-in-the-browser-using-webgpu/

Demo?

- We haven’t yet seen any (JavaScript) code!

Demo: Tensorflow.js

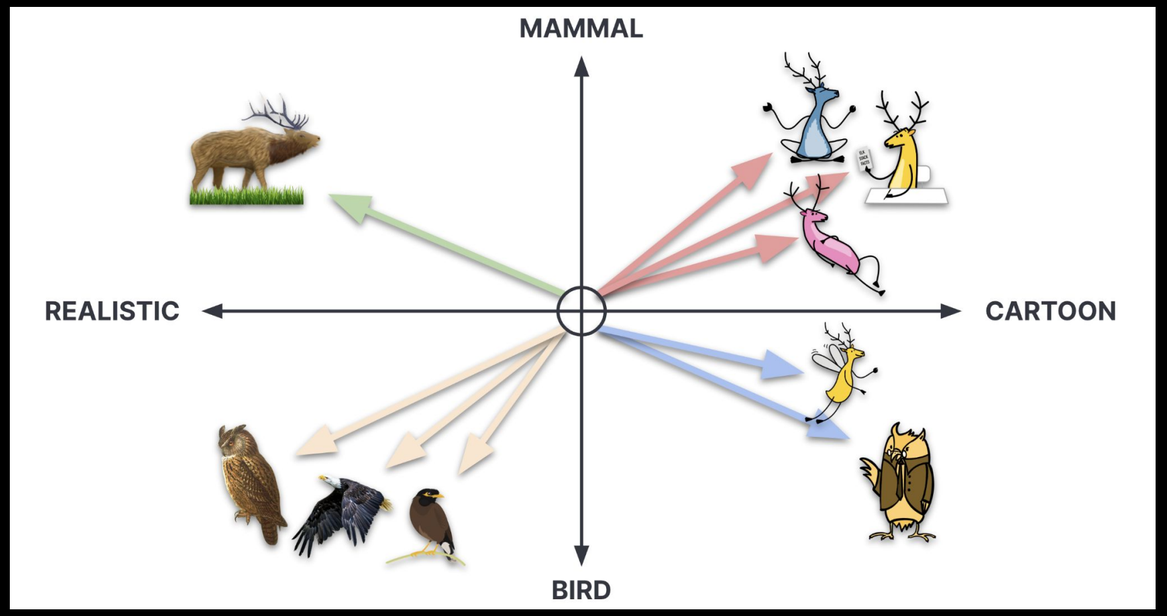
- https://www.tensorflow.org/js/models
- Posenet
  - https://github.com/tensorflow/tfjs-models/tree/master/pose-detection
  - https://storage.googleapis.com/tfjs-models/demos/pose-detection/index.html?model=posenet
  - On localhost: clock.js
- Face landmarks
  - https://github.com/tensorflow/tfjs-models/tree/master/face-landmarks-detection
  - https://storage.googleapis.com/tfjs-models/demos/face-landmarks-detection/index.html?model=mediapipe_face_mesh
- Object detection
  - https://github.com/tensorflow/tfjs-models/tree/master/coco-ssd
  - https://tensorflow-js-object-detection.glitch.me/

Demo: Hugging Face Transformers.js

- https://huggingface.co/docs/transformers.js
- Object detection
  - https://huggingface.co/posts/Xenova/804343794091633
  - https://huggingface.co/spaces/Xenova/video-object-detection
- On localhost Node.js:
  - object-detection.ts
  - image-to-text.ts
  - vectorize.ts (“feature extraction”)

Demo Bonus: Semantic Search by Transformers.js and PostgreSQL pgvector

- Let’s combine the vectorize.ts (“feature extraction”)
- ...with the PostgreSQL database with pgvector extension
- Yes, this is implemented in many “special” databases – for example the image there in right is stolen from ElasticSearch, but...



Thank you!

- Questions?