The Evolution of A.I.

The 1st. Period 1950-1970

Trivial problem solving, no practicality, GOFAI - Good Old Fashioned Artificial Intelligence

1942 The 3 Laws of Robotics by Isaac Asimov. Other sets of laws have been proposed by researchers since then.

1950 **The Turing Test -** proposed by Alan Turing

1952 The first self learning game program

1956 **Dartmouth Conference** First Use of the term "**Artificial Intelligence/A.I.**"

1957 General Problem Solver (GPS) (by Newell)

1958 McCarthy developed **Lisp** programming language

1959 **The MIT AI Lab** (McCarthy and Minsky)

1959 The term "**Machine Learning**" by Samuel

1961 **First Industrial Robot** (Unimate) working at GM

1961 **SAINT** - the first expert system by Slagle (MIT)

1964 **STUDENT** The first A.I. program which understands natural language

1965 **ELIZA -** The first A.I. based chatbot and expert system

1966 **Shakey** - the first locomotive and intelligent robot (SRI)

1966 **MAC HACK-** chess-playing program (by Greenblatt, MIT)

1968 **SHRDLU**, an early natural language understanding computer program

1970 **WABOT-1** the first anthropomorphic robot (Waseda University)

1972 **Prolog** logic programming language

1973 **Lighthill Report** The poor progress report caused the **First A.I. winter** (Reduced funding for AI research)

1974 **MYCIN** - The first rule based AI expert system for medical diagnostics

1974 The first **autonomous vehicle**, a mechanical "slider" (Stanford)

1980 LISP based machines developed and marketed

1980 **INTERNIST-I** The first Commercial Expert System

1986 A **driverless van by Mercedes-Benz**, with cameras and sensors

1988 **Bayesian Network** (BNs or belief nets) is invented by Pearl

1988 The chatbots, Jabberwacky and Cleverbot invented by Carpenter

1989 The first **autonomous vehicle** created by CMU using **neural network**

1993 **Polly**, the tour guide robot; behavior-based robotics (MIT)

The 2nd. Period 1980-2000

Researchers feeding machines with labeled data. Projects: ICOT – Japan '82, MCC – US '83, Alvey – UK '84. Algorithms began to appear as parts of larger systems. AI solutions proved to be useful throughout the technology industry, such as data mining, industrial robotics, logistics, speech recognition, banking software, medical diagnosis and search engines

1997 IBM's Deep Blue beats Gary Kasparov in chess

1998 **Furby** – The first "pet" toy robot for children

1999 **Kismet** - Emotional AI, (MIT AI Lab)

1999 **AIBO** introduced The first AI domestic robot by Sony

2000 ASIMO, humanoid robot released by Honda

2002 Roomba, autonomous robot vacuum is released by i-Robot

2004 The first challenge for autonomous vehicles by DARPA

2004 NASA rovers **Spirit** and **Opportunity** exploring Mars

2005 AI based recommendation engines

2006 "**Machine reading**" unsupervised autonomous understanding of text

2007 **ImageNet** – visual database for object recognition software research

2007 NVIDIA launches **CUDA**, a parallel computing platform and programming interface

2009 Self Driving Car built by Google. By 2014 it passed Nevada's self-driving test.

2009 AI researchers discover **GPU** (Graphics Processing Unit) for DL

2010 Democratize Data Access begins for Image Recognition

2010 Narrative Science's AI demonstrates ability to write reports

2011 Apple released **Siri**

2011 IBM's Watson wins Jeopardy clash

2013 **NEIL** by CMU, a semantic image analyzer ML system

1913 "**Vicarious**" passes first Turing test - CAPTCHA

2014 Microsoft - **Cortana** Amazon – **Alexa**

2015 Google Brain releases **TensorFlow**, ML Library, (TPU)

2015 **Open AI-** open source initiative to develop AI benefit of all humanity

2016 Google released **Google Home**

2016 Google's Deepmind AlphaGo has defeated Go's No1. champion

2016 NVIDIA announces supercomputer for DL and AI

The 3rd. Period 2010-

The age of machine learning. Computers acquire knowledge from data, not humans. Large tech companies invest in commercial applications of AI/ML.

2016 **Sophia** humanoid robot by Hanson Robotics, the first "robot citizen"

2016 **PyTorch** Open source ML Library

2017 The Facebook AI Research Lab trained two chatbots to communicate with each other in order to learn how to negotiate. The chatbots diverged from human language and invented their own language to communicate with one another

2017 **Caffe** Open source DL framework

2018 **BERT** (Google), the first bidirectional, unsupervised language representation

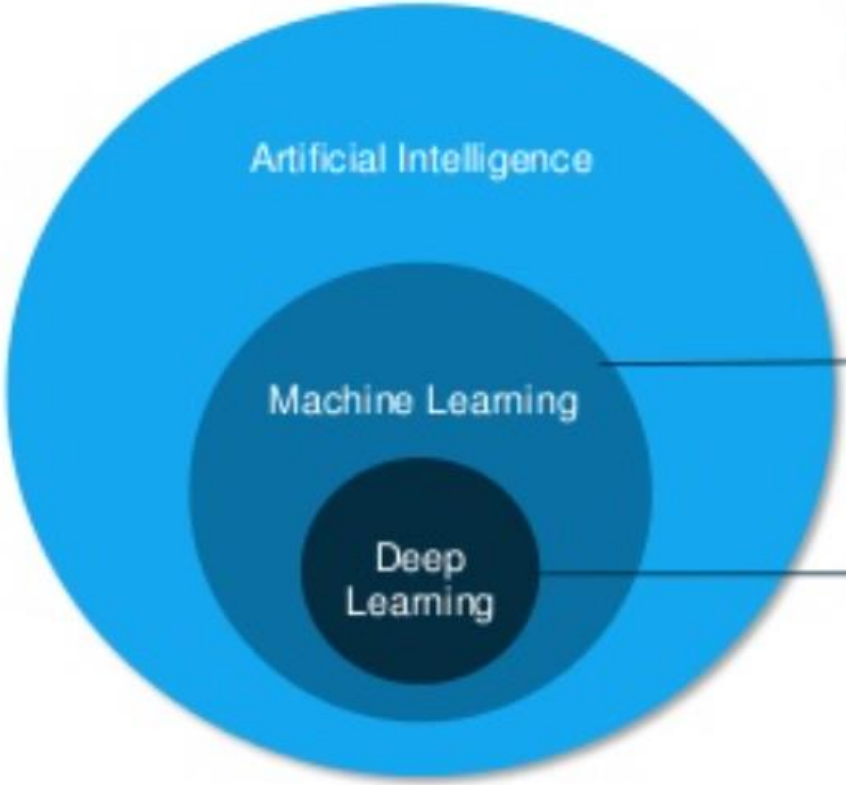
2018 Samsung introduces **Bixby**

2018 Facebook detects faces and shares photos with friends to whom those photos belong

2018 Alibaba language processing AI outscored human intellect at a Stanford reading and comprehension test

2020 DeepMind team uses DL algorithms (Agent57) that outperforms humans at Atari games with deep reinforcement learning

2020 Widespread **5G** network deployments worldwide

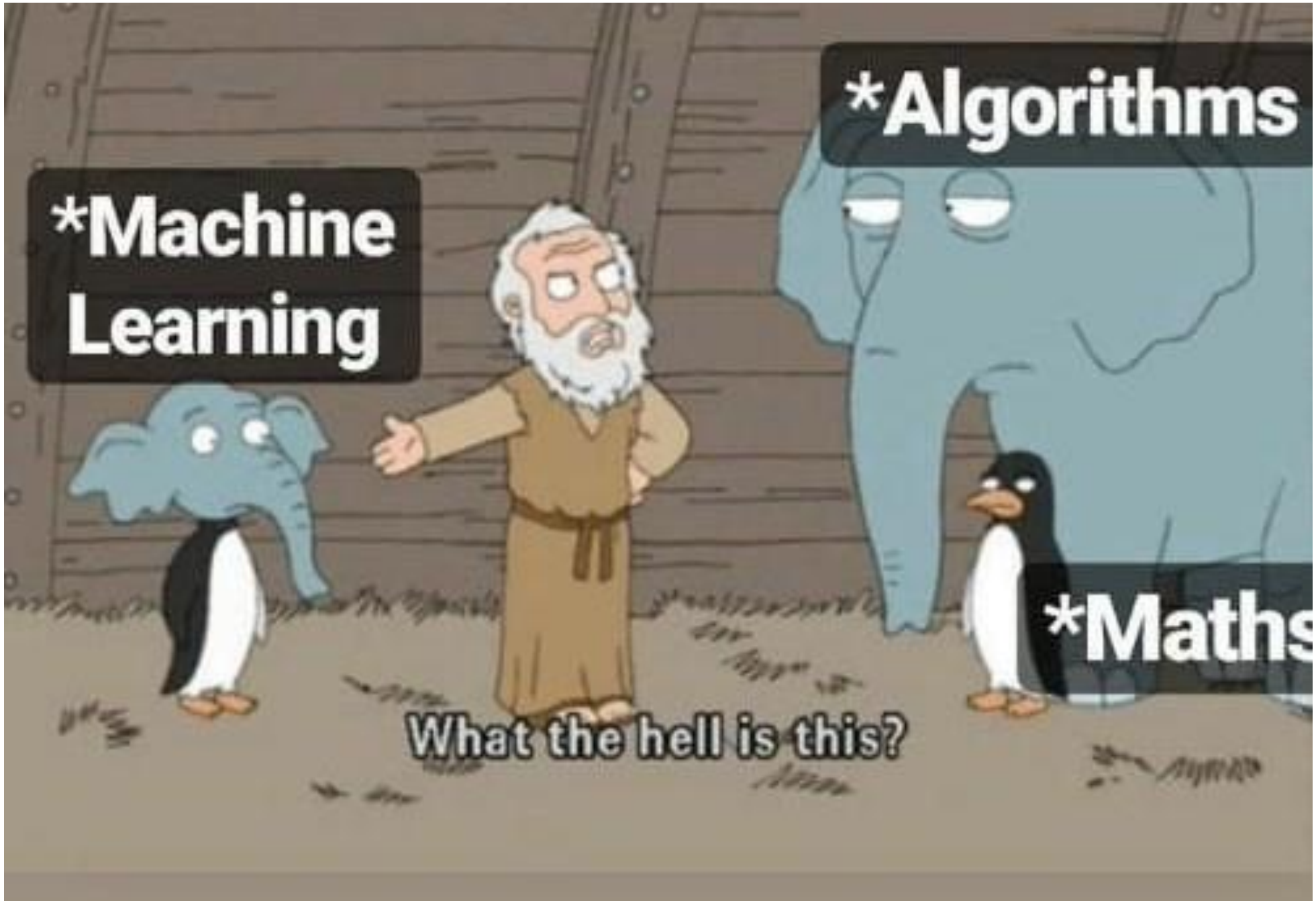


We'll learn more about Deep Learning when we discuss Deep networks and Neural networks in module 2 and 3

Machine Learning is a subset of AI

Deep Learning is a subset of Machine Learning

Deep Learning uses neural networks to simulate human like decision making


*Machine Learning
*Algorithms
*Maths
What the hell is this?

Analisi di rischio

Recommender system

Elaborazione del linguaggio parlato

Riconoscimento di oggetti

Veicoli a guida autonoma

Fraud detection

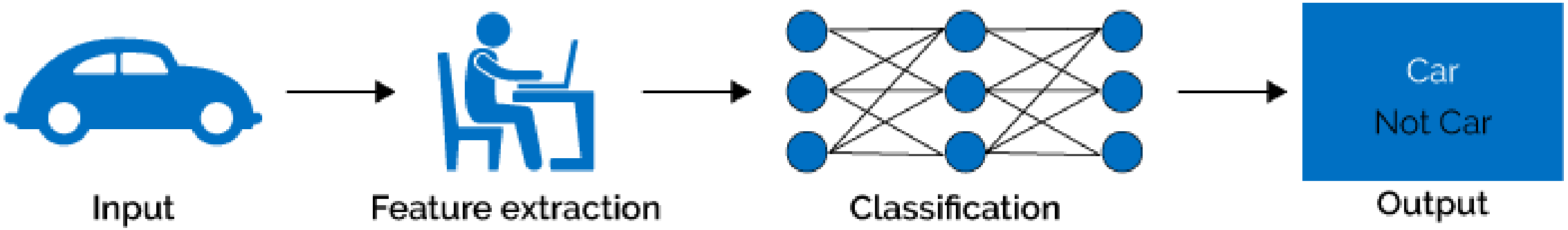
Diagnosi mediche

Customer segmentation

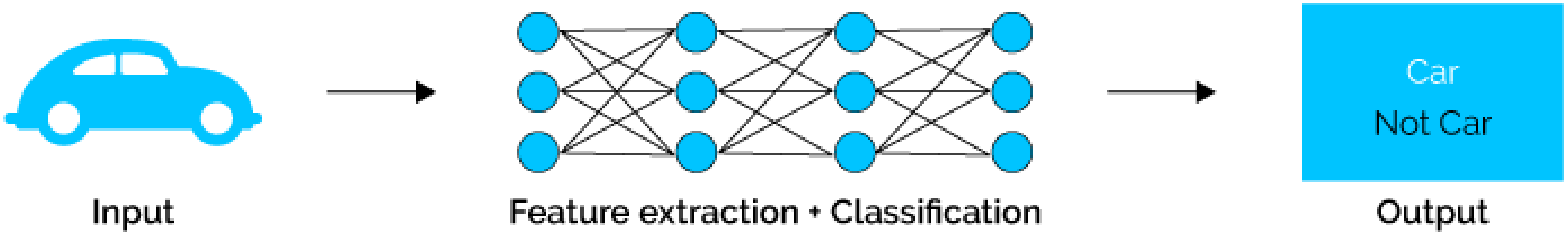

becomes
continue
models
machine
AI
use
still
COVID
IoT
applications
year
work
about
technology
used
time
people
data
science
value
model
start
healthcare
GPT
digital
being
business
new
management
another
NLP
Deep
Language
real
life
need
MLOps
Analytics
issues
learning
organizations


CAN I KNOW WHAT IS
DEEP LEARNING?

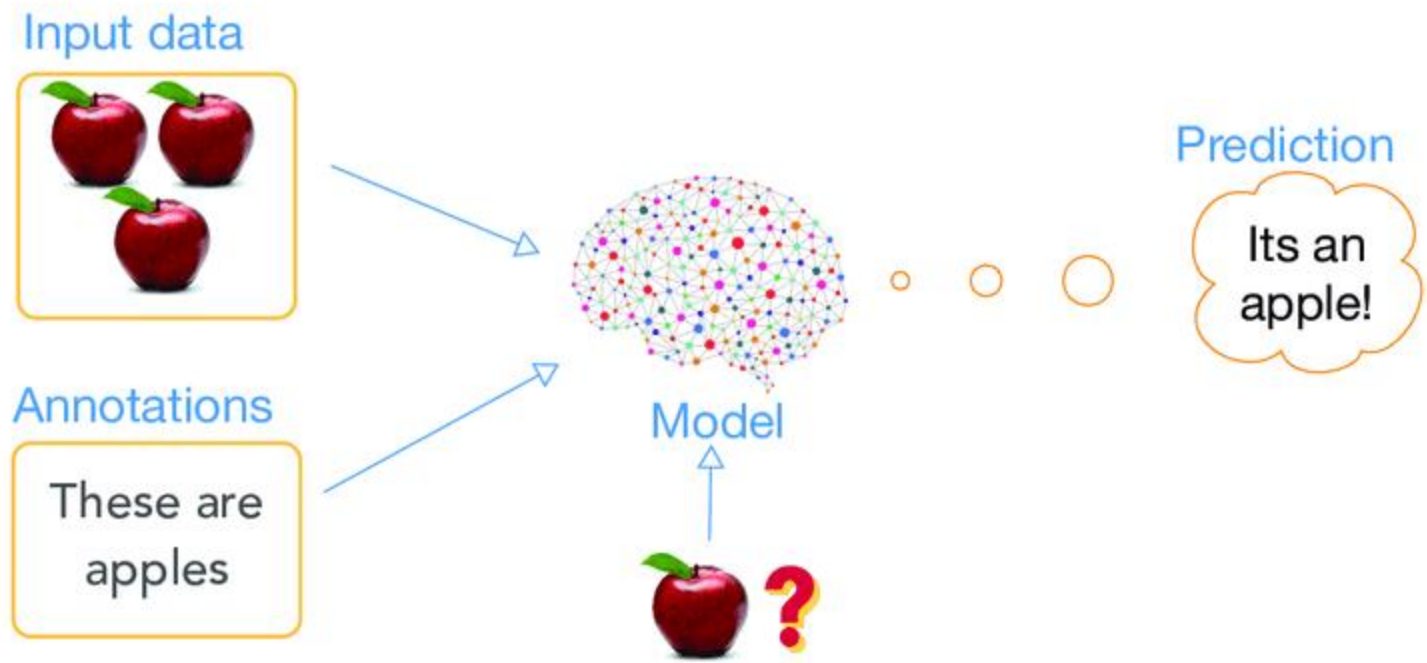
Machine Learning



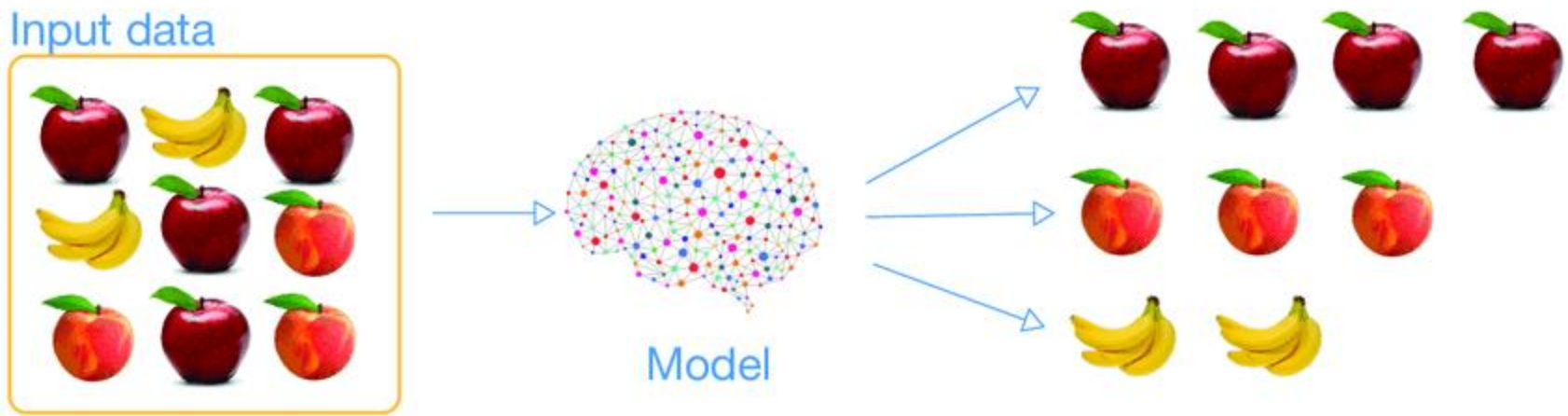
Deep Learning

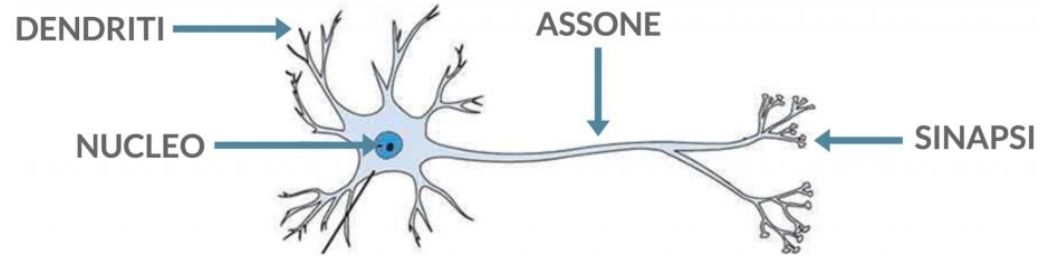


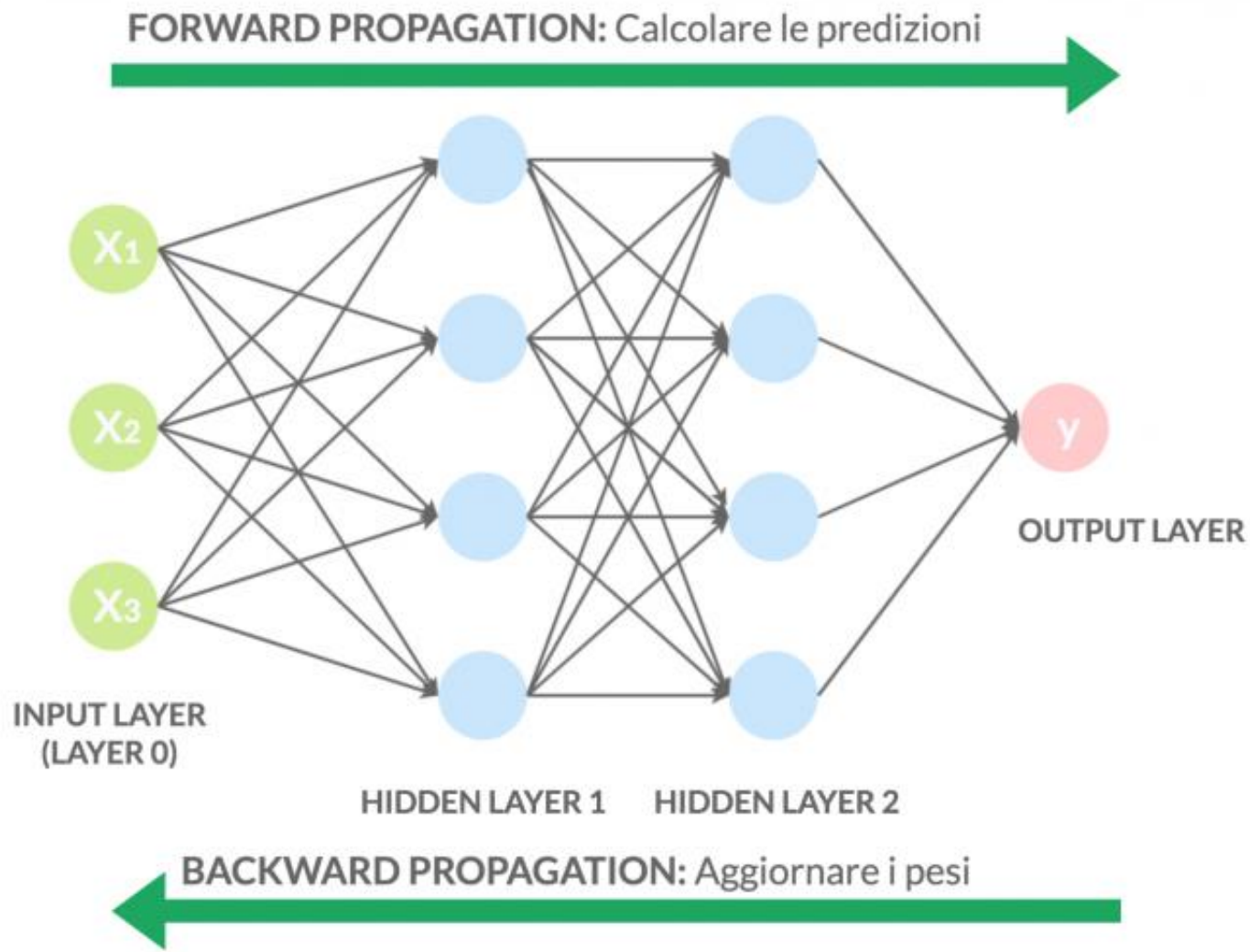
supervised learning



unsupervised learning

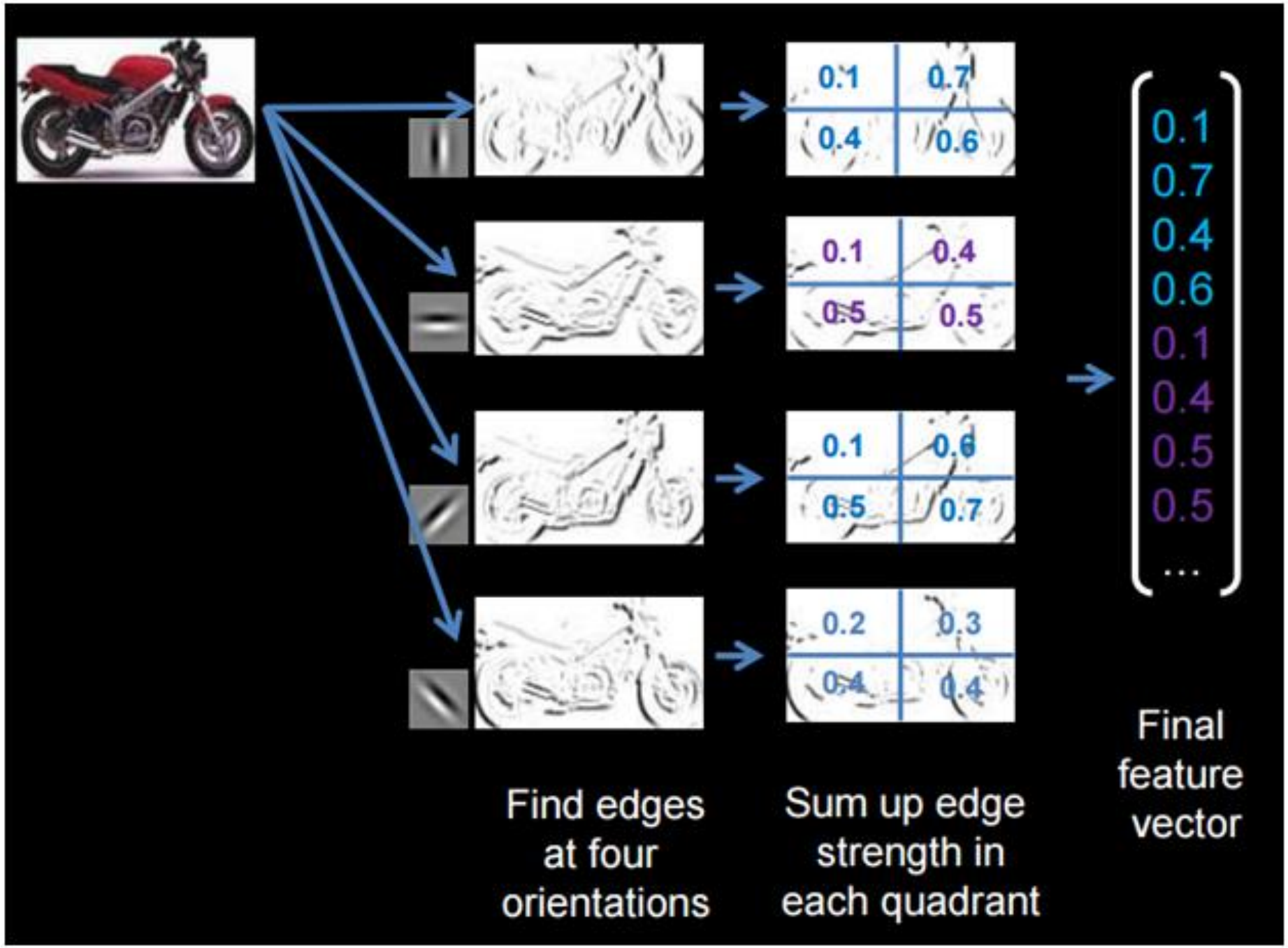




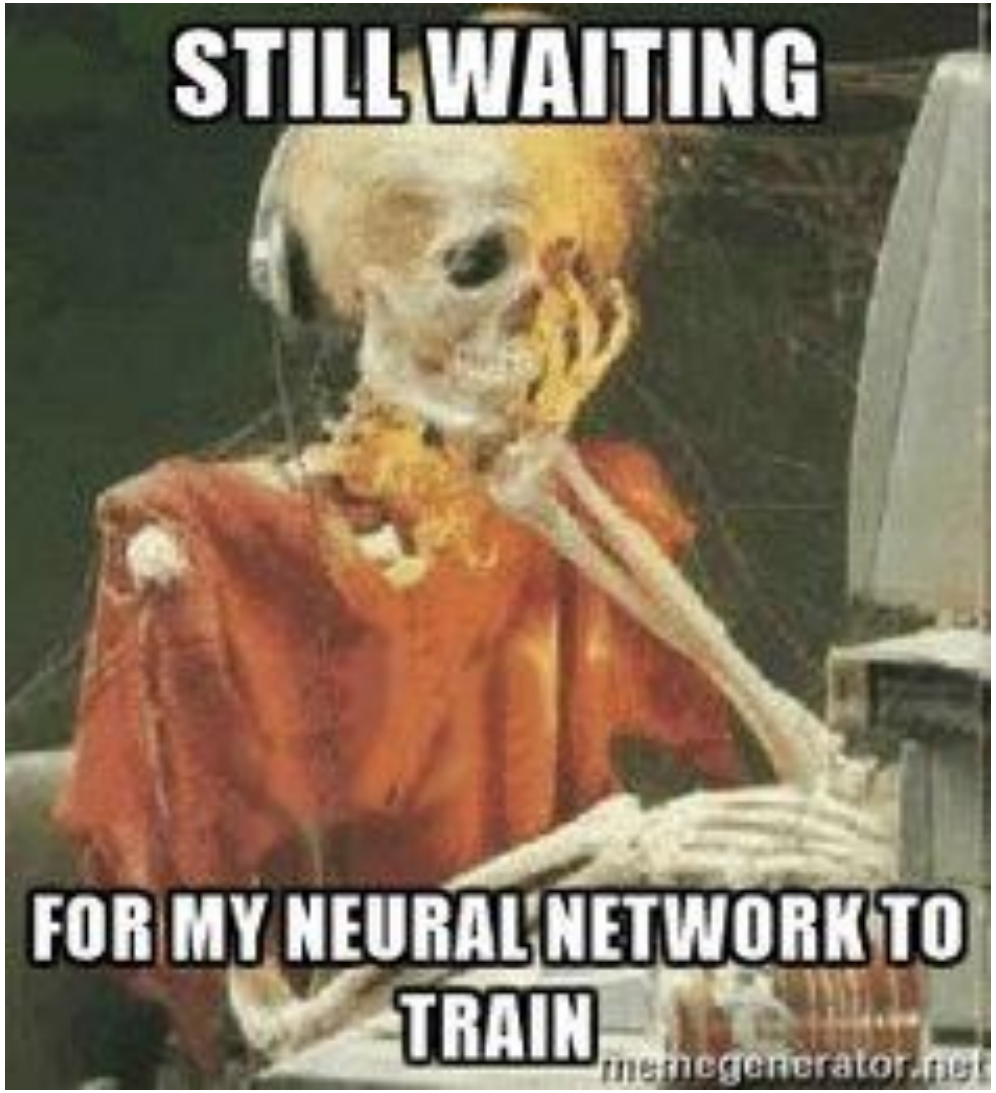


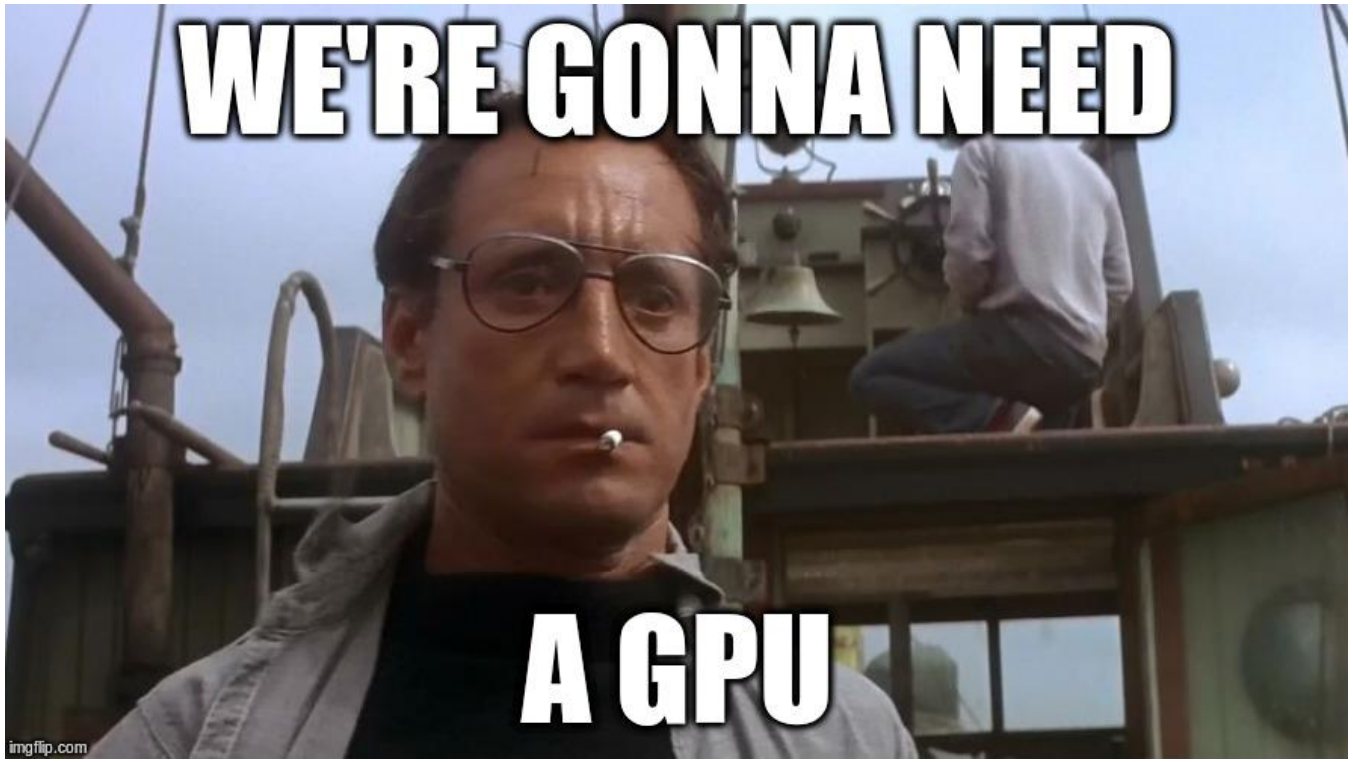
PROBLEMA
Una rete neurale ha un enorme numero di pesi, per eseguire il gradient descend dobbiamo sapere quanto ogni peso di ogni layer ha influenzato l'errore.

SOLUZIONE: BACKPROPAGATION
Propagando all'indietro l'errore la backpropagation ci permette di sapere in che quantità ogni nodo di ogni layer ha influito sull'errore e possiamo utilizzare questi valori per aggiornare i pesi eseguendo il gradient descend.


0.1
0.7
0.4
0.6
0.1
0.4
0.5
0.5
0.1
0.6
0.5
0.7
0.2
0.3
0.4
0.4
0.1
0.7
0.4
0.6
0.1
0.4
0.5
0.5
0.1
0.6
0.5
0.7
0.2
0.3
0.4
0.4
…
Find edges at four orientations
Sum up edge strength in each quadrant
Final feature vector

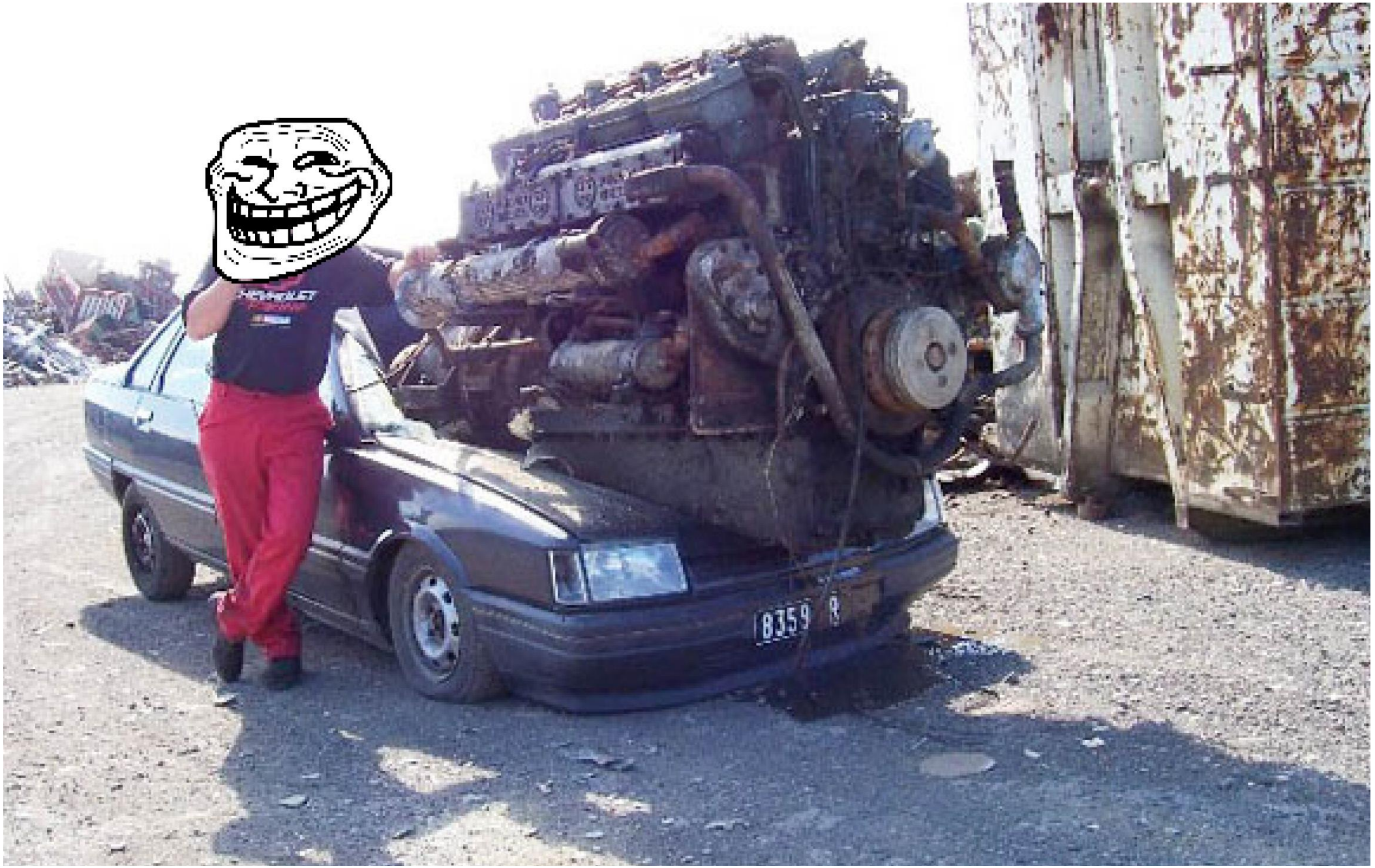

HEAVY

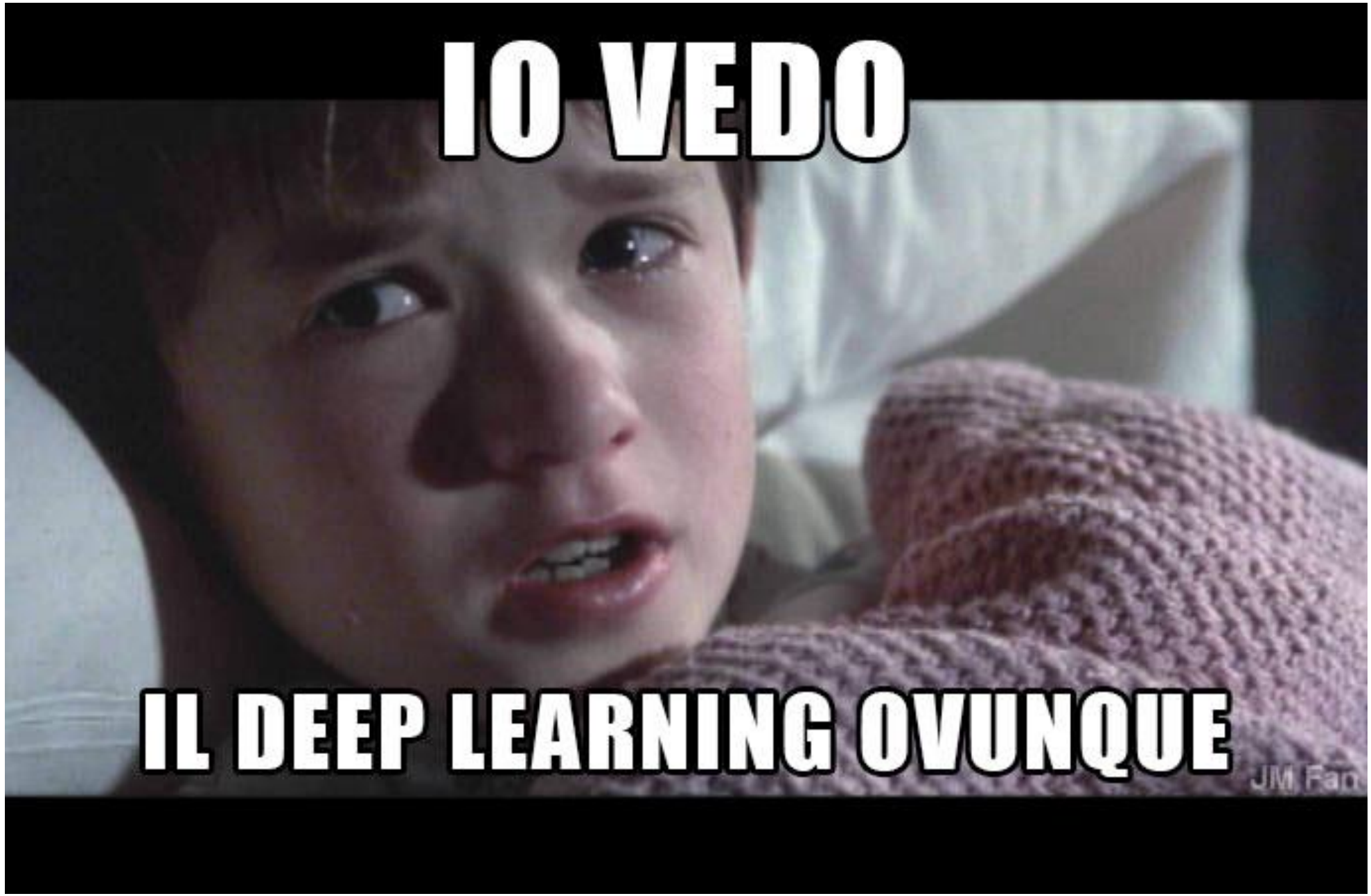

STILL WAITING
FOR MY NEURAL NETWORK TO TRAIN
memegenerator.net

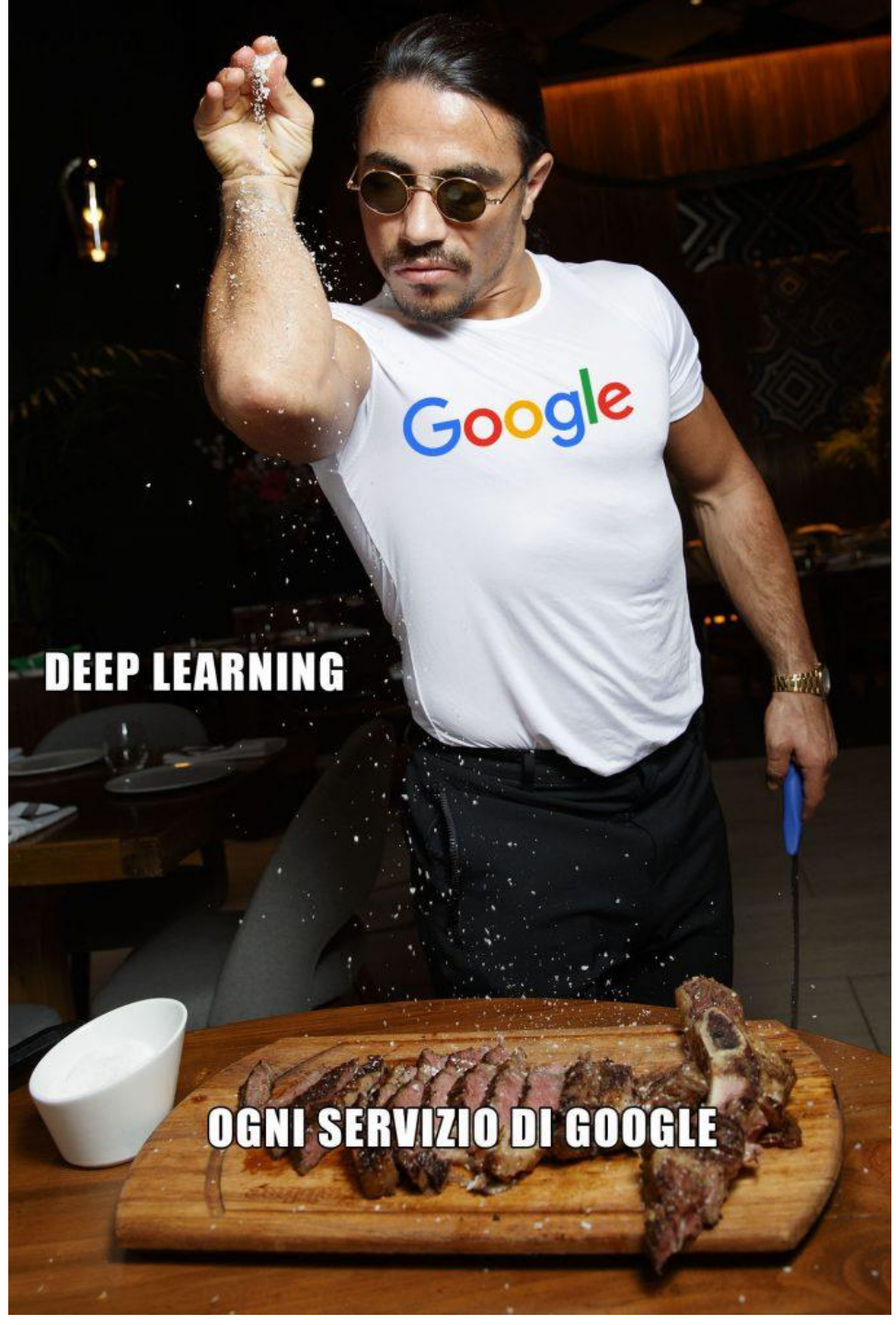

WE'RE GONNA NEED
A GPU
imgflip.com

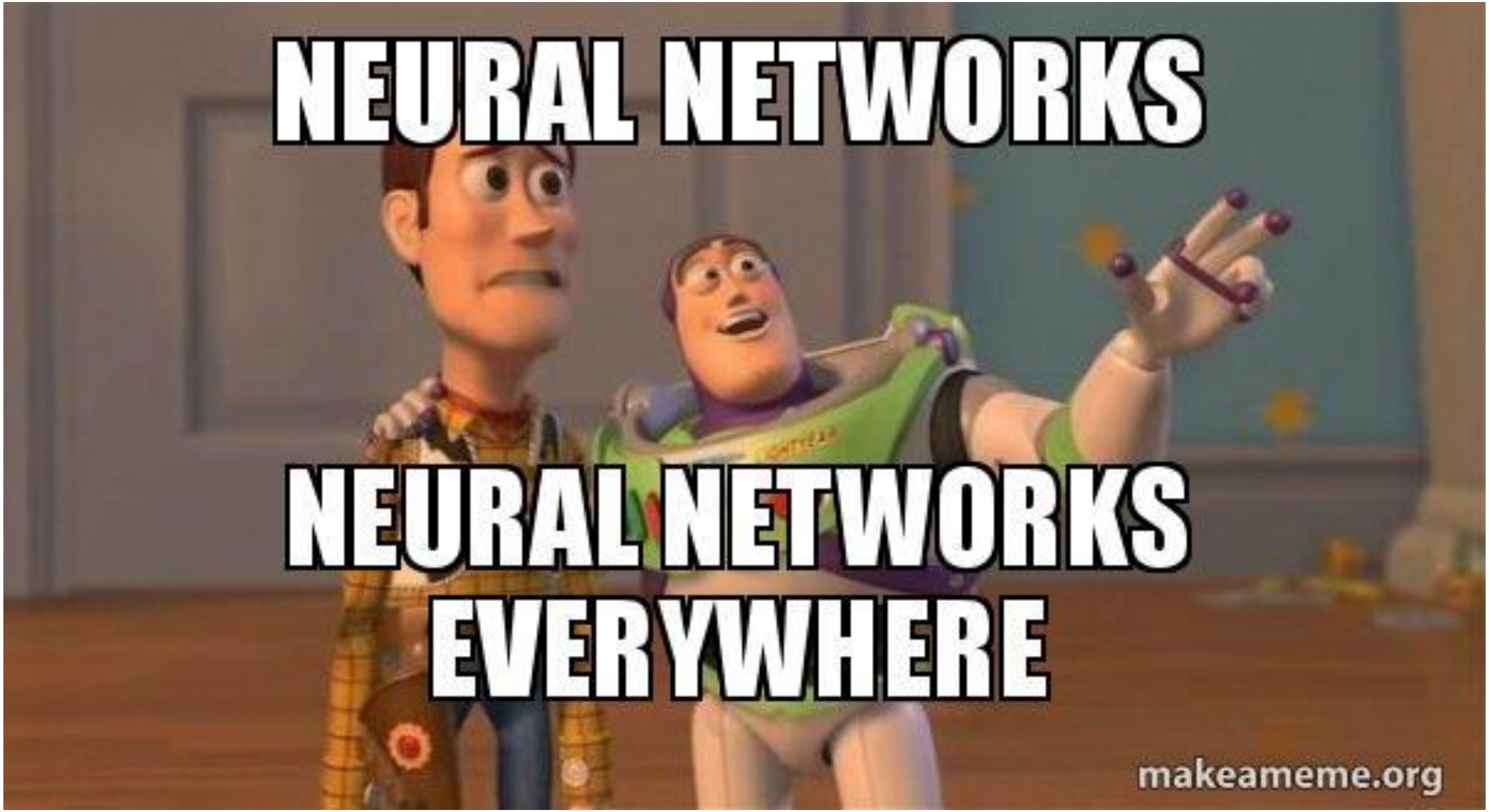
The GPU War

WHAT DOES NEURAL NETWORKS MEAN TO ME?




IO VEDO
IL DEEP LEARNING OVUNQUE
JM Fan

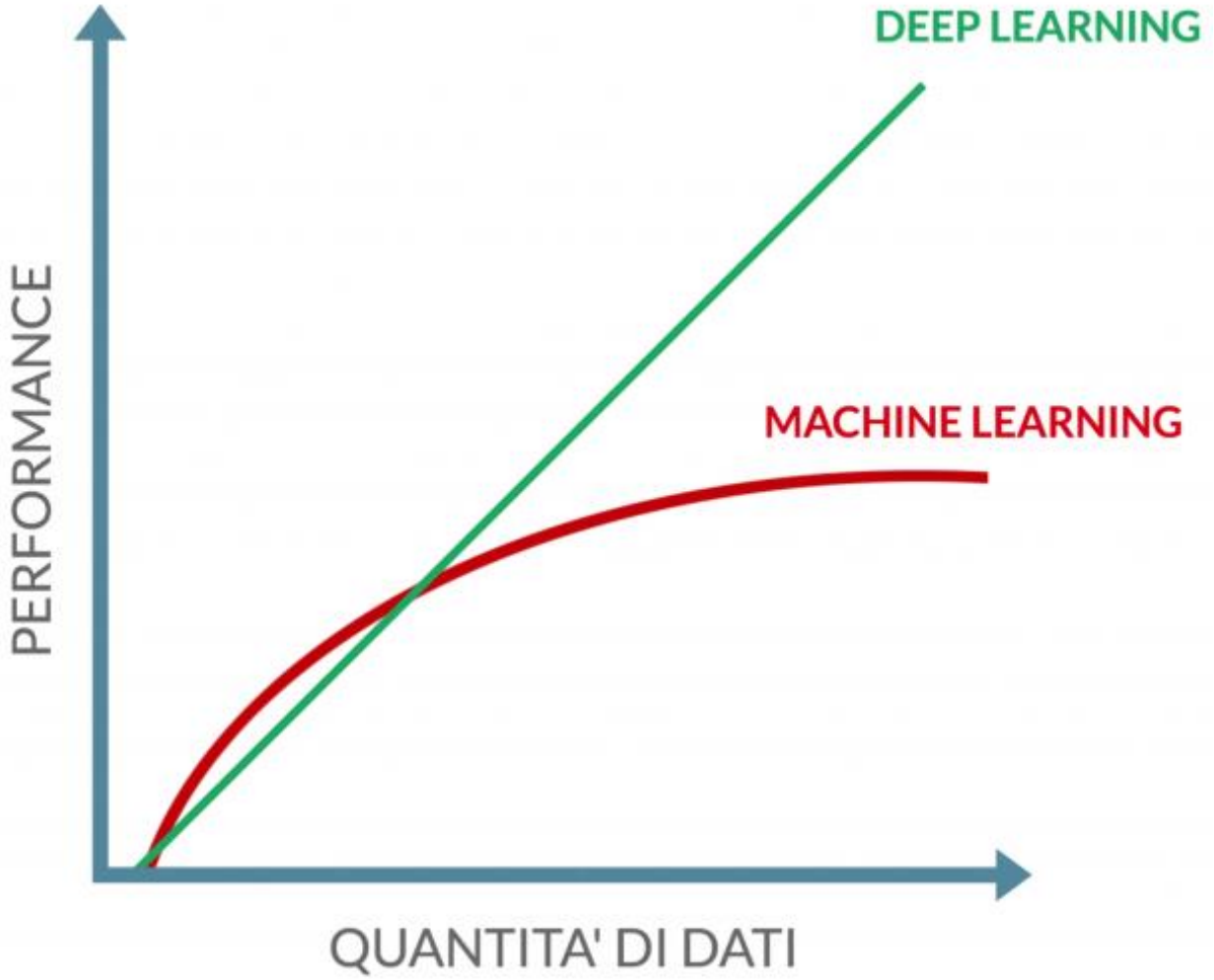

Google
DEEP LEARNING
OGNI SERVIZIO DI GOOGLE


NEURAL NETWORKS
NEURAL NETWORKS EVERYWHERE
makeameme.org

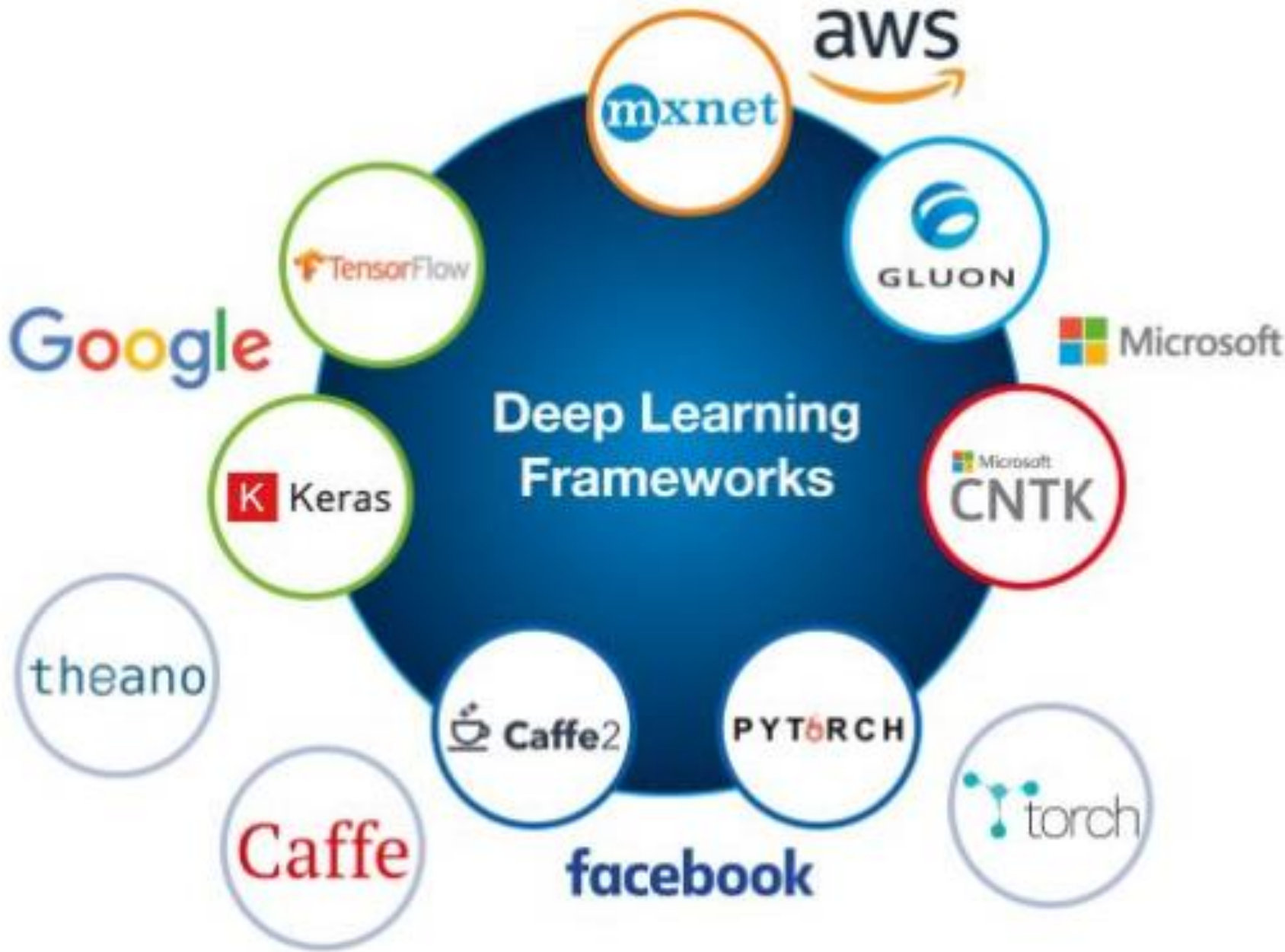
My Answer.....

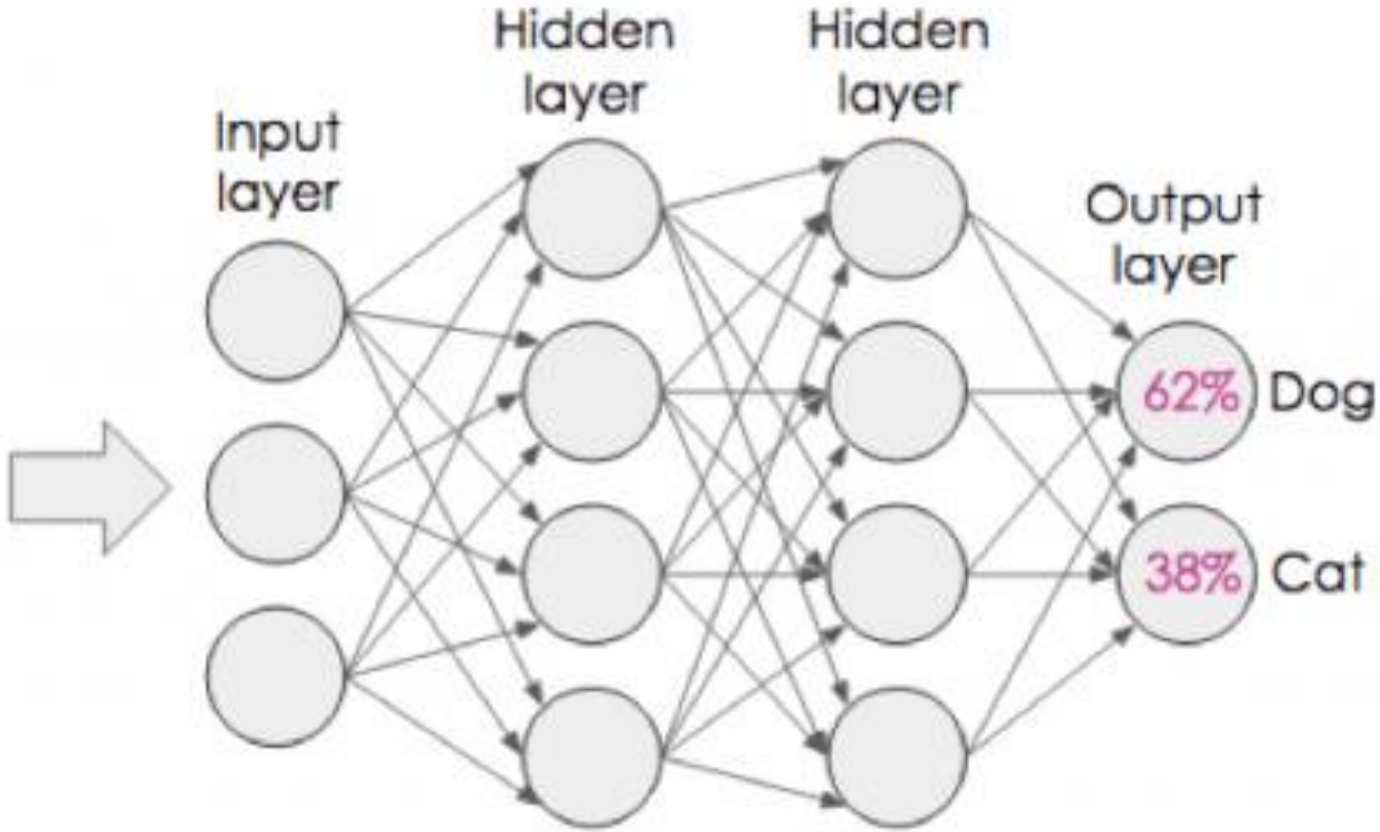

BAD DATA
BAD DATA EVERYWHERE
makeameme.org

SIMPLE PROBLEM WITH NEURAL NETWORKS

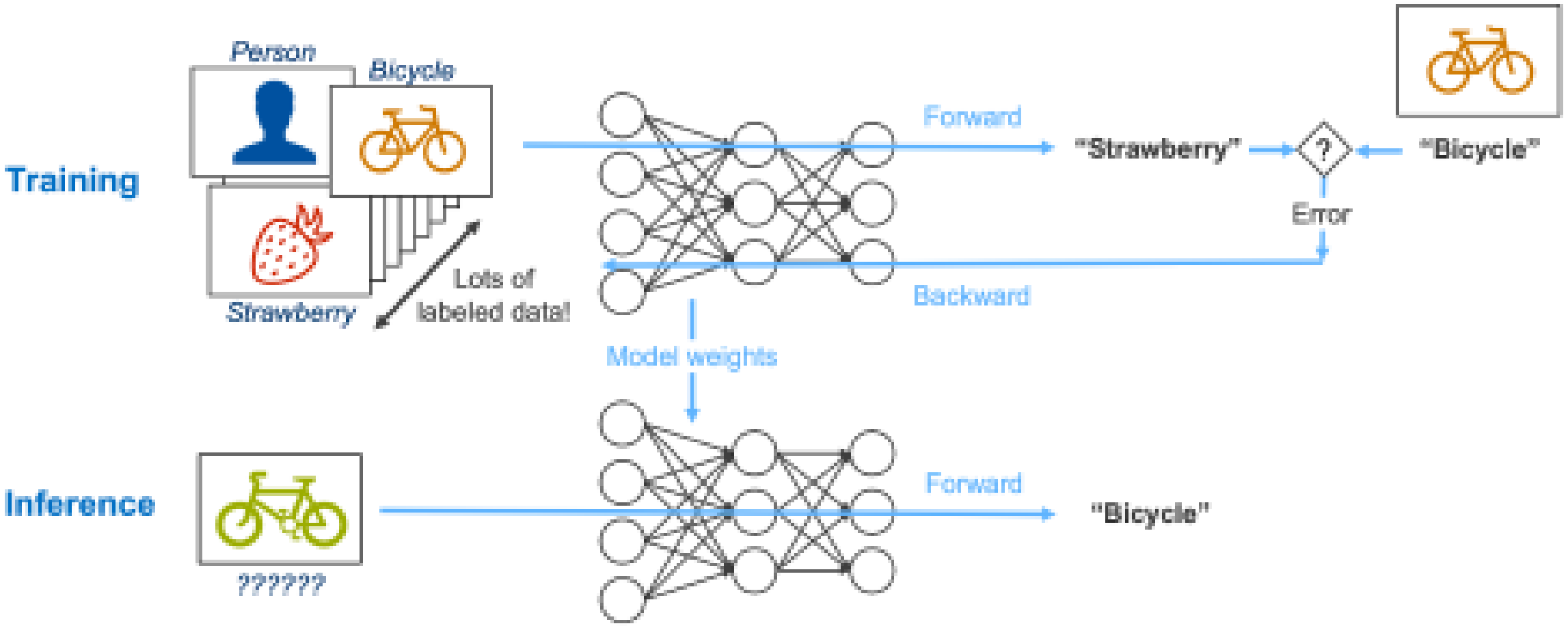


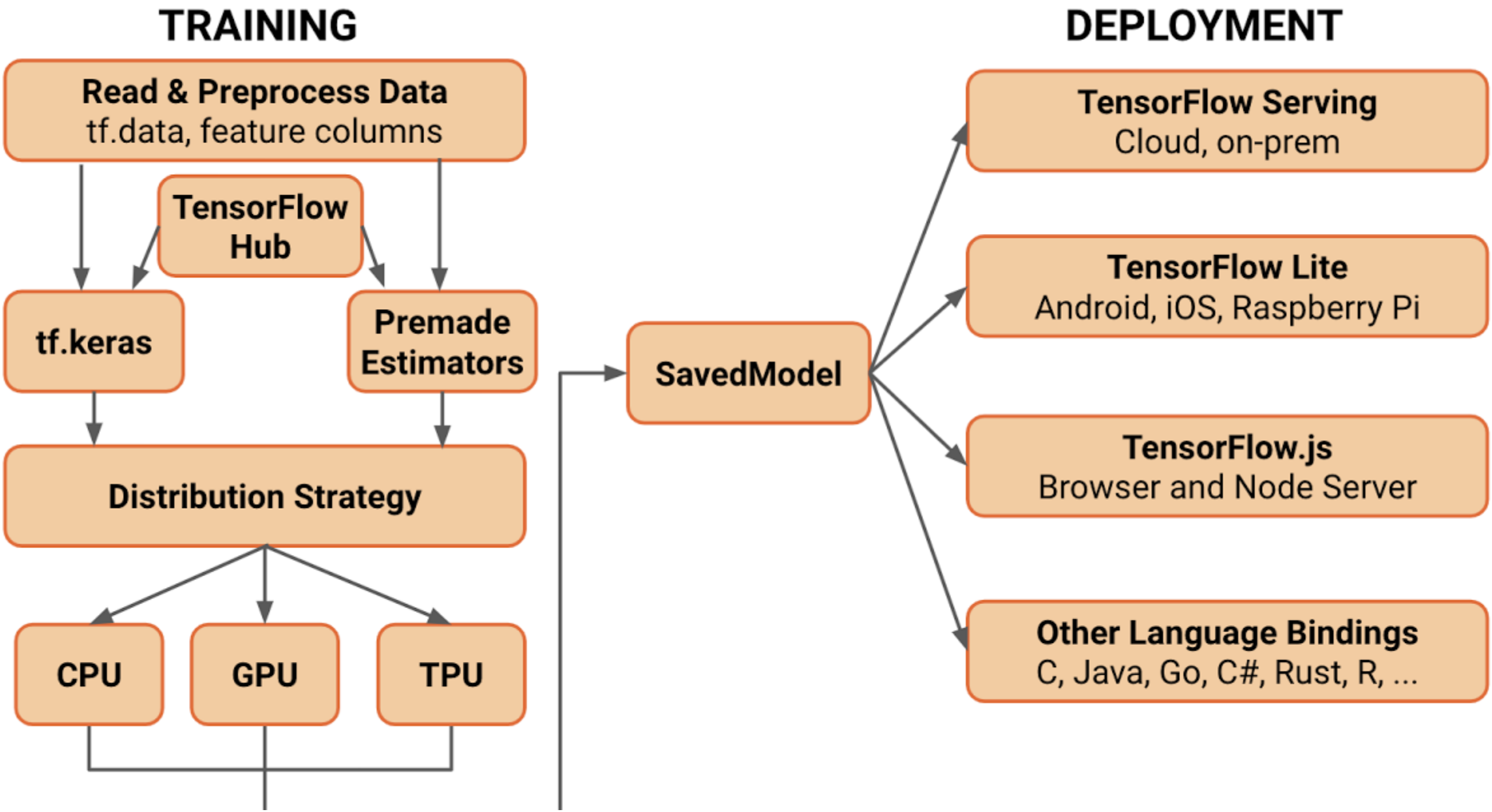

SAY DEEP LEARNING
ONE MORE TIME

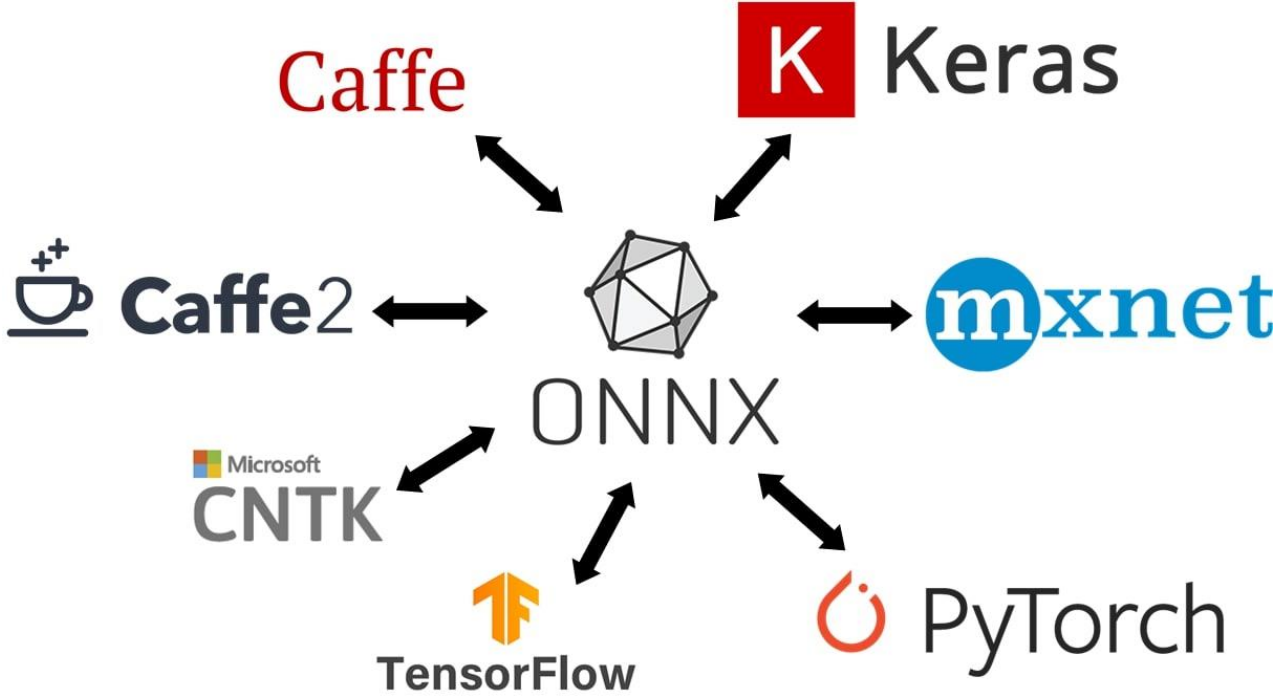

aws
mxnet
GLUON
Microsoft
TensorFlow
Google
Deep Learning
Frameworks
K Keras
Microsoft
CNTK
theano
Caffe2
PYTORCH
torch
Caffe
facebook



It should be
100% Cat :(

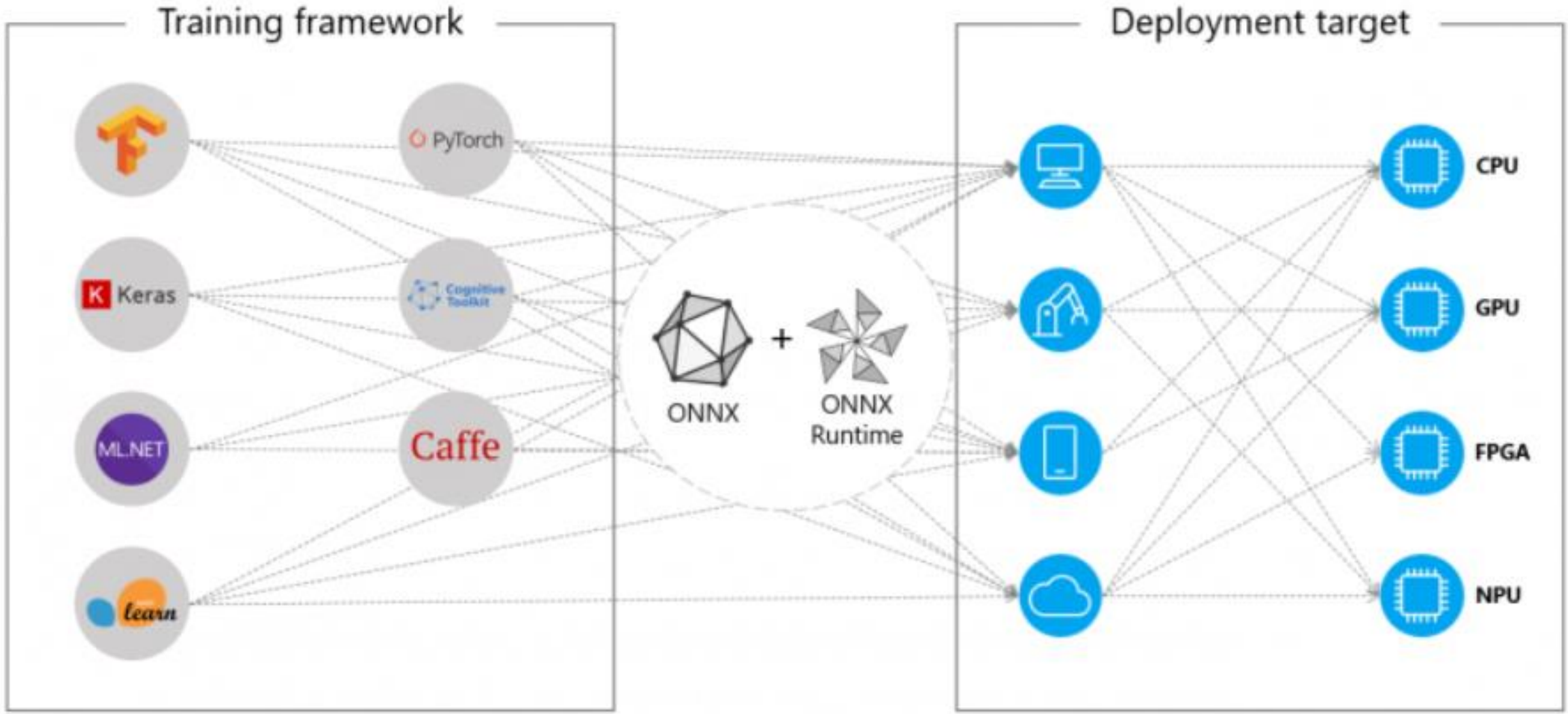

Training
Person
Bicycle
Strawberry
Lots of
labeled data!
Forward
"Strawberry"
?
"Bicycle"
Error
Backward
Model weights
Inference
??????
Forward
"Bicycle"

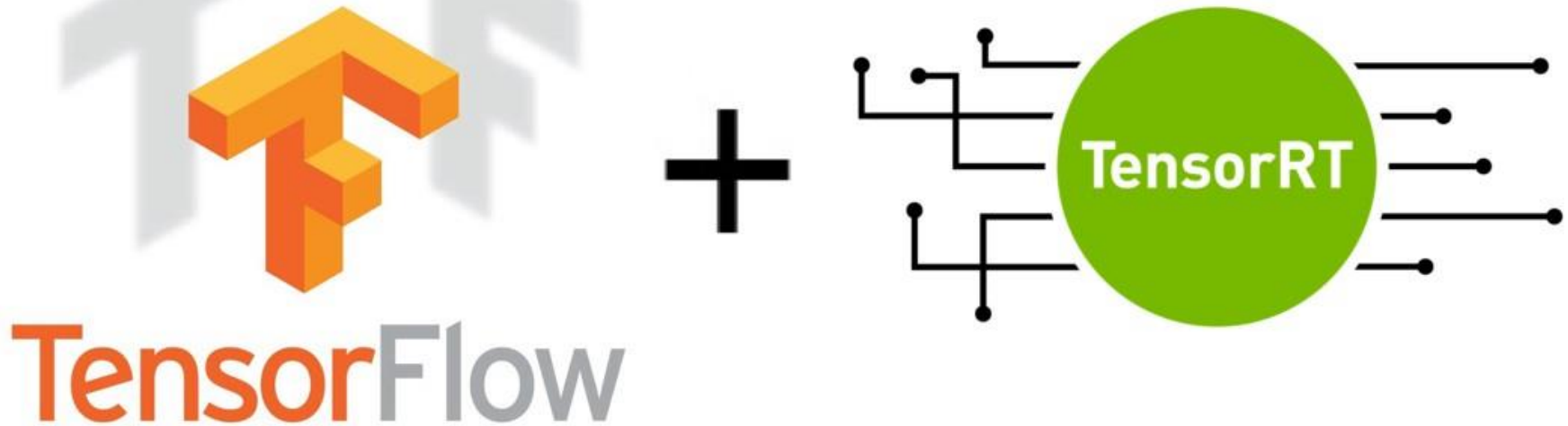

TRAINING
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
DEPLOYMENT
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...



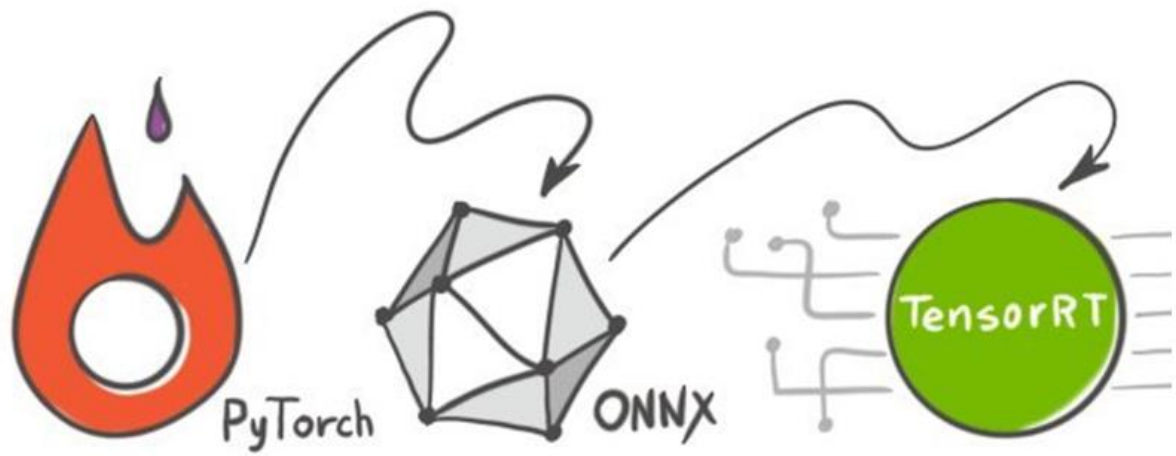
ONNX converters for popular frameworks

- Native Support
  - Pytorch
  - CNTK
- Open sourced Converter Tools
  - Tensorflow: onnx/tensorflow-onnx
  - Keras: onnx/keras-onnx
  - Scikit-learn: onnx/sklearn-onnx
  - CoreML: onnx/onnxmltools
  - LightGBM: onnx/onnxmltools
  - LibSVM: onnx/onnxmltools
  - XGBoost: onnx/onnxmltools


Training framework
PyTorch
Keras
Cognitive Toolkit
ML.NET
Caffe
learn
ONNX
+
ONNX Runtime
Deployment target
CPU
GPU
FPGA
NPU



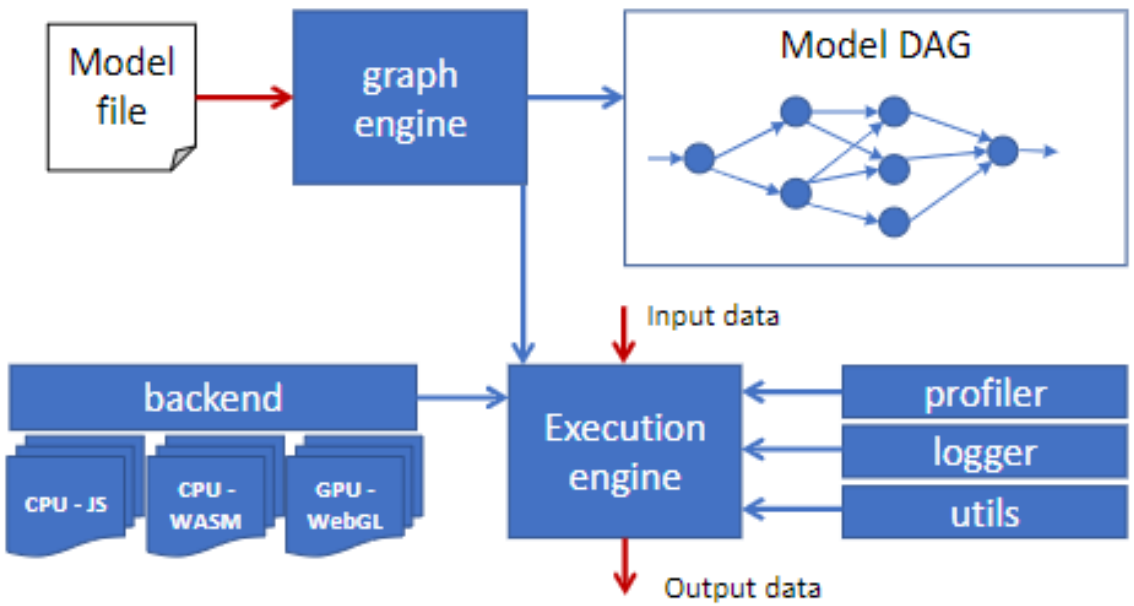
LEARNOPENCV.COM



How To Run Inference Using TensorRT C++ API

ONNX.JS

- A pure JavaScript implementation of ONNX framework
- Optimize ONNX model inference on both CPUs and GPUs
- Support a variety of browsers on major OSes



Desktop Platforms

OS/Browser	Chrome	Edge	FireFox	Safari	Opera	Electron	Node.js
Windows 10	✔	✔	✔	-	✔	✔	✔
macOS	✔	-	✔	✔	✔	✔	✔
Ubuntu LTS 18.04	✔	-	✔	-	✔	✔	✔

Mobile Platforms

OS/Browser	Chrome	Edge	FireFox	Safari	Opera
iOS	✔	✔	✔	✔	✔
Android	✔	✔	Coming soon	-	✔

HTML example to use ONNX.js

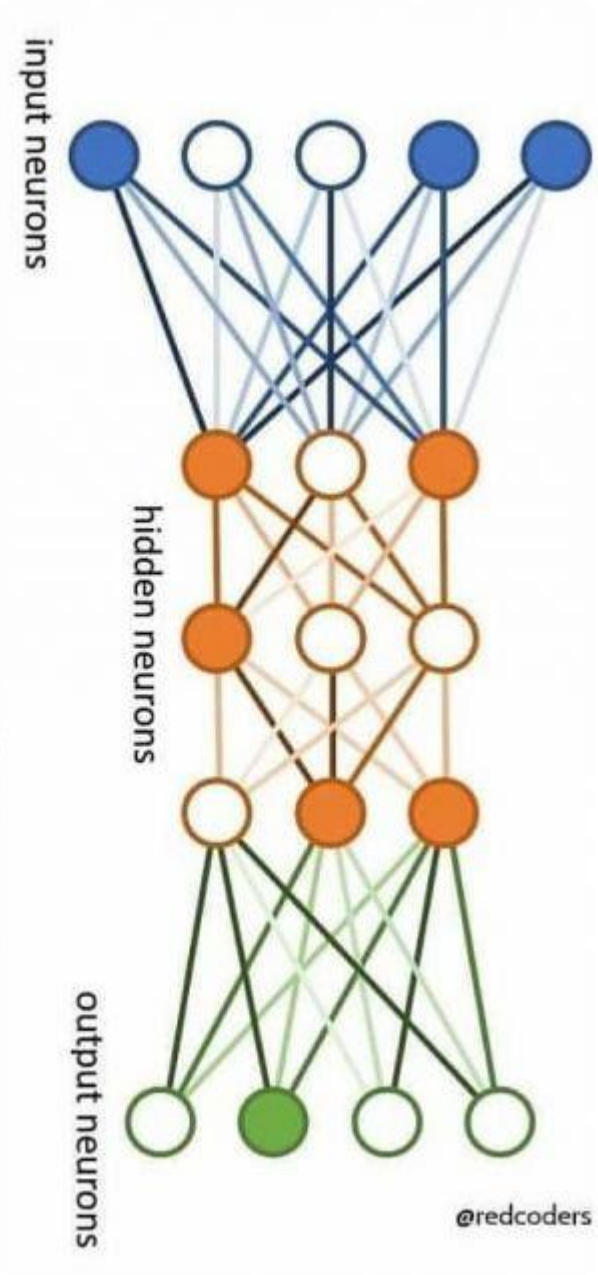
```
<html>
  <head>
  </head>

  <body>
    <!-- load ONNX.js -->
    <script src="https://cdn.jsdelivr.net/npm/onnxjs/dist/onnx.min.js"></script>
    <!-- Code that consume ONNX.js -->
    <script>
      // create a session
      const myOnnxSession = new onnx.InferenceSession();
      // load the ONNX model file
      myOnnxSession.loadModel("./my-model.onnx").then(()=>{
        // generate model input
        const inferenceInputs = getInputs();
        // execute the model
        session.run(inferenceInputs).then(output=>{
          // consume the output
          const outputTensor = output.values().next().value;
          console.log(`model output tensor: ${outputTensor.data}.`);
        });
      })
    </script>
  </body>
</html>
```

THIS IS A NEURAL NETWORK.

IT MAKES MISTAKES.
IT LEARNS FROM THEM.

BE LIKE A NEURAL NETWORK.




SO YOU'RE TELLING ME
MEMES ARE REAL?


That's all Folks!


It's Just
The Beginning