



공개소프트웨어를 활용한 기계학습 (Machine Learning)

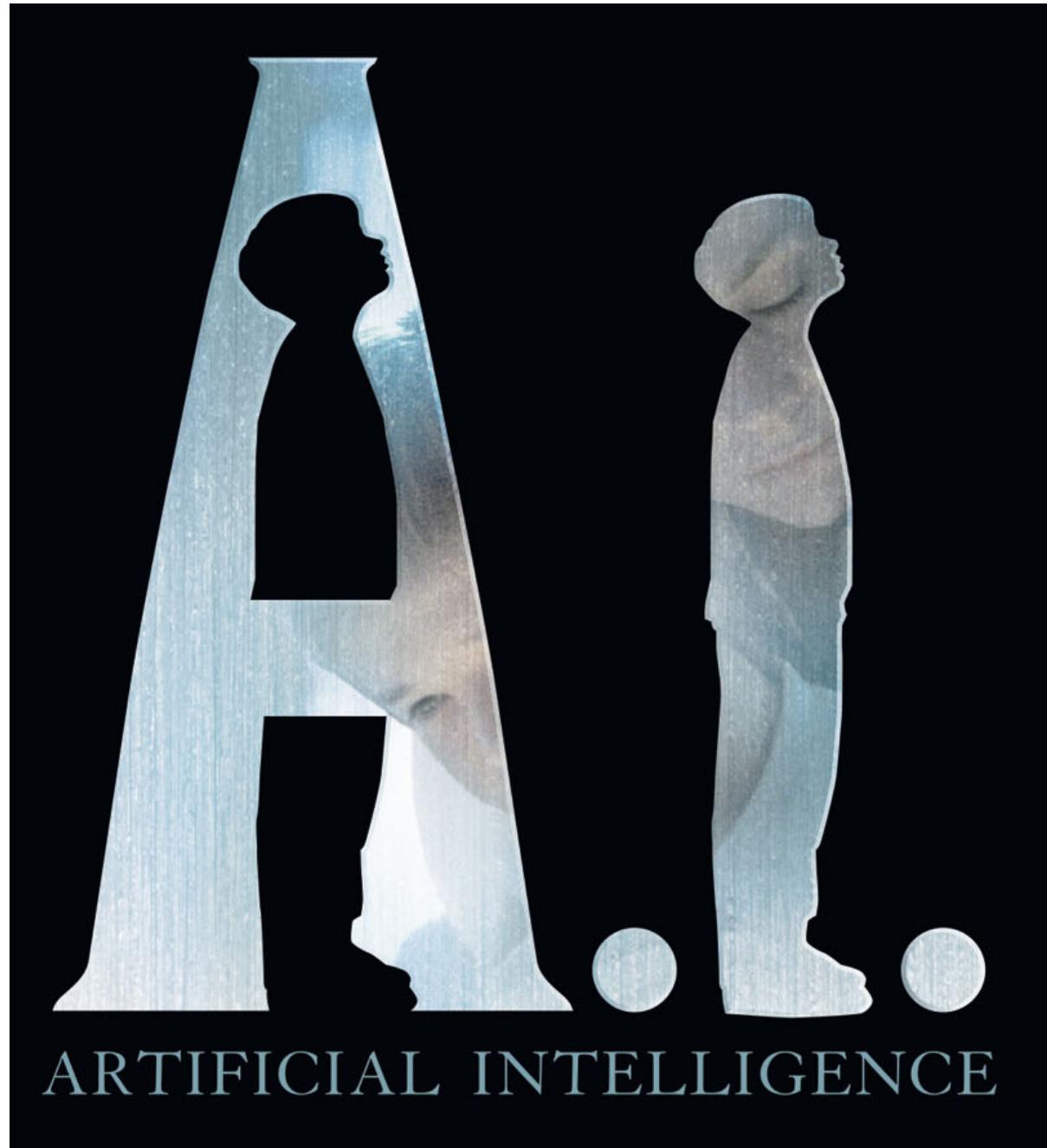
공개소프트웨어 개발자 센터 2기

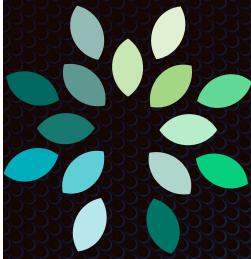
Mario Cho (조만석)

hephaex@gmail.com

Contents

- 기계학습 개요
- 딥러닝 기술
- 오프소스 딥러닝 기술
- 활용방안
- Q&A





Who am I ?

Development Experience

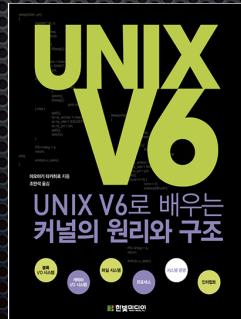
- ◆ Bio-Medical Data Processing based on HPC for Human Brain Mapping
- ◆ Bio-medical data processing based on Neural Network (Machine Learning)
- ◆ Medical Image Reconstruction (Computer Tomography)
- ◆ Enterprise System
- ◆ Open Source Software Developer

Open Source Software Enigeer

- ◆ Linux Kernel & LLVM
- ◆ OpenStack & OPNFV (NFV&SDN)
- ◆ Machine Learning

Technical Book

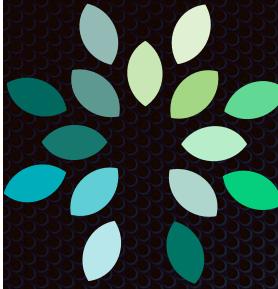
- ◆ Unix V6 Kernel



Open Frontier Lab.

Manseok (Mario) Cho

hephaex@gmail.com



Open Source S/W

developer community

<http://kernelstudy.net>

- Linux Kernel (ARM, x86)
- Machine Learning

120주차 (2015.10.24) vfs_caches_init();

Kernel A10C : 120 주차

hephaex

ARM10C : 120 주차

일시 : 2015.10.24 (120 주차 스터디 진행)

모임명 : NAVER_개발자커뮤니티지원_10차ARM-C

장소 : 토즈 타워점

장소지원 : NAVER 개발자 커뮤니티 지원 프로그램

참여인원 : 3명

=====

120 주차 진도

- 119주차 진도를 복습하였습니다.
- buffer_init()
 - buffer_head 를 사용하기 위한 kmem_cache 할당자 및 max_buffer_heads 값 초기화 수행
- key_init()
 - null function()
- security_init()



The Future of Jobs

The image shows the cover of the 'Global Challenge Insight Report' from the World Economic Forum. The report is titled 'The Future of Jobs: Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution'. It is dated January 2016. The cover features the WEF logo and a grid of small photographs depicting various professionals and workers in different settings.

WORLD
ECONOMIC
FORUM
COMMITTED TO
IMPROVING THE STATE
OF THE WORLD

Global Challenge Insight Report

The Future of Jobs

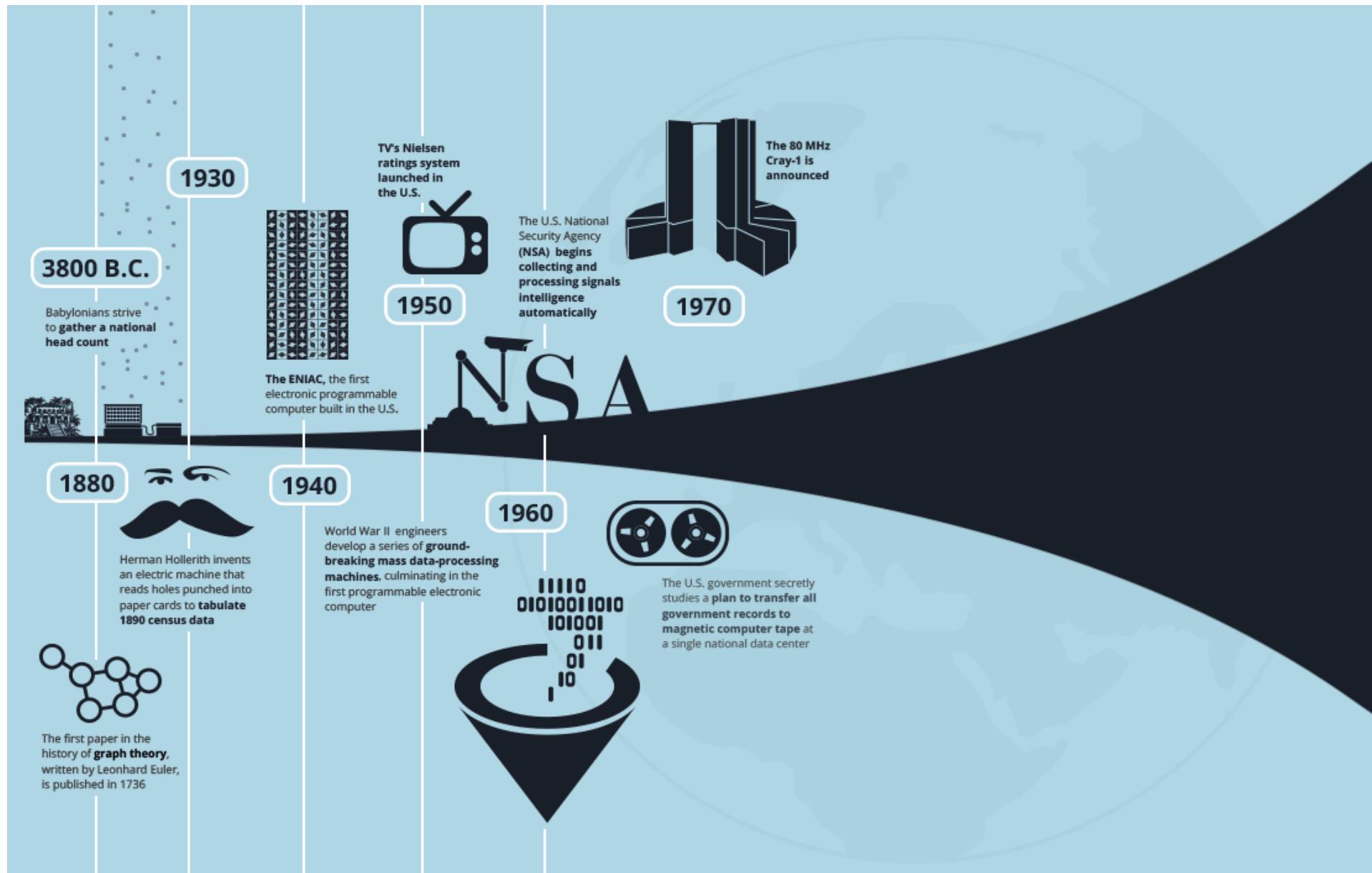
Employment, Skills and
Workforce Strategy for the
Fourth Industrial Revolution

January 2016

“**The Fourth Industrial Revolution**, which includes developments in previously disjointed fields such as **artificial intelligence & machine-learning**, **robotics**, **nanotechnology**, **3-D printing**, and **genetics & biotechnology**, will cause widespread disruption not only to business models but also to labor market over the next **five years**, with enormous change predicted in the skill sets needed to thrive in the new landscape.”

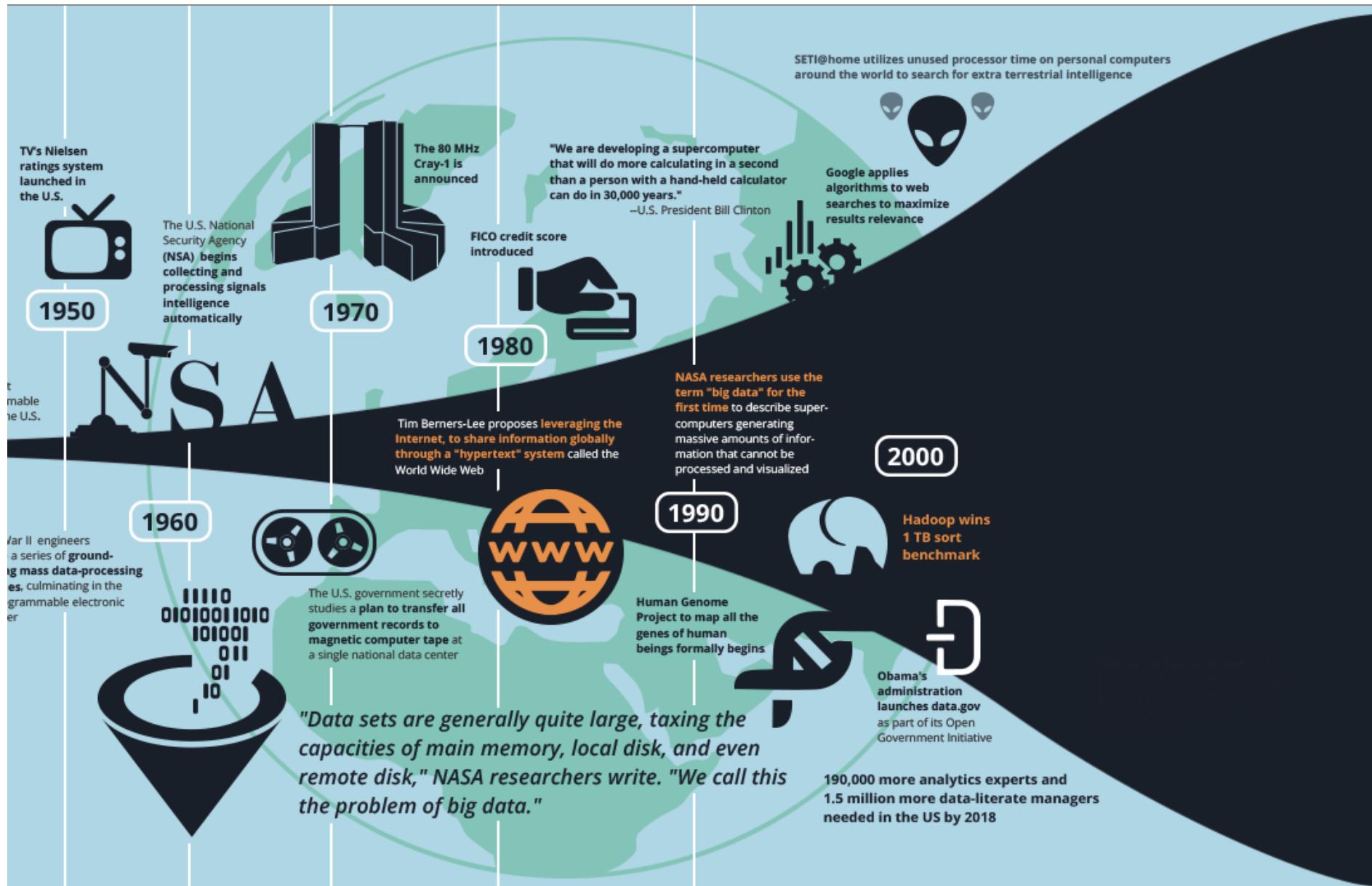


Evolution of information technology



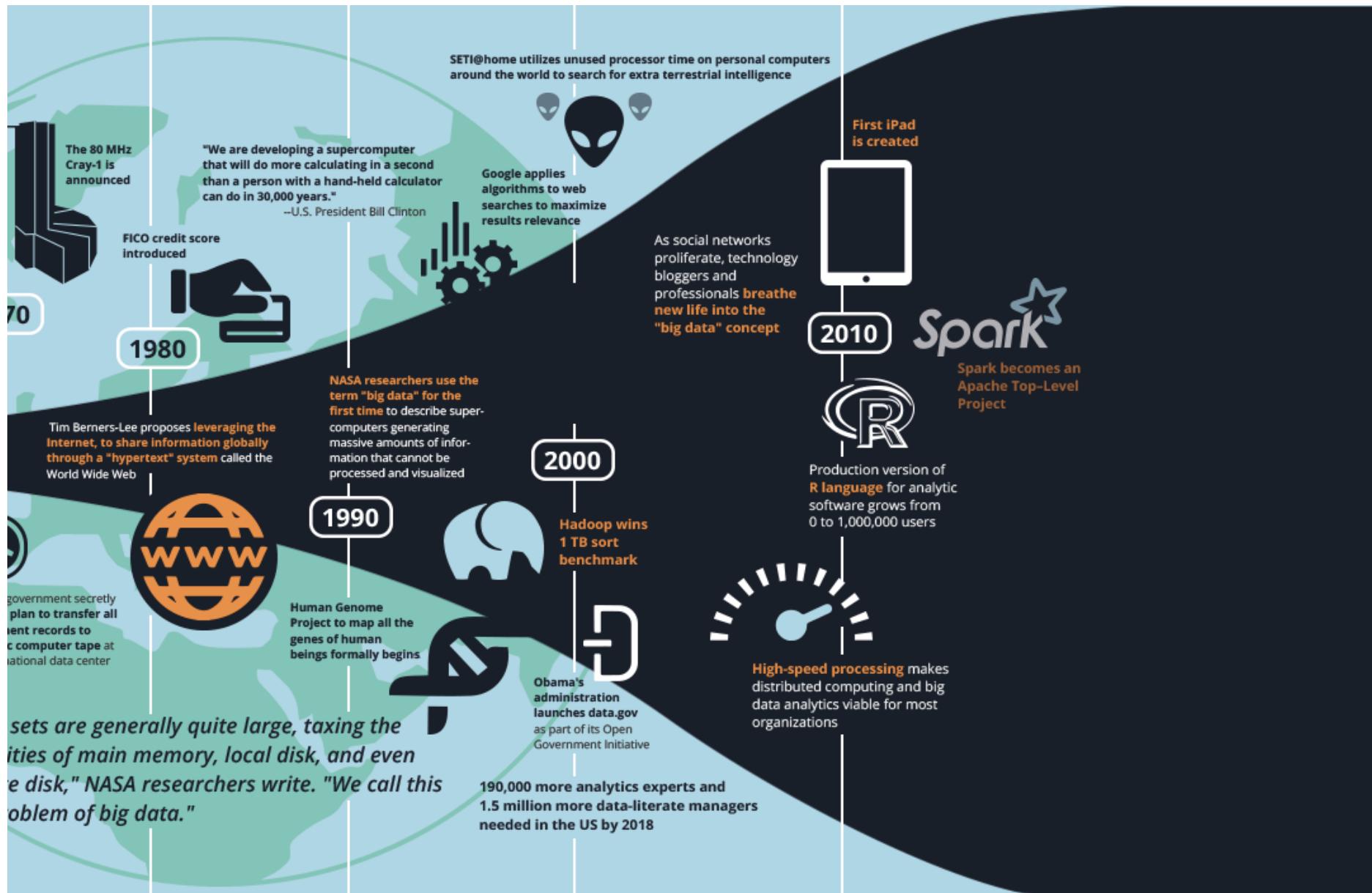
* <http://www.cray.com/Assets/Images/urika/edge/analytics-infographic.html>

Evolution of information technology



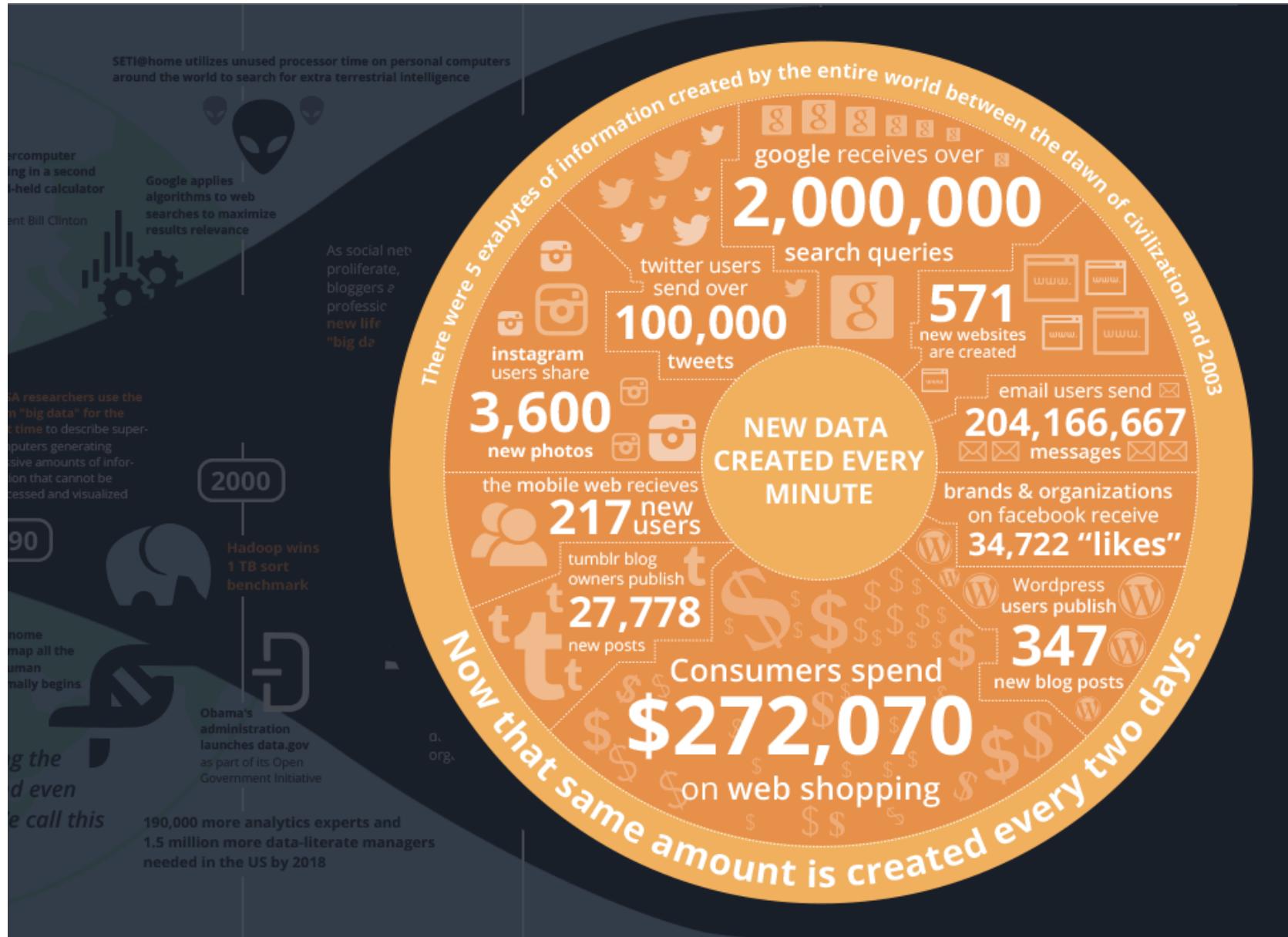
* <http://www.cray.com/Assets/Images/urika/edge/analytics-infographic.html>

Evolution of information technology



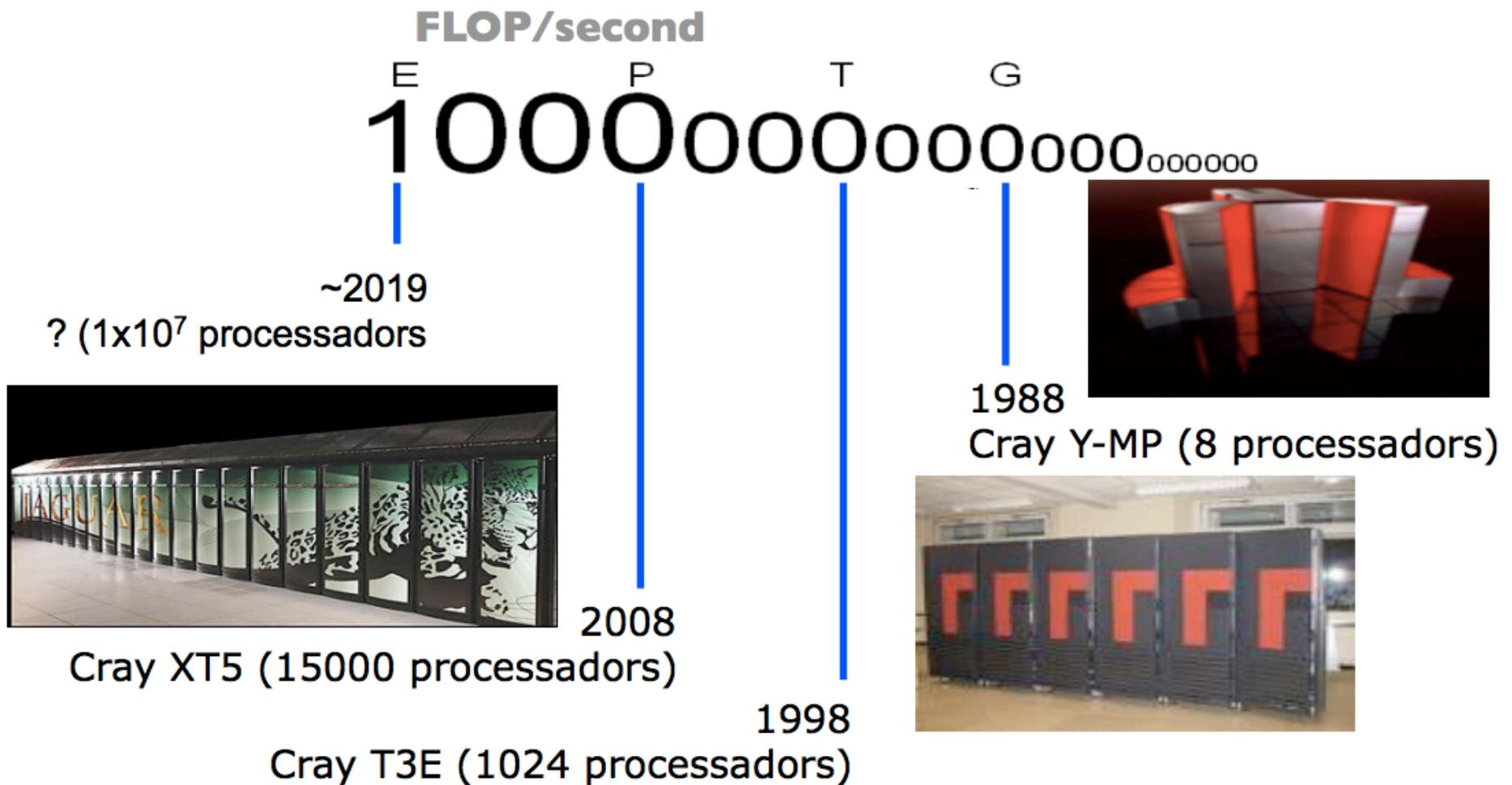
* <http://www.cray.com/Assets/Images/urika/edge/analytics-infographic.html>

Today's information



* <http://www.cray.com/Assets/Images/urika/edge/analytics-infographic.html>

Evolution of Computer hardware

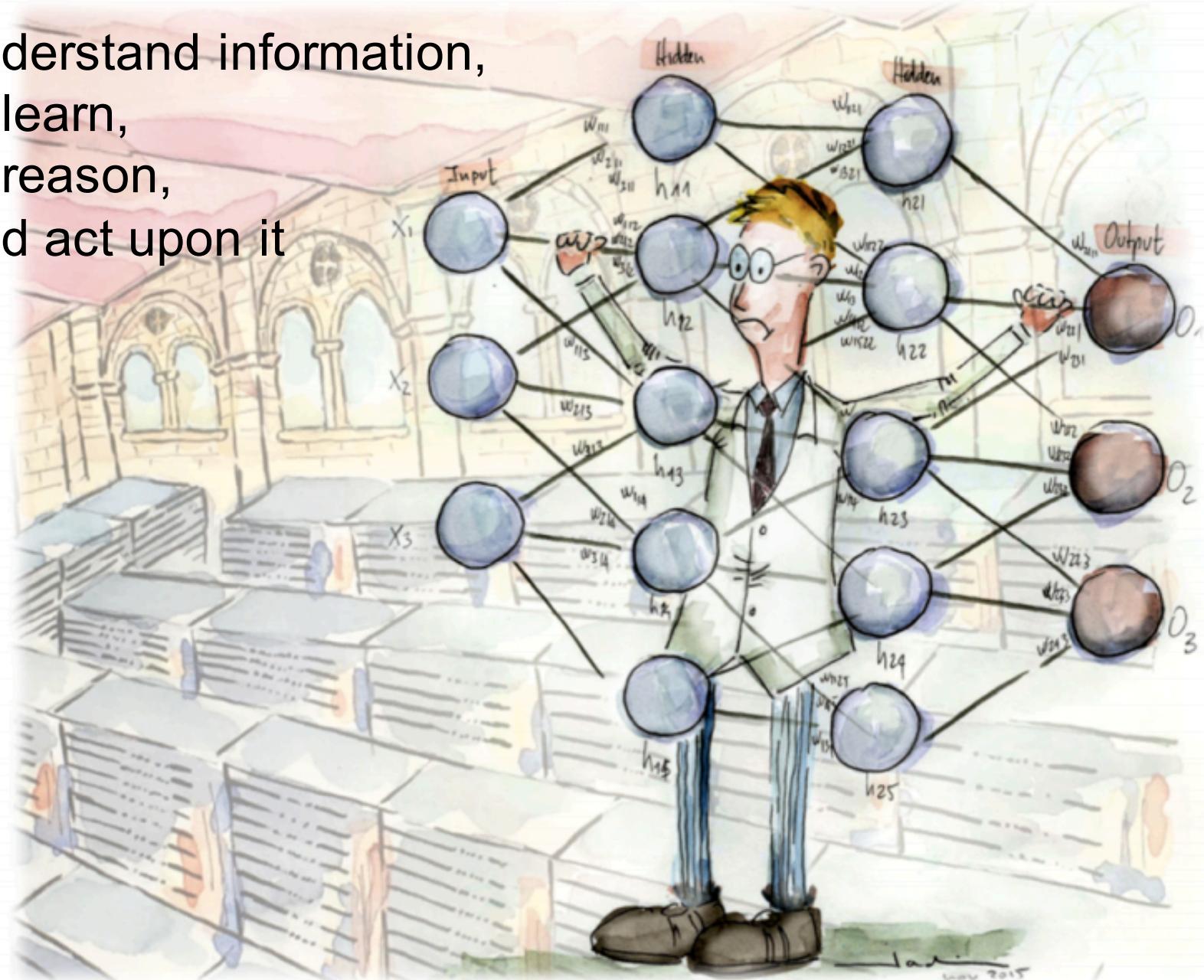


기계학습? 인공지능?



Required of New type of Computing

understand information,
to learn,
to reason,
and act upon it





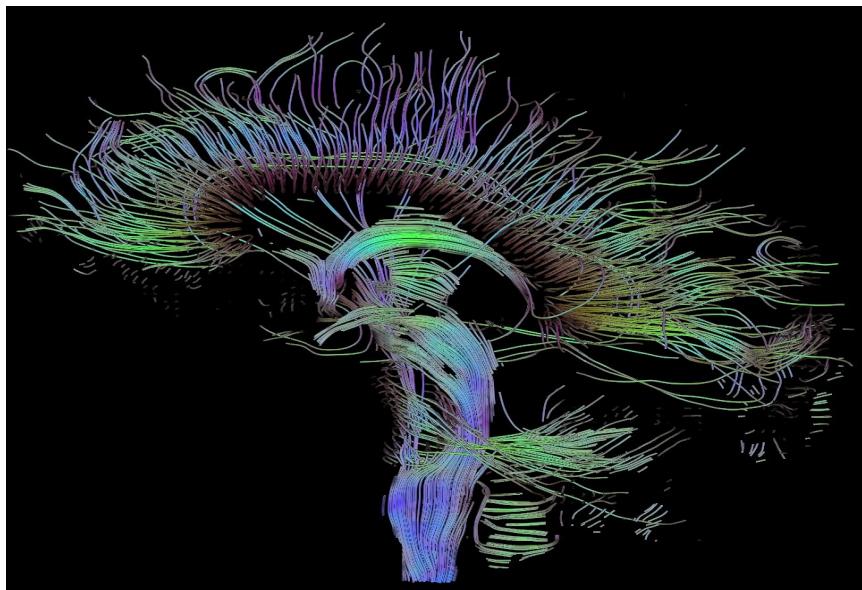
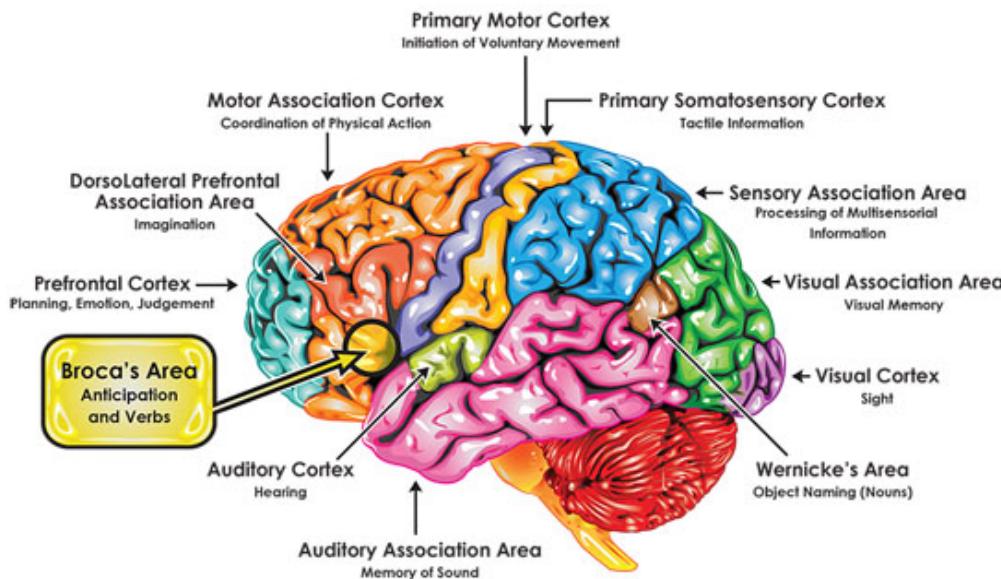
넌

뭐

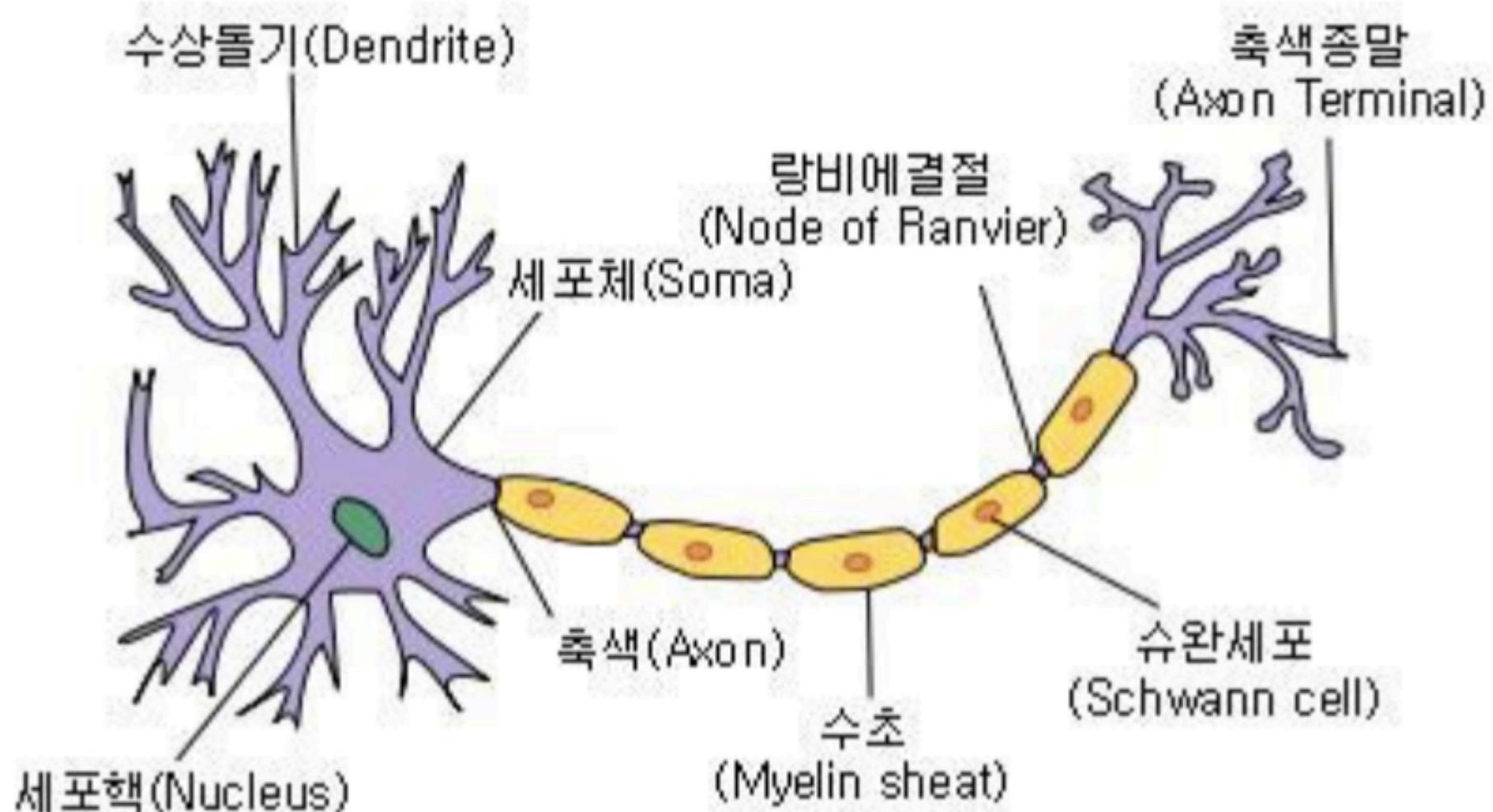
냐

?

Human Intelligence

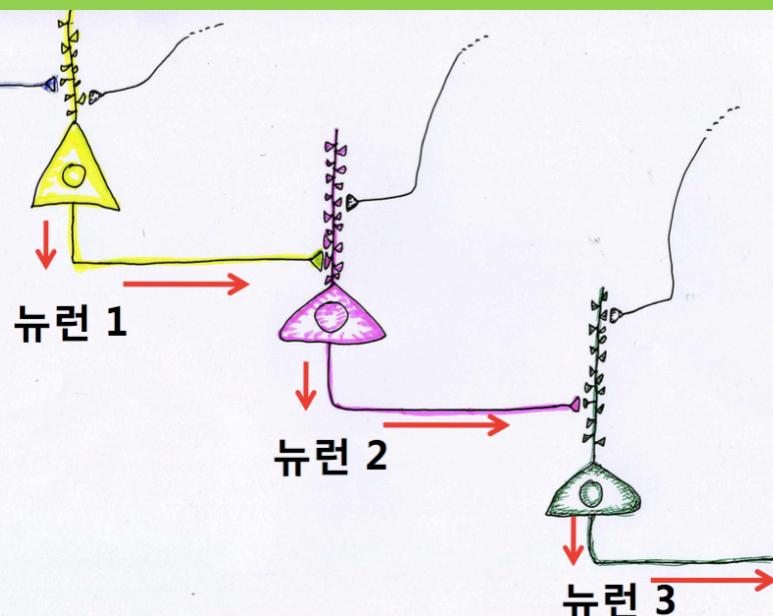
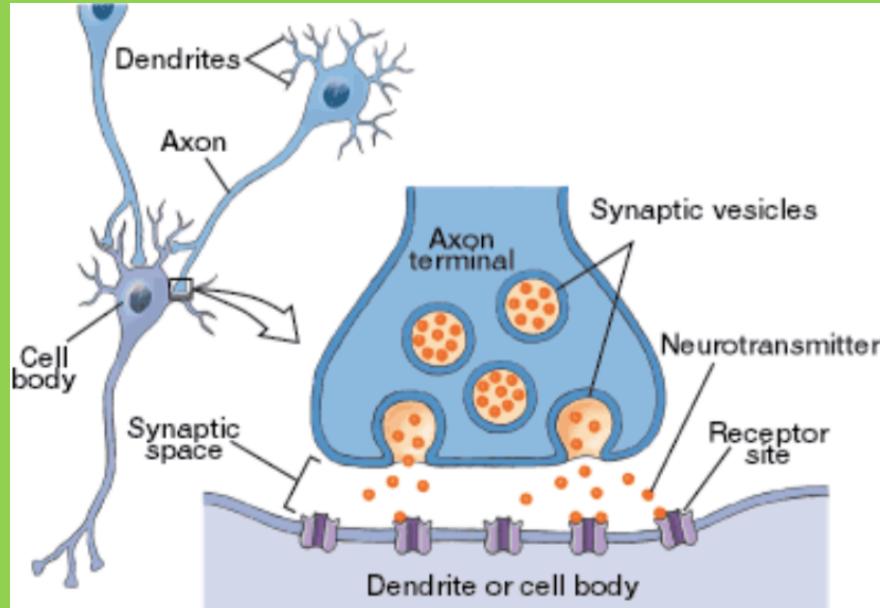


Neuron

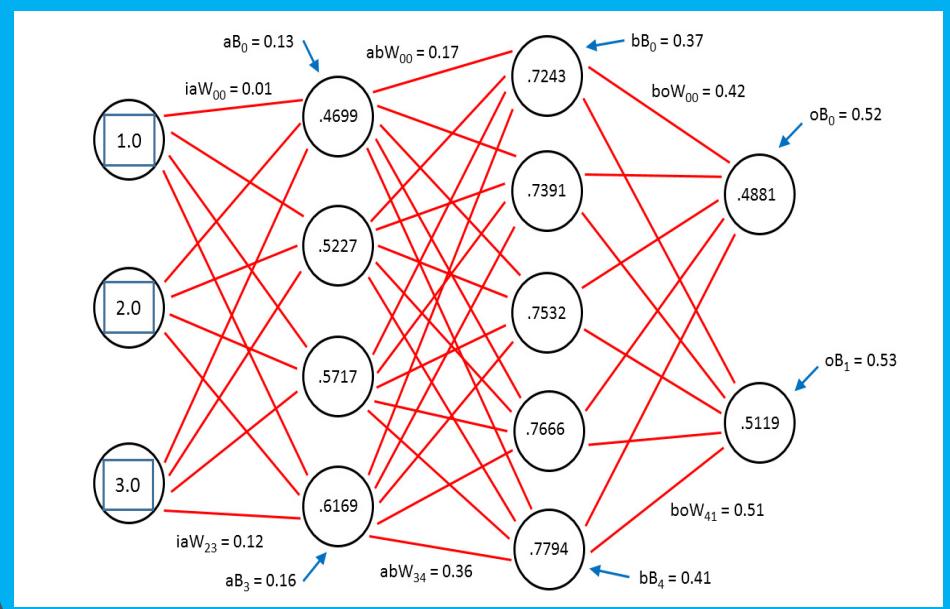
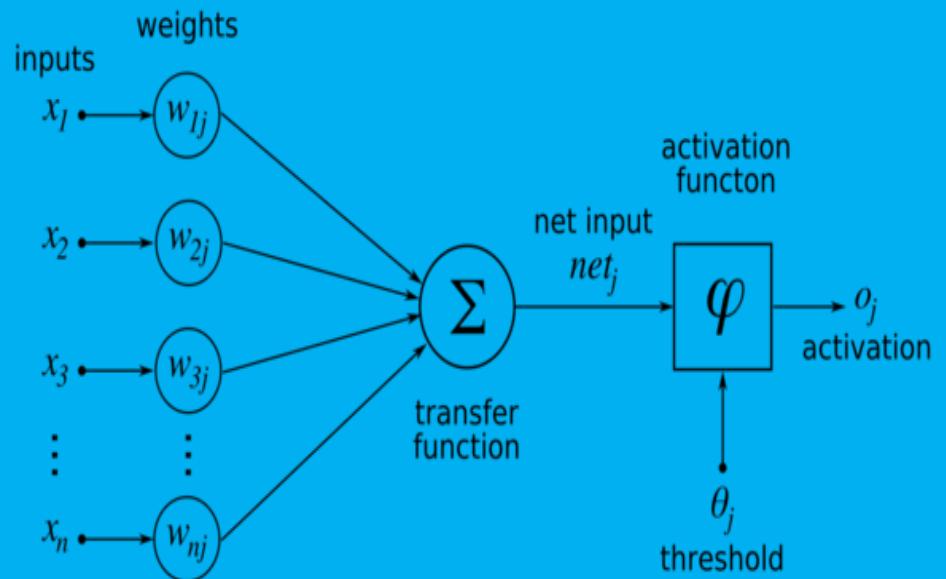


Neural network vs Learning network

Neural Network



Deep Learning Network



What is the Machine Learning ?

- Field of Computer Science that evolved from the study of **pattern recognition** and **computational learning theory** into **Artificial Intelligence**.
- Its goal is to **give computers the ability to learn** without being explicitly programmed.
- For this purpose, Machine Learning uses **mathematical / statistical techniques** to construct models from a set of observed data rather than have specific set of instructions entered by the user that define the model for that set of data.

Today's Data Analysis

Applications

Artificial neural network

Latent Dirichlet Allocation

Gaussian process regression

Support vector machines

Random Forests

Linear classifiers

k-nearest neighbor

Bayesian networks

Naive Bayes

Hidden Markov models

Unsupervised learning

Expectation-maximization alg.

K-means algorithm

DBSCAN

Deep learning

Linear regression

Logistic regression

...

Distributed Processing layer

Distributed Data Management layer

Distributed Systems

Hardware

Advanced computing provide cognitive

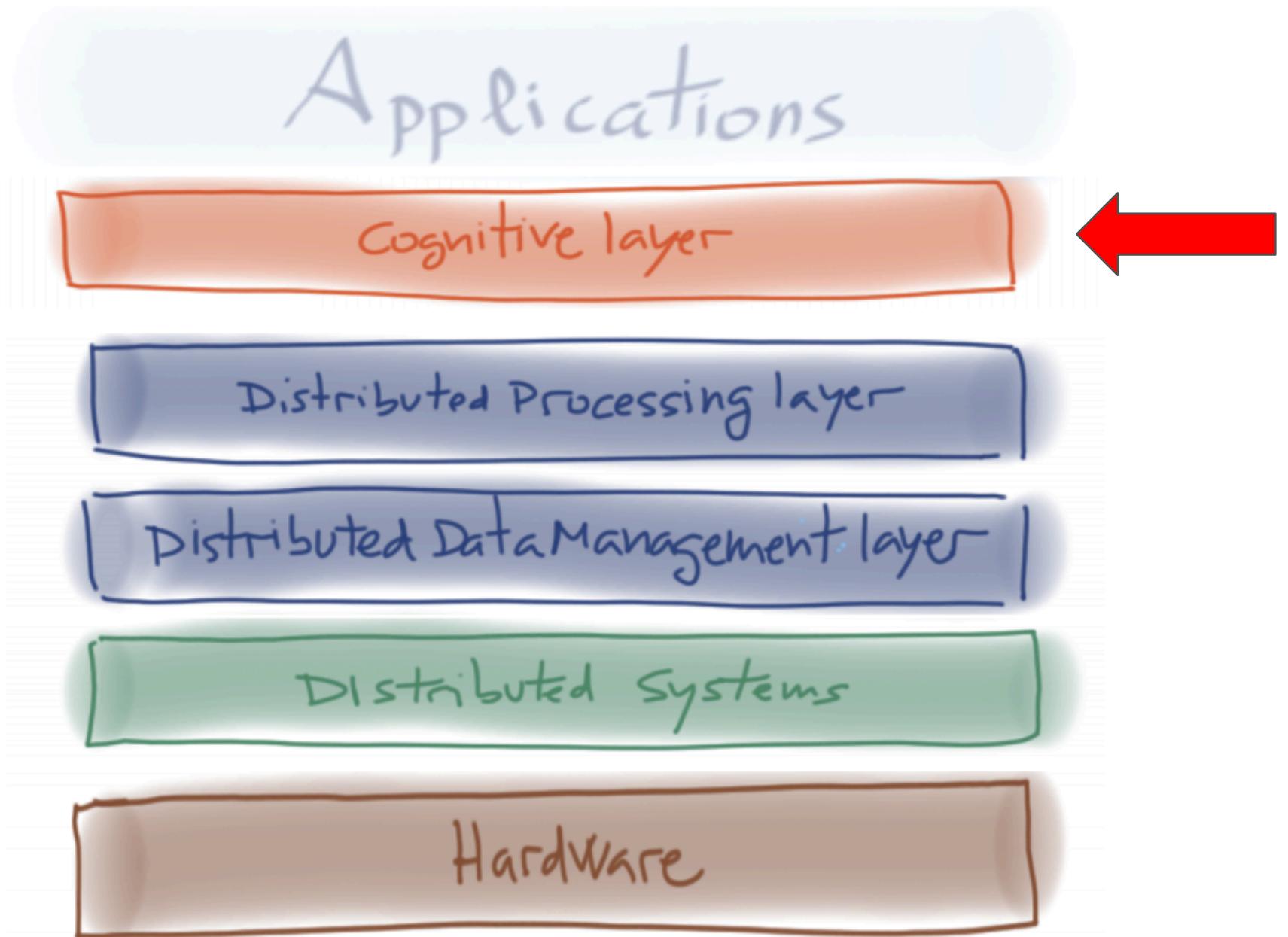


Image Recognition

stone wall [0.95, [web](#)]



dishwasher [0.91, [web](#)]



car show [0.99, [web](#)]



judo [0.96, [web](#)]



judo [0.92, [web](#)]



judo [0.91, [web](#)]



tractor [0.91, [web](#)]



tractor [0.91, [web](#)]



tractor [0.94, [web](#)]



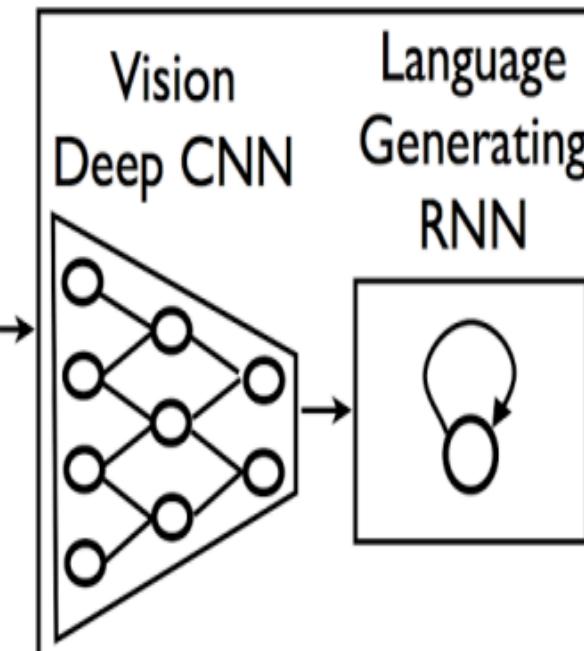
* Source: Oriol Vinyals – Research Scientist at Google Brain

Image recognition in Google Map



* Source: Oriol Vinyals – Research Scientist at Google Brain

Language Generating

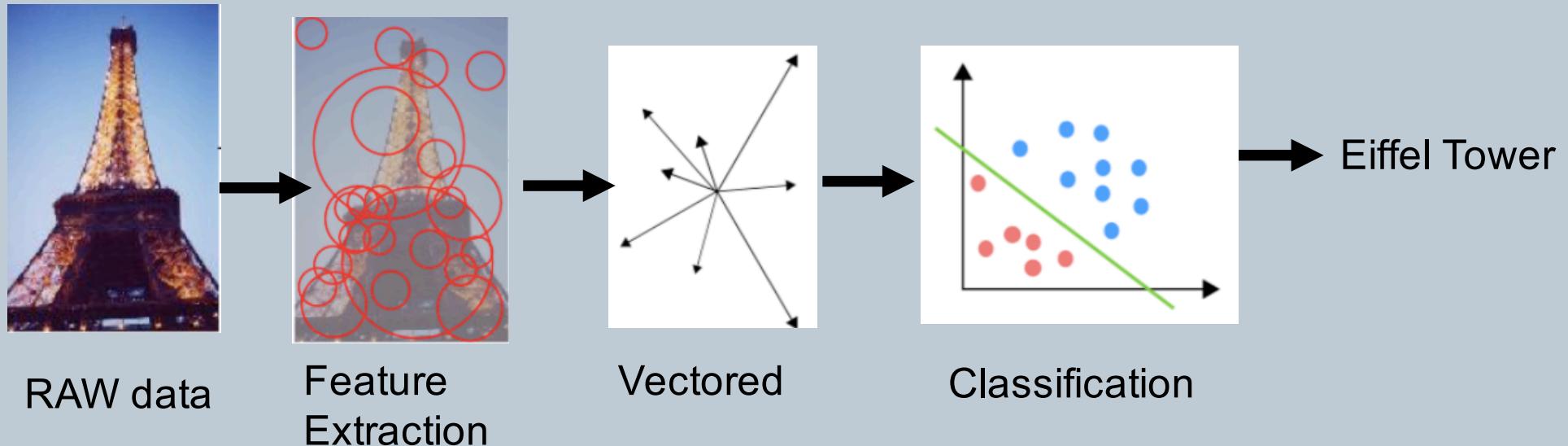


**A group of people
shopping at an
outdoor market.**

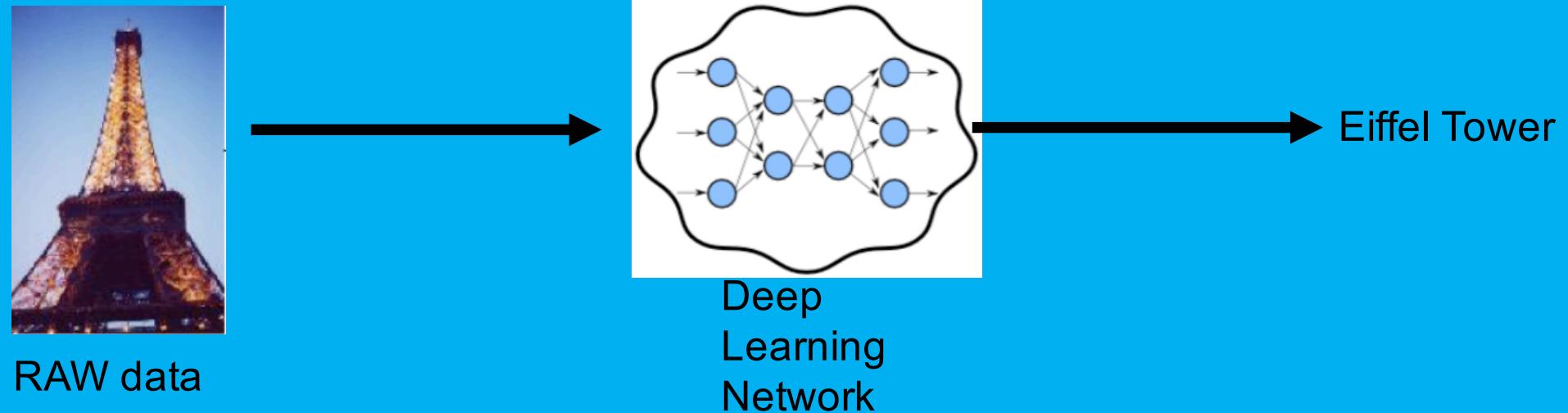
**There are many
vegetables at the
fruit stand.**

Traditional learning **vs** Deep Machine Learning

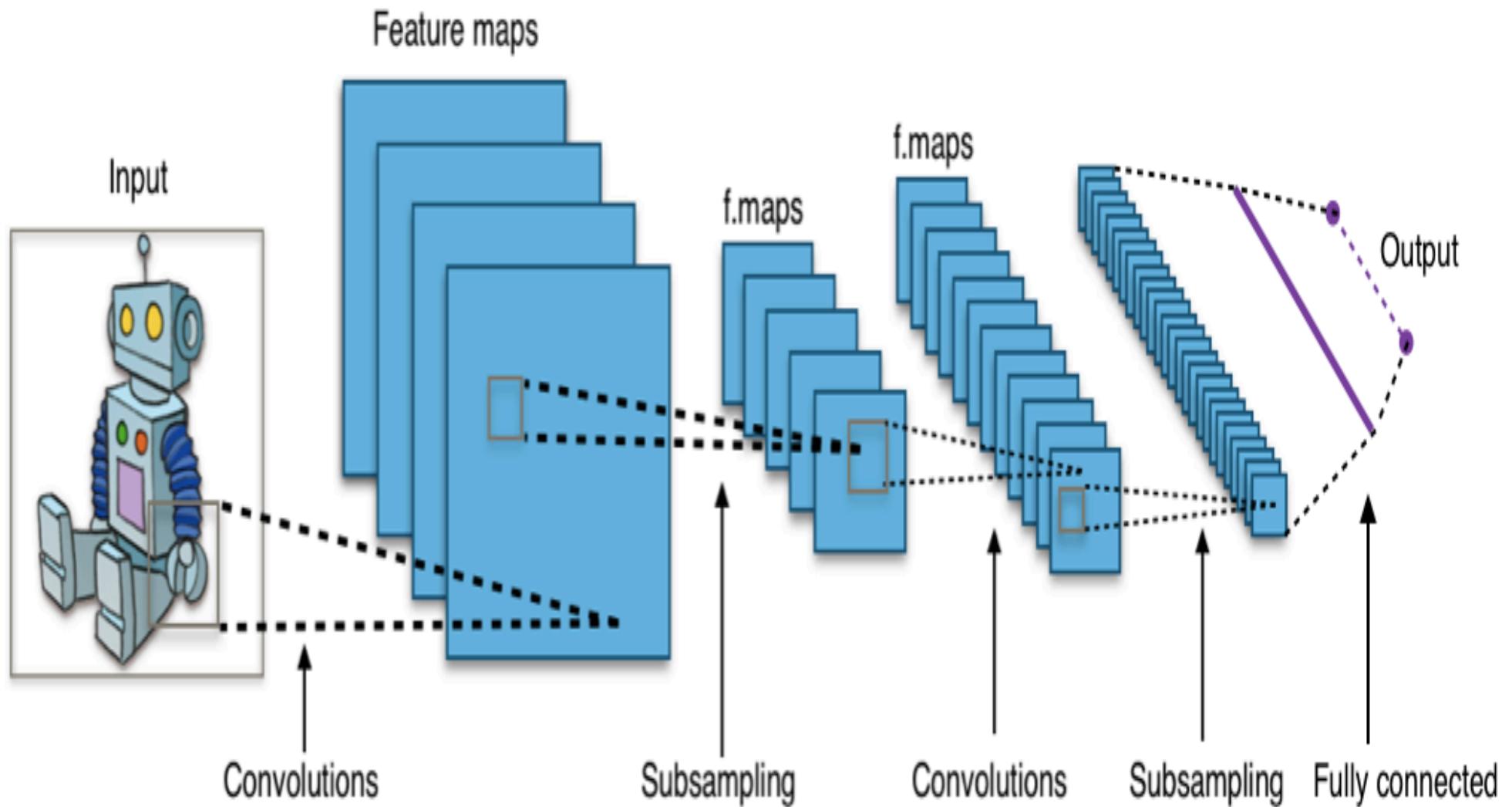
Traditional Learning



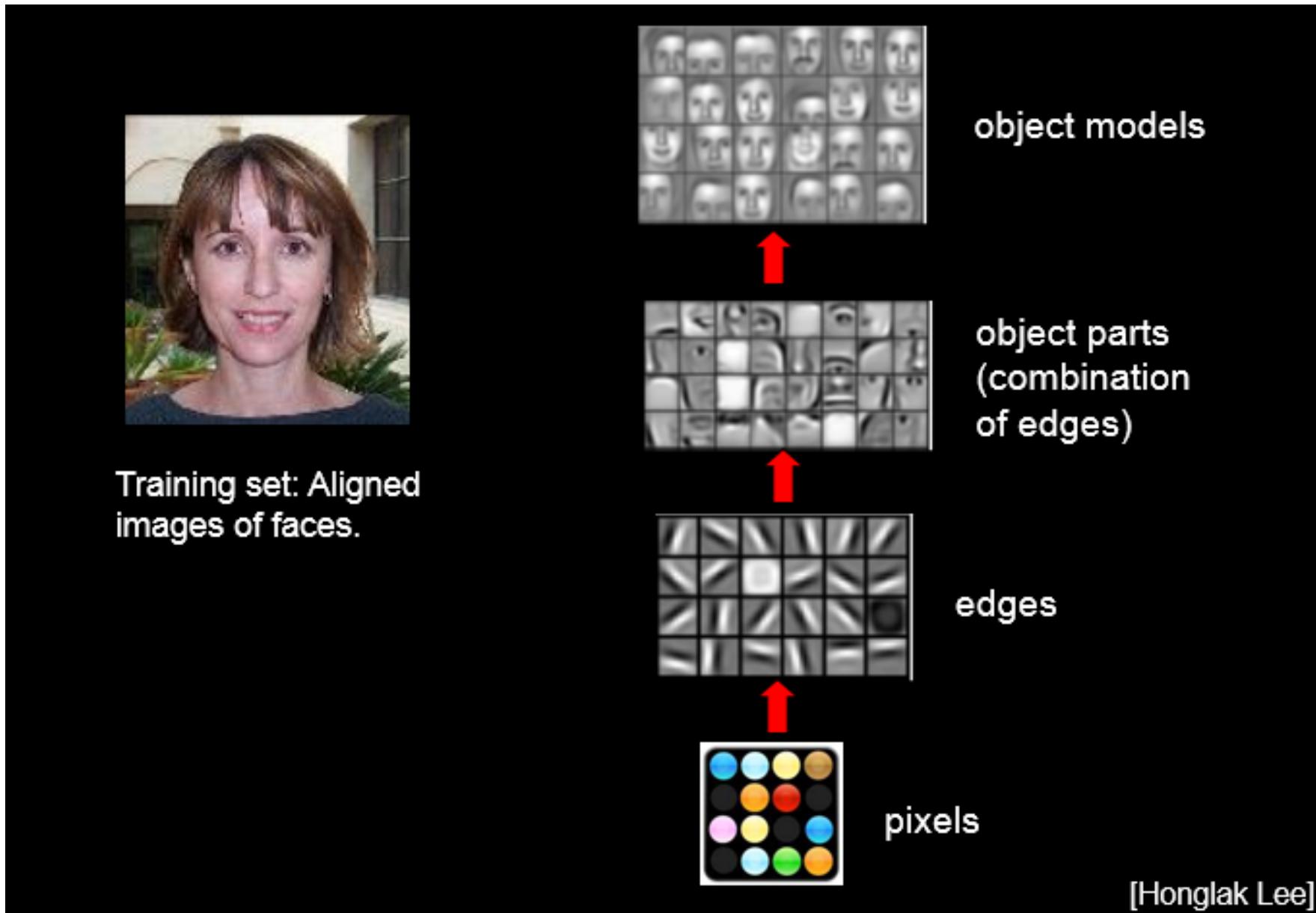
Deep Learning



Deep learning - CNN



Hierarchical Representation of Deep Learning



* Source: Honglak Lee and colleagues (2011) as published in “Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks”.

Deep Learning (object parts)

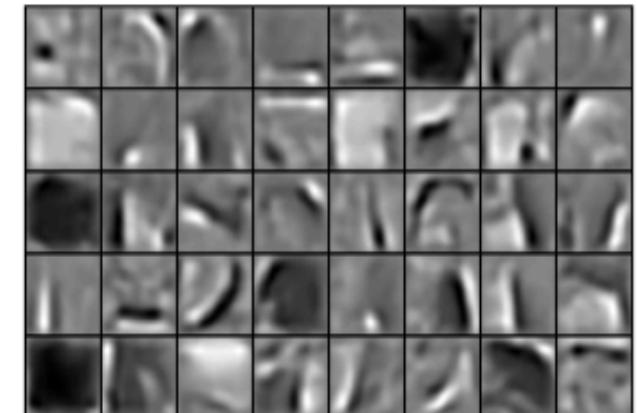
faces



cars

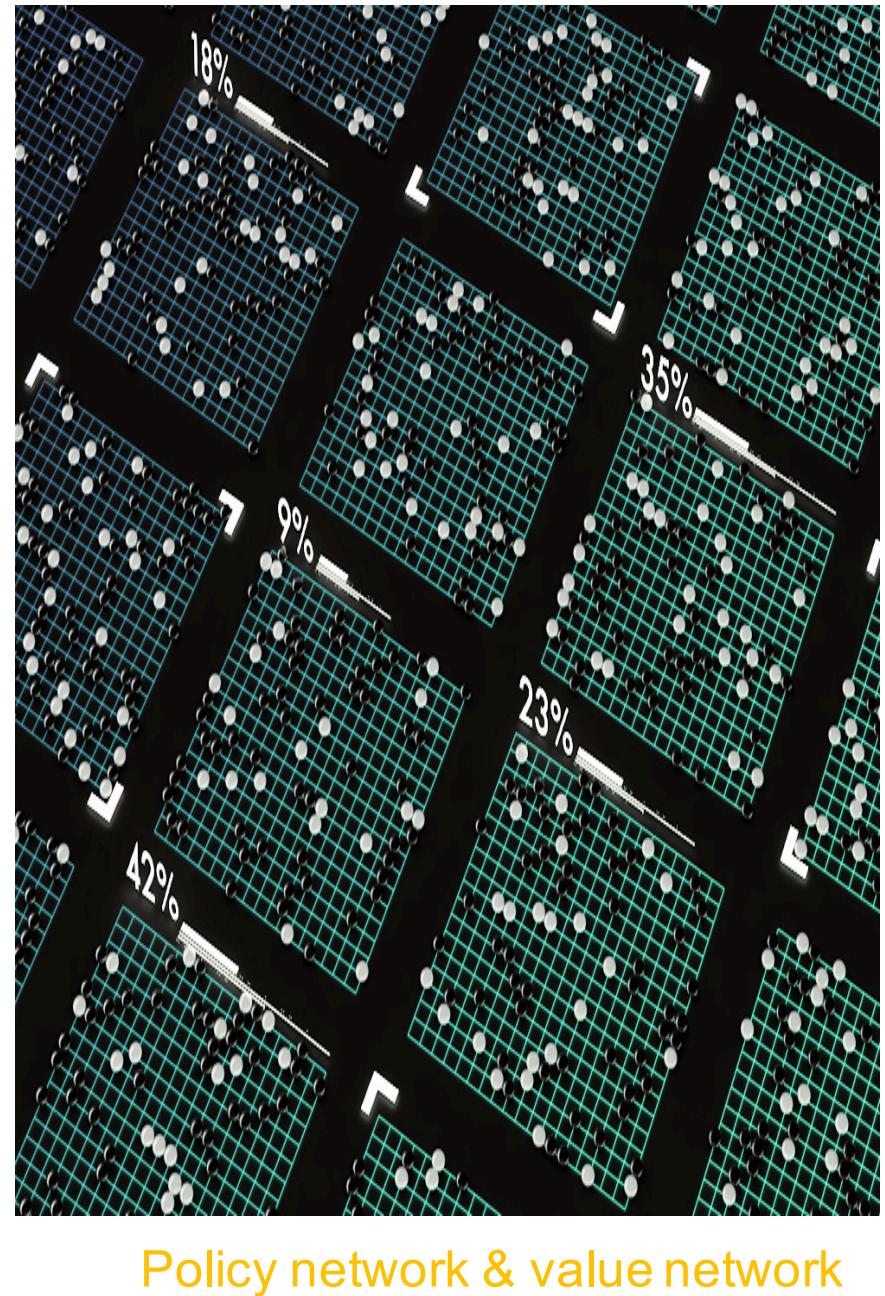
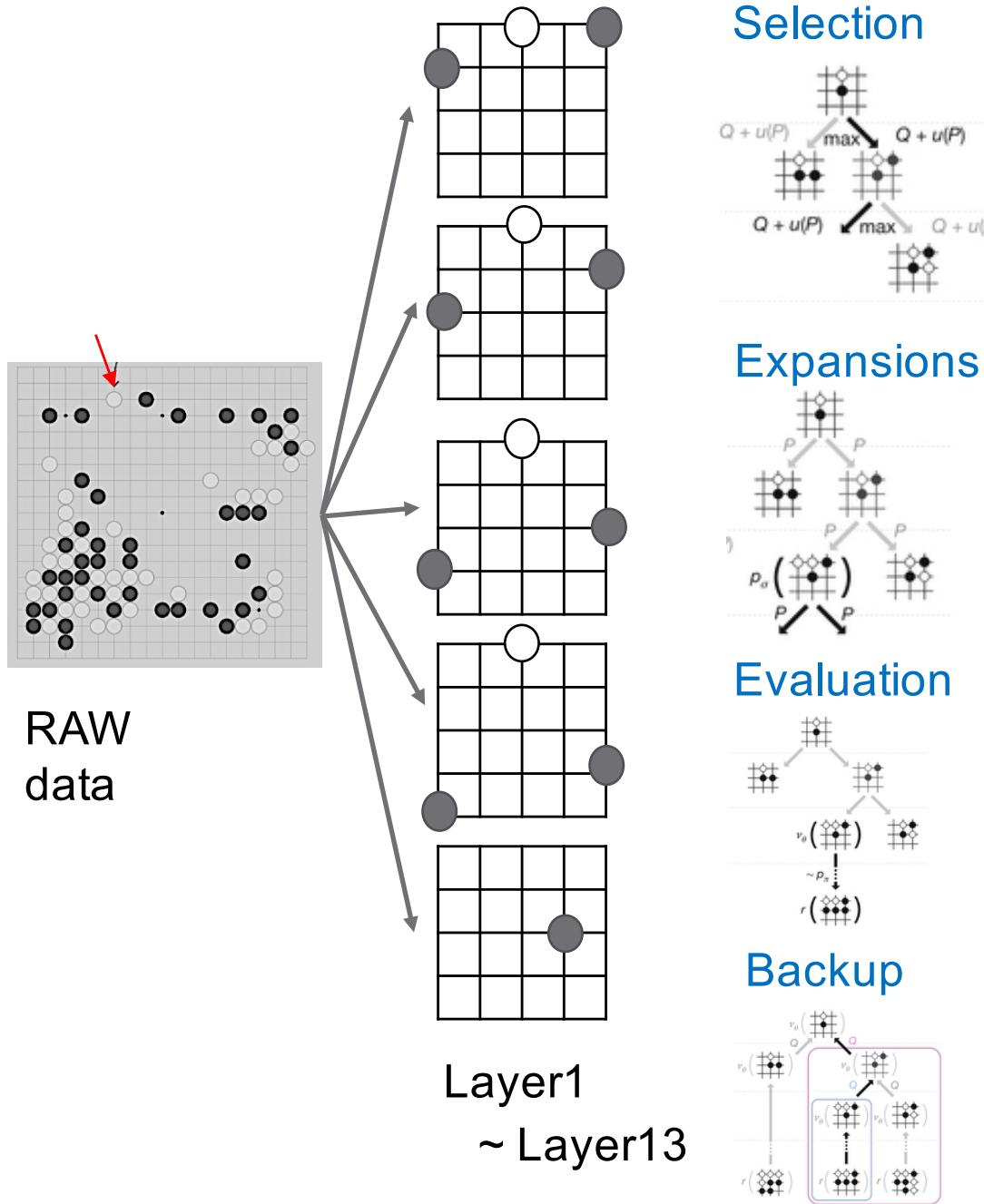


elephants



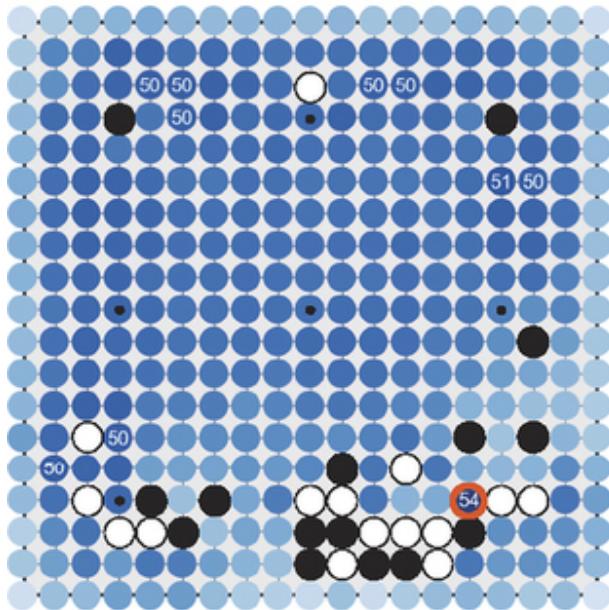
* Source: : Honglak Lee and colleagues (2011) as published in “Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks”.

AlphaGo

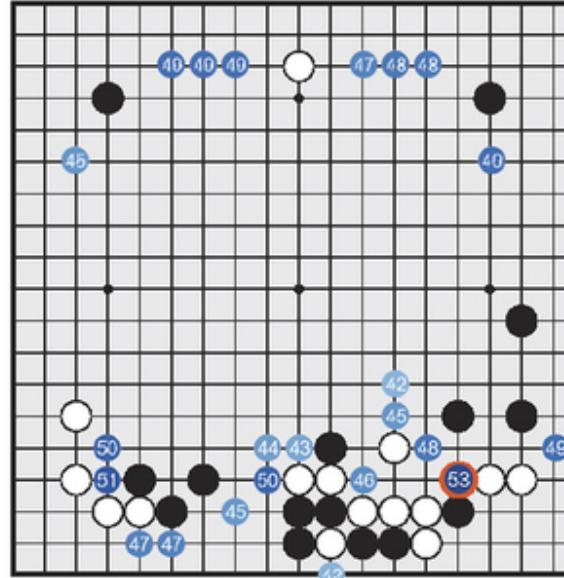


AlphaGo

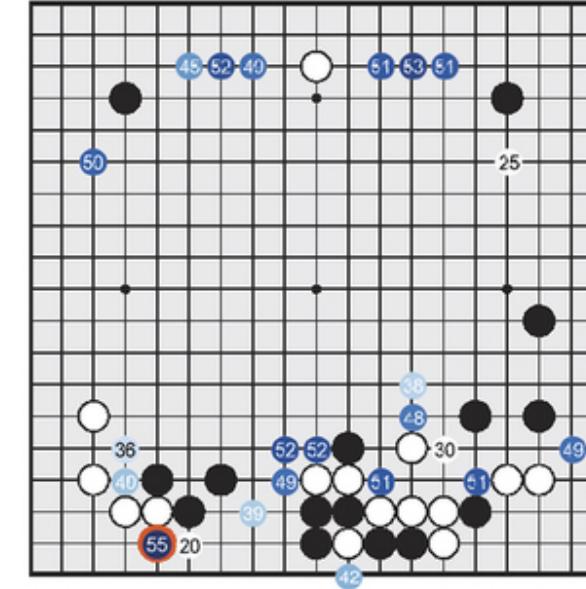
a Value network



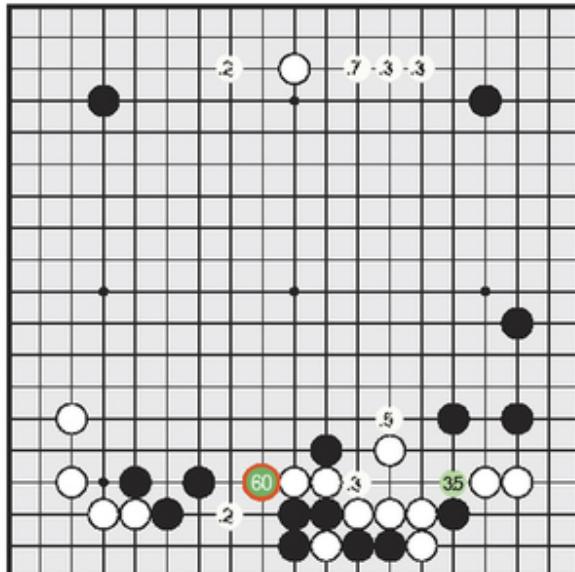
b Tree evaluation from value net



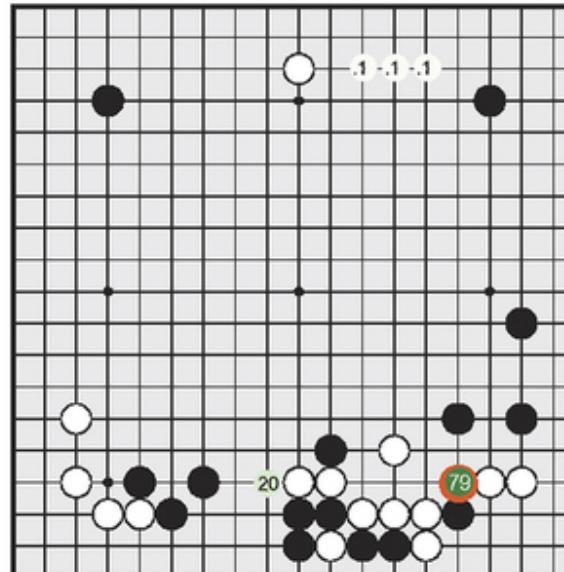
c Tree evaluation from rollouts



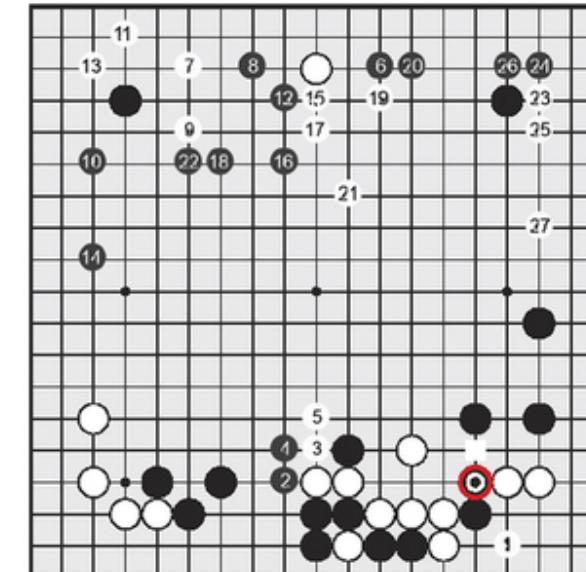
d Policy network



e Percentage of simulations



f Principal variation



Open Source Software for Machine Learning

Deep
Learning

Stream
Analytics

Big Data
Machine Learning

Data
Mining

Machine Learning
As a Service

TensorFlow

Apache SA

Spark MLlib

Knife

Amazon ML

Caffe

Summingbird

Mahout

Weka

Google
prediction API

Pylearn2

SANOA

Flink ML

RapidMiner

IBM Watson

Chainer

Scikit-learn

BigML

Torch7

DataRobot

Theano

HPE haven
OnDemand

Neon

PurePredictive

Convnet.js

Yottamine

DL4J

FICO

Caffe

Caffe

Deep learning framework
by the [BVLC](#)

Created by
[Yangqing Jia](#)

Lead Developer
[Evan Shelhamer](#)

 [View On GitHub](#)

Caffe

Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by the Berkeley Vision and Learning Center ([BVLC](#)) and by community contributors. [Yangqing Jia](#) created the project during his PhD at UC Berkeley. Caffe is released under the [BSD 2-Clause license](#).

Check out our web image classification [demo!](#)

Why Caffe?

Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

Extensible code fosters active development. In Caffe's first year, it has been forked by over 1,000 developers and had many significant changes contributed back. Thanks to these contributors the framework tracks the state-of-the-art in both code and models.

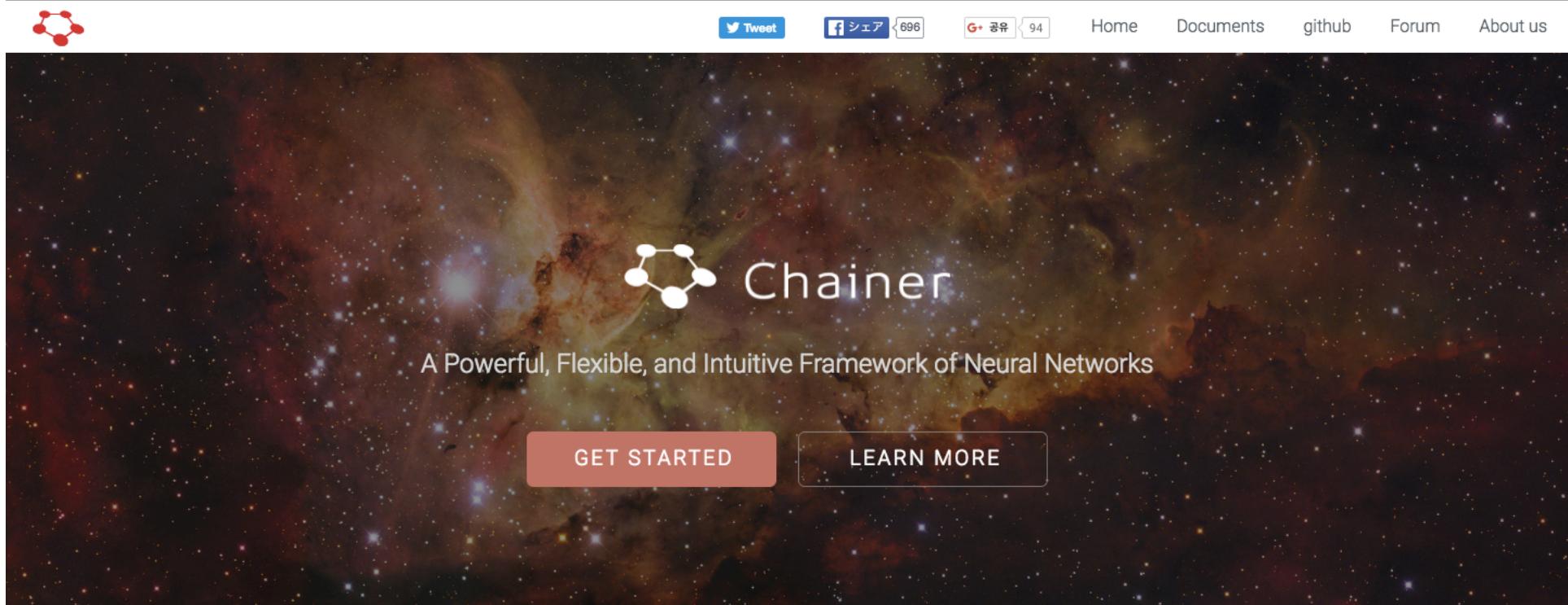
Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU*. That's 1 ms/image for inference and 4 ms/image for learning. We believe that Caffe is the fastest convnet implementation available.

Community: Caffe already powers academic research projects, startup prototypes, and even large-scale industrial applications in vision, speech, and multimedia. Join our community of brewers on the [caffe-users group](#) and [Github](#).

* With the ILSVRC2012-winning [SuperVision](#) model and caching IO. Consult performance [details](#).

U.C. Berkley 개발
C++, Python, MATLAB
대표적인 기계학습을 위한 공개 소프트웨어
* Source: <http://caffe.berkeleyvision.org>

Chainer



The banner features a dark, star-filled nebula background. In the center, the Chainer logo (a white molecular-like icon) is positioned next to the word "Chainer". Below the logo, the text "A Powerful, Flexible, and Intuitive Framework of Neural Networks" is displayed. At the bottom, there are two buttons: a red "GET STARTED" button and a white "LEARN MORE" button.

Twitter Tweet Facebook シェア 696 Google+ 공유 94

Home Documents github Forum About us

BRIDGE THE GAP BETWEEN ALGORITHMS AND IMPLEMENTATIONS
OF DEEP LEARNING

Preferred Networks (PFN)

Python

Windows에서도 동작

* Source: <http://chainer.org>

Pylearn2

Pylearn2 dev documentation »

[next](#) | [modules](#) | [index](#)

Welcome

Warning

This project does not have any current developer. We will continue to review pull requests and merge them when appropriate, but do not expect new development unless someone decides to work on it.

There are other machine learning frameworks built on top of Theano that could interest you, such as: [Blocks](#), [Keras](#) and [Lasagne](#).

Don't expect a clean road without bumps! If you find a bug please write to pylearn-dev@googlegroups.com. If you're a Pylearn2 developer and you find a bug, please write a unit test for it so the bug doesn't come back!

Pylearn2 is a machine learning library. Most of its functionality is built on top of [Theano](#). This means you can write Pylearn2 plugins (new models, algorithms, etc) using mathematical expressions, and Theano will optimize and stabilize those expressions for you, and compile them to a backend of your choice (CPU or GPU).

Pylearn2 Vision

- Researchers add features as they need them. We avoid getting bogged down by too much top-down planning in advance.
- A machine learning toolbox for easy scientific experimentation.
- All models/algorithms published by the LISA lab should have reference implementations in Pylearn2.
- Pylearn2 may wrap other libraries such as scikit-learn when this is practical
- Pylearn2 differs from scikit-learn in that Pylearn2 aims to provide great flexibility and make it possible for a researcher to do almost anything, while scikit-learn aims to work as a "black box" that can produce good results even if the user does not understand the implementation
- Dataset interface for vector, images, video, ...
- Small framework for all what is needed for one normal MLP/RBM/SDA/Convolution experiments.
Easy reuse of sub-component of Pylearn2.
- Using one sub-component of the library does not force you to use / learn to use all of the other sub-components if you choose not to.

Python 기반
GPU support

풍부한 종류의 영상 처리 라이브러리,

* Source: <http://deeplearning.net/software/pylearn2/>

Table Of Contents

[Welcome](#)
[Pylearn2 Vision](#)
[Download and installation](#)
▪ [Data path](#)
▪ [Other methods](#)
 ▪ [Vagrant \(any OS\)](#)
[Dependencies](#)
[License and Citations](#)
[Documentation](#)
[Community](#)
[Developer](#)

[Next topic](#)

[Library Documentation](#)

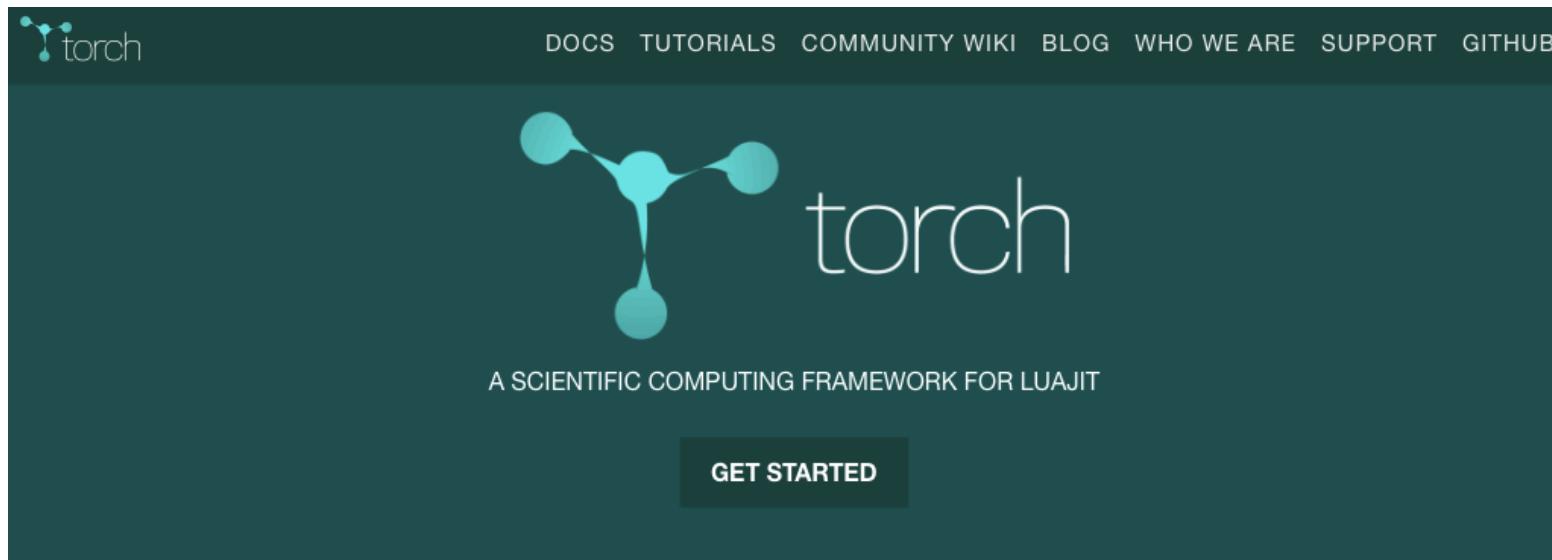
[This Page](#)

[Show Source](#)

[Quick search](#)

Enter search terms or a module, class or function name.

torch



What is Torch?

Torch is a scientific computing framework with wide support for machine learning algorithms that puts GPUs first. It is easy to use and efficient, thanks to an easy and fast scripting language, LuaJIT, and an underlying C/CUDA implementation.

A summary of core features:

- a powerful N-dimensional array
- lots of routines for indexing, slicing, transposing, ...
- amazing interface to C, via LuaJIT
- linear algebra routines
- neural network, and energy-based models
- numeric optimization routines
- Fast and efficient GPU support

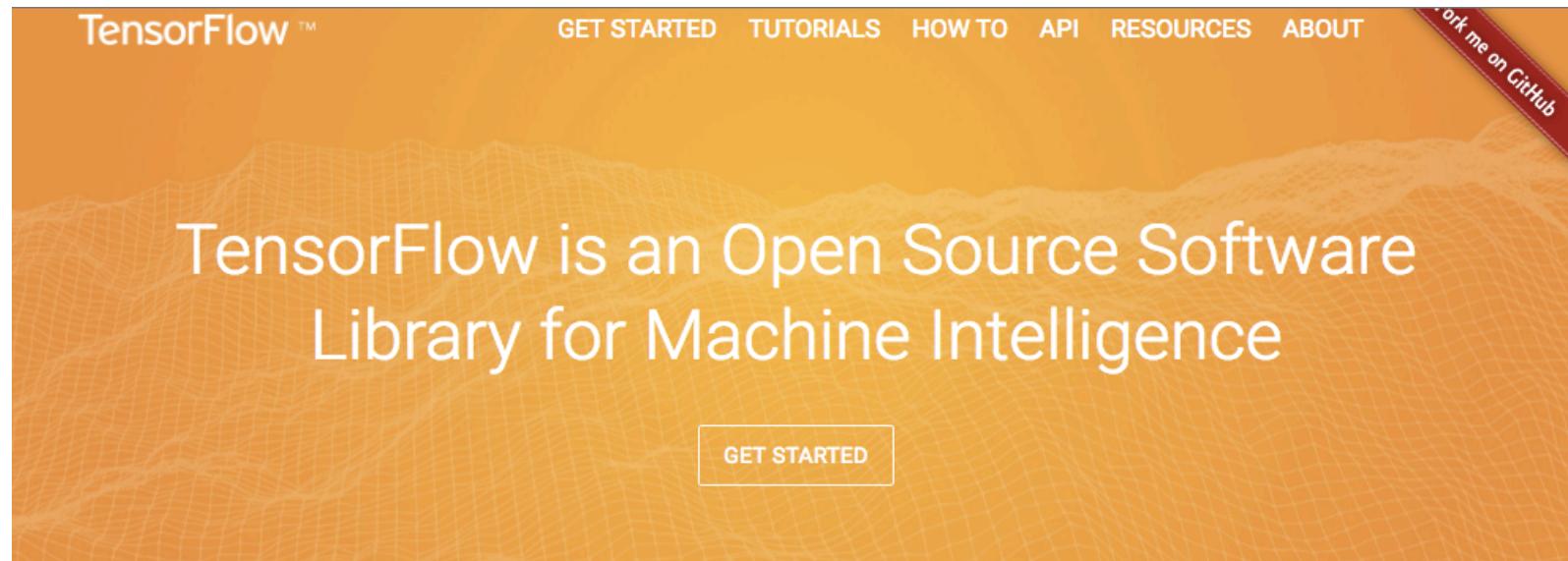
Lua 기반

GPU support

Matlab과 유사, JIT로 빠른 실행,

* Source: <http://torch.ch>

TensorFlow

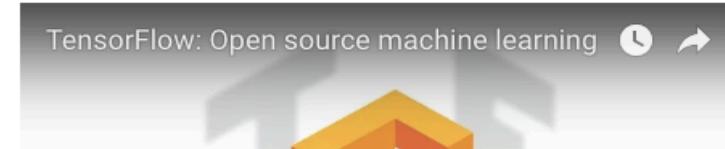


About TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device

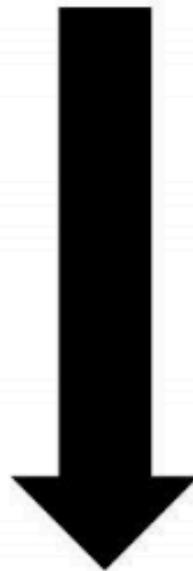
with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain
Google 개발
C++, Python
GPU support, 개발자 참여 활발

* Source: <https://www.tensorflow.org/>



TensorFlow

1st Generation : *DistBelief*



- *Dean et al. 2011*
- *Major Output Products*
 - *Inception (Image Categorization)*
 - *Google Search*
 - *Google Translate*
 - *Google Photos*

2nd Generation : *TensorFlow*

- *Dean et al. 2015 (November, 1st)*
- *Most of DistBelief users at Google have already switched to TensorFlow*

* Source: Oriol Vinyals – Research Scientist at Google Brain

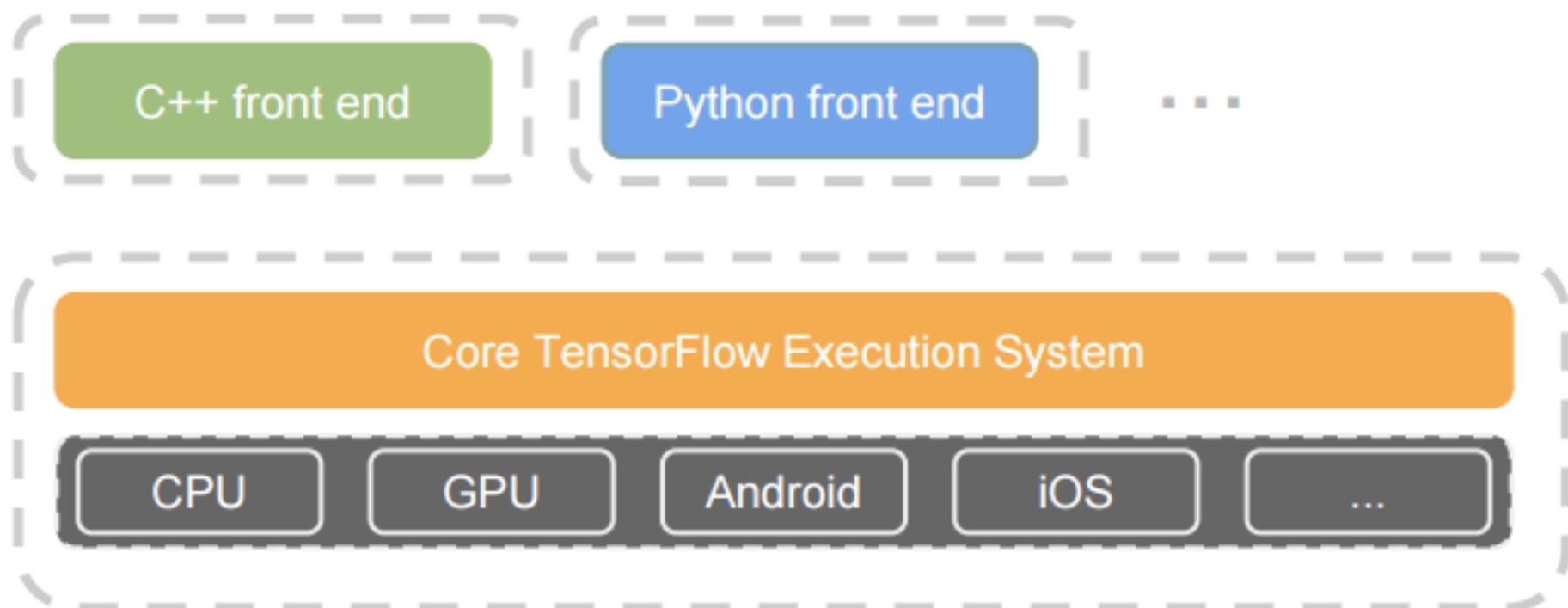
About TensorFlow

- TensorFlow was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well. •
- Open source software library for numerical computation using data flow graphs.
- The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

* Source: Jeff Dean– Research Scientist at Google Brain

Expressing High-Level ML Computations

- Core in C++
- Different front ends for specifying/driving the computation
 - Python and C++ today, easy to add more

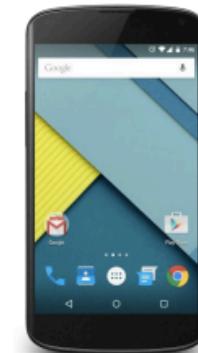


* Source: Jeff Dean – Research Scientist at Google Brain

TensorFlow Platforms

Automatically runs models on range of platforms:

from phones ...



to single machines (CPU and/or GPUs) ...



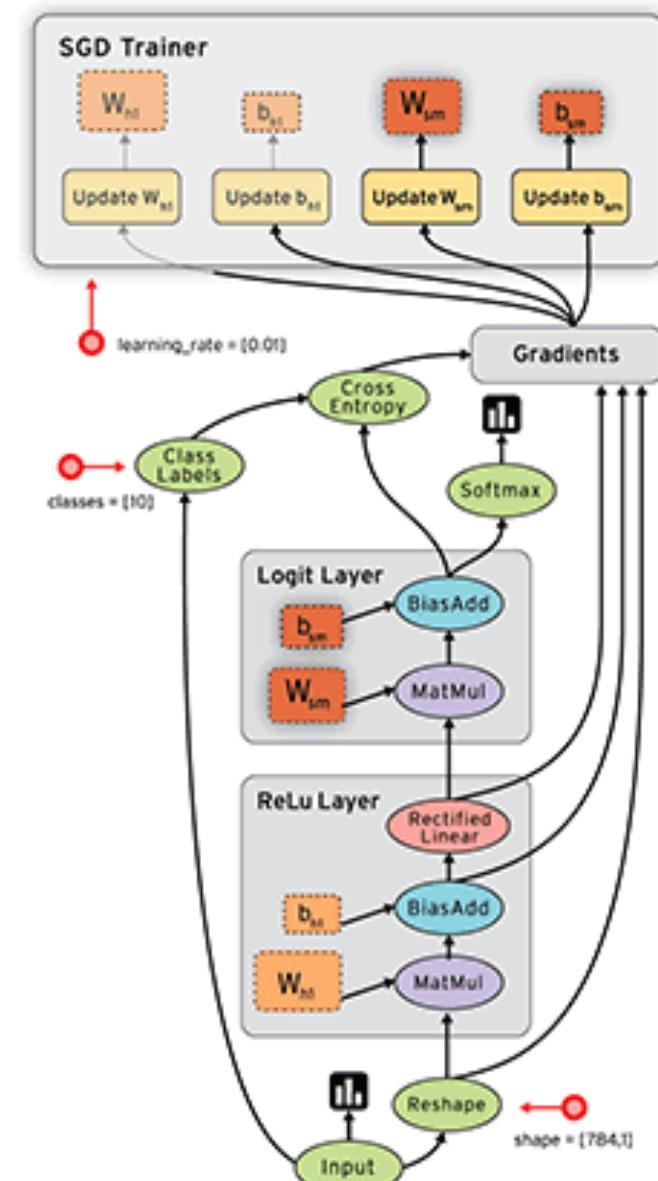
to distributed systems of many 100s of GPU cards



* Source: TensorFlow: Jeff-Oriol-NIPS-Tutorial-2015.pdf [Accessed: 20/12/2015].

TensorFlow == Data Flow

- Describe mathematical computation with a directed graph of nodes & edges.
 - Nodes in the graph represent mathematical operations,
 - Edges describe the i/o relationships between nodes.
 - Data edges carry dynamically-sized multidimensional data arrays, or tensors.
- The flow of tensors through the graph is where TensorFlow gets its name. Nodes are assigned to computational devices and execute asynchronously and in parallel once all the tensors on their incoming edges becomes available.



* Source: Jeff Dean – Research Scientist at Google Brain

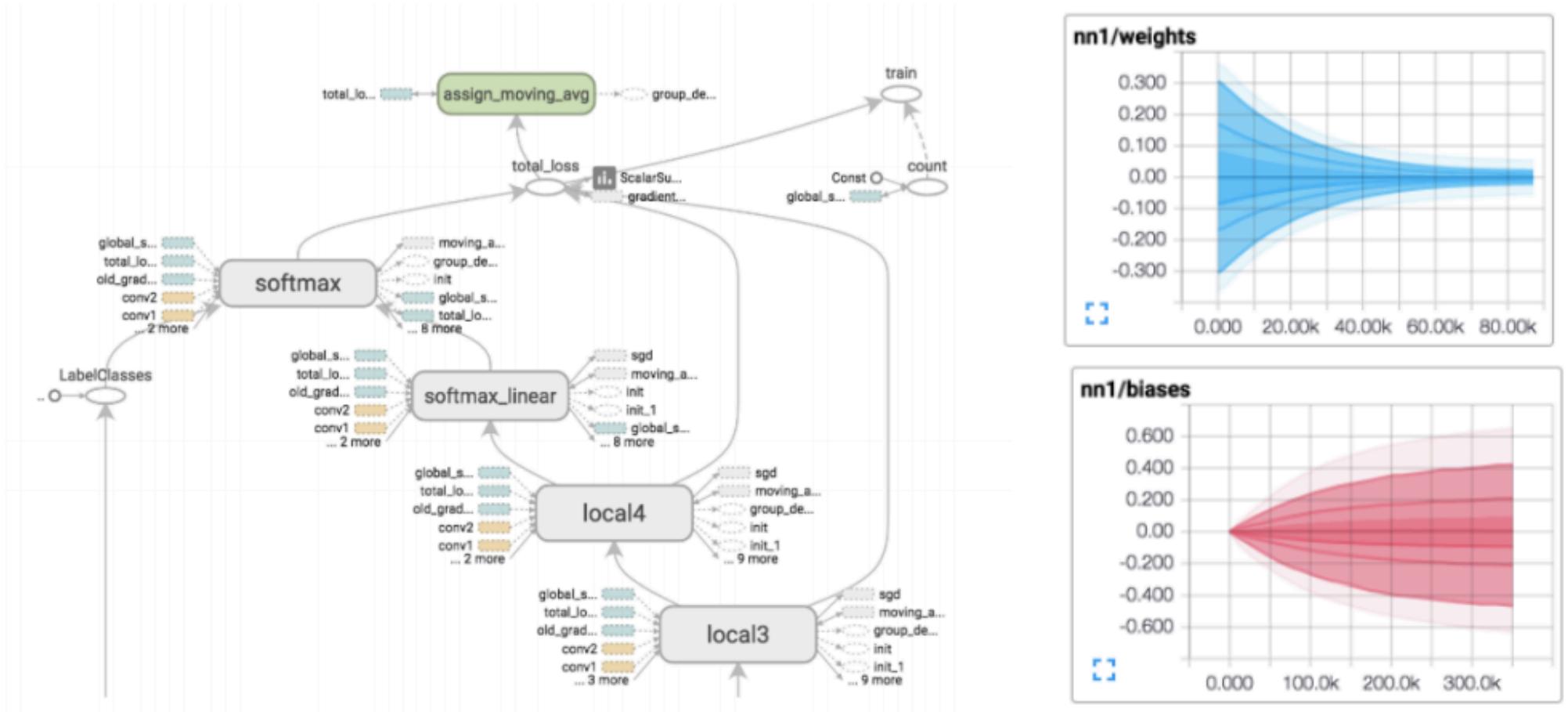
TensorFlow Kernel

| Operations groups | Operations |
|----------------------------|--|
| Maths | Add, Sub, Mul, Div, Exp, Log, Greater, Less, Equal |
| Array | Concat, Slice, Split, Constant, Rank, Shape, Shuffle |
| Matrix | MatMul, MatrixInverse, MatrixDeterminant |
| Neuronal Network | SoftMax, Sigmoid, ReLU, Convolution2D, MaxPool |
| Checkpointing | Save, Restore |
| Queues and sincronizations | Enqueue, Dequeue, MutexAcquire, MutexRelease |
| Flow control | Merge, Switch, Enter, Leave, NextIteration |

- Source: TensorFlow: Large-scale machine learning on heterogeneous systems, (2015)

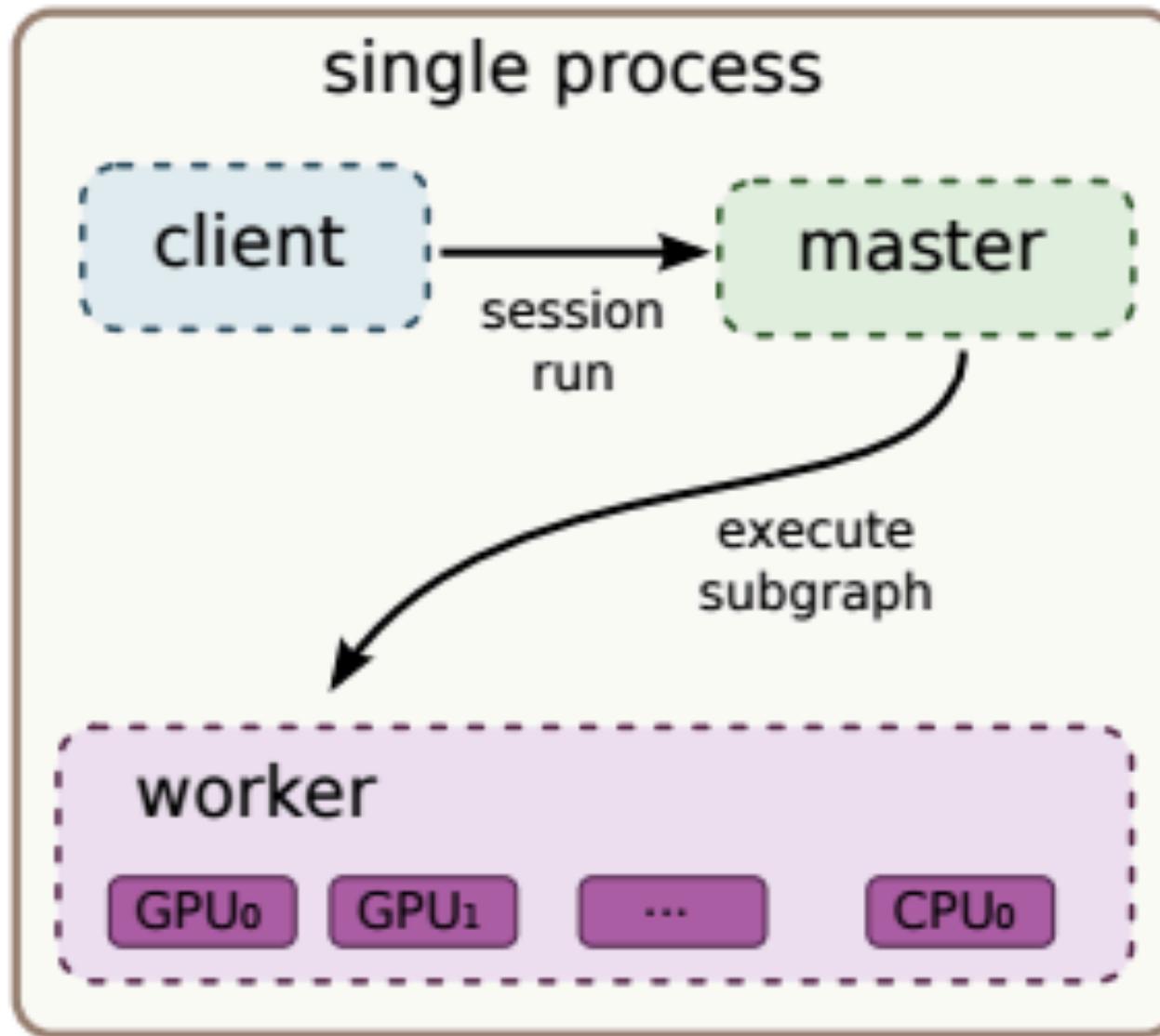
TensorBoard

- Visualization of graph structures and summary statistics
- In order to help users understand the structure of their computation graphs and also to understand the overall behavior of machine learning models.



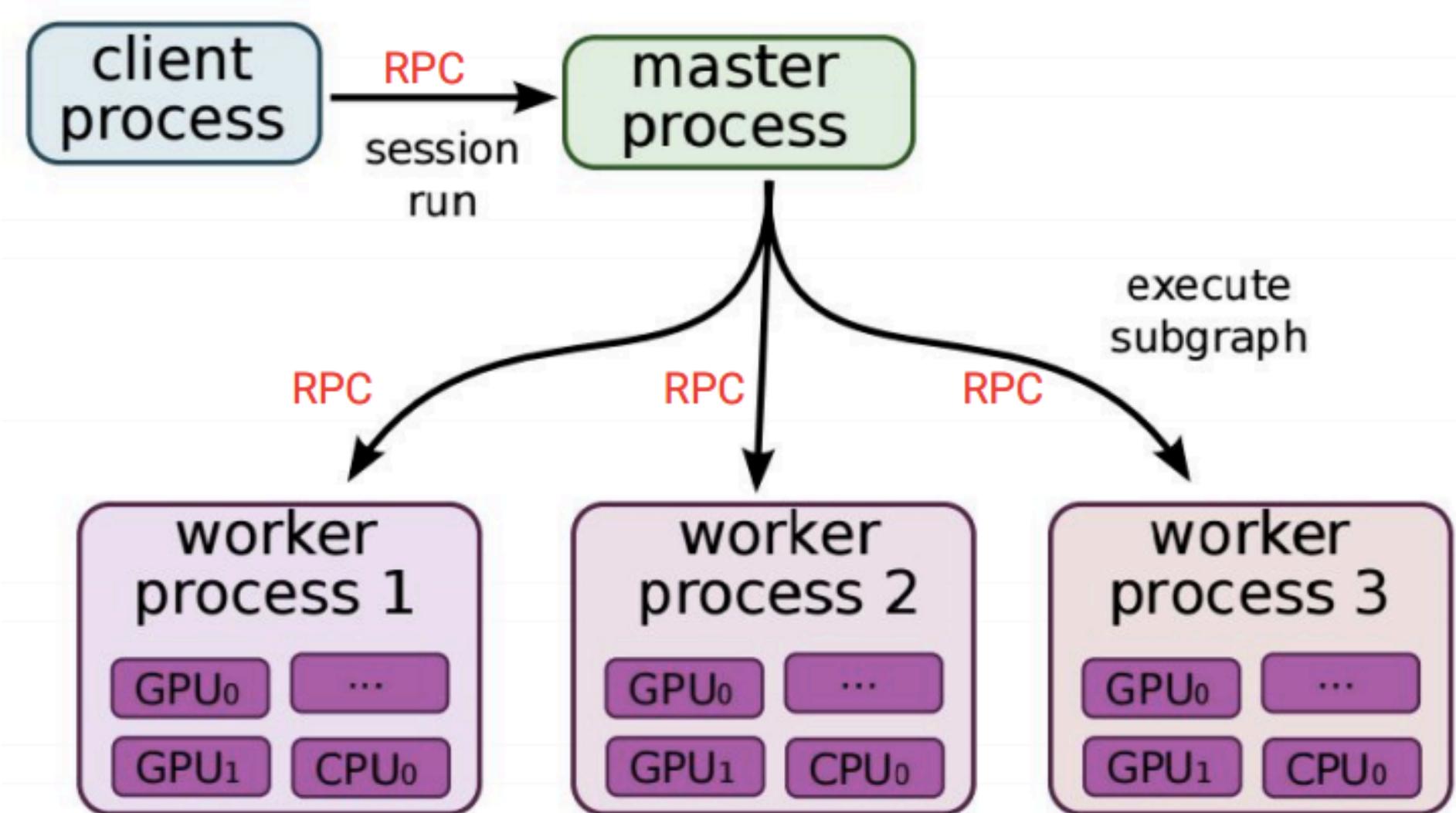
- Source: TensorFlow: Large-scale machine learning on heterogeneous systems, (2015)

TensorFlow on Single Processor



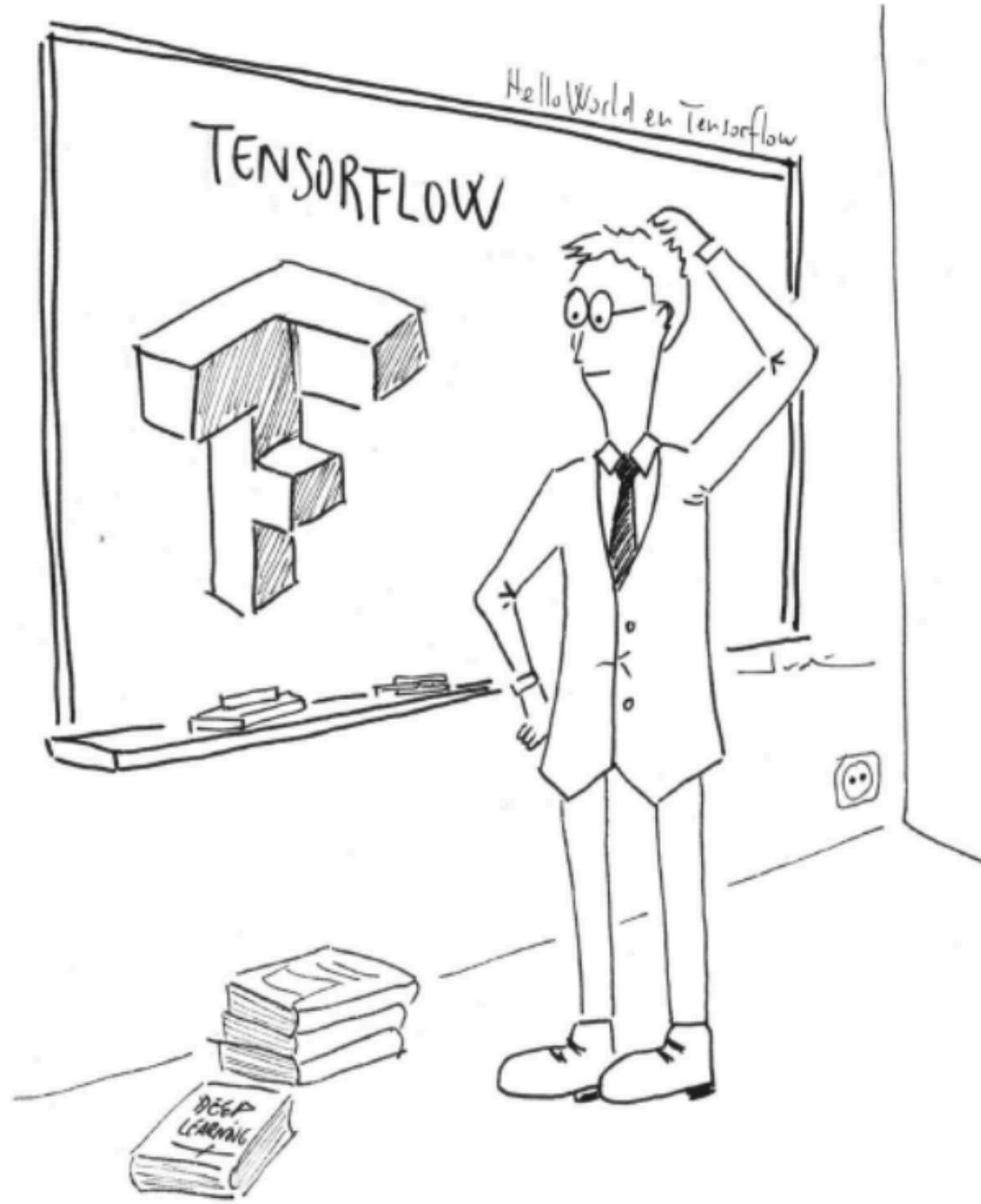
- Source: TensorFlow: Large-scale machine learning on heterogeneous systems, (2015)

TensorFlow on Distribute Processor



- Source: TensorFlow: Large-scale machine learning on heterogeneous systems, (2015)

Hello World on TensorFlow



Hello World == MNIST

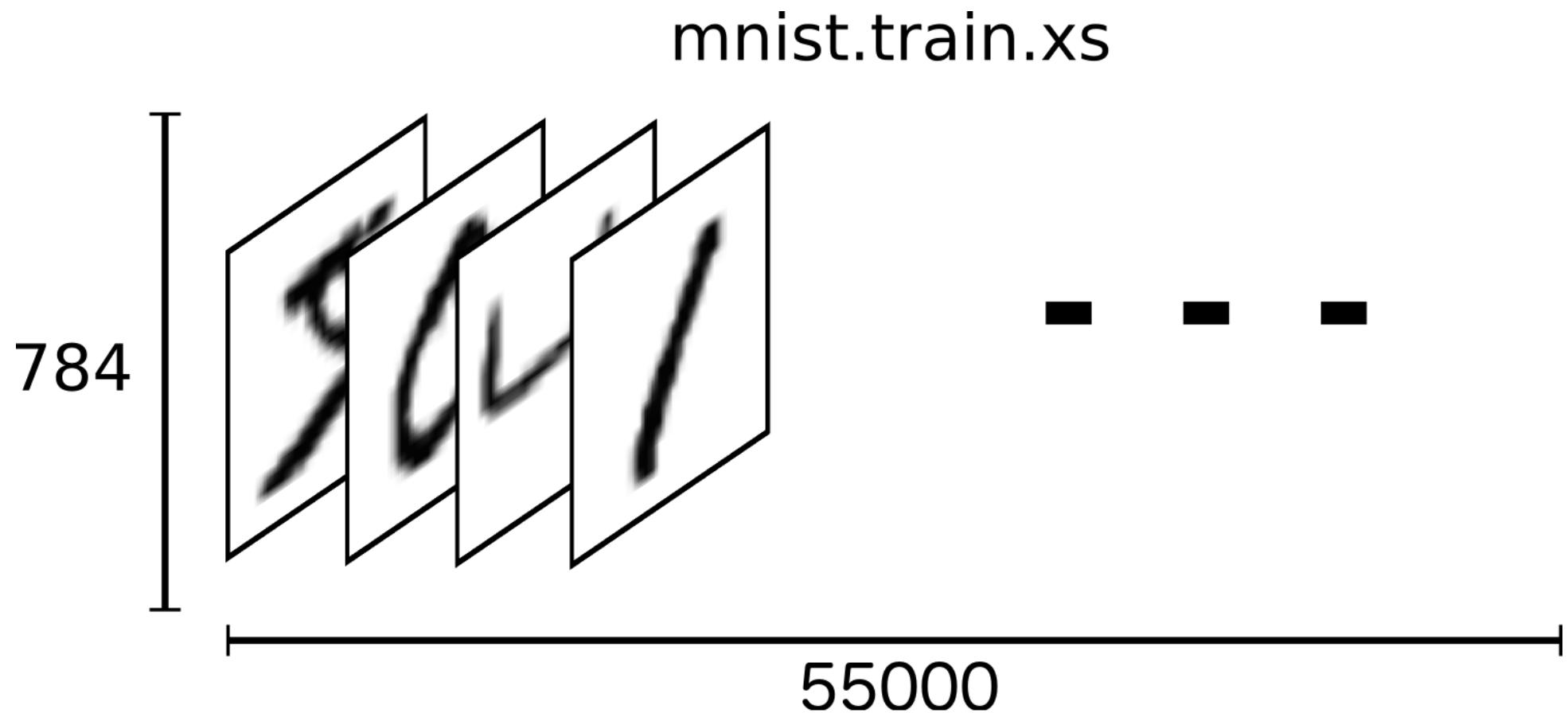


MNIST (predict number of image)

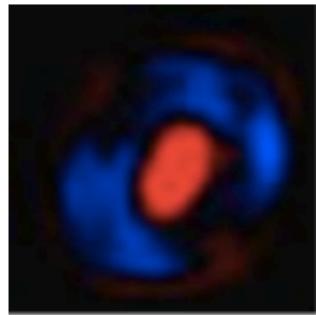


2

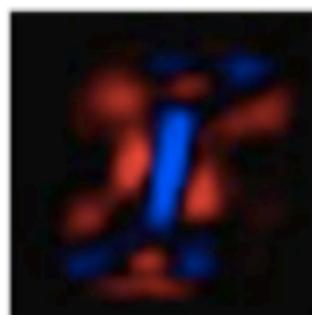
CNN (convolution neural network) training



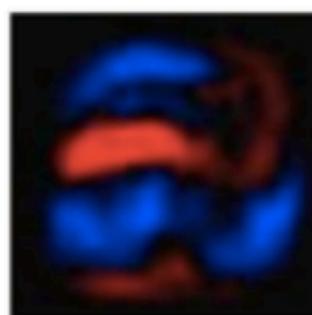
MNIST



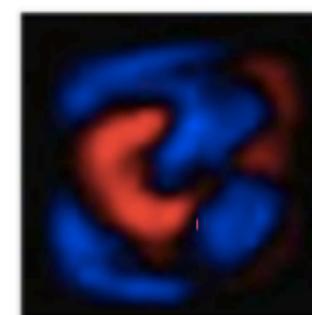
0



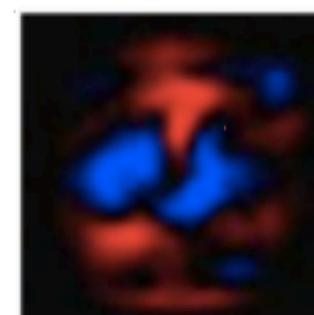
1



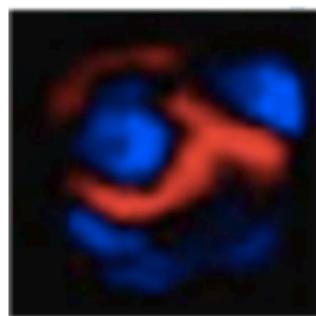
2



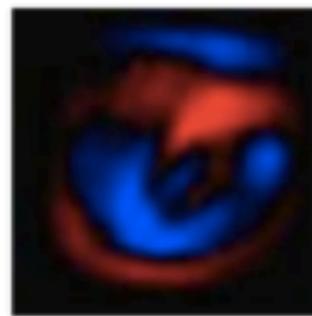
3



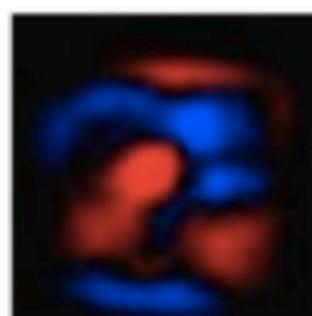
4



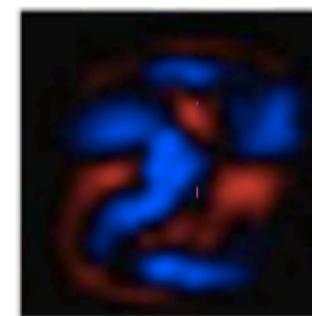
5



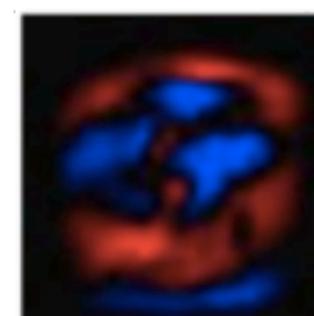
6



7



8



9

MNIST code

```
import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)

import tensorflow as tf
x = tf.placeholder("float", [None, 784])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))

matm=tf.matmul(x,W)
y = tf.nn.softmax(tf.matmul(x,W) + b)
y_ = tf.placeholder("float", [None,10])

cross_entropy = -tf.reduce_sum(y_*tf.log(y))
train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)

sess = tf.Session()
sess.run(tf.initialize_all_variables())

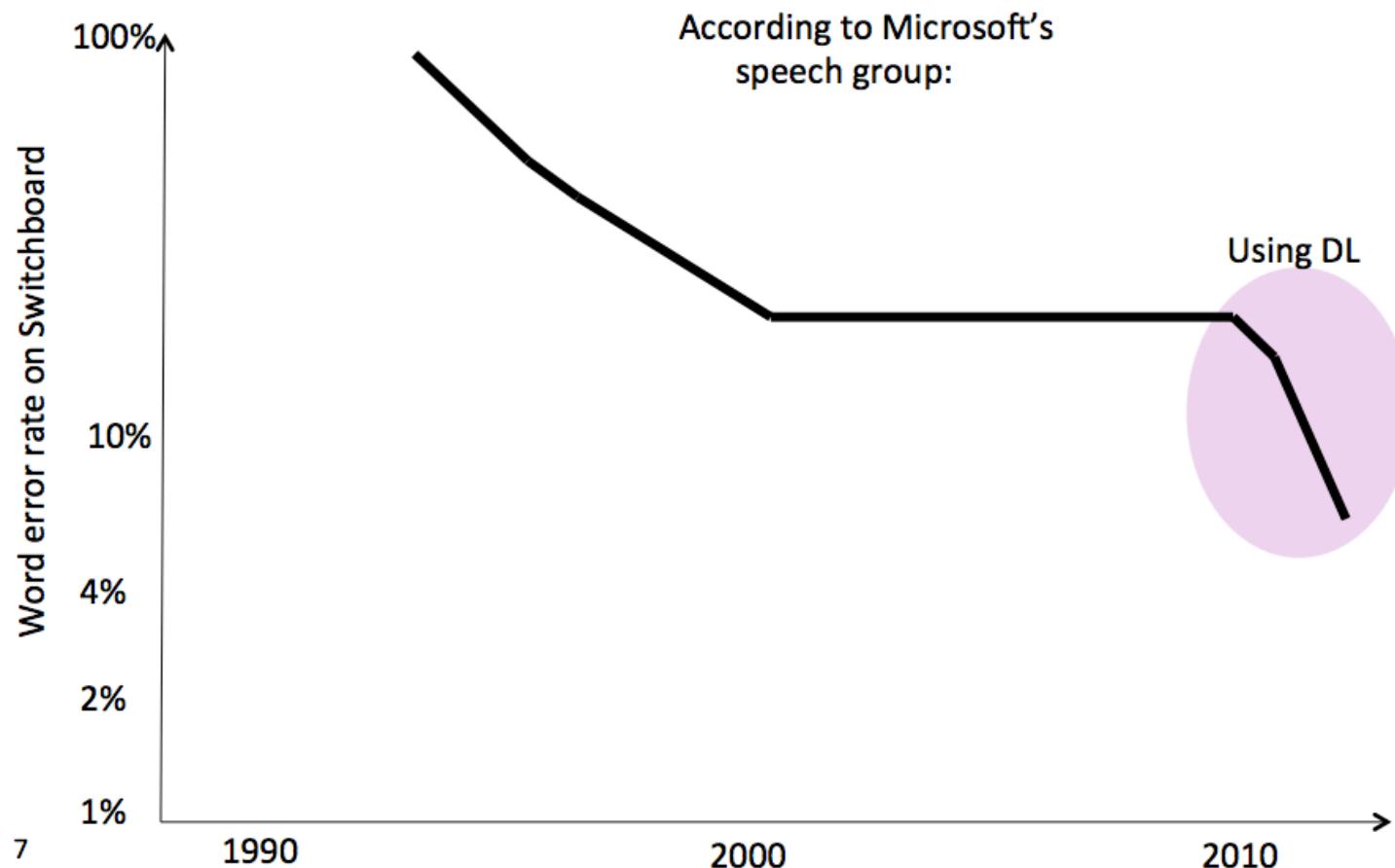
for i in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})
    correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
    accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
    print sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test.labels})
```

기계학습의 응용분야

| 응용 분야 | 적용 사례 |
|-----------|--|
| 인터넷 정보검색 | 텍스트 마이닝, 웹로그 분석, 스팸필터, 문서 분류, 여과, 추출, 요약, 추천 |
| 컴퓨터 시각 | 문자 인식, 패턴 인식, 물체 인식, 얼굴 인식, 장면전환 검출, 화상 복구 |
| 음성인식/언어처리 | 음성 인식, 단어 모호성 제거, 번역 단어 선택, 문법 학습, 대화 패턴 분석 |
| 모바일 HCI | 동작 인식, 제스쳐 인식, 휴대기기의 각종 센서 정보 인식, 떨림 방지 |
| 생물정보 | 유전자 인식, 단백질 분류, 유전자 조절망 분석, DNA 칩 분석, 질병 진단 |
| 바이오메트릭스 | 홍채 인식, 심장 박동수 측정, 혈압 측정, 당뇨치 측정, 지문 인식 |
| 컴퓨터 그래픽 | 데이터기반 애니메이션, 캐릭터 동작 제어, 역운동학, 행동 진화, 가상현실 |
| 로보틱스 | 장애물 인식, 물체 분류, 지도 작성, 무인자동차 운전, 경로 계획, 모터 제어 |
| 서비스업 | 고객 분석, 시장 클러스터 분석, 고객 관리(CRM), 마켓팅, 상품 추천 |
| 제조업 | 이상 탐지, 에너지 소모 예측, 공정 분석 계획, 오류 예측 및 분류 |

Speech Recognition (Acoustic Modeling)

- ~2010 GMM-HMM (Dynamic Bayesian Models)
- ~2013 DNN-HMM (Deep Neural Networks)
- ~Current LSTM-RNN (Recurrent Neural Networks)



Movie Recommendations in Netflix

Smart Models



- Regression models (Logistic, Linear, Elastic nets)
- GBDT/RF
- SVD & other MF models
- Factorization Machines
- Restricted Boltzmann Machines
- Markov Chains & other graphical models
- Clustering (from k-means to HDP)
- Deep ANN
- LDA
- Association Rules
- ...



Human-Level Object Recognition

- ImageNet
- Large-Scale Visual Recognition Challenge
 - Image Classification / Localization
 - 1.2M labeled images, 1000 classes
 - Convolutional Neural Networks (CNNs) has been dominating the contest since..
 - 2012 non-CNN: 26.2% (top-5 error)
 - 2012: (Hinton, AlexNet) 15.3%
 - 2013: (Clarifai) 11.2%
 - 2014: (Google, GoogLeNet) 6.7%
 - 2015: (Google) 4.9%
 - Beyond human-level performance

ILSVRC (Image-net Large Scale Visual Recognition Challenge)



a woman is playing tennis on a tennis court



a wooden bench sitting in the middle of a forest



a close up of a plate of food with broccoli



a microwave oven sitting on top of a counter



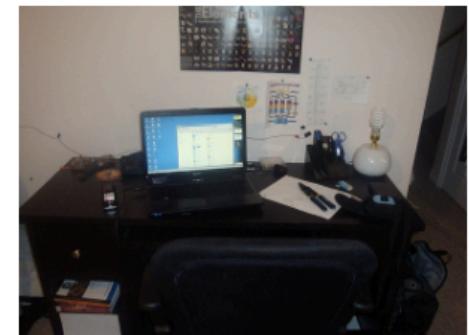
a young man holding a tennis racquet on top of a tennis court



a bathroom with a sink and a toilet

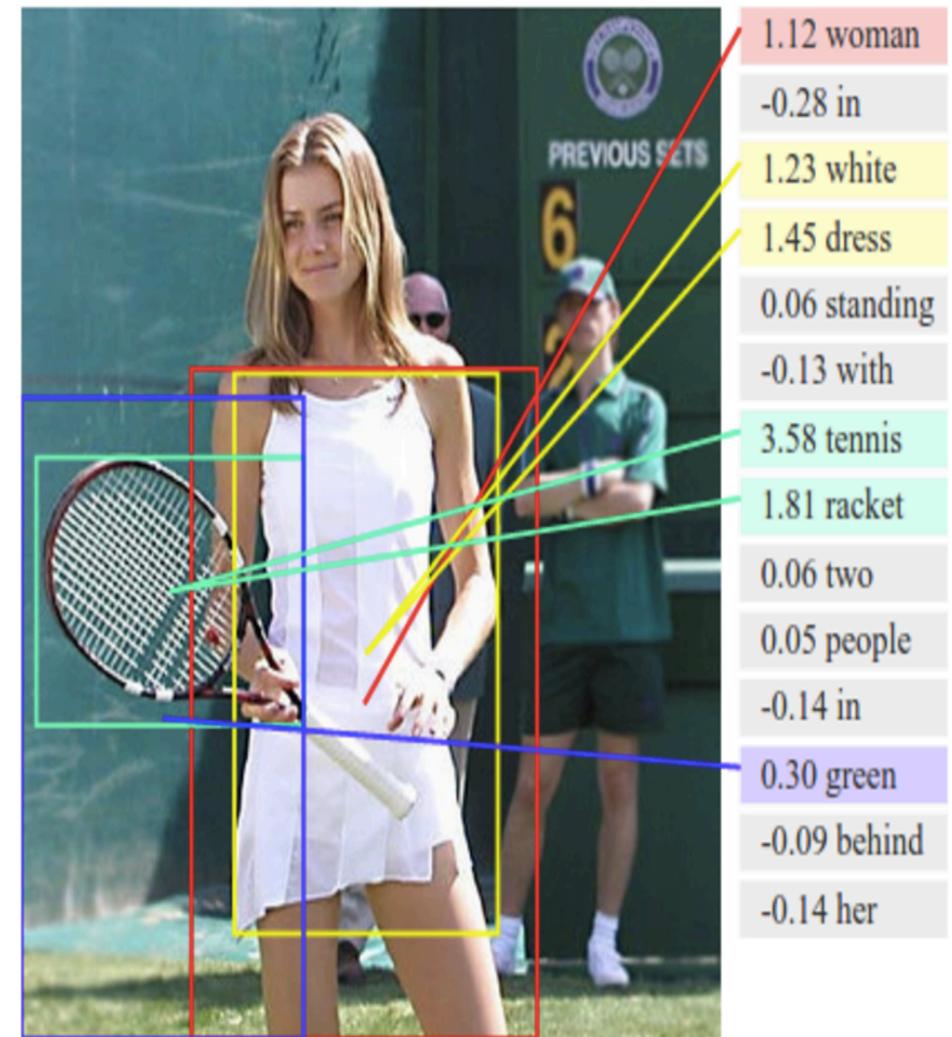
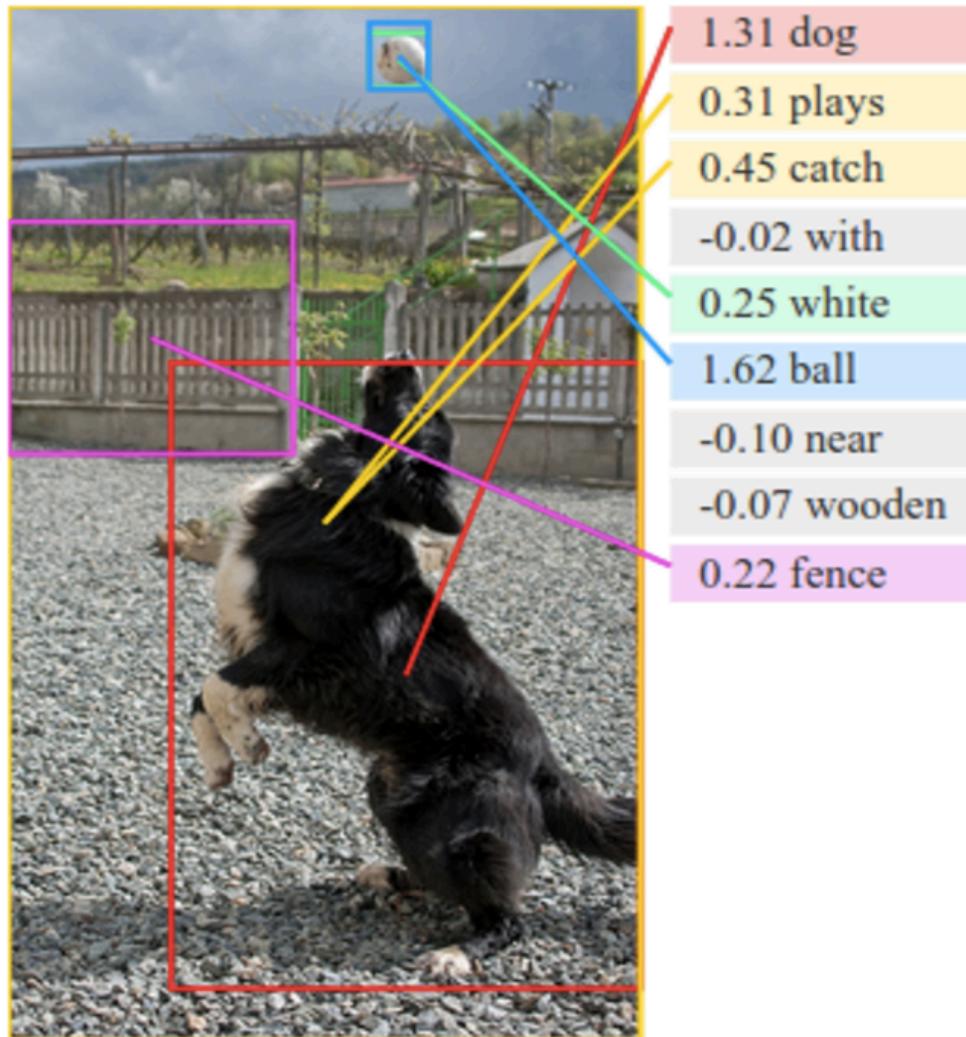


a man riding on the back of a brown horse



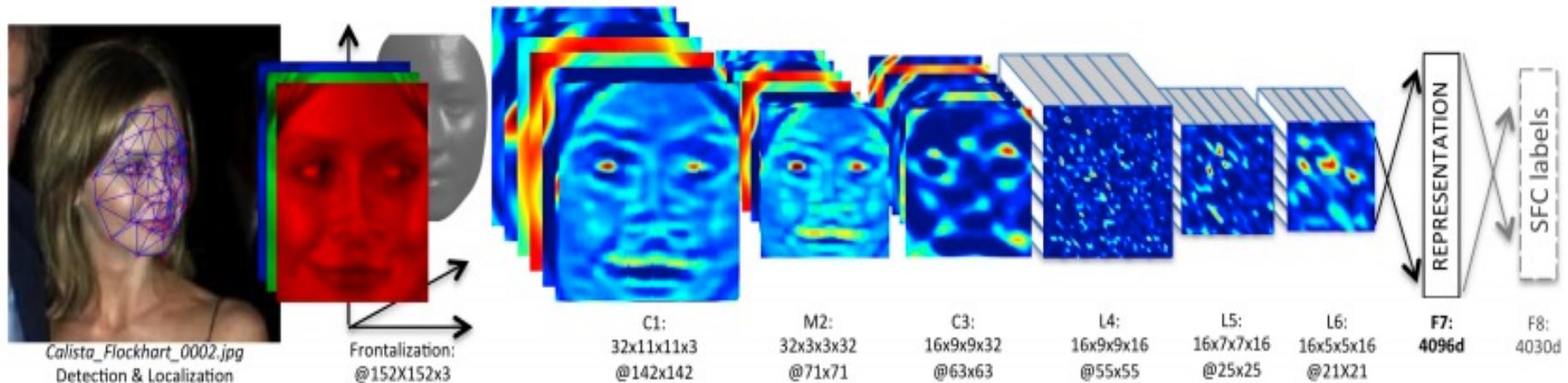
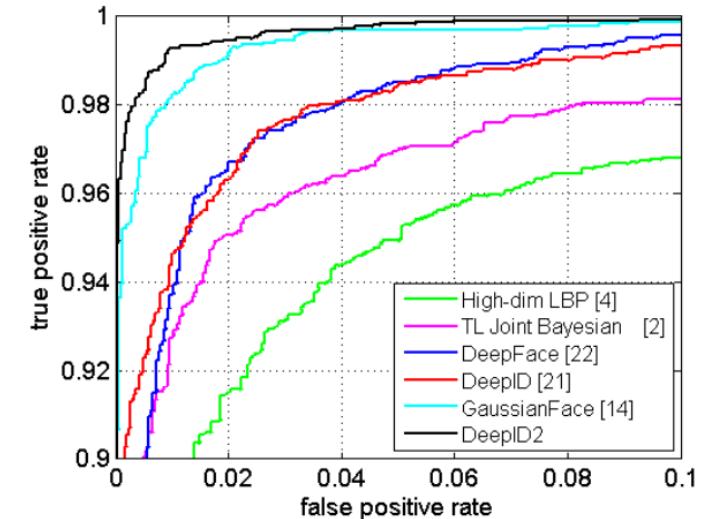
a desk with a laptop and a monitor

How to the Object recognition ?



Human-Level Face Recognition

- Convolutional neural networks based face recognition system is dominant
- 99.15% face verification accuracy on LFW dataset in DeepID2 (2014)
 - Beyond human-level recognition



Source: Taigman et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR'14

Emphasis?



How the TensorFlow is used today





Thanks you!

Q&A