Development of Deep Learning & Attitude of Sharing

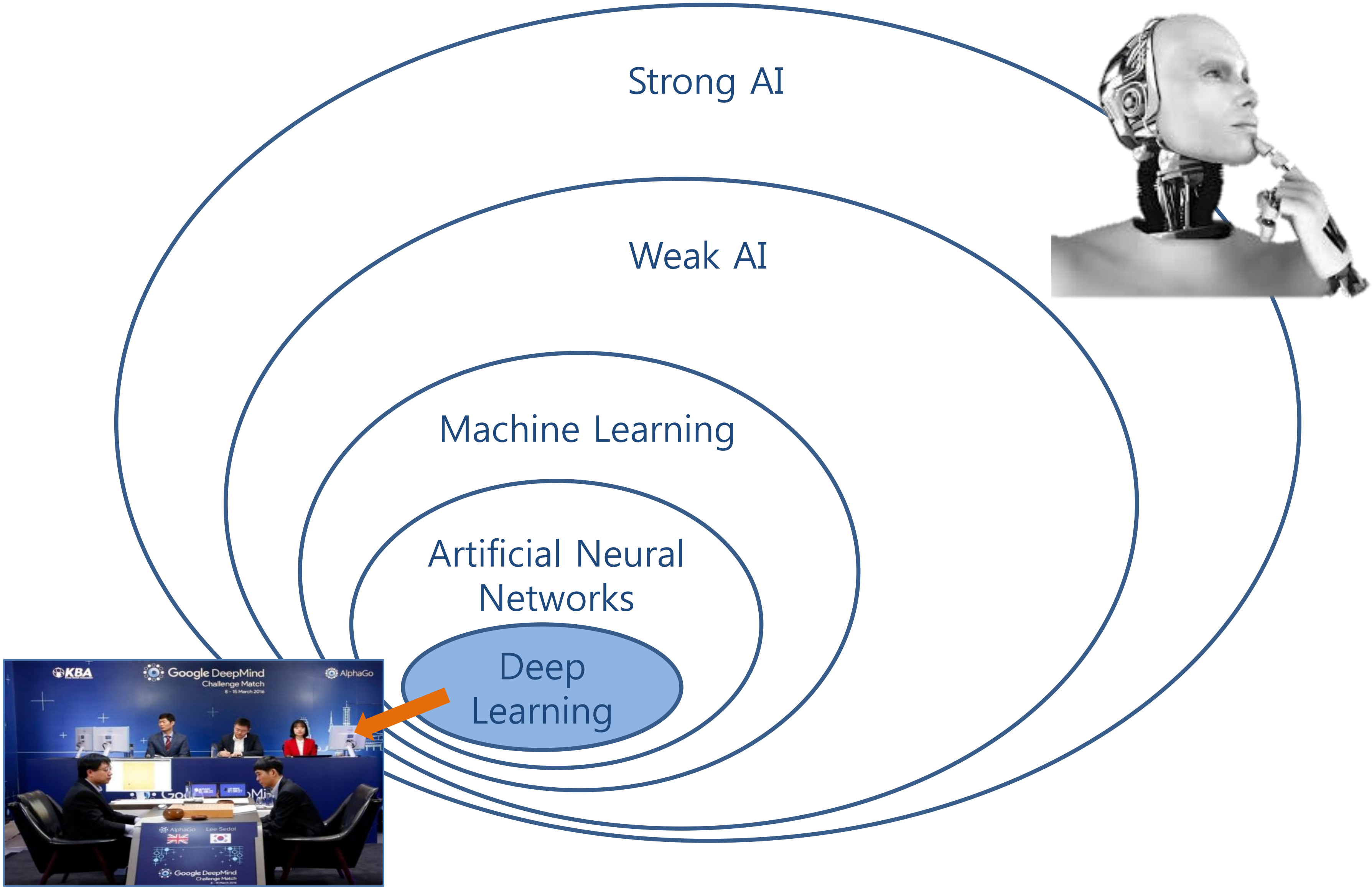
Jooyoul Lee
LG CNS

Agenda

1. Deep Learning overview
2. Why is Deep Learning growing so fast?
3. Deep Learning Tools & Open Source
4. Wrap-Up

1. Deep Learning overview

AI, Machine Learning, Deep Learning



Strong AI

Weak AI

Machine Learning

Artificial Neural Networks

Deep Learning

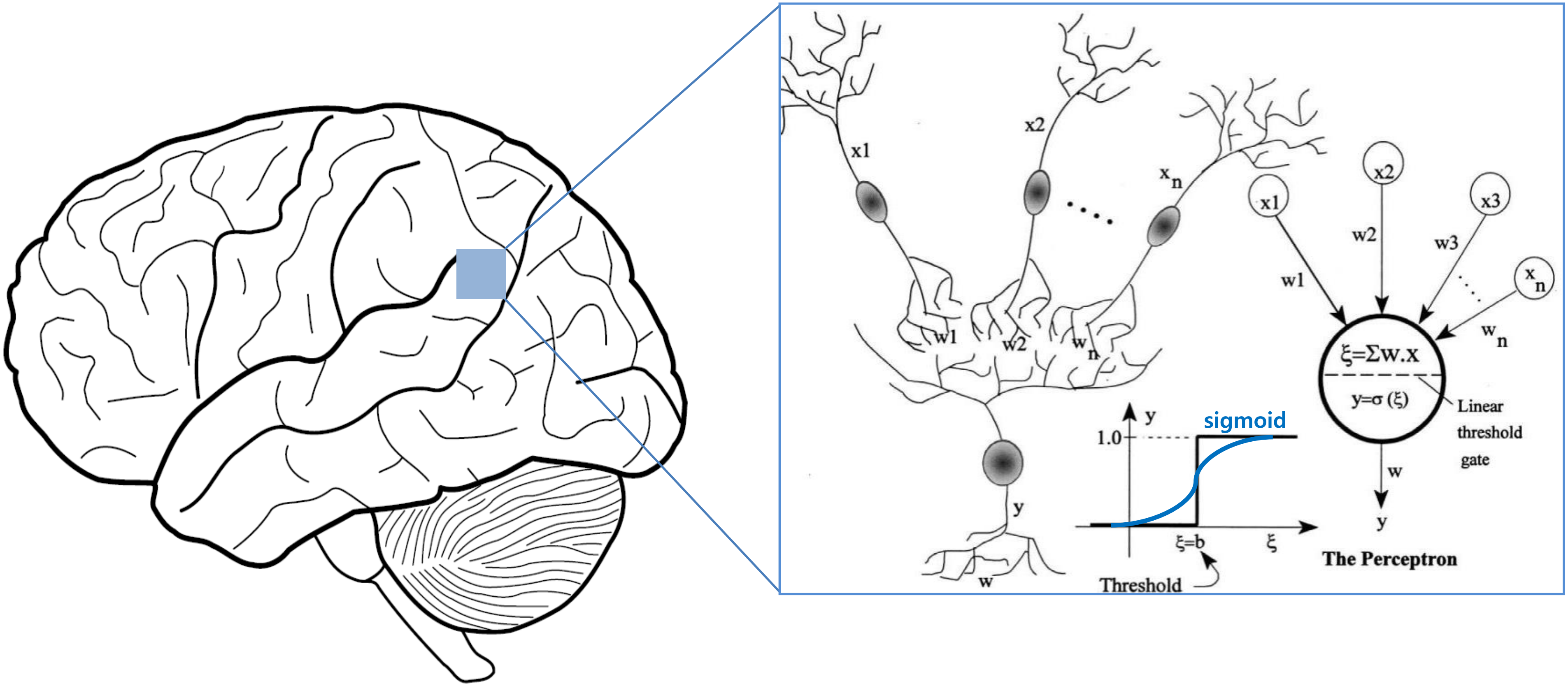
1.1 Machine Learning & Neural Nets

- Machine Learning Algorithms

	Unsupervised	Supervised
Continuous	• Clustering & Dimensionality reduction – K-means – LDA(Latent Dirichlet Allocation) – SVD(Singular Value Decomposition) – PCA(Principal Component Analysis)	• Regression – Liner – Polynomial • Decision Tree • Random Forests • Artificial Neural Networks
Categorical	• Gaussian Mixture • HMM(Hidden Markov Model)	• Classification – Logistic Regression – Naïve-Bayes – SVM(Support Vector Machine)

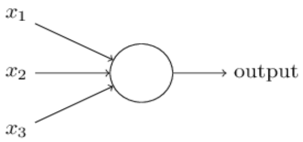
1.1 Machine Learning & Neural Nets

- Artificial Neural Networks (ANN)

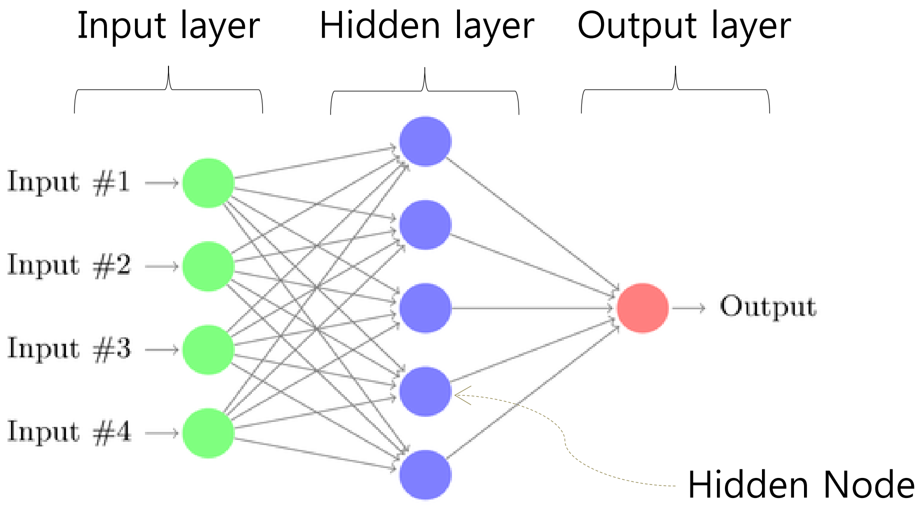


1.1 Machine Learning & Neural Nets

- More hidden layer, more powerful



[Perceptron]

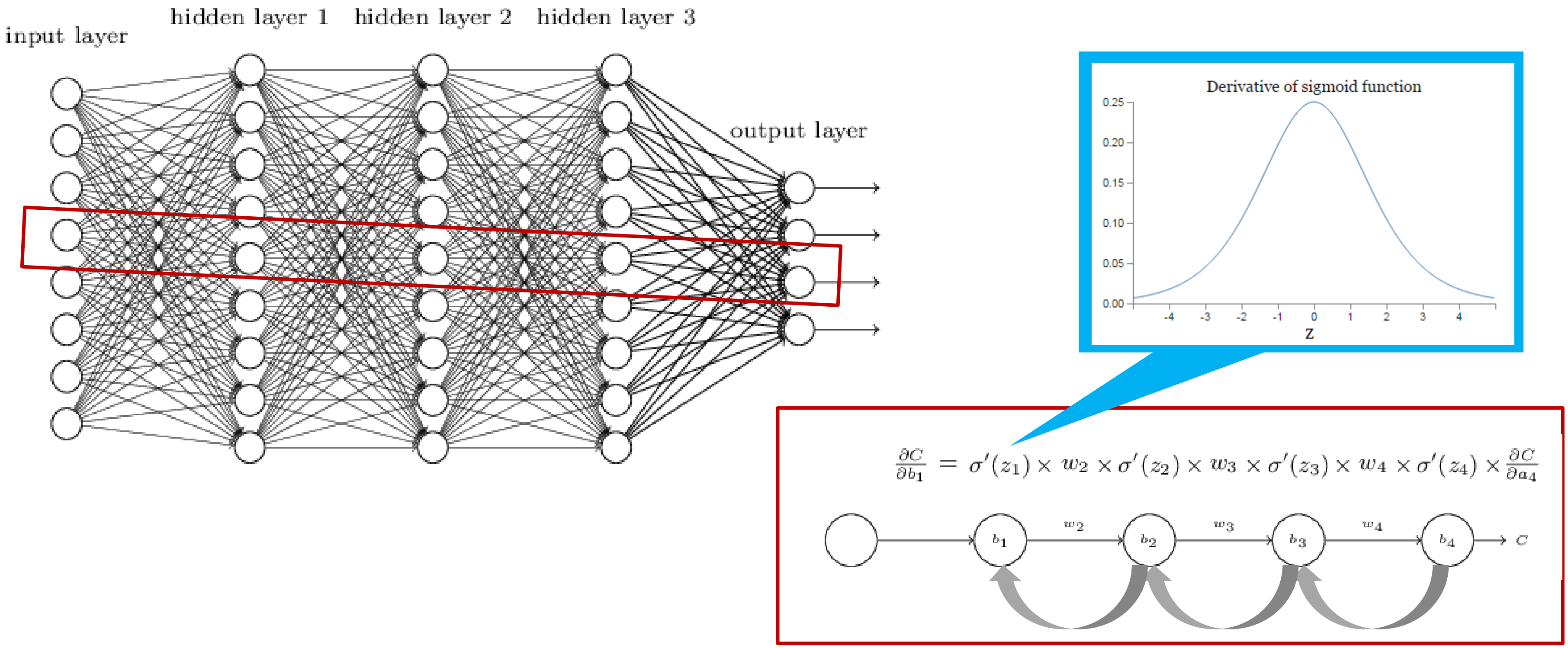


[Multi-Layer Perceptron]

1.1 Machine Learning & Neural Nets

- But, More hidden layer, hard to train!

  ➔Why? The vanishing gradient problem & so on…



$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$

Michael Nielsen (2015), http://neuralnetworksanddeeplearning.com/

1.1 Machine Learning & Neural Nets

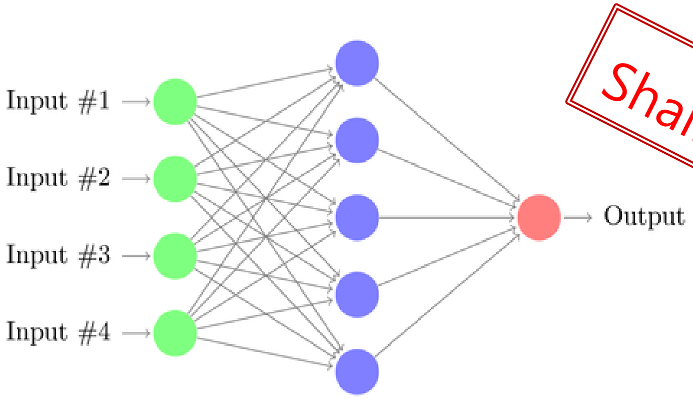
- Breakthrough!

- Hinton, Osindero & Teh « A Fast Learning Algorithm for Deep Belief Nets », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », *NIPS'2006*

1.2 What is Deep Learning

Deep Neural Networks + Machine Learning = Deep Learning

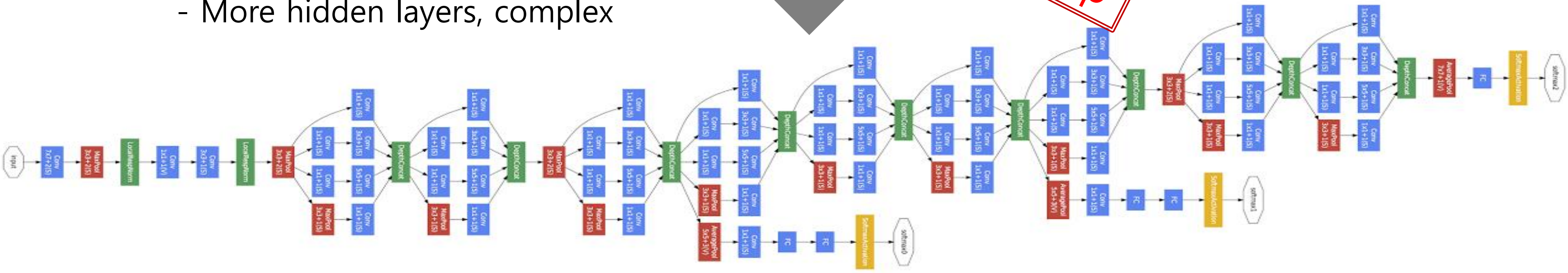
- Neural Networks
  - 1~2 hidden layers



Shallow

- Deep Neural Networks
  - More hidden layers, complex

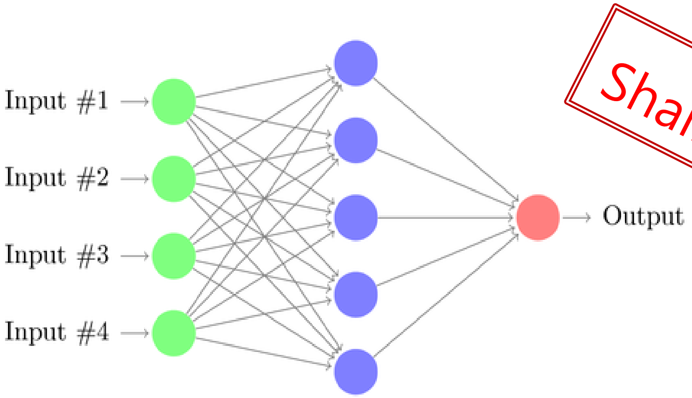
Deep



1.2 What is Deep Learning

Deep Neural Networks + Machine Learning = Deep Learning

- Neural Networks
  - 1~2 hidden layers



Shallow

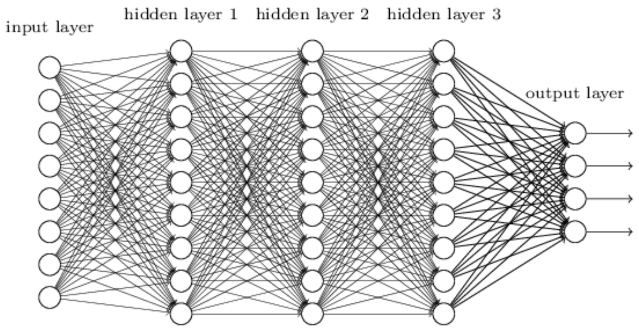
- Deep Neural Networks
  - More hidden layers, complex

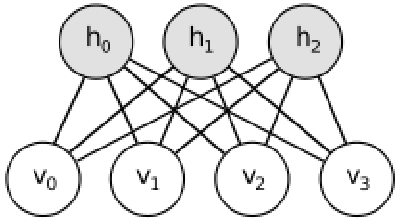
Ultra Deep!!!

1.2 What is Deep Learning

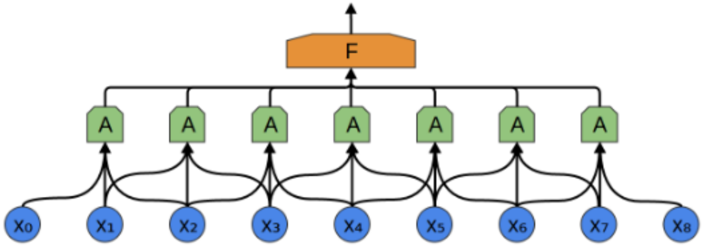
Deep Neural Networks + Machine Learning = Deep Learning

- Fully-Connected / Undirected
  - DNN(Deep Neural Networks)
  - RBM(Restricted Boltzmann Machine)
  - DBN(Deep Belief Network)

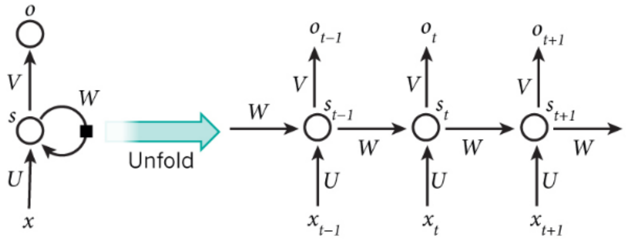




- Convolutional
  - LeNet
  - AlexNet, VGGNet
  - GoogleNet
  - ResNet



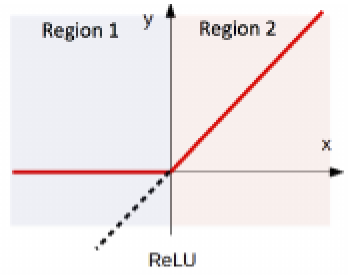
- Recurrent
  - LSTM(Long Short-Term Memory)
  - GRU(Gated Recurrent Unit)
  - Memory Networks



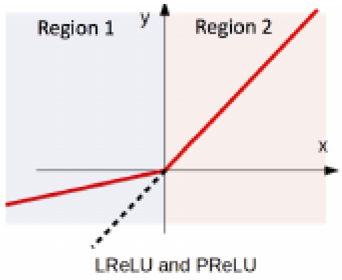
1.2 What is Deep Learning

Deep Neural Networks + Machine Learning = Deep Learning

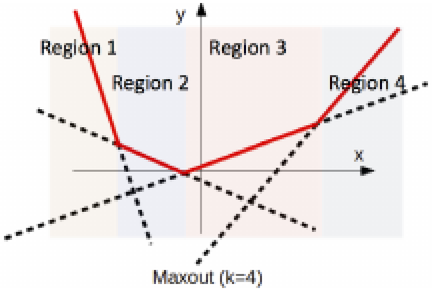
- Activation Function
  - ReLU(Rectified Linear Units)
  - Leaky ReLU
  - Maxout



ReLU

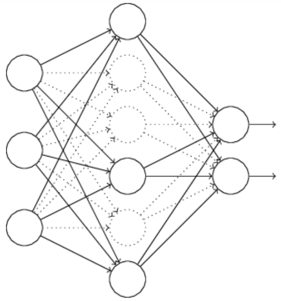


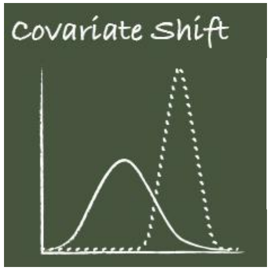
LReLU and PReLU



Maxout (k=4)

- Regularization
  - Drop-Out
  - Batch Normalization

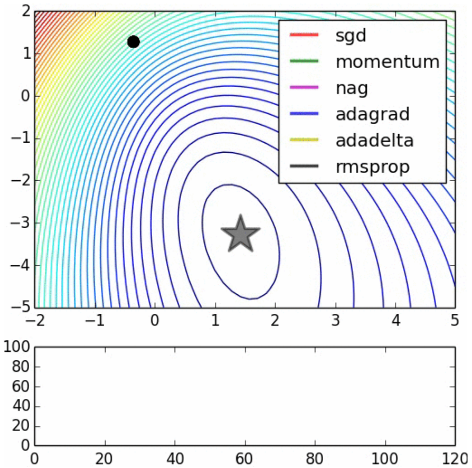




$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}}$$

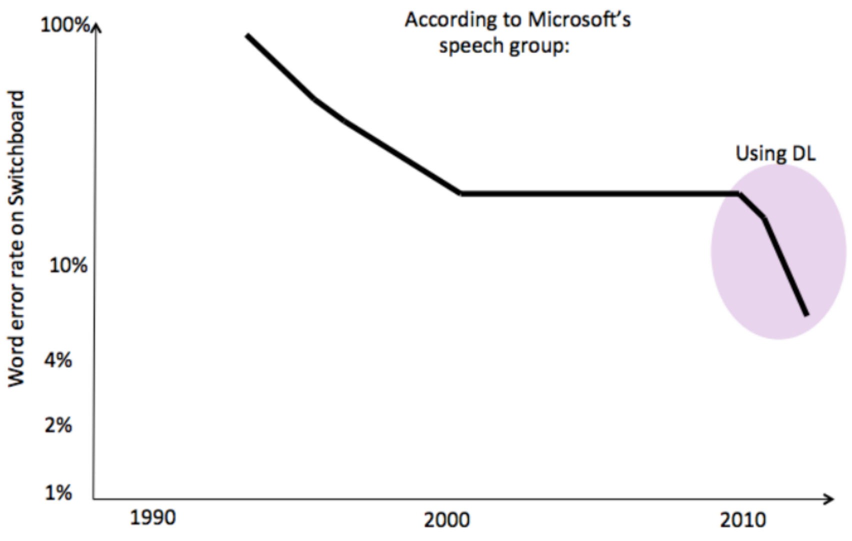
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$$

- Optimization
  - SGD(Stochastic Gradient Descent)
  - AdaGrad
  - RMSprop
  - Adam

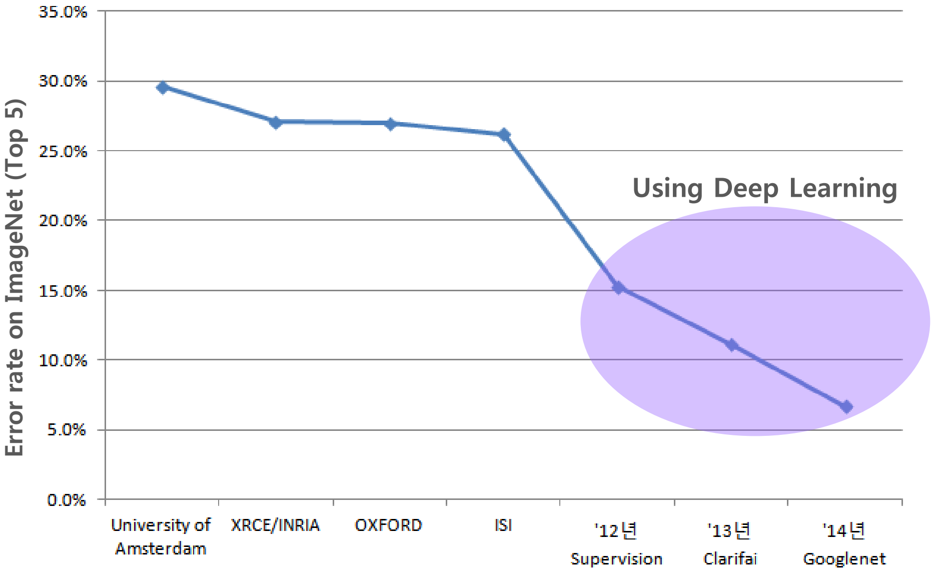


1.3 Why Deep Learning?

- The state-of-the-art in speech recognition, visual object recognition



Speech Recognition

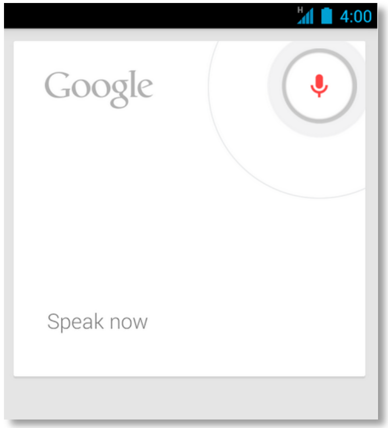


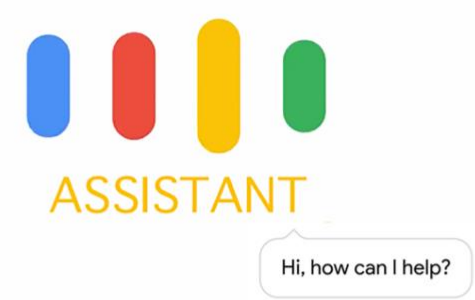
Visual Object Recognition (ILSVRC[1])

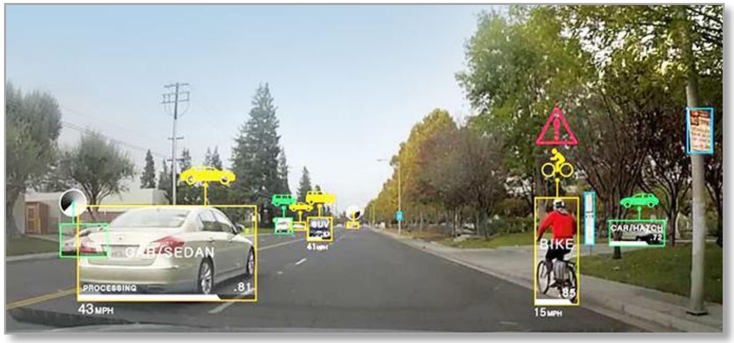
[1] ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

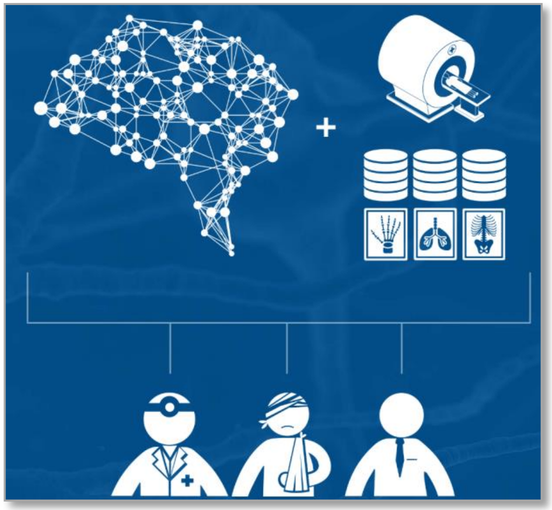
1.3 Why Deep Learning?

- Deep Learning Everywhere



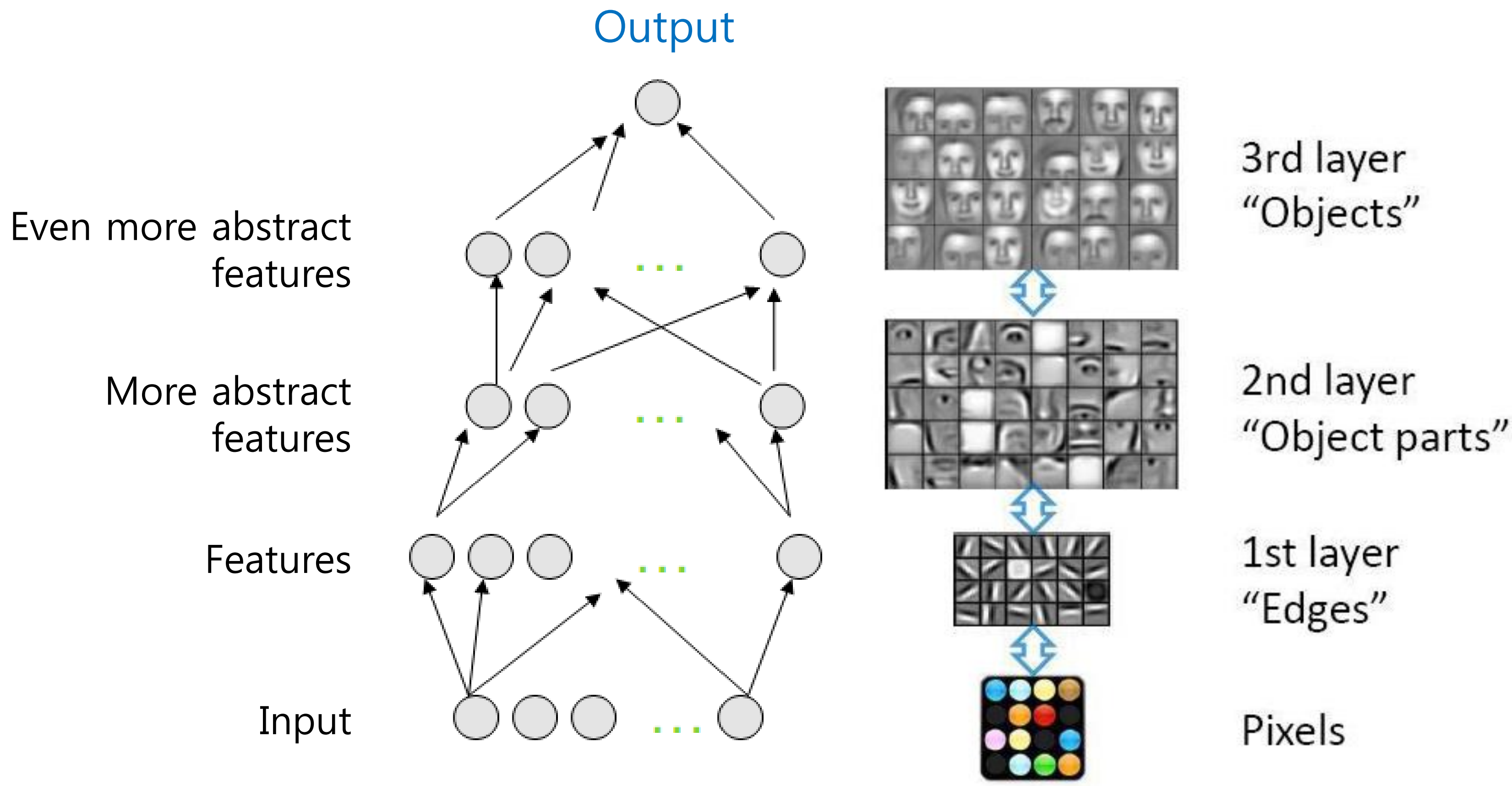






1.3 Why Deep Learning?

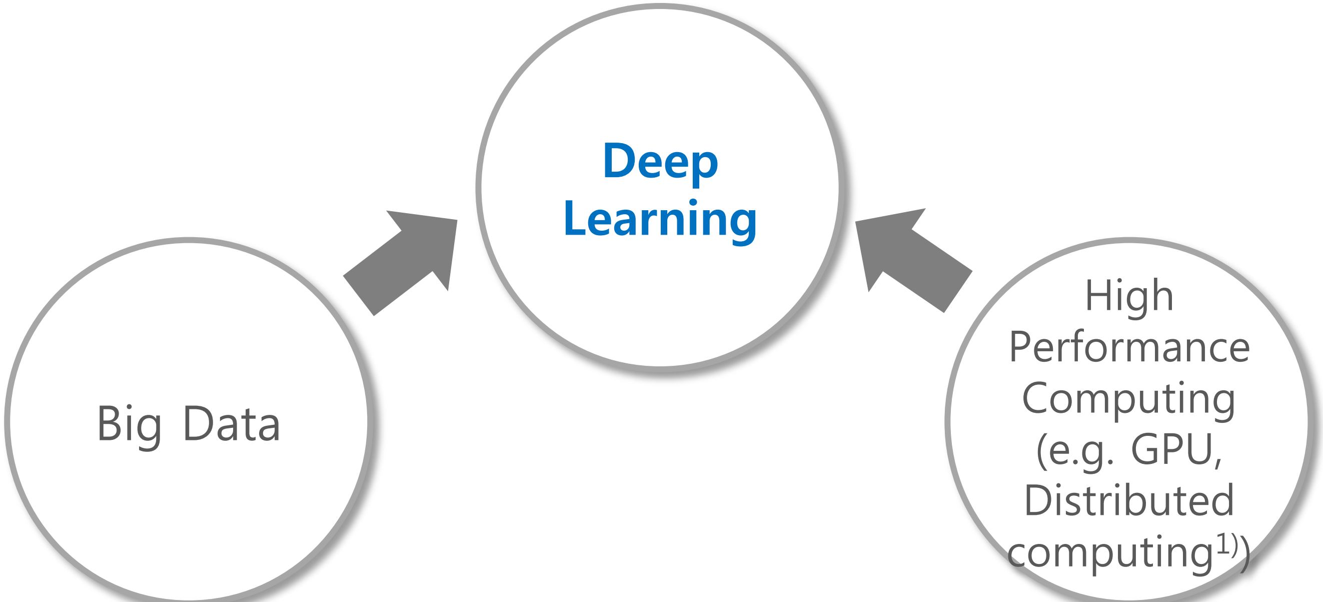
- Why is Deep Learning working so well?
  ➔ Learning multiple levels of representation/abstraction



2. Why is Deep Learning growing so fast?

2.1 Why is Deep Learning growing so fast?

- Big Data & High Performance Computing(HPC)



Deep Learning

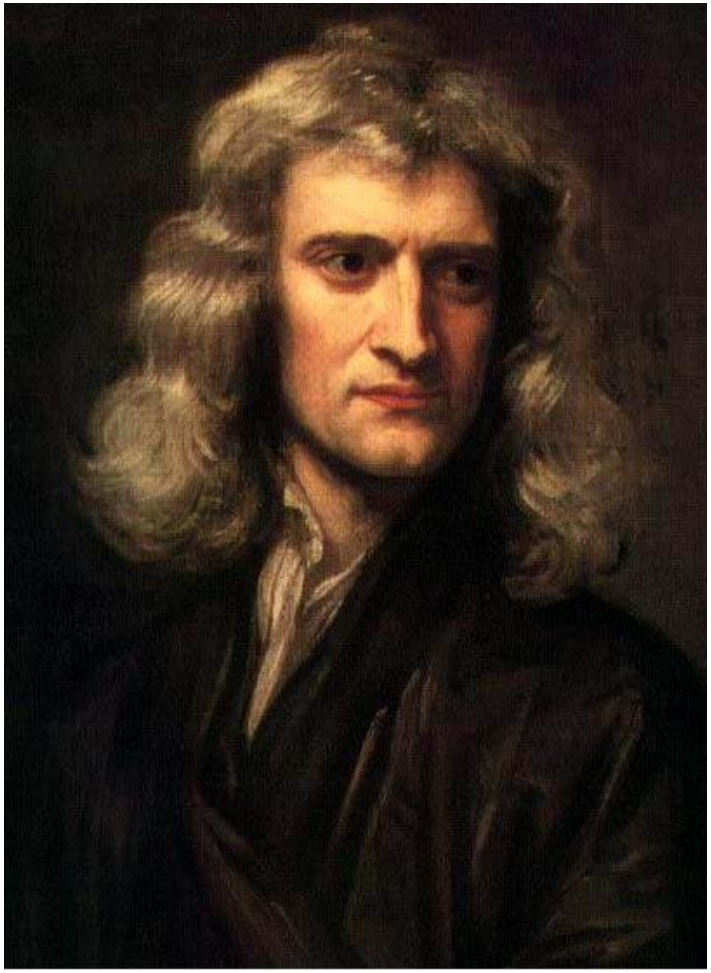
Big Data

High Performance Computing (e.g. GPU, Distributed computing[1])

[1] Hadoop, Spark

2.1 Why is Deep Learning growing so fast?

- The shoulders of giants

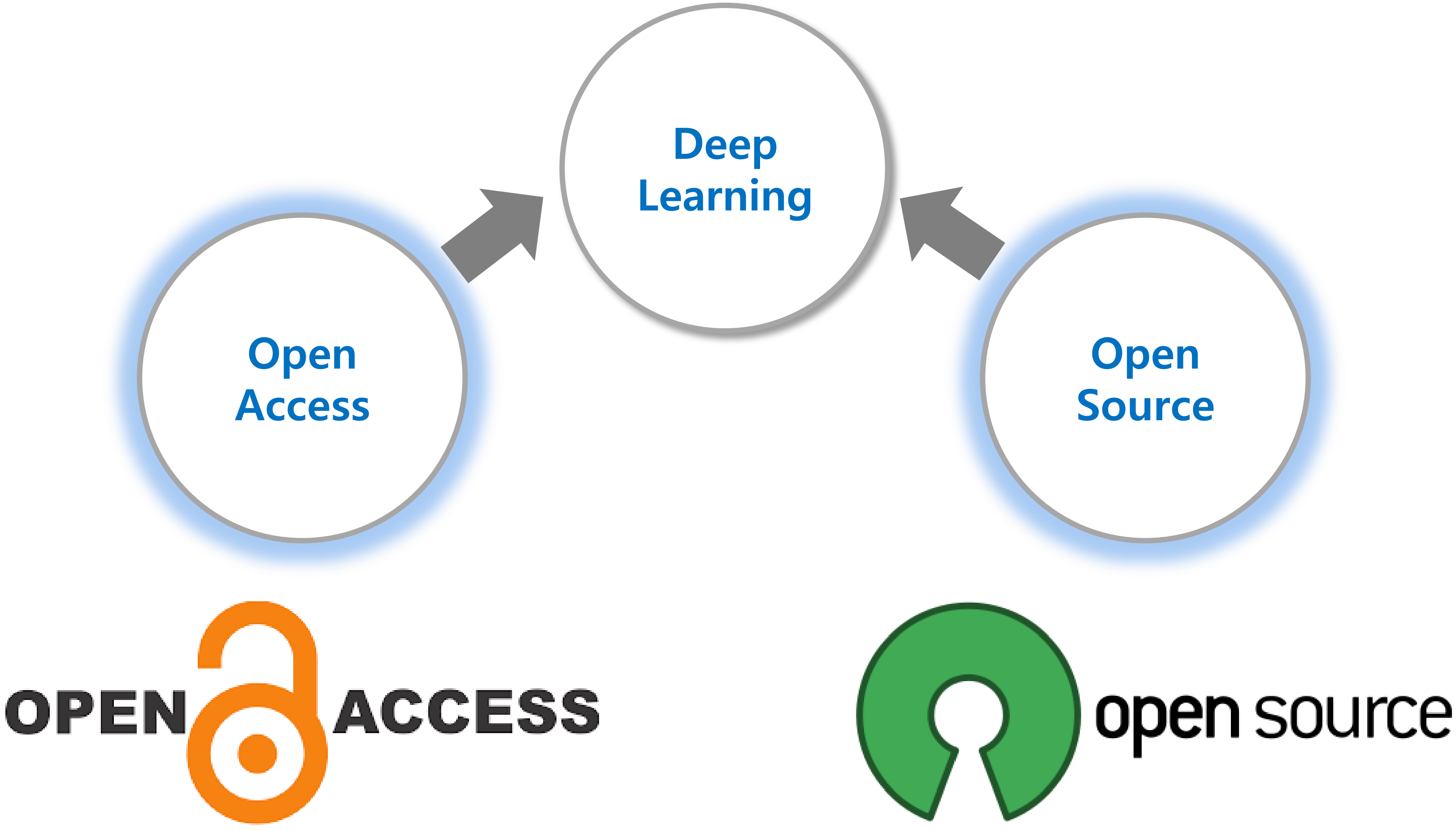


"If I have seen further than others,
it is by standing Upon the shoulders of giants."

– Issac Newton

2.1 Why is Deep Learning growing so fast?

- Open Access & Open Source



Deep Learning

Open Access

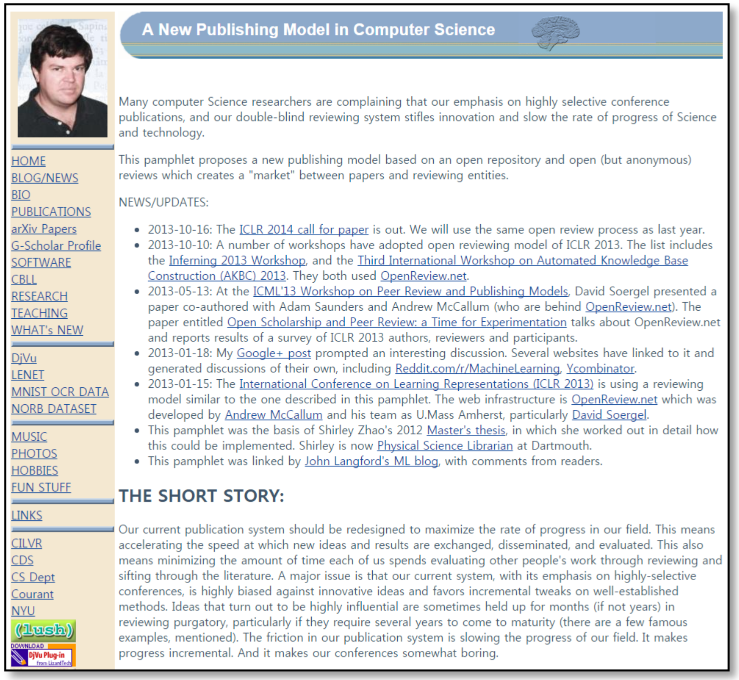
Open Source

2.2 Open Access

- "Open access refers to online research outputs that are free of all restrictions on access (e.g. access tolls) and free of many restrictions on use (e.g. certain copyright and license restrictions)."_from Wikipedia

2.2 Open Access

- arXiv.org : repository of e-prints of scientific papers
  - Open access to e-prints in Physics, Mathematics, **Computer Science**, Quantitative Biology, Quantitative Finance and Statistics
  - Many Deep Learning researchers submit papers
  - So, you can find NIPS, ICML, ICLR[1] papers at arXiv.org



A New Publishing Model in Computer Science

Many computer Science researchers are complaining that our emphasis on highly selective conference publications, and our double-blind reviewing system stifles innovation and slow the rate of progress of Science and technology.

This pamphlet proposes a new publishing model based on an open repository and open (but anonymous) reviews which creates a "market" between papers and reviewing entities.

NEWS/UPDATES:

- 2013-10-16: The ICLR 2014 call for paper is out. We will use the same open review process as last year.
- 2013-10-10: A number of workshops have adopted open reviewing model of ICLR 2013. The list includes the Inferning 2013 Workshop, and the Third International Workshop on Automated Knowledge Base Construction (AKBC) 2013. They both used OpenReview.net.
- 2013-05-13: At the ICML'13 Workshop on Peer Review and Publishing Models, David Soergel presented a paper co-authored with Adam Saunders and Andrew McCallum (who are behind OpenReview.net). The paper entitled Open Scholarship and Peer Review: a Time for Experimentation talks about OpenReview.net and reports results of a survey of ICLR 2013 authors, reviewers and participants.
- 2013-01-18: My Google+ post prompted an interesting discussion. Several websites have linked to it and generated discussions of their own, including Reddit.com/r/MachineLearning, Ycombinator.
- 2013-01-15: The International Conference on Learning Representations (ICLR 2013) is using a reviewing model similar to the one described in this pamphlet. The web infrastructure is OpenReview.net which was developed by Andrew McCallum and his team as U.Mass Amherst, particularly David Soergel.
- This pamphlet was the basis of Shirley Zhao's 2012 Master's thesis, in which she worked out in detail how this could be implemented. Shirley is now Physical Science Librarian at Dartmouth.
- This pamphlet was linked by John Langford's ML blog, with comments from readers.

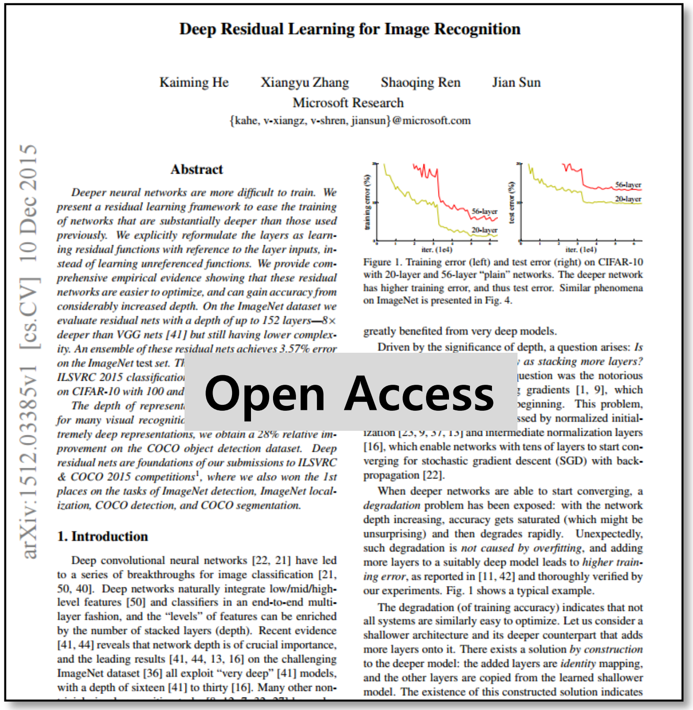
THE SHORT STORY:

Our current publication system should be redesigned to maximize the rate of progress in our field. This means accelerating the speed at which new ideas and results are exchanged, disseminated, and evaluated. This also means minimizing the amount of time each of us spends evaluating other people's work through reviewing and sifting through the literature. A major issue is that our current system, with its emphasis on highly-selective conferences, is highly biased against innovative ideas and favors incremental tweaks on well-established methods. Ideas that turn out to be highly influential are sometimes held up for months (if not years) in reviewing purgatory, particularly if they require several years to come to maturity (there are a few famous examples, mentioned). The friction in our publication system is slowing the progress of our field. It makes progress incremental. And it makes our conferences somewhat boring.

[1] NIPS : Conference on Neural Information Processing Systems
ICML : International Conference on Machine Learning
ICLR : International Conference on Learning Representation

2.3 Open Source

- GitHub : a web-based Git repository hosting service
  - Many Deep Learning researchers open the algorithm implementation codes
  - The codes are mainly released under BSD, MIT, Apache license



Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

arXiv:1512.03385v1 [cs.CV] 10 Dec 2015

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test *set. Th*
ILSVRC 2015 classificatio
on CIFAR-10 with 100 and

The depth of represent
for many visual recognitic
tremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions[1]*, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.*

1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 40]. Deep networks naturally integrate low/mid/high-level features [50] and classifiers in an end-to-end multi-layer fashion, and the "levels" of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [41, 44, 13, 16] on the challenging ImageNet dataset [36] all exploit "very deep" [41] models, with a depth of sixteen [41] to thirty [16]. Many other non-

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

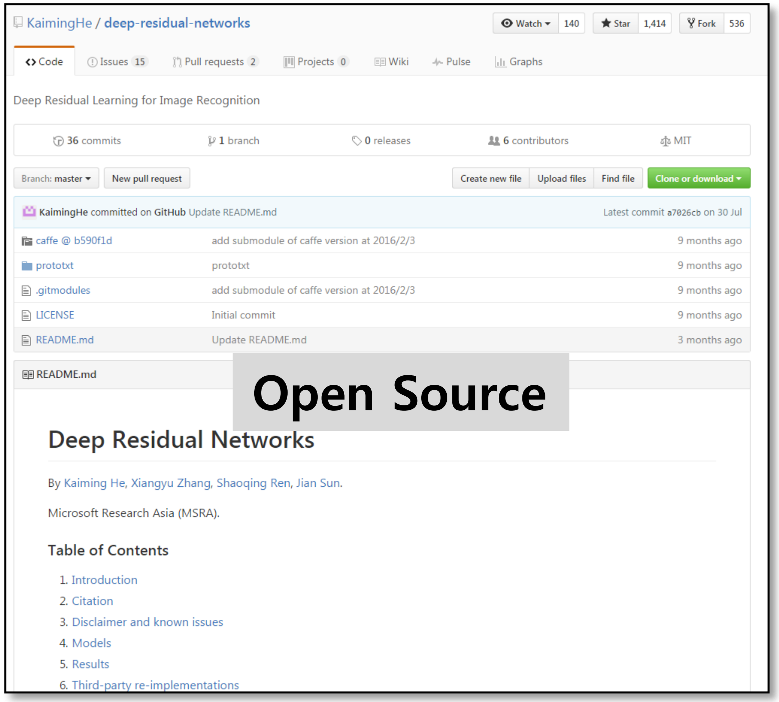
greatly benefited from very deep models.

Driven by the significance of depth, a question arises: *Is*
as stacking more layers?
question was the notorious
gradients [1, 9], which
beginning. This problem,
ssed by normalized initialization [23, 9, 37, 13] and intermediate normalization layers [16], which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a *degradation* problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is *not caused by overfitting*, and adding more layers to a suitably deep model leads to *higher training error*, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution *by construction* to the deeper model: the added layers are *identity* mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates

Open Access



KaimingHe / **deep-residual-networks**

Deep Residual Learning for Image Recognition

36 commits 1 branch 0 releases 6 contributors MIT

KaimingHe committed on GitHub Update README.md Latest commit a7026cb on 30 Jul

caffe @ b590f1d	add submodule of caffe version at 2016/2/3	9 months ago
prototxt	prototxt	9 months ago
.gitmodules	add submodule of caffe version at 2016/2/3	9 months ago
LICENSE	Initial commit	9 months ago
README.md	Update README.md	3 months ago

README.md

Deep Residual Networks

By Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.

Microsoft Research Asia (MSRA).

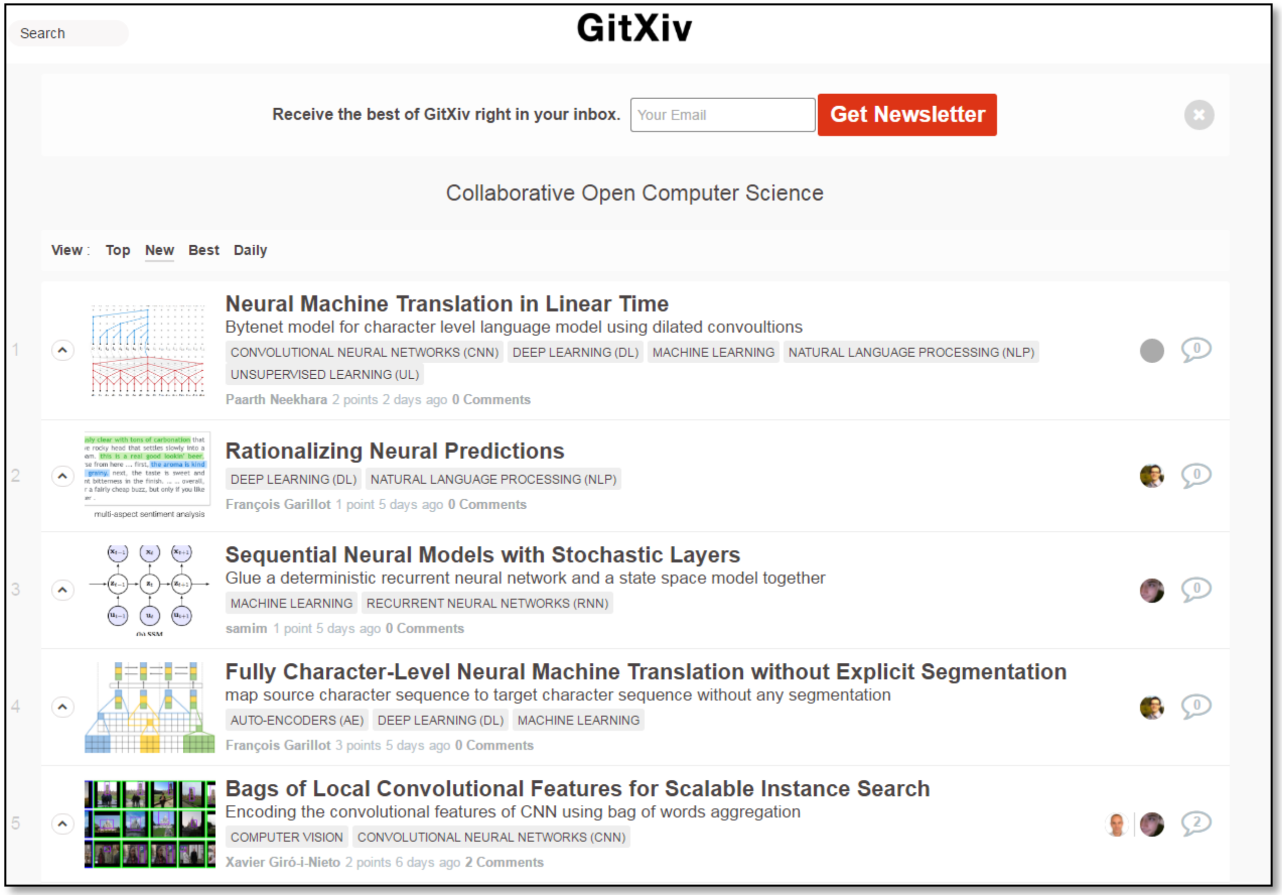
Table of Contents

1. Introduction
2. Citation
3. Disclaimer and known issues
4. Models
5. Results
6. Third-party re-implementations

Open Source

2.4 GitHub + arXiv

- GitXiv.com : a space to share collaborative open computer science projects



Search

GitXiv

Receive the best of GitXiv right in your inbox. Your Email **Get Newsletter**

Collaborative Open Computer Science

View : Top New Best Daily

1 **Neural Machine Translation in Linear Time**
Bytenet model for character level language model using dilated convoultions
CONVOLUTIONAL NEURAL NETWORKS (CNN) DEEP LEARNING (DL) MACHINE LEARNING NATURAL LANGUAGE PROCESSING (NLP) UNSUPERVISED LEARNING (UL)
Paarth Neekhara 2 points 2 days ago 0 Comments

2 **Rationalizing Neural Predictions**
DEEP LEARNING (DL) NATURAL LANGUAGE PROCESSING (NLP)
François Garillot 1 point 5 days ago 0 Comments

3 **Sequential Neural Models with Stochastic Layers**
Glue a deterministic recurrent neural network and a state space model together
MACHINE LEARNING RECURRENT NEURAL NETWORKS (RNN)
samim 1 point 5 days ago 0 Comments

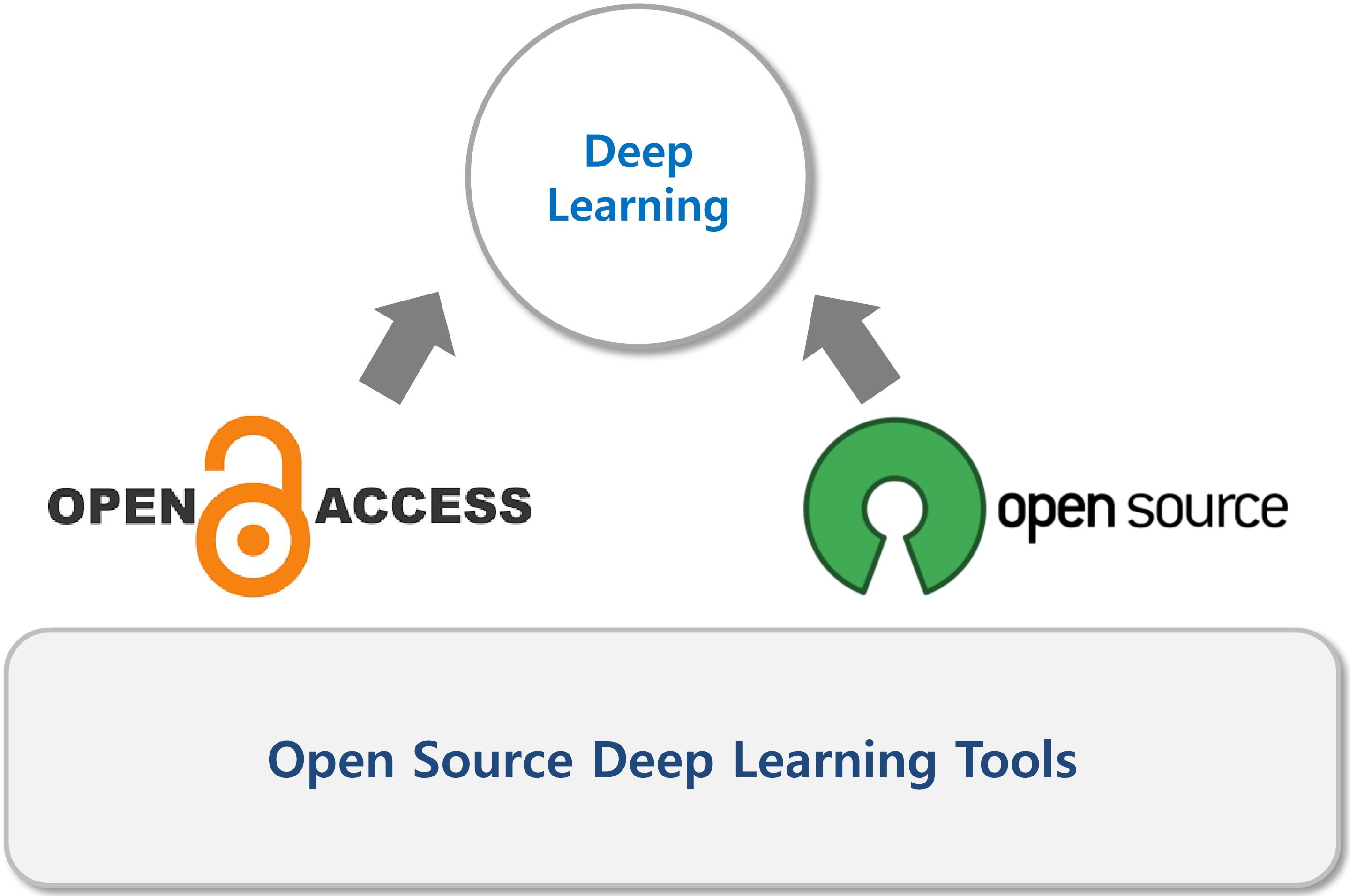
4 **Fully Character-Level Neural Machine Translation without Explicit Segmentation**
map source character sequence to target character sequence without any segmentation
AUTO-ENCODERS (AE) DEEP LEARNING (DL) MACHINE LEARNING
François Garillot 3 points 5 days ago 0 Comments

5 **Bags of Local Convolutional Features for Scalable Instance Search**
Encoding the convolutional features of CNN using bag of words aggregation
COMPUTER VISION CONVOLUTIONAL NEURAL NETWORKS (CNN)
Xavier Giró-i-Nieto 2 points 6 days ago 2 Comments

3. Deep Learning Tools & Open Source

3.1 Why is Deep Learning growing so fast?

- Remind “The shoulders of giants”



Deep Learning

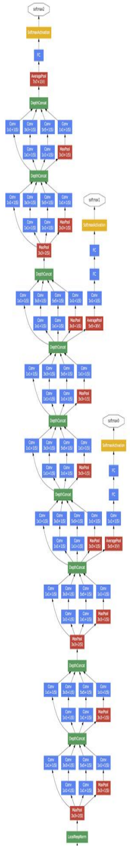
OPEN ACCESS

open source

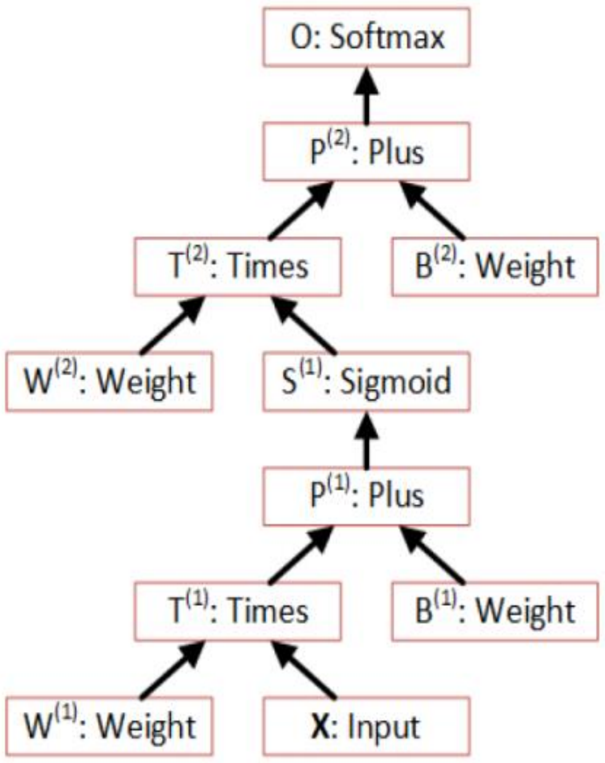
Open Source Deep Learning Tools

3.1 Why is Deep Learning growing so fast?

- Implement a deep neural networks



Deep Neural Nets



O: Softmax

$P^{(2)}$: Plus

$T^{(2)}$: Times

$B^{(2)}$: Weight

$W^{(2)}$: Weight

$S^{(1)}$: Sigmoid

$P^{(1)}$: Plus

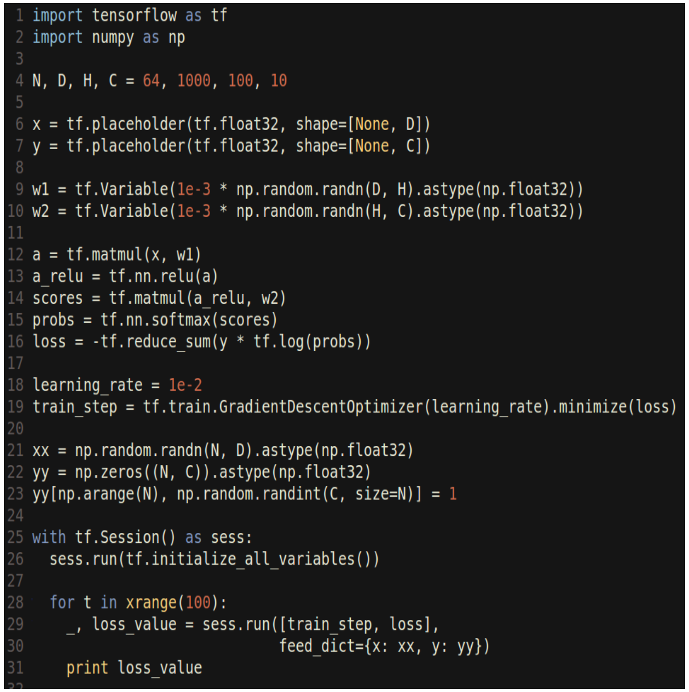
$T^{(1)}$: Times

$B^{(1)}$: Weight

$W^{(1)}$: Weight

X: Input

Computational Graphs

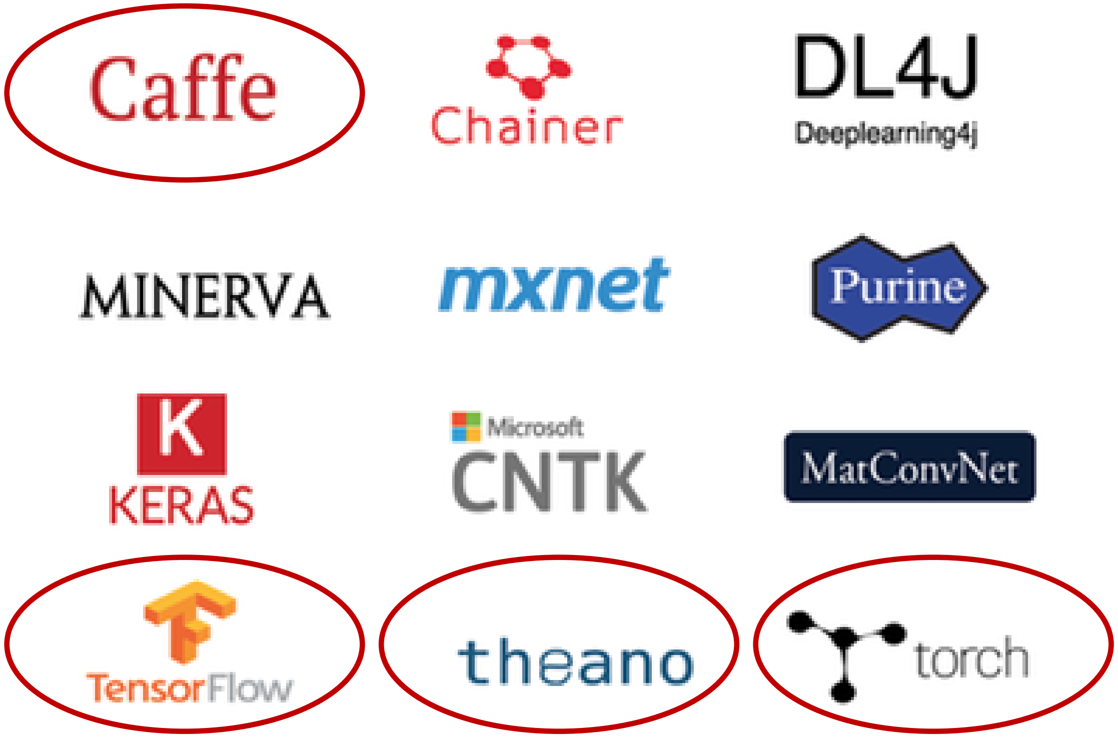


```
import tensorflow as tf
import numpy as np

N, D, H, C = 64, 1000, 100, 10

x = tf.placeholder(tf.float32, shape=[None, D])
y = tf.placeholder(tf.float32, shape=[None, C])

w1 = tf.Variable(1e-3 * np.random.randn(D, H).astype(np.float32))
w2 = tf.Variable(1e-3 * np.random.randn(H, C).astype(np.float32))

a = tf.matmul(x, w1)
a_relu = tf.nn.relu(a)
scores = tf.matmul(a_relu, w2)
probs = tf.nn.softmax(scores)
loss = -tf.reduce_sum(y * tf.log(probs))

learning_rate = 1e-2
train_step = tf.train.GradientDescentOptimizer(learning_rate).minimize(loss)

xx = np.random.randn(N, D).astype(np.float32)
yy = np.zeros((N, C)).astype(np.float32)
yy[np.arange(N), np.random.randint(C, size=N)] = 1

with tf.Session() as sess:
  sess.run(tf.initialize_all_variables())

  for t in xrange(100):
    _, loss_value = sess.run([train_step, loss],
                             feed_dict={x: xx, y: yy})
    print loss_value
```

Code (call API of Deep Learning Tool)

3.2 Deep Learning Tools

- So many Open Source Deep Learning Tools



3.2 Deep Learning Tools

- Caffe : **C**onvolutional **A**rchitecture for **F**ast **F**eature **E**mbedding
  - From U.C. Berkeley, BSD 2-Clause license, Written in C++
  - Pros and Cons[1] :

(+) Good for feedforward networks and image processing

(+) Good for finetuning existing networks

(+) Train models without writing any code

(+) Python interface is pretty useful

(-) Need to write C++ / CUDA for new GPU layers

(-) Not good for recurrent neural networks

(-) Not extensible, bit of a hairball

[1] Stanford CS231n Andrej Karpathy, https://deeplearning4j.org/compare-dl4j-torch7-pylearn#caffe

3.2 Deep Learning Tools

- Torch
  - From NYU, BSD license, Written in C/C++, Lua
  - Used a lot in Facebook and DeepMind
  - Pros and Cons[1] :

(+) Lots of modular pieces that are easy to combine

(+) Easy to write your own layer types and run on GPU

(+) Lots of pre-trained models

(-) Lua

(-) You usually write your own training code (Less plug and play)

[1] Stanford CS231n Andrej Karpathy, https://deeplearning4j.org/compare-dl4j-torch7-pylearn#torch

3.2 Deep Learning Tools

- Theano
  - From University of Montreal, Written in Python
  - High-level wrappers: Keras, Lasagne
  - Pros and Cons[1] :

(+) Python + Numpy

(+) Computational graph is nice abstraction

(+) RNNs fit nicely in computational graph

(-) Hard to navigate, debug, refactor

(-) Large models can have long compile times

(-) Not enough pre-trained models

[1] Stanford CS231n Andrej Karpathy, https://deeplearning4j.org/compare-dl4j-torch7-pylearn#theano

3.2 Deep Learning Tools

- TensorFlow
  - From Google, Apache 2.0 license, Written in C++, Python
  - Very similar to Theano
  - Pros and Cons[1] :

(+) Python + Numpy

(+) Computational graph abstraction, like Theano

(+) Easy Visualizations

(+) Multi-node training

(-) Hard to hack into the networks

(-) Training time is slower than others right now

(-) Not enough pre-trained models (but it will get better)

[1] Stanford CS231n Andrej Karpathy, https://deeplearning4j.org/compare-dl4j-torch7-pylearn#tensorflow

3.2 Deep Learning Tools

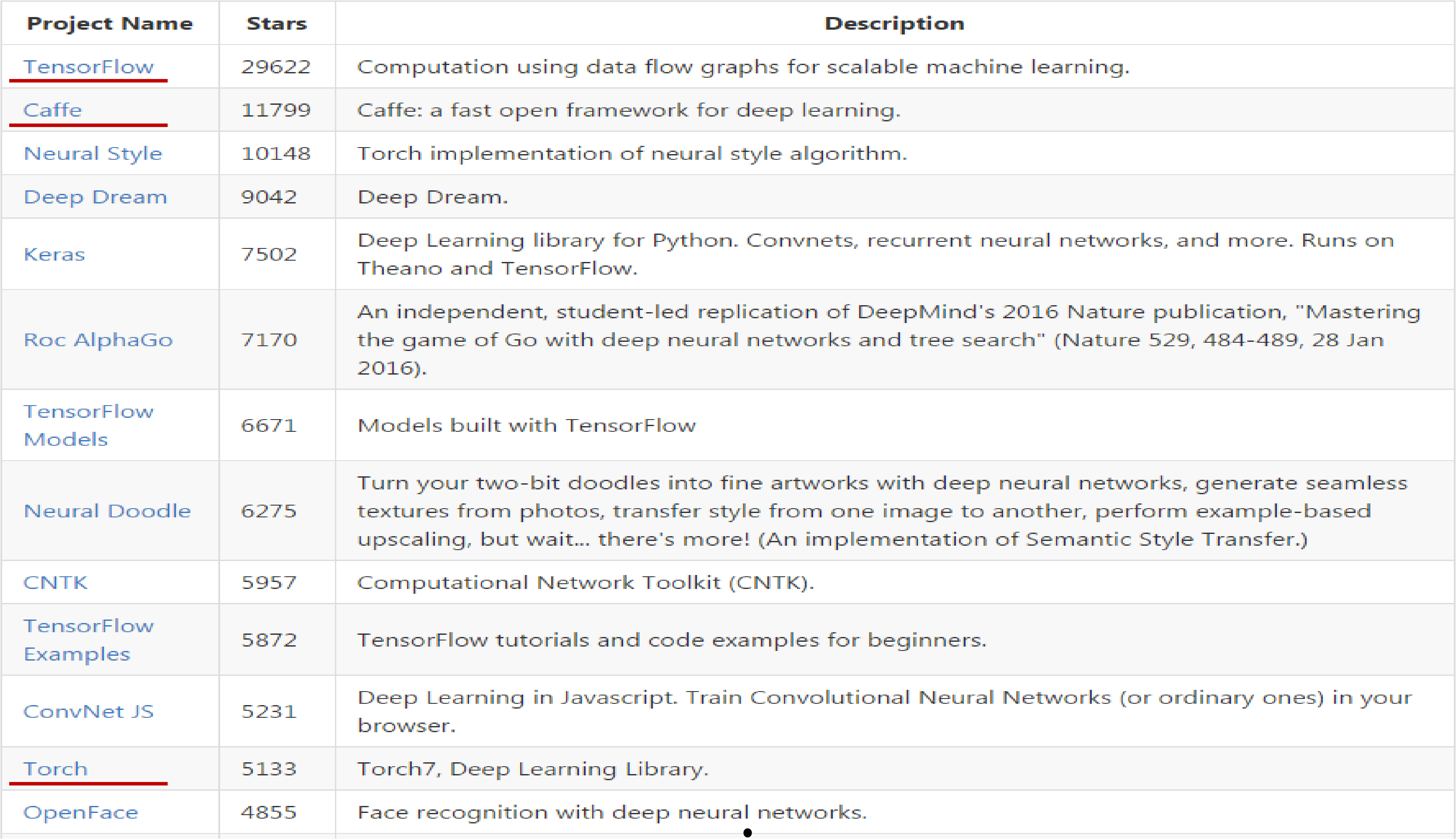
- Overview[1)]

	Caffe	Torch	Theano	TensorFlow
Language	C++, Python	Lua	Python	Python
Pre-trained Model	Yes ++	Yes ++	Yes (Lasagne)	Not enough
Multi-GPU: Data parallel	Yes	Yes	Yes	Yes
Multi-GPU: Model parallel	No	Yes	Experimental	Yes (best)
Readable source code	Yes (C++)	Yes (Lua)	No	No
Good at RNN	No	Mediocre	Yes	Yes (best)

1) Stanford CS231n Andrej Karpathy

3.3 Top Deep Learning Projects

- Popular **GitHub** Projects related to Deep Learning[1]



Project Name	Stars	Description
TensorFlow	29622	Computation using data flow graphs for scalable machine learning.
Caffe	11799	Caffe: a fast open framework for deep learning.
Neural Style	10148	Torch implementation of neural style algorithm.
Deep Dream	9042	Deep Dream.
Keras	7502	Deep Learning library for Python. Convnets, recurrent neural networks, and more. Runs on Theano and TensorFlow.
Roc AlphaGo	7170	An independent, student-led replication of DeepMind's 2016 Nature publication, "Mastering the game of Go with deep neural networks and tree search" (Nature 529, 484-489, 28 Jan 2016).
TensorFlow Models	6671	Models built with TensorFlow
Neural Doodle	6275	Turn your two-bit doodles into fine artworks with deep neural networks, generate seamless textures from photos, transfer style from one image to another, perform example-based upscaling, but wait... there's more! (An implementation of Semantic Style Transfer.)
CNTK	5957	Computational Network Toolkit (CNTK).
TensorFlow Examples	5872	TensorFlow tutorials and code examples for beginners.
ConvNet JS	5231	Deep Learning in Javascript. Train Convolutional Neural Networks (or ordinary ones) in your browser.
Torch	5133	Torch7, Deep Learning Library.
OpenFace	4855	Face recognition with deep neural networks.

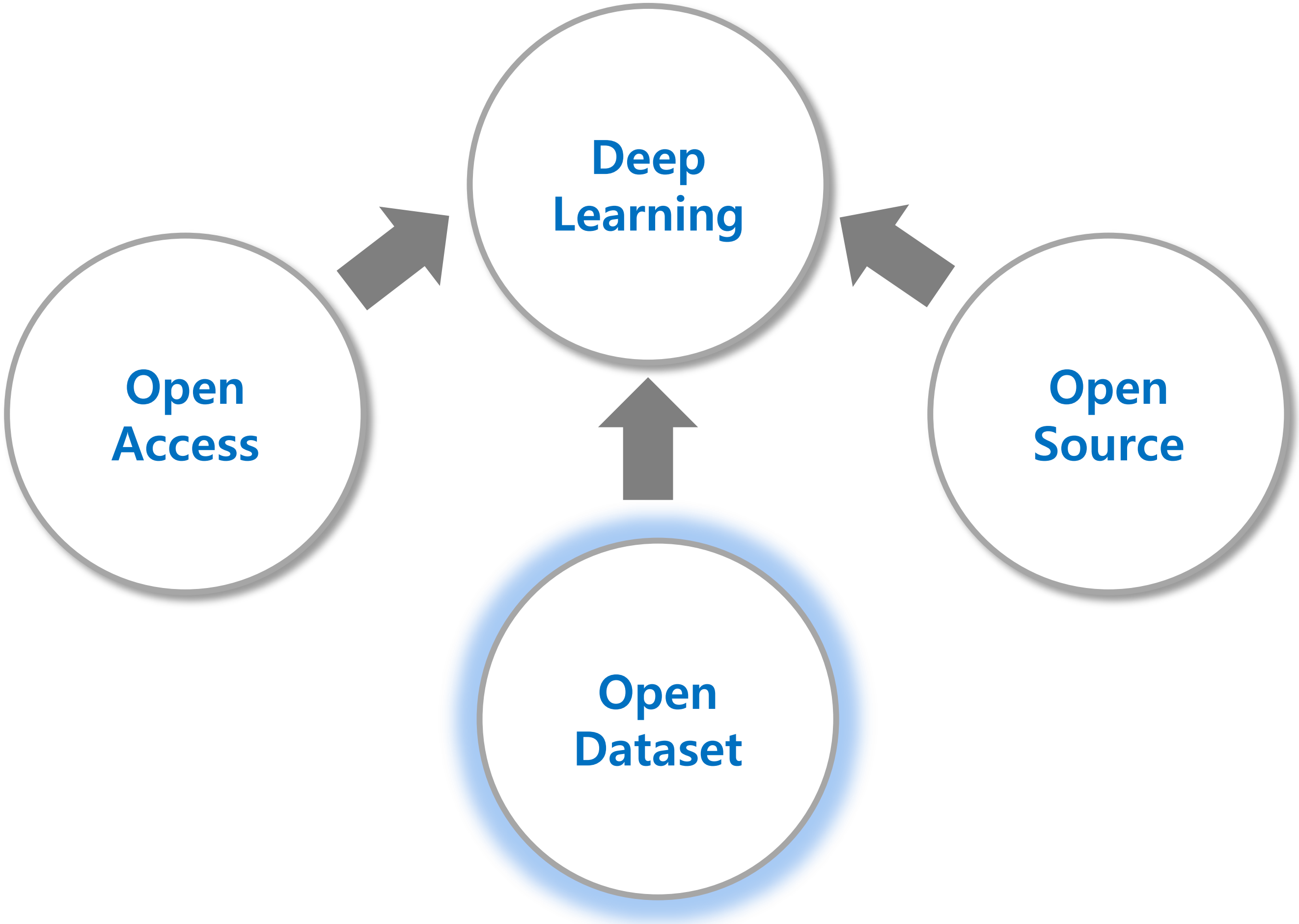
⋮

[1] https://github.com/aymericdamien/TopDeepLearning

4. Wrap-Up

4.1 Open Access, Source & Dataset

- Open Dataset



4.1 Open Access, Source & Dataset

- Open Dataset
  - ImageNet
  - Microsoft COCO Dataset
  - Google Open Images Dataset
  - Youtube 8M Dataset
  - Facebook Question Answering Dataset
  - And so on…

4.2 Become a Deep Learning Hacker

- You can become a Deep Learning Hacker!
  - Open Access
  - Open Source
  - Open Dataset
  - And Open Courseware (MOOC)

    . Coursera

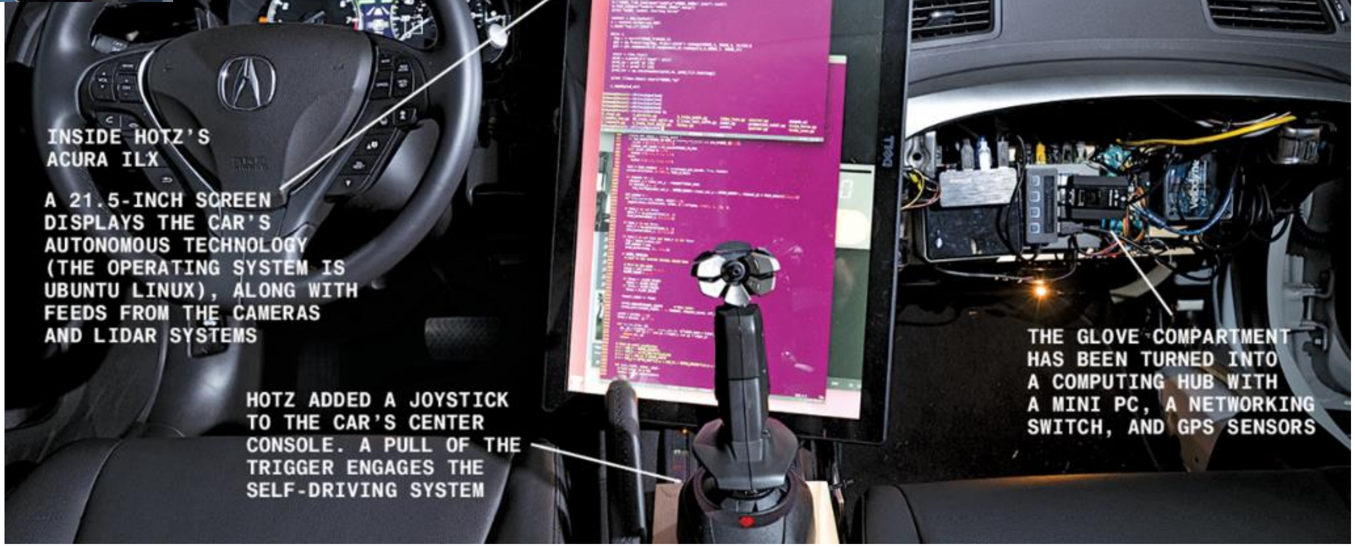
    . Udacity

    . edx

4.2 Become a Deep Learning Hacker

- George Hotz (comma.ai CEO)

- Hacked the iOS device, Playstation3 …
- Developed Deep Learning based self-driving car



INSIDE HOTZ'S ACURA ILX

A 21.5-INCH SCREEN DISPLAYS THE CAR'S AUTONOMOUS TECHNOLOGY (THE OPERATING SYSTEM IS UBUNTU LINUX), ALONG WITH FEEDS FROM THE CAMERAS AND LIDAR SYSTEMS

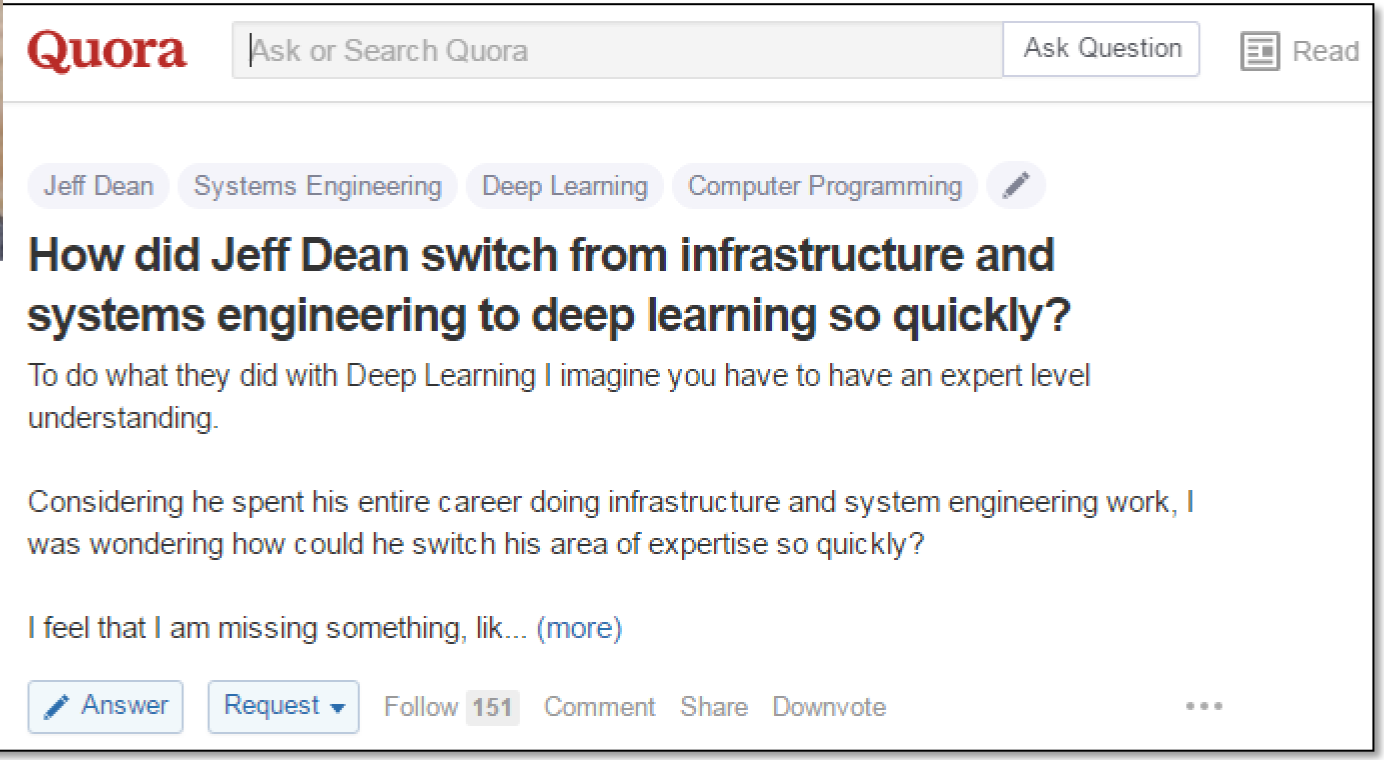
HOTZ ADDED A JOYSTICK TO THE CAR'S CENTER CONSOLE. A PULL OF THE TRIGGER ENGAGES THE SELF-DRIVING SYSTEM

THE GLOVE COMPARTMENT HAS BEEN TURNED INTO A COMPUTING HUB WITH A MINI PC, A NETWORKING SWITCH, AND GPS SENSORS

4.2 Become a Deep Learning Hacker

- Jeff Dean (Google Senior Fellow)

  - GFS(Google File System), MapReduce, BigTable …
  - TensorFlow, Google Brain Team Leader



How did Jeff Dean switch from infrastructure and systems engineering to deep learning so quickly?

To do what they did with Deep Learning I imagine you have to have an expert level understanding.

Considering he spent his entire career doing infrastructure and system engineering work, I was wondering how could he switch his area of expertise so quickly?

I feel that I am missing something, lik... (more)

Thank you.