SHARCNET™

SUMMER SCHOOL 2020

MACHINE LEARNING DAY 2

DEEP LEARNING

Session I: Introduction to DL

Isaac Ye, HPTC @ York University

Isaac@sharcnet.ca

Schedule

Session	Time	Topic	Hands-on
1	9:30 - 10:30	Introduction to Deep Learning	Lab 1
2	10:45 - 12:30	Linear regression problem	Lab 2A, 2B
3	14:00 - 15:30	Multi-layer perceptron	Lab 3A, 3B, 3C
4	15:45 - 17:00	Convolutional neural network	Lab 4A, 4B

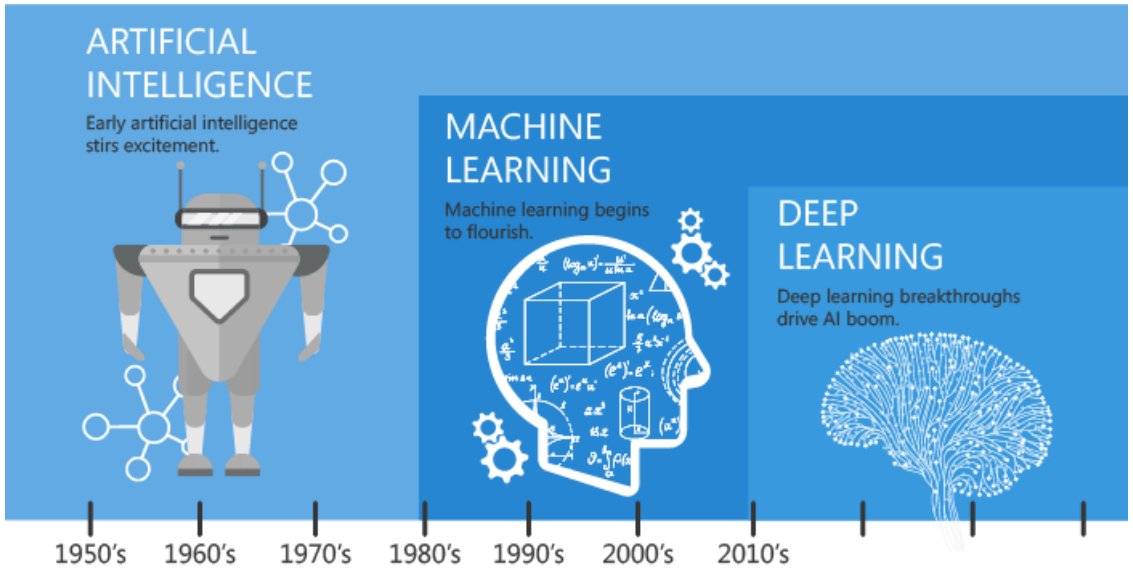
Session I

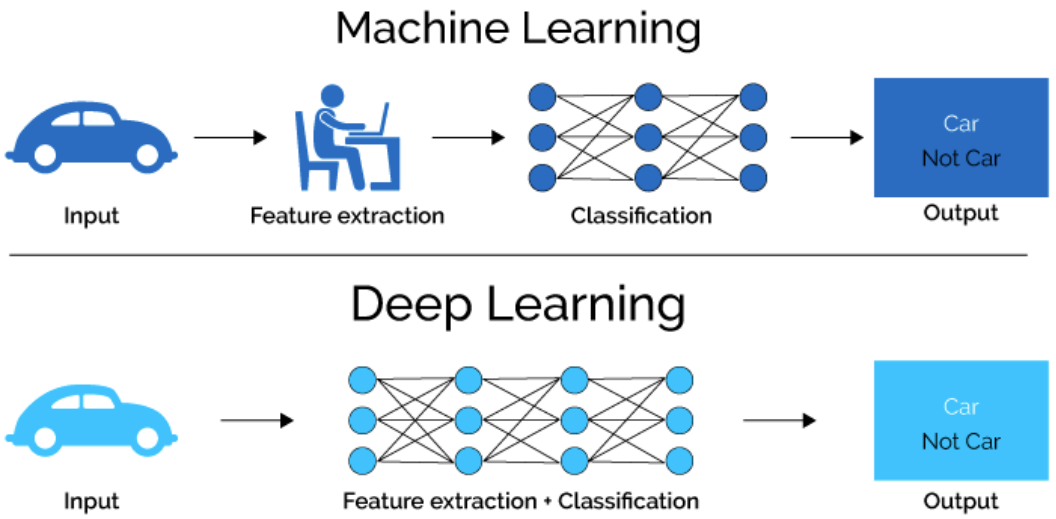
- What is AI/Deep Learning (DL)
- DL applications
- DL trends
- DL frameworks
- *Lab 1: playing around working environment in Google Colab*

  ** code comparison*

AI / Deep Learning

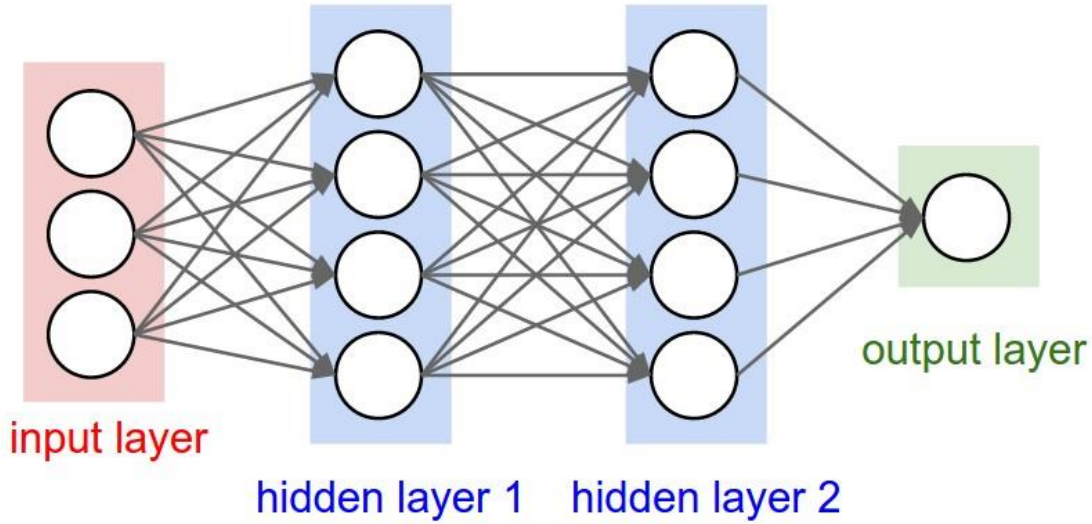
Deep learning (DL) is a class of machine learning algorithms in which multiple layers of nonlinear processing units are used for feature extraction and transformation, with each successive layer taking the output from the previous layer as input.





Deep Neural Network (DNN)

“A family of parametric, non-linear and hierarchical representation learning functions, which are massively optimized with stochastic gradient descent to encode domain knowledge, i.e. domain invariances, stationarity.” -- Efstratios Gavves

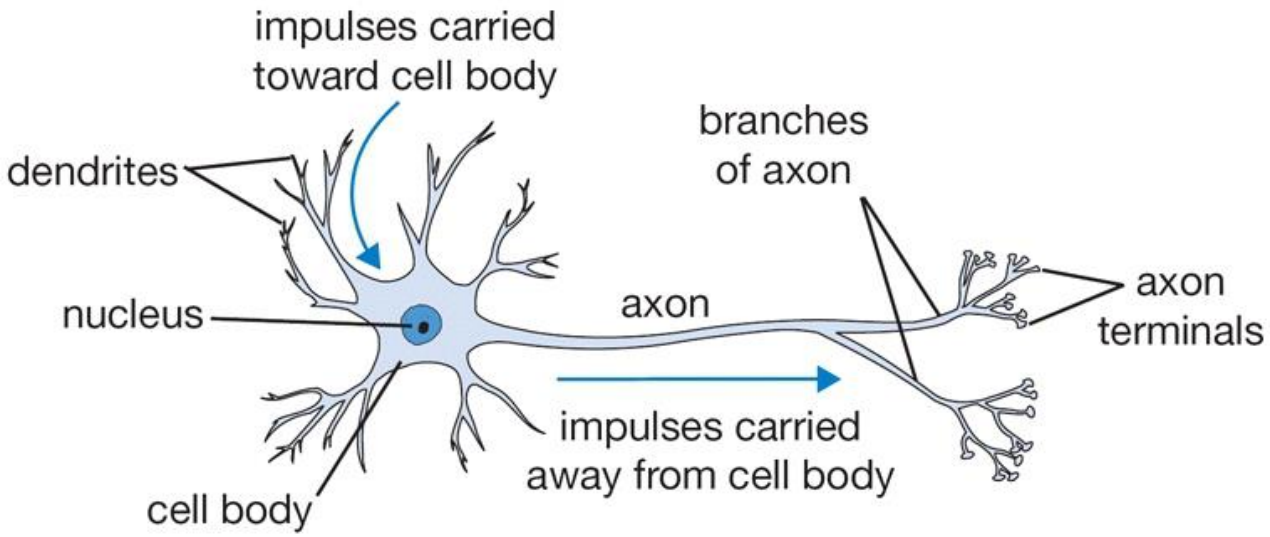


Example of a 3-layer Deep Neural Network (DNN)

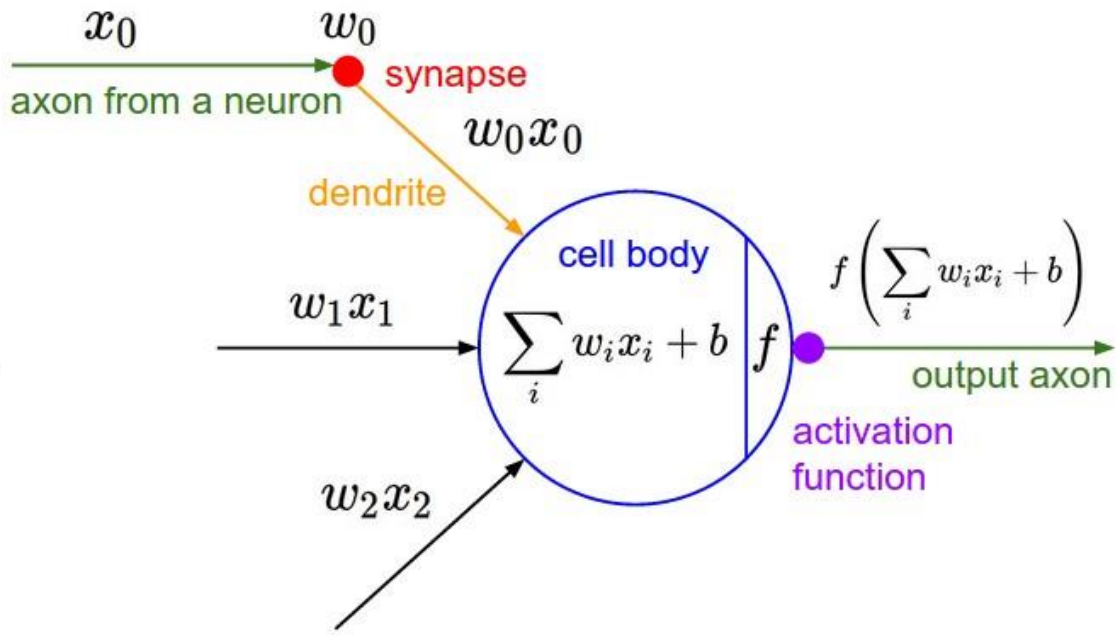
http://cs231n.github.io/neural-networks-1/

Neural Network

http://cs231n.github.io/neural-networks-1/

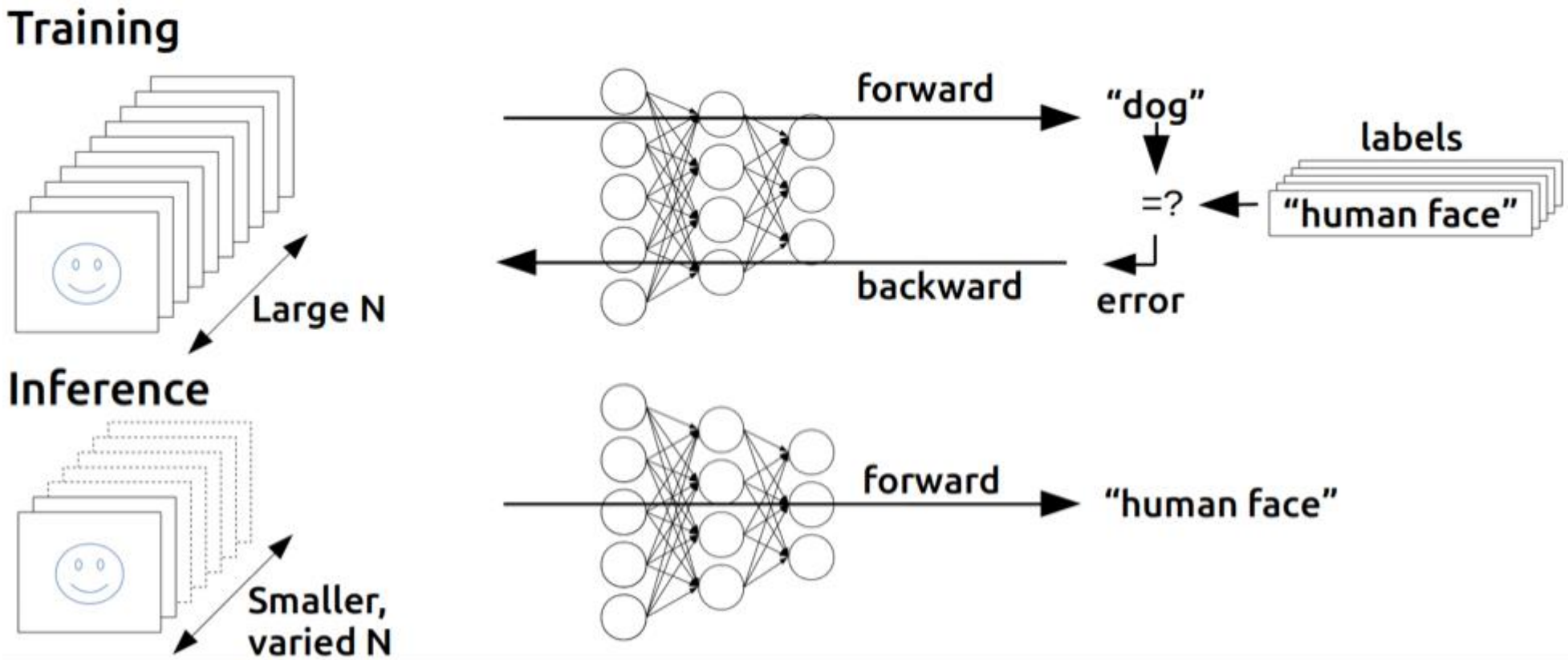


Biological neuron



Mathematical model

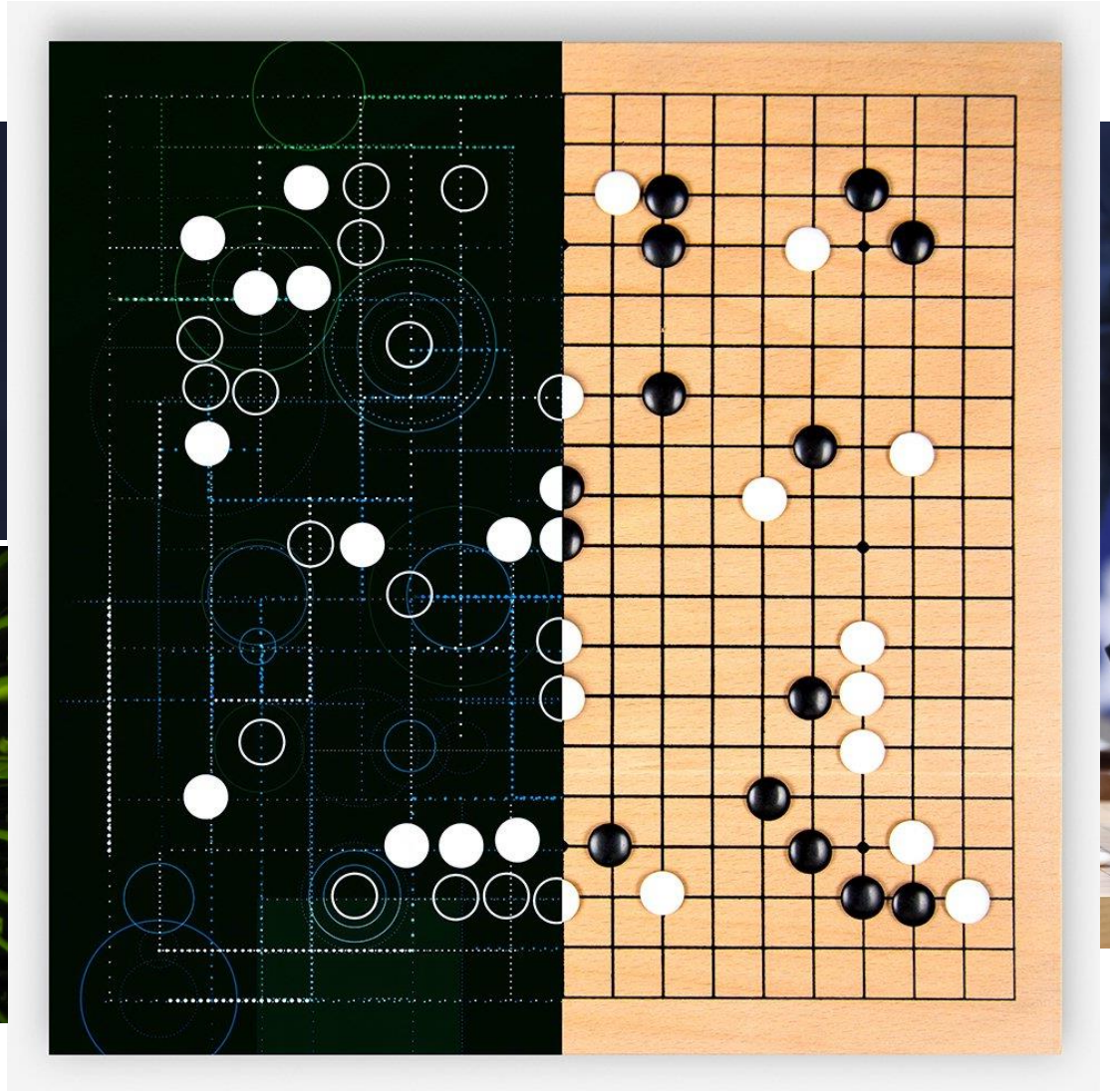
DL: Training / Inference



https://devblogs.nvidia.com/inference-next-step-gpu-accelerated-deep-learning/

DL is powering many real-world problems

AlphaGo



https://deepmind.com/research/case-studies/alphago-the-story-so-far

DL is powering many real-world problems

Image classification

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

https://www.image-net.org

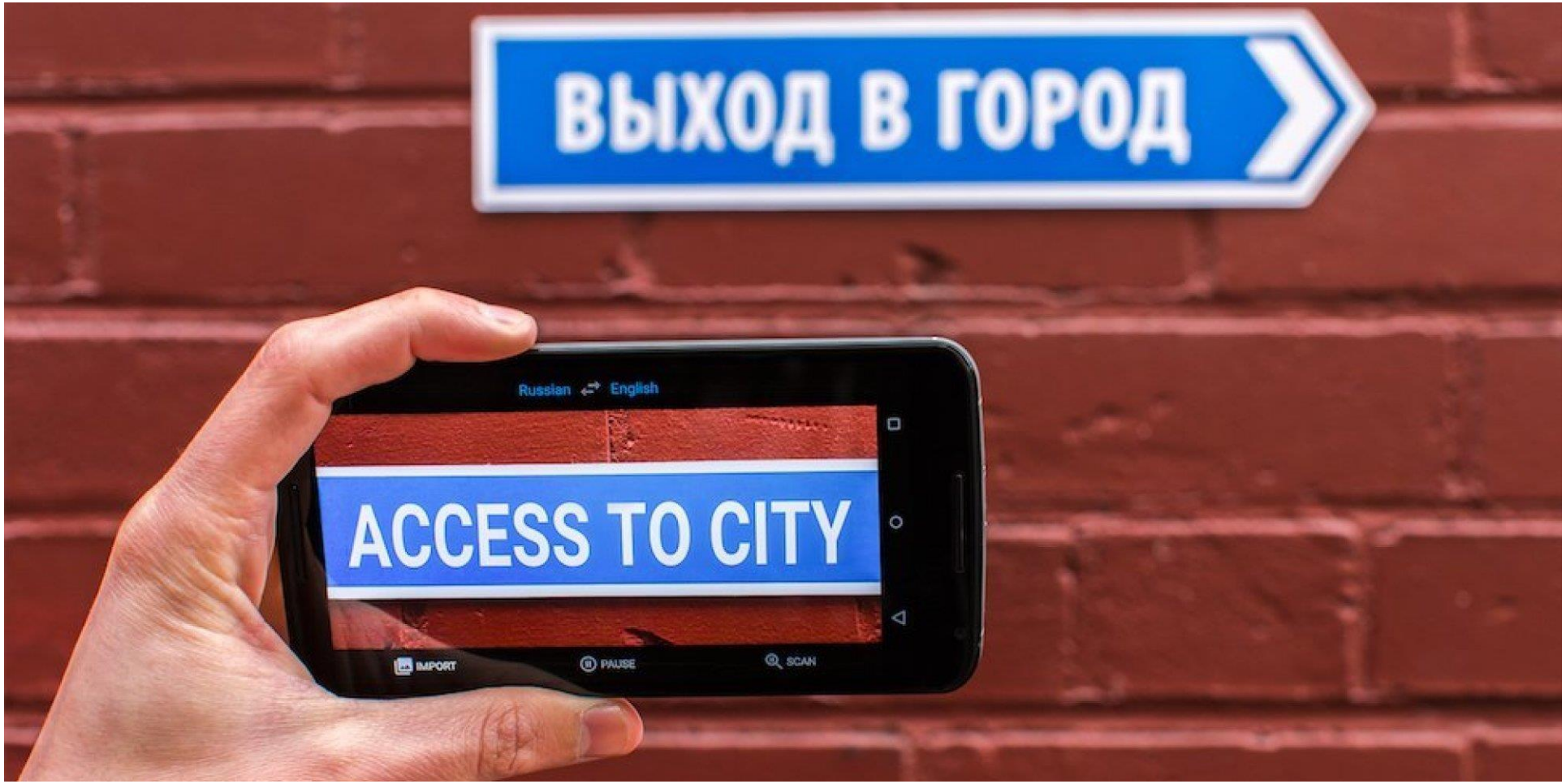
DL is powering many real-world problems

Object detection

https://www.teslarati.com/teslas-full-self-driving-capability-arrive-3-months-definitely-6-months-says-musk/

DL is powering many real-world problems

Language Translation



http://didarc.com/en/news/deep-learning-mechanism

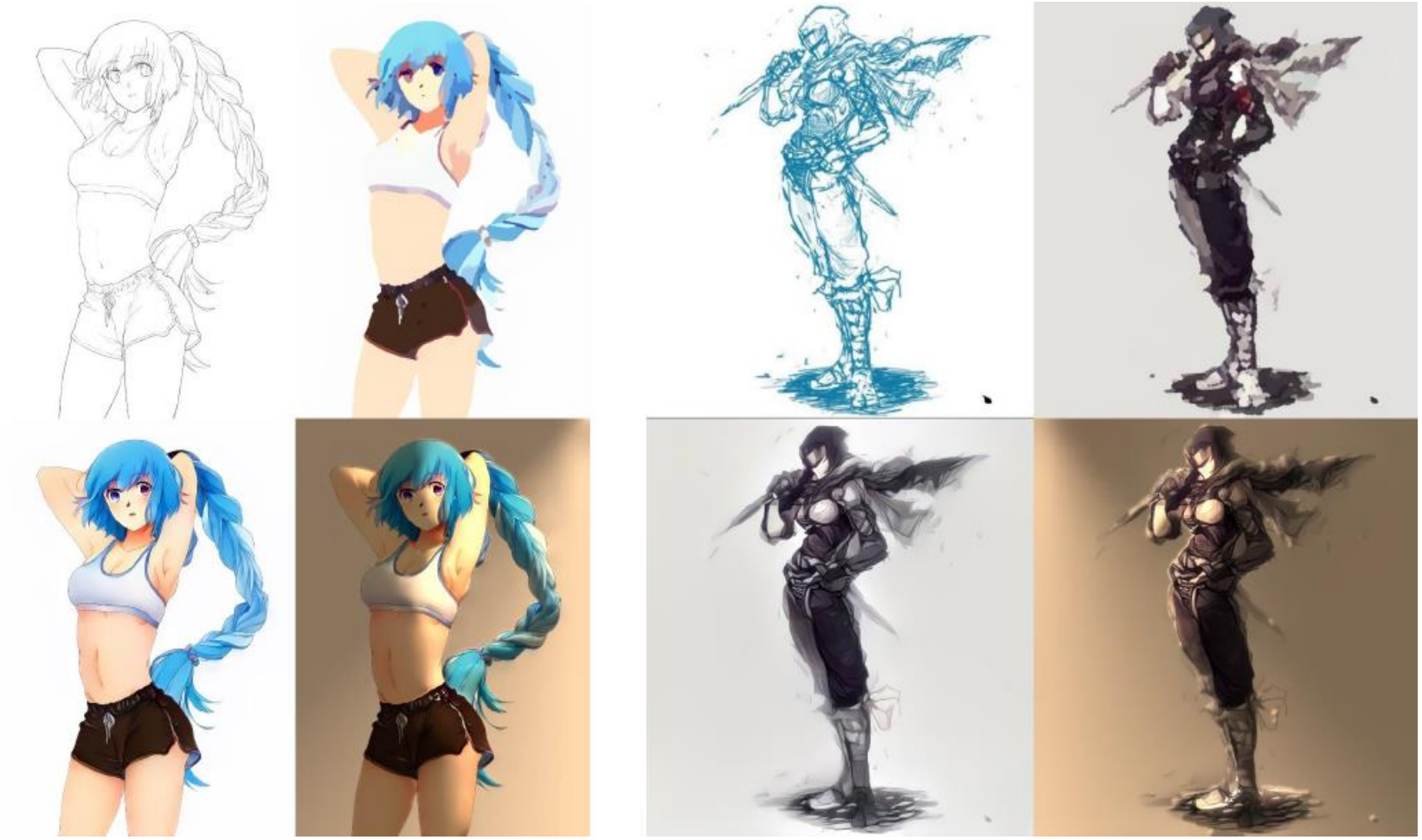
DL is powering many real-world problems

Image generation (Style transfer)

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

DL is powering many real-world problems

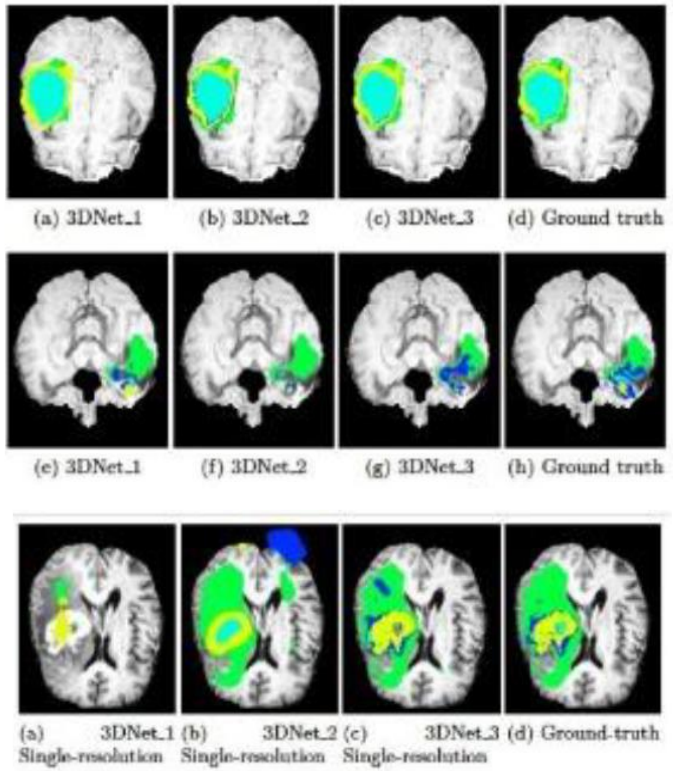
Image generation (Style2Paints)



https://github.com/lllyasviel/style2paints

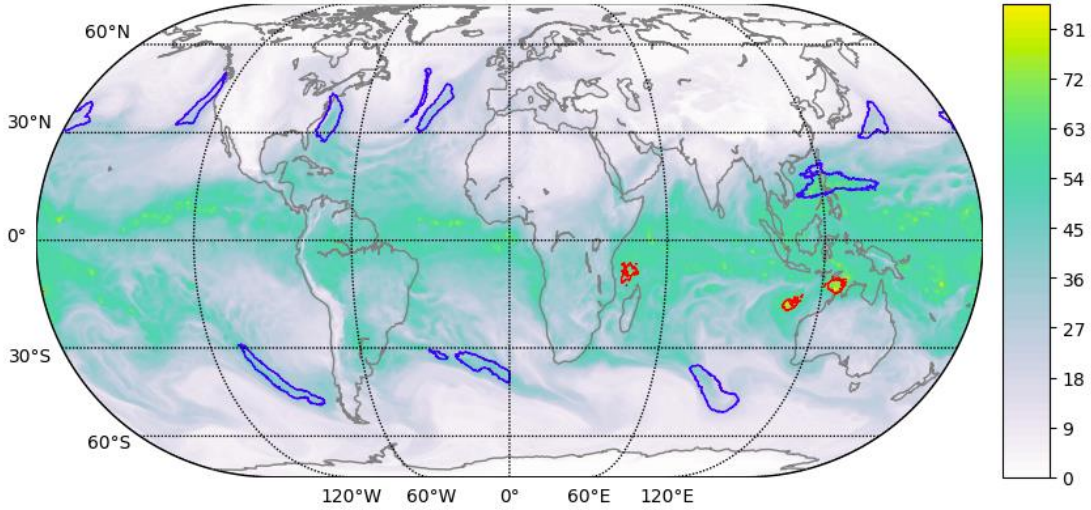
DL is powering many real-world problems

Segmentation



Tumor segmentation

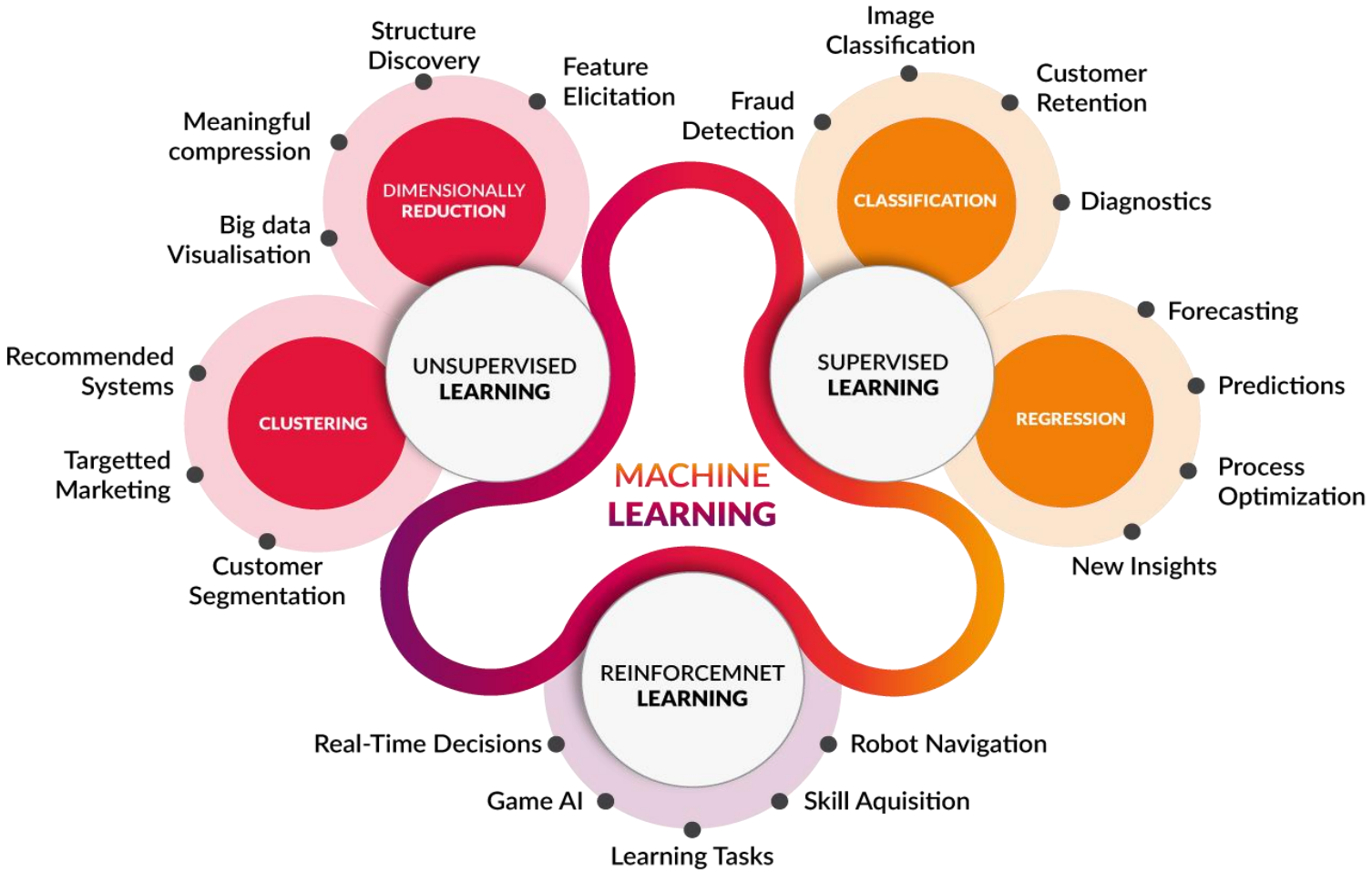
https://www.slideshare.net/xavigiro/medical-imaging-d3l3-2017-upc-deep-learning-for-computer-vision



Climate segmentation

Exascale Deep Learning for Climate Analytics, Thorsten K. https://arxiv.org/abs/1810.01993

What can we do with AI?



http://www.cognub.com/index.php/cognitive-platform/

Landscape of Science problems

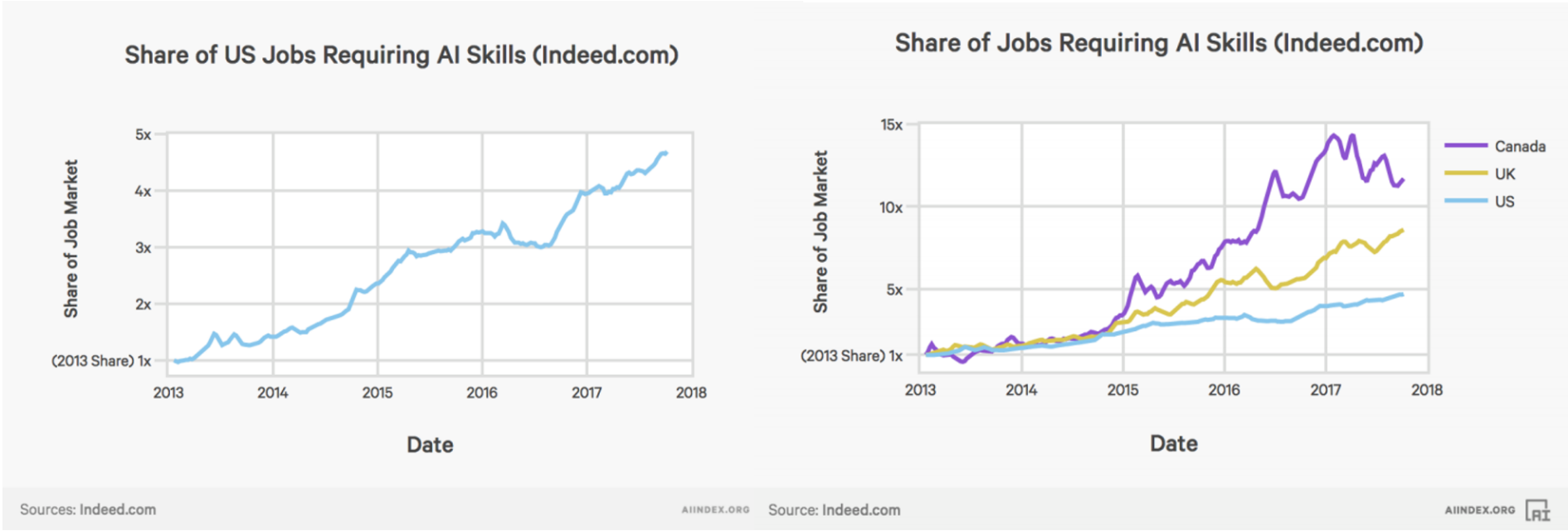
	High Energy Physics			Biological and Environment		Basic Energy		Nuclear Physics	Fusion Energy
	Astronomy	Cosmology	Particle Physics	Climate	Genomics	Light Sources	Materials	Heavy Ion Colliders	Plasma Physics
Classification	X		X	X	X	X	X	X	X
Regression	X	X	X	X	X	X	X	X	X
Clustering	X	X	X	X	X	X	X	X	X
Dimensionality Reduction				X				X	
Surrogate Models	X	X	X	X			X	X	X
Design of Experiments		X		X		X	X		X

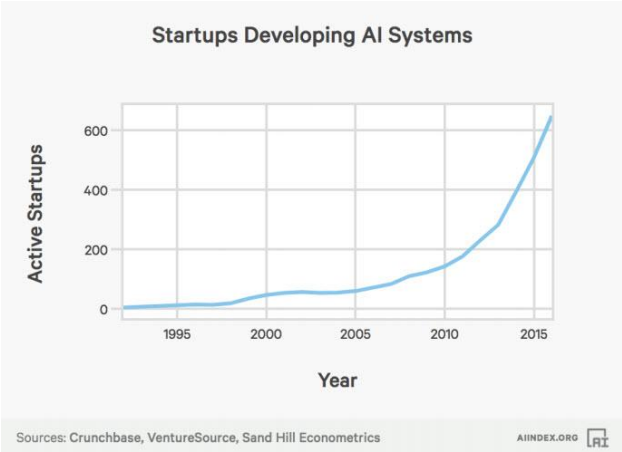
Analytics

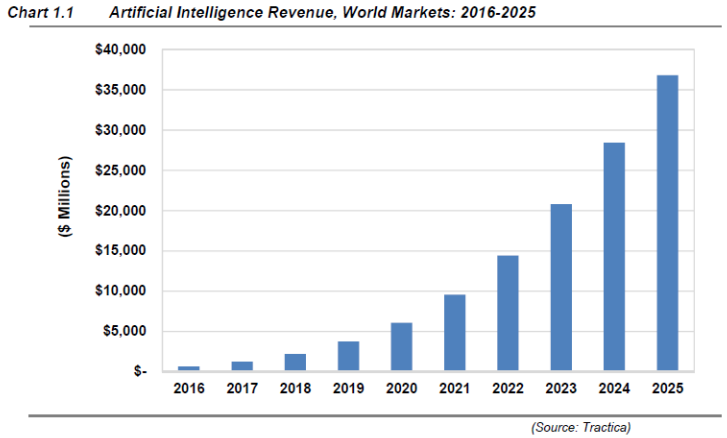
Simulations

Control

DL use cases and growth trends

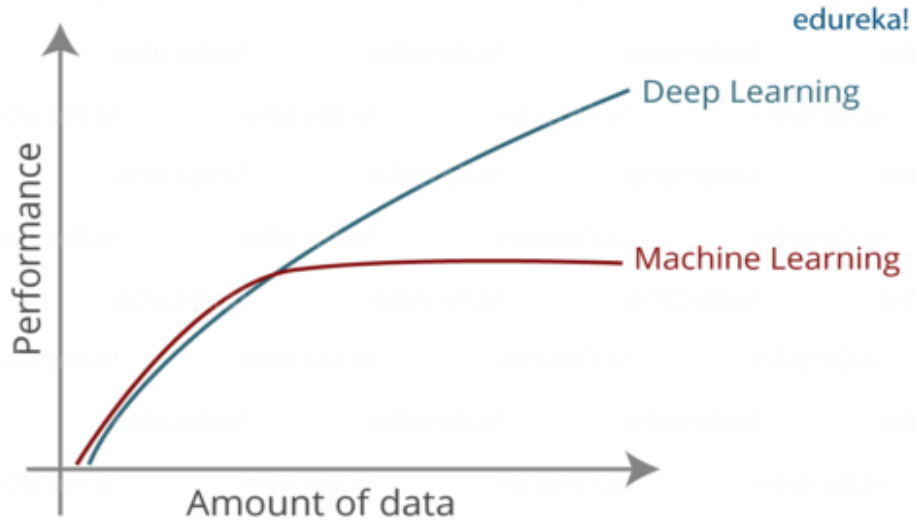




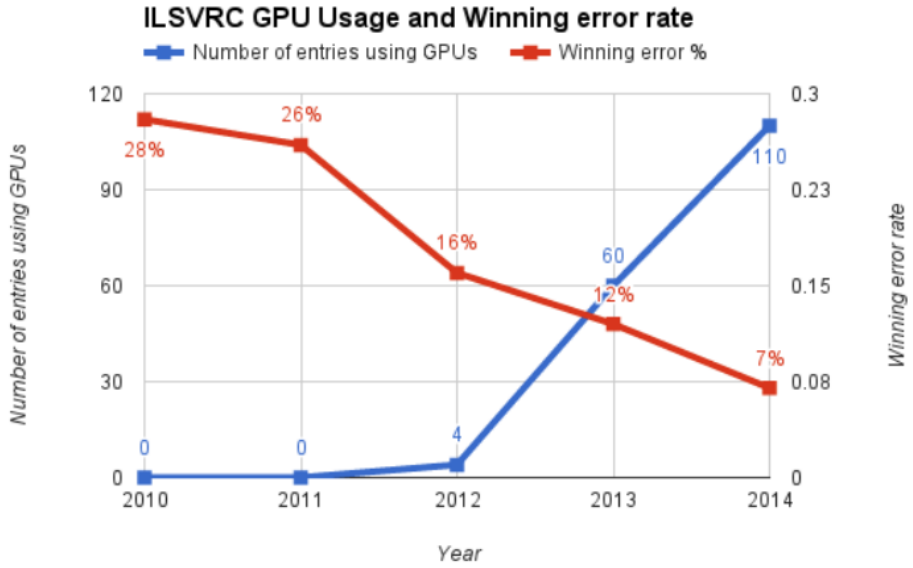


Why now?

1) Data: large curated datasets



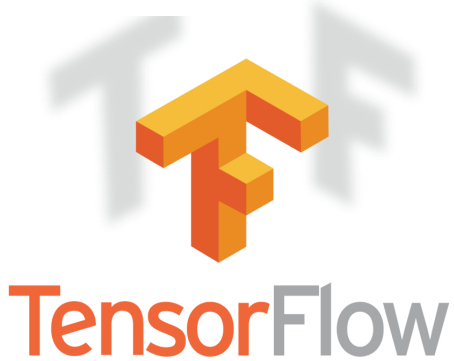
2) GPUs: linear algebra accelerators

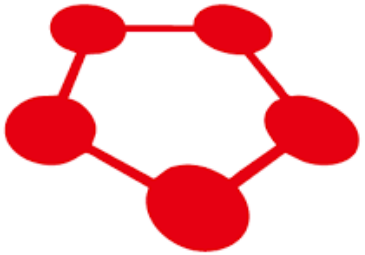


https://devblogs.nvidia.com/nvidia-ibm-cloud-support-imagenet-large-scale-visual-recognition-challenge/

3) Algorithmic advances: optimizers, regularization, normalization … etc.

DL frameworks

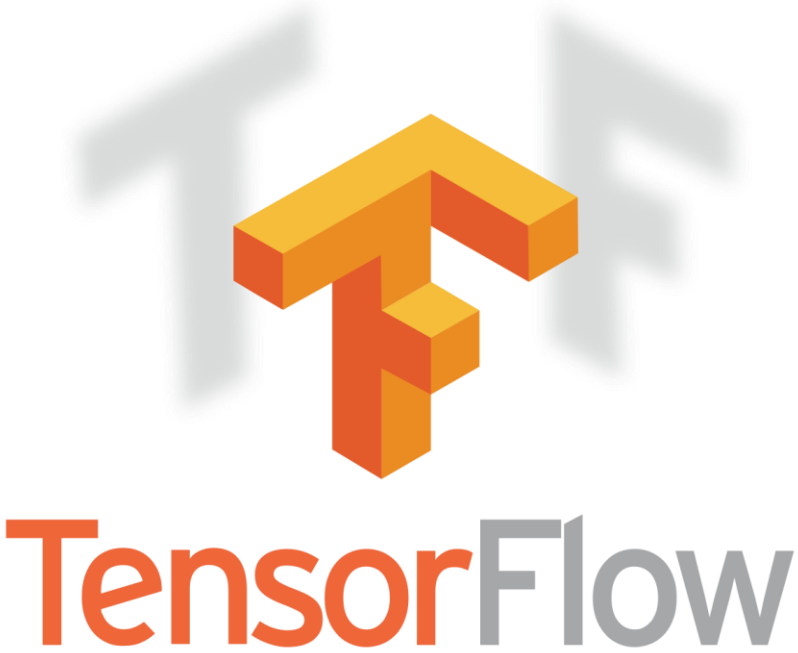




TensorFlow

- The most widely used framework open-sourced by Google
- Replaced Google's DistBelief framework
- Runs on almost all architectures (CPU/GPU/TPU/etc)
- Define-and-Run type for neural networks
- Version 2.0 has Define-by-Run component(Eager execution)
- https://github.com/tensorflow/tensorflow

```
[isaac@cedar1 ~]$ avail_wheels "tensorflow*"
name                    version    build    python    arch
----------------------  ---------  -------  --------  -------
tensorflow_cpu          2.1.0              cp37      generic
tensorflow_cpu          2.1.0              cp36      generic
tensorflow_cpu          2.1.0              cp35      generic
tensorflow_estimator    2.1.0              py2.py3   generic
tensorflow_gpu          2.1.0              cp37      generic
tensorflow_gpu          2.1.0              cp36      generic
tensorflow_gpu          2.1.0              cp35      generic
tensorflow_tensorboard  1.5.1              py3       generic
tensorflow_tensorboard  1.5.1              py2       generic
[isaac@cedar1 ~]$
```



PyTorch

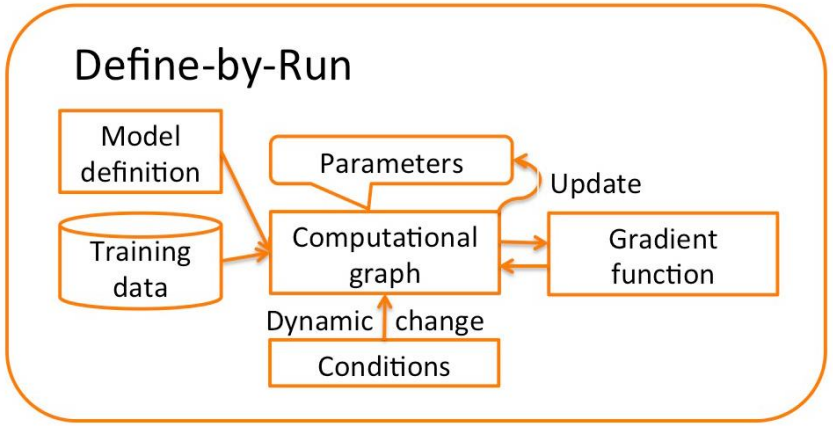
- Rapidly growing in research community for deep learning framework developed by Facebook
- A Python adaptation of Torch
- Caffe2 has been merged to PyTorch
- Define-by-Run type for neural networks
- Ease of expression and use
- https://github.com/pytorch/pytorch

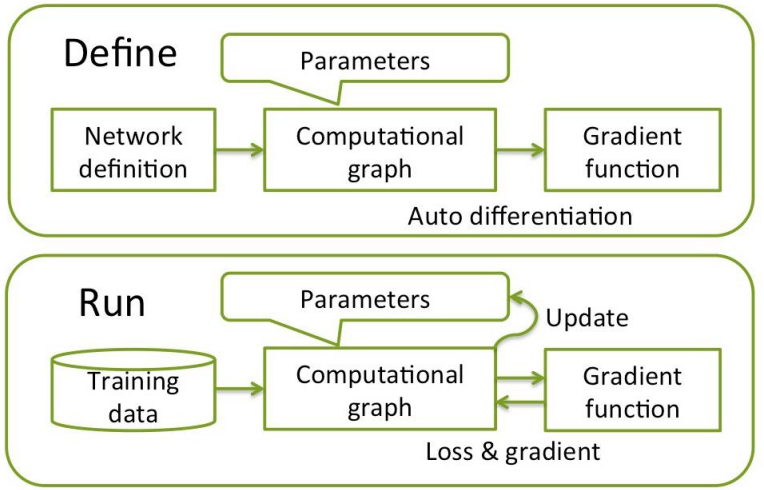

```
[isaac@cedar1 ~]$ avail_wheels "torch_*"
name               version    build    python    arch
-----------------  ---------  -------  --------  -------
torch_cluster      1.4.5                cp37      generic
torch_cluster      1.4.5                cp36      generic
torch_cluster      1.4.5                cp35      generic
torch_cpu          1.0.0                cp37      avx2
torch_cpu          1.0.0                cp36      avx2
torch_cpu          1.0.0                cp35      avx2
torch_cpu          1.0.0                cp27      avx2
torch_geometric    1.4.2                py3       generic
torch_gpu          1.0.0                cp37      avx2
torch_gpu          1.0.0                cp36      avx2
torch_gpu          1.0.0                cp35      avx2
torch_gpu          1.0.0                cp27      avx2
```

PyTorch

TensorFlow

Pros	Easy to use (Python support) Intuitive Dynamic graphs Research community prefers	Large community Heterogeneous architecture TF 2.0: Eager execution(Define-by-Run) Tensorboard (visualizing), Keras
Cons	Small community Less additional tools	Verbose Static graphs



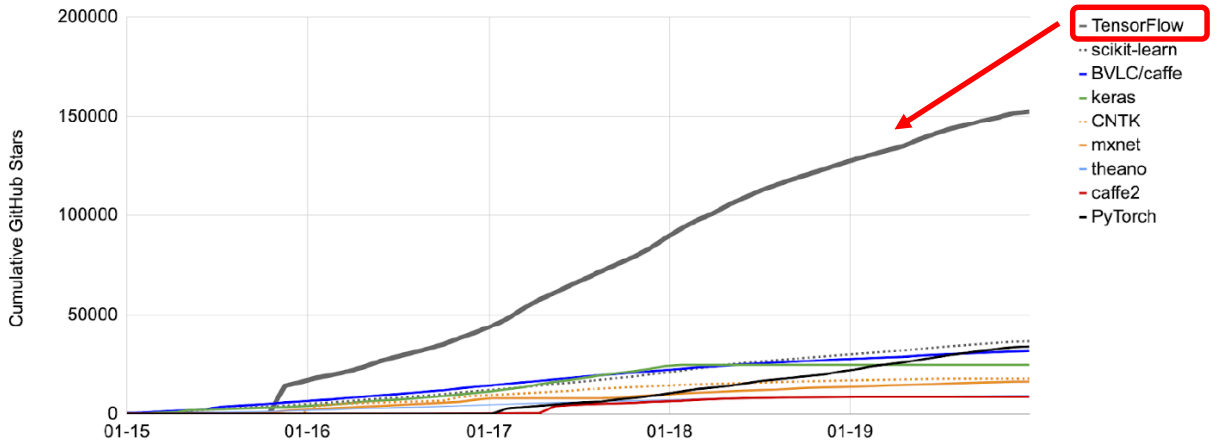


https://www.oreilly.com/content/complex-neural-networks-made-easy-by-chainer/

DL frameworks trend

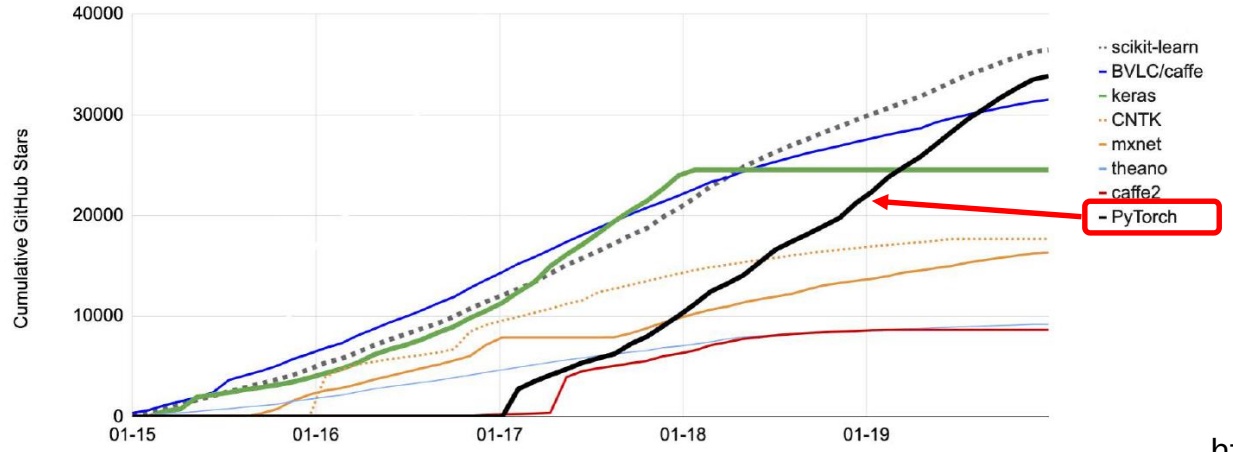
Cumulative GitHub stars by AI library (2015—2019)

Source: Github, 2019.



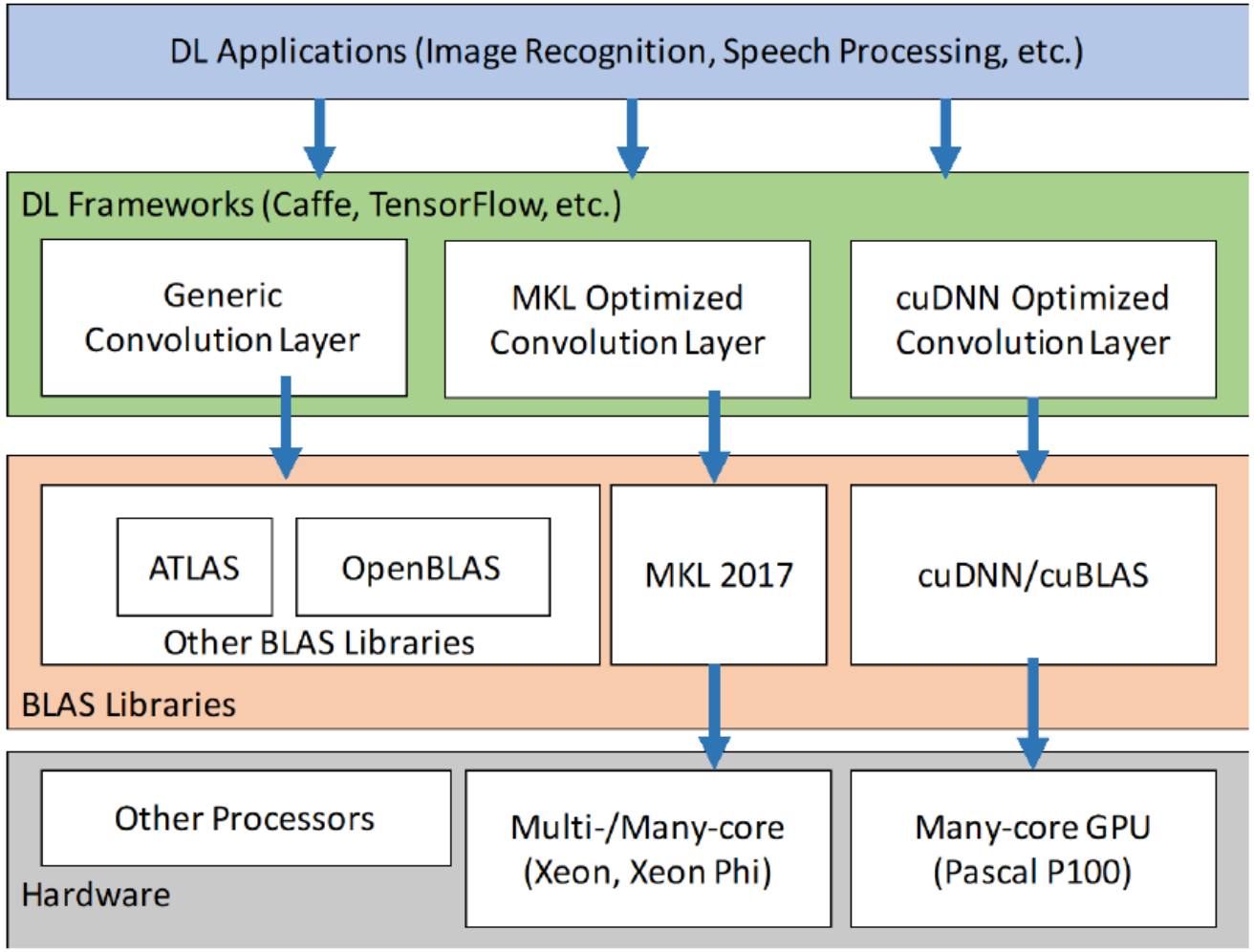
Cumulative GitHub stars by AI library, not including TensorFlow (2015—2019)

Source: Github, 2019.



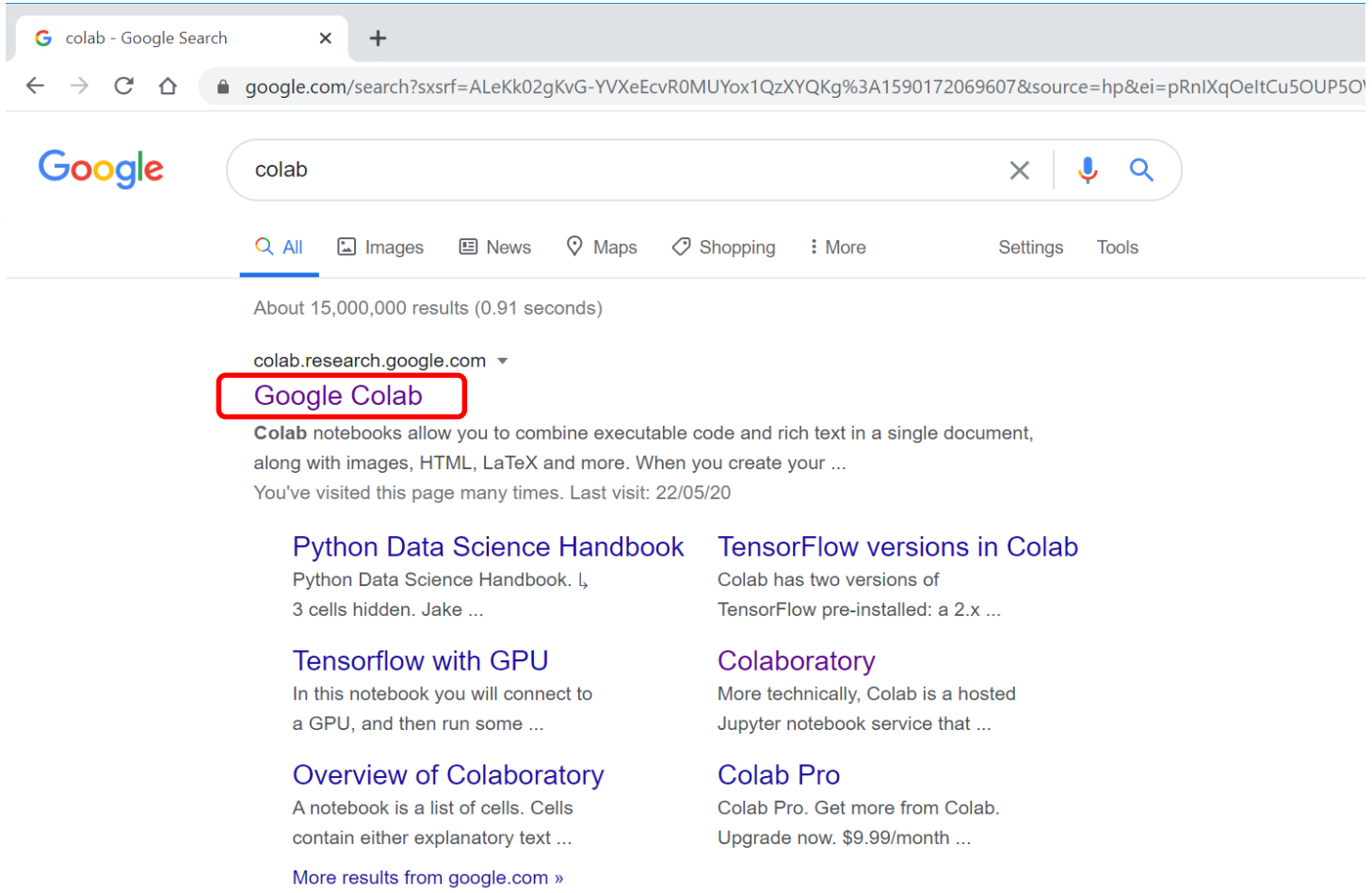
https://aiindex.org

DL framework outline



A. A. Awan, H. Subramoni, and Dhabaleswar K. Panda. "An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures", In Proceedings of the Machine Learning on HPC Environments (MLHPC'17). ACM, New York, NY, USA, Article 8.

Lab 1: Working environment



We will use Google Colab for most of simple runs!

Lab 1: playing around Google Colab

1. Go to https://colab.research.google.com
2. Open a new Jupyter notebook
3. Check Runtime type (GPU/TPU) and settings
4. Editor (Code/text block)
5. Be careful of running order
6. Make sure where you can find your code

Lab 1: Code comparison(calculating gradient)

NumPy

```python
import numpy as np
np.random.seed(0)

N, D = 3, 4

x = np.ones((N,D))
y = 2*np.ones((N,D))
z = 3*np.ones((N,D))

print(x)
print(y)
print(z)

a = x * y
b = a + z
c = np.sum(b)
print(c)

grad_c = 1.0
grad_b = grad_c * np.ones((N,D))
grad_a = grad_b.copy()
grad_z = grad_b.copy()

grad_x = grad_a*y
grad_y = grad_a*x

print(grad_x)
```

PyTorch

```python
import torch
print(torch.__version__)

N, D = 3, 4

x = torch.ones(N,D, requires_grad=True)
y = 2*torch.ones(N,D)
z = 3*torch.ones(N,D)

print(x)
print(y)
print(z)

a = x*y
b = a+z
c = torch.sum(b)
print(c)

c.backward()
print(x.grad)
```

TensorFlow

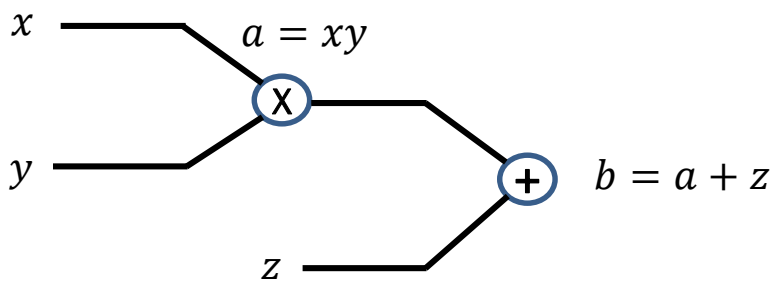
```python
import numpy as np
import tensorflow as tf
print(tf.__version__)

N, D = 3, 4

x = tf.ones([N,D])
y = 2*tf.ones([N, D])
z = 3*tf.ones([N, D])

print(x)
print(y)
print(z)

with tf.GradientTape(persistent=True) as g:
  g.watch(x)
  a = x * y
  b = a + z
  c = tf.reduce_sum(b)
grad_x = g.gradient(b,x)
print(c)
print(grad_x)
```



Break room

Session break:

Please come back by 10:45 AM