

Bài 3: Giới thiệu tổng quan về Machine Learning. Thư viện TensorFlow

Tuần 2A
22-08-2019

Giới thiệu tổng quan về Machine Learning

Nội dung chính

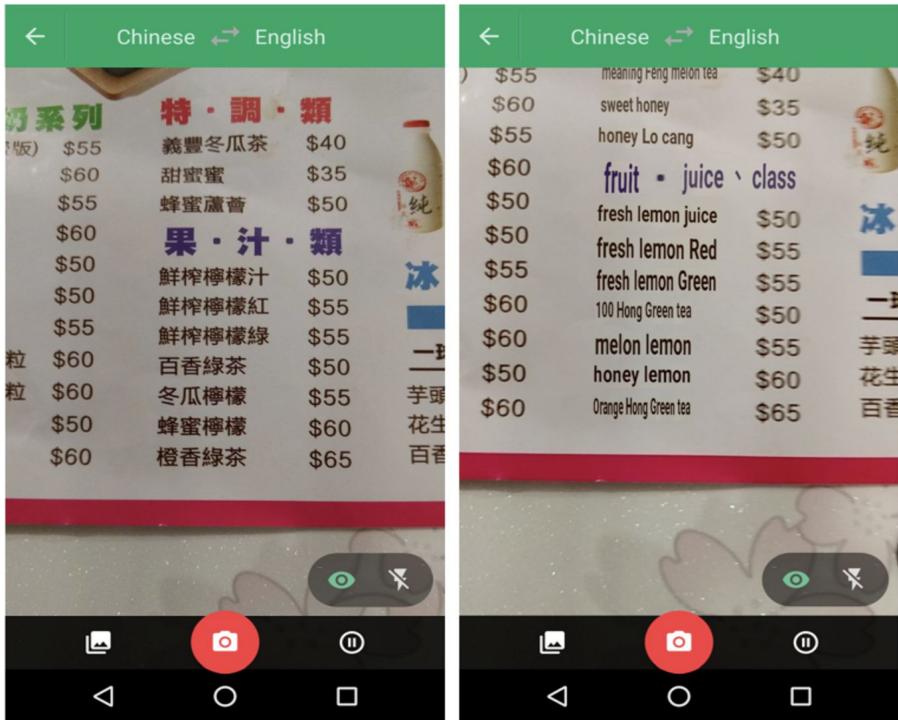
1. Giới thiệu chung & Định nghĩa
2. Các loại bài toán Machine Learning
 - a. Supervised Learning - Unsupervised Learning - Reinforcement Learning
 - b. Sơ lược các bước xây dựng mô hình / thuật toán Machine Learning
3. Một số hướng phát triển trong Machine Learning

1. Giới thiệu chung & Định nghĩa

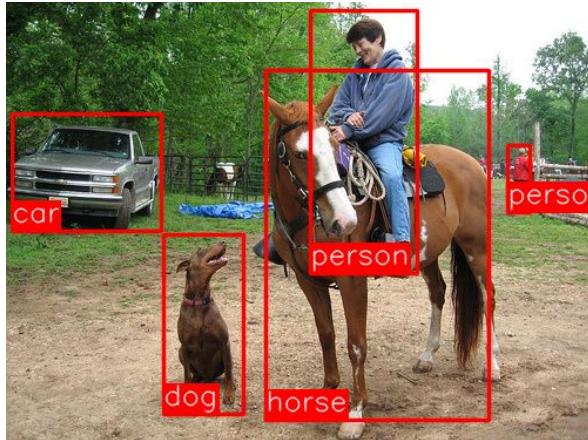
AI/ML trong cuộc sống

Google Translate

- Word detection (nhận diện chữ)
- Machine translation (dịch tự động)

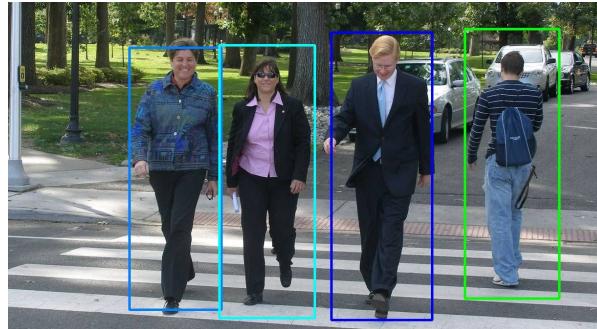


AI/ML trong cuộc sống



Object detection
(nhận diện đối tượng)

Hệ thống nhận diện / giám sát



Pedestrian detection (nhận diện người đi đường)



Face detection
(nhận diện khuôn mặt)



Face recognition / identification
(định danh khuôn mặt)

AI/ML trong cuộc sống

Hệ gợi ý - Recommender System

amazon.com

Recommended for You

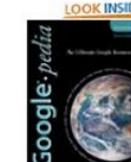
Amazon.com has new recommendations for you based on items you purchased or told us you own.



Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop



Google Apps Administrator Guide: A Private-Label Web Workspace



Googlepedia: The Ultimate Google Resource (3rd Edition)

YouTube

What to Watch My Subscriptions Music

Recommended



'Antennagate'
Not unique to iPhone 4
33:54

Steve Jobs at the Antennagate Press...
by EverySteveJobsVideo
18,990 views • 9 months ago



Last Week Tonight with John Oliver: Online Harassment...
by LastWeekTonight
2,895,113 views • 1 week ago



BASTILLE feat. Ella - No Angels (HD)
by dancenationedm
5,836,404 views • 2 years ago



Bill Maher with Charlie Rose
by DarkLordDabc
88,923 views • 8 months ago



The Attack That Could Disrupt The Whole Internet ...
by Computerphile
345,806 views • 1 year ago



START! Walking at Home American Heart Association...
by Leslie Sansone's Walk at Home
1,863,839 views • 7 months ago



The 5 Best Shia LaBeouf 'DO IT' Motivation Videos
by The VGC
1,390,971 views • 4 weeks ago



Elon Musk and Bill Gates discuss AI, entrepreneurship..
by Every Elon Musk Video
105,539 views • 2 months ago

AI/ML trong cuộc sống

Chatbot



Spell correction

Intent detection

Entity detection

Conversation Management

Third Party API

Tìm vé máy bay tới Sài Gòn vào ngày này tuần sau

Intent: find_flight

To: Sai Gon
Time: 29 - 12 - 2018

From: Ha Noi
To: Sai Gon
Time: 29 - 12 - 2018

AI/ML trong cuộc sống

Image generation: “Style transfer” (conditional image-to-image translation)



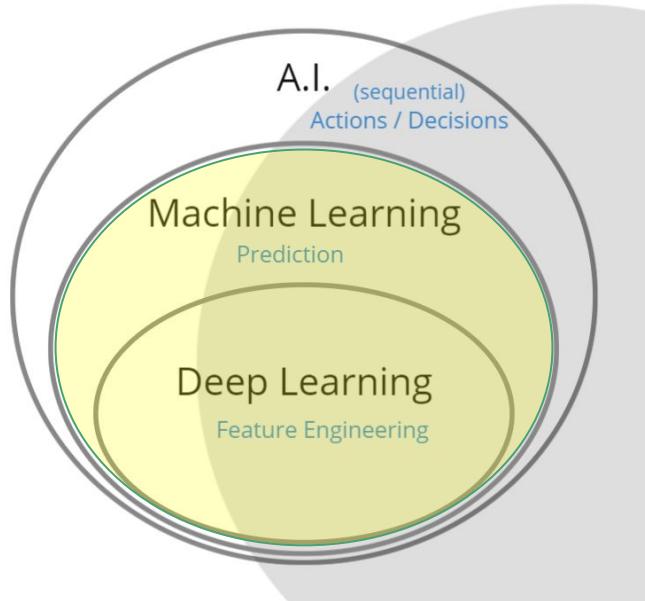
(image credits: from [Justin Johnson's style-transfer implementation](#))

AI/ML thúc đẩy sự phát triển trong các lĩnh vực liên quan

- Image Processing
 - Image Understanding i.e. Computer Vision: Image Classification, Object Detection, Face Recognition, Image Segmentation
 - Image Generation i.e. Computer Graphics: (conditional) Image Synthesis
- Language Processing
 - Language Understanding e.g. Dependency Parsing, Named Entity Recognition, Text Classification, Sentiment Analysis
 - Language Generation e.g. Machine Translation, Text Summarization, Image Captioning
- Speech Processing
 - Speech Recognition
 - Speech Synthesis (e.g. Google Duplex)
- Recommender System, Gaming

Machine Learning - Nhắc lại

(từ lecture trước ...)



A.I. is about *automation*

ML is (mostly) about *automated prediction with high accuracy* from data*

DL is (mostly) about *automated feature engineer from data*

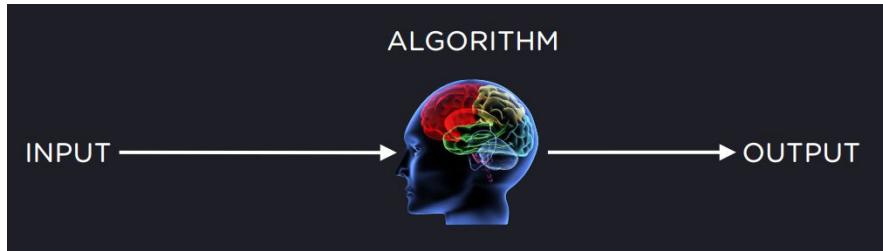
Machine Learning - Định nghĩa

“Field of study that gives computers the ability to learn
without being explicitly programmed”

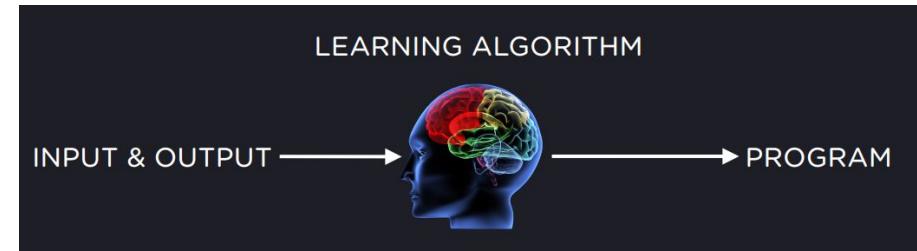
(“... tạo cho máy tính khả năng tự học hỏi mà **không cần phải được lập trình / chỉ dẫn cụ thể.**”)

- Arthur Samuel (1959)

Traditional programming



Machine Learning

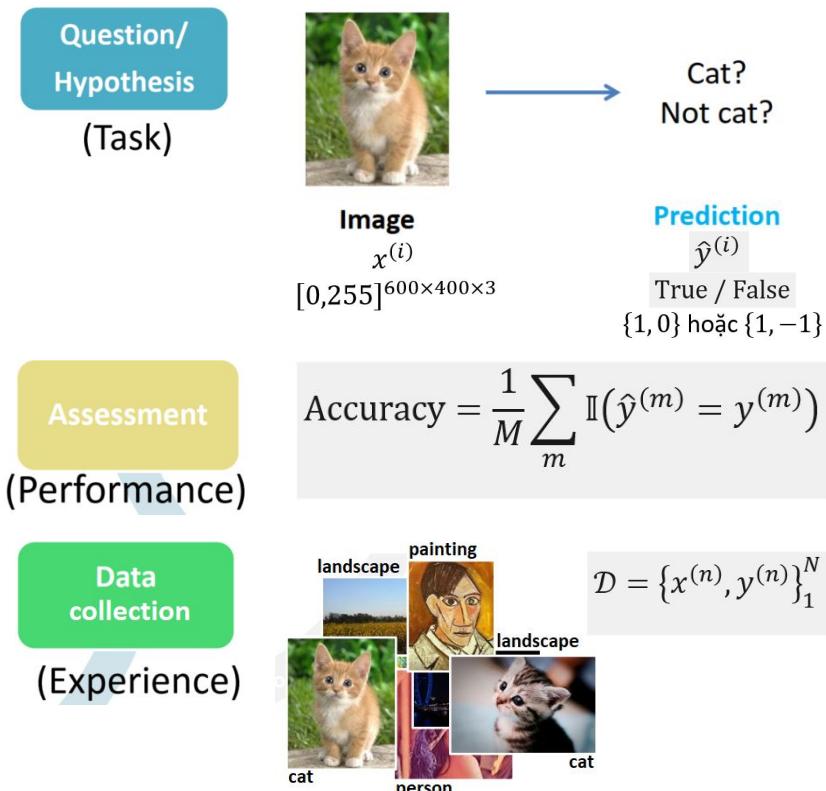


Machine Learning - Định nghĩa

“A computer program is said to learn from **experience E** with respect to **some task T** and some **performance measure P**, if its performance on T, as measured by P, improves with experience E.”

- Tom Mitchell (1998)

- T: nhiệm vụ/bài toán cần giải quyết
- E: dữ liệu cần thu thập (collect)
- P: cách đánh giá kết quả

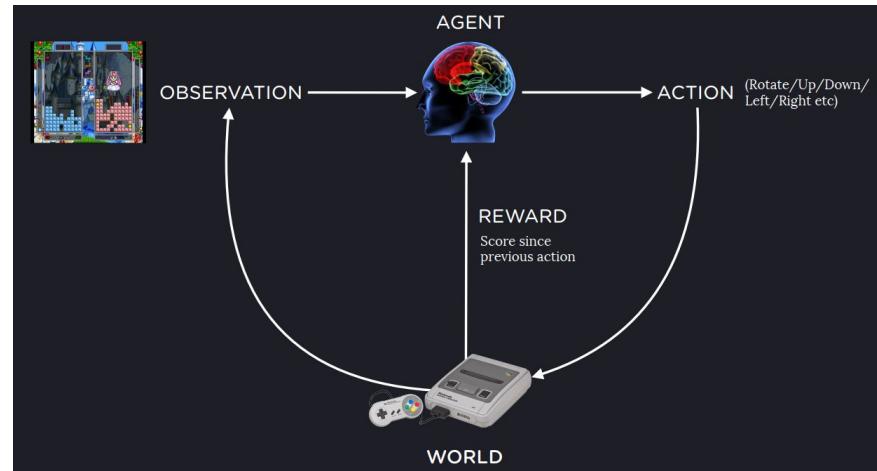


Lý do nên áp dụng các thuật toán Machine Learning

- Khó lập trình lời giải theo cách đơn thuần
 - Rule-set nào cho bài toán Nhận diện hình ảnh?
- Mong muốn hệ thống hoạt động tốt hơn lập trình viên (con người nói chung)
 - Câu chuyện AlphaGo: “Nước đi thứ 37”
- Hệ thống vừa phức tạp, vừa phải thích nghi với sự thay đổi (vd: spam detection, recommender system)

2. Các loại bài toán Machine Learning

- **Supervised Learning:** huấn luyện mô hình với dữ liệu có gán nhãn $\mathcal{D} = \{x^{(n)}, y^{(n)}\}_1^N$
 (Tuần 3-4-5)
- **Unsupervised Learning:** huấn luyện mô hình với dữ liệu *không* có nhãn $\mathcal{D} = \{x^{(n)}\}_1^N$
 (Tuần 6 – Word2Vec)
- **Reinforcement Learning:** hệ thống (agent) sẽ nhận được “phần thưởng” (reward) khi tương tác với môi trường (environment), nhiệm vụ của nó là cực đại số “phần thưởng” nhận được
 (ngoài phạm vi khóa học)



(image credit: from Max Pagels's [presentation on Bandit Algorithm at sc5.io](#))

Supervised Learning

Supervised Learning

Mô hình Supervised Learning

- Input: $x \in X$, Output: $y \in Y$
- Giả sử tồn tại map $f : X \rightarrow Y$
- Hàm giả thiết (hypothesis): $h : X \rightarrow Y$
- Data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) , \dots , (x^{(m)}, y^{(m)})$

Một **Machine Learning Model** quy định tập giả thiết (Hypothesis set) H và Thuật toán học (Learning Algorithm) A

- Chú ý 1: “**Mô hình ML**” và “**Thuật toán ML**” thường được dùng với ý nghĩa tương đương. Vì thế khi nói “thuật toán” trong AI/ML, cần nói rõ “thuật toán” đang được đề cập là “thuật toán ML” hay “thuật toán học”
- Chú ý 2: “Mô hình / Thuật toán ML” = “Tập giả thiết” + “Thuật toán học”. Khi nói “**Mô hình**” mà không nói gì thêm, ta hiểu “Mô hình” = “Tập giả thiết H ”.

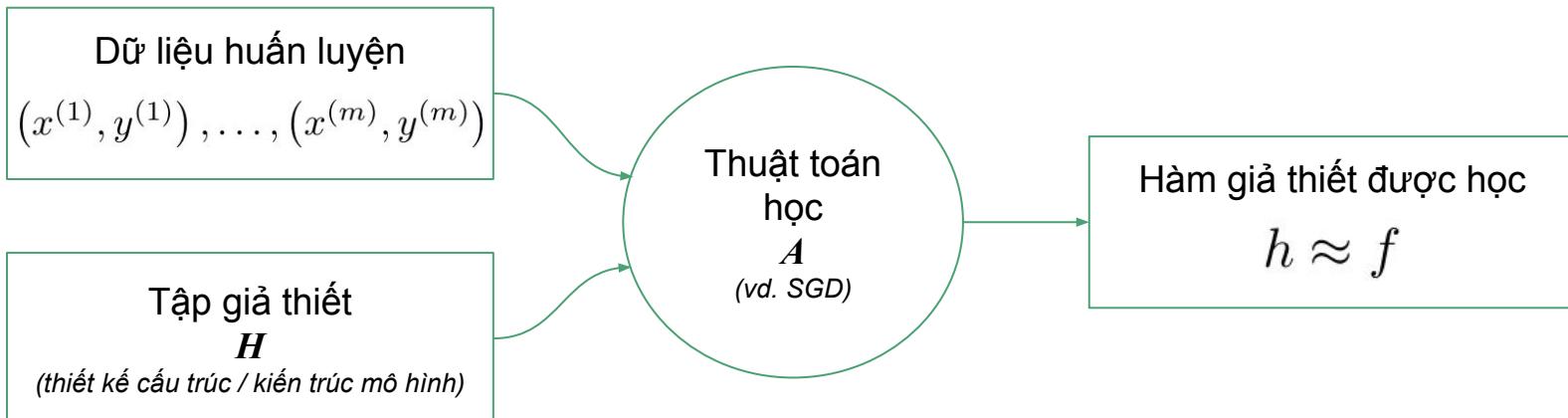
Supervised Learning

Mô hình Supervised Learning

$$f : X \rightarrow Y$$

(chưa biết, cần tìm)

- Nếu Y là liên tục, bài toán được gọi là **Regression**
- Nếu Y là rời rạc, bài toán được gọi là **Classification**



Supervised Learning

Mô hình Supervised Learning - Ví dụ

Classification: C classes

Input: Image

Output: Class label

Evaluation metric: Accuracy



Localization:

Input: Image

Output: Box in the image (x, y, w, h)

Evaluation metric: Intersection over Union



**Classification
+ Localization**



Chú ý: Một mô hình Supervised Learning có thể kết hợp cả bài toán Regression và Classification trong cùng một mô hình

Supervised Learning

Ứng dụng - Image Recognition



flamingo



cock



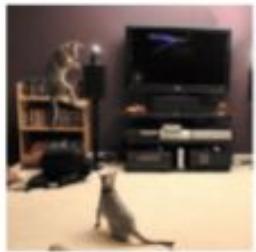
ruffed grouse



quail



partridge



Egyptian cat



Persian cat



Siamese cat



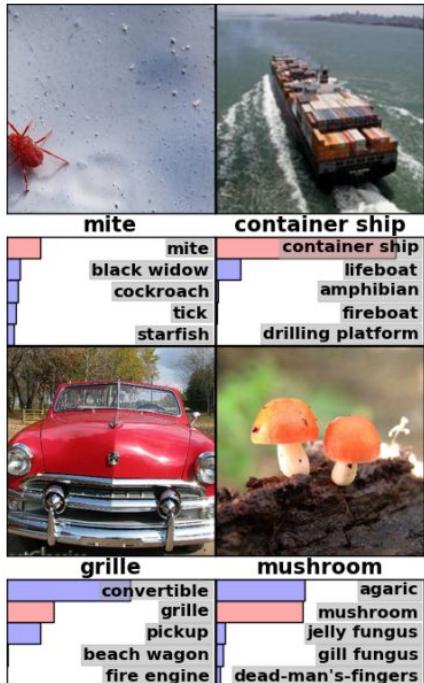
tabby



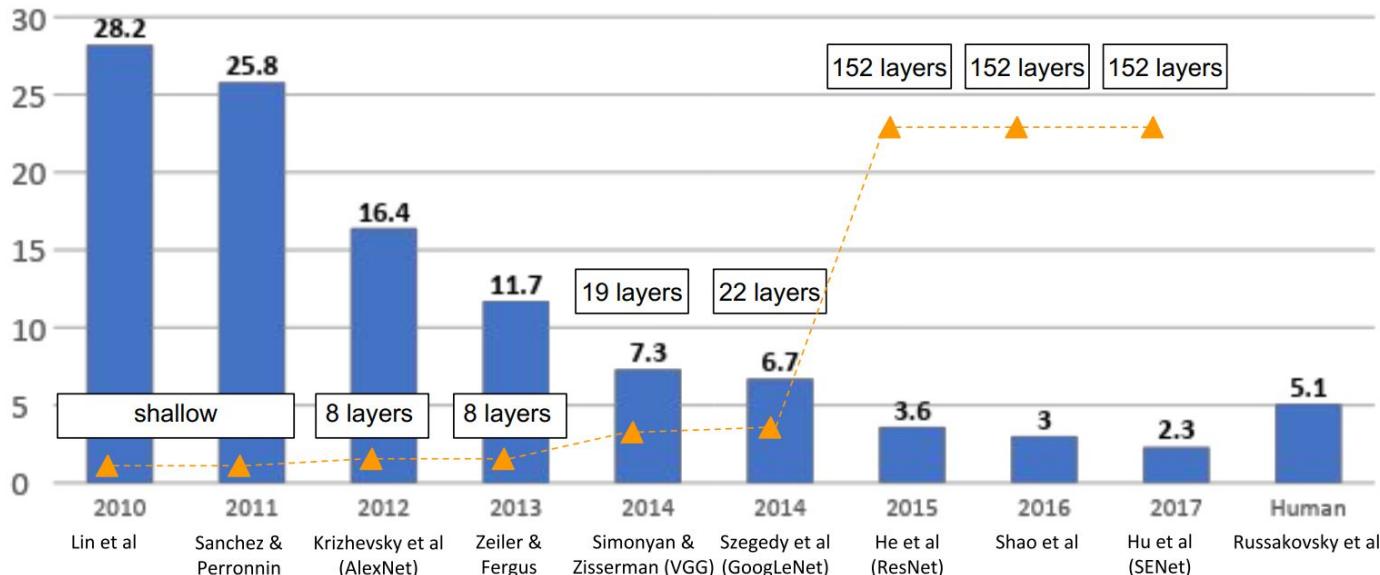
lynx

Supervised Learning

Ứng dụng - Image Recognition



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

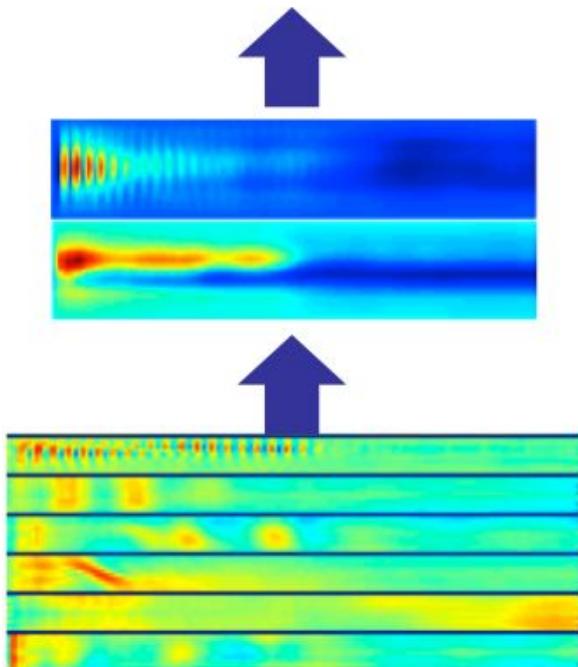


(image credits: [ImageNet Classification with Deep Convolutional Neural Networks](#) (Krizhevsky, 2012); Stanford's [CS231n](#))

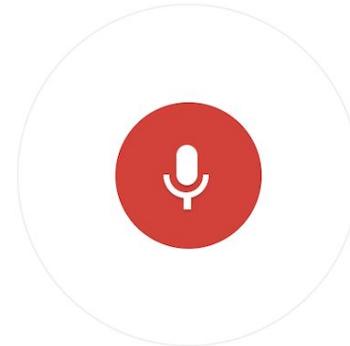
Supervised Learning

Ứng dụng - Speech Recognition

Phonemes/Words



Listening...



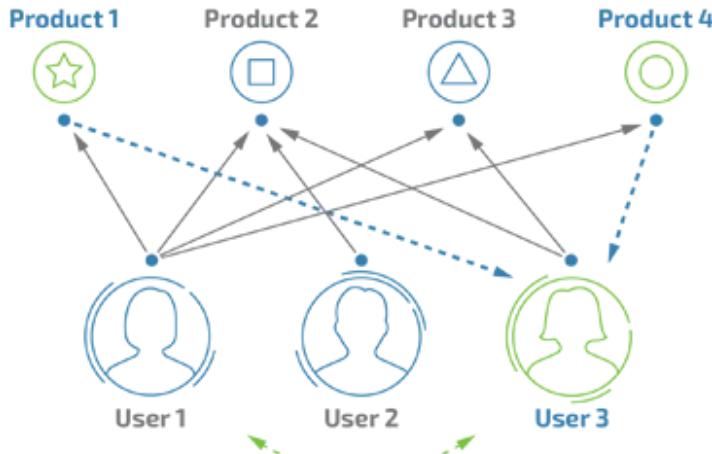
Acoustic model	Recog WER	RT03S FSH	Hub5 SWB
Traditional features	1-pass -adapt	27.4	23.6
Deep Learning	1-pass -adapt	18.5 (-33%)	16.1 (-32%)

Supervised Learning

Ứng dụng - Recommender System

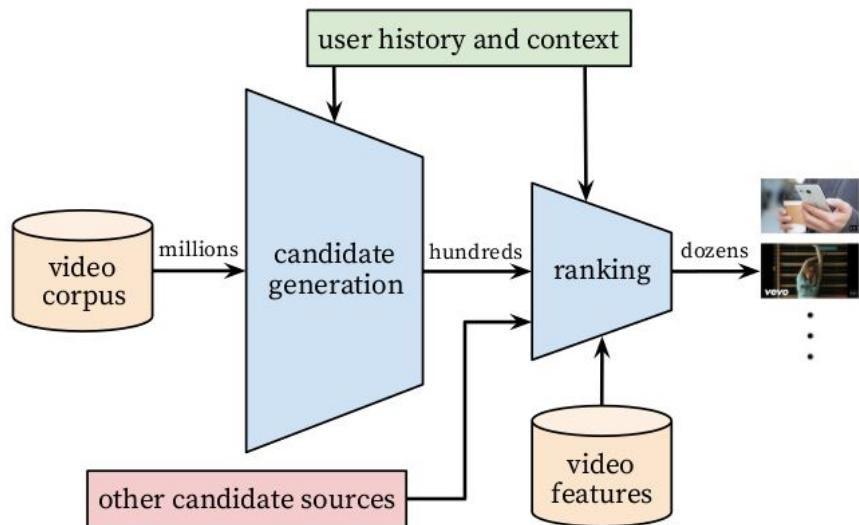
Hai kỹ thuật chính thường được sử dụng trong Recommender System:

- Collaborative Filtering
- Content-based Filtering

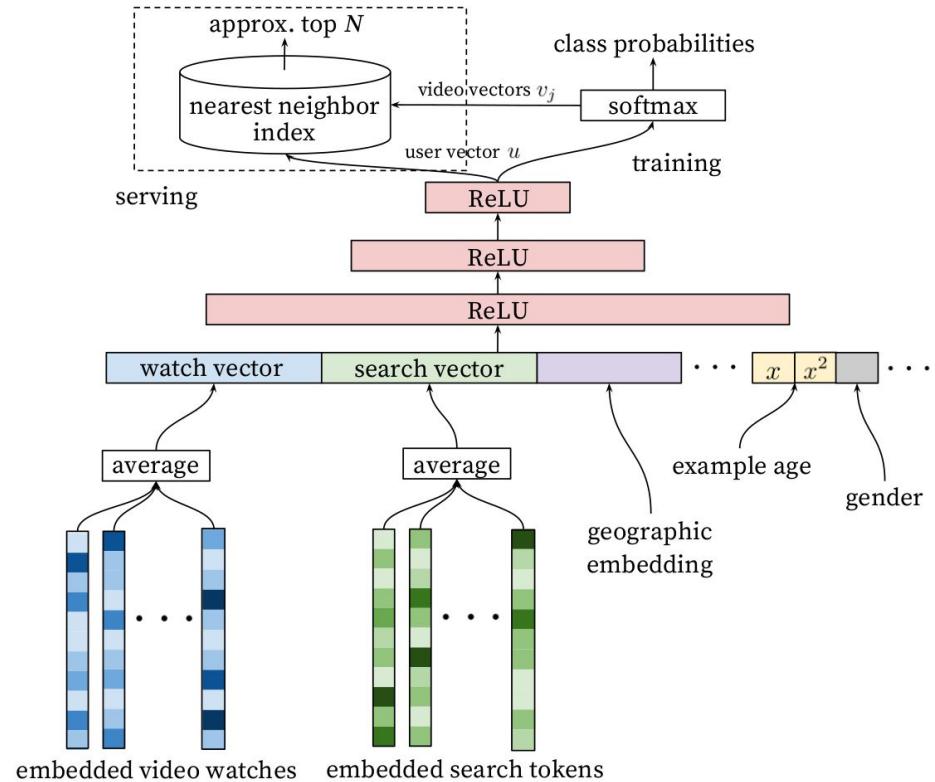


Supervised Learning

Ứng dụng - Recommender System



[Deep Neural Networks for YouTube Recommendations](#)

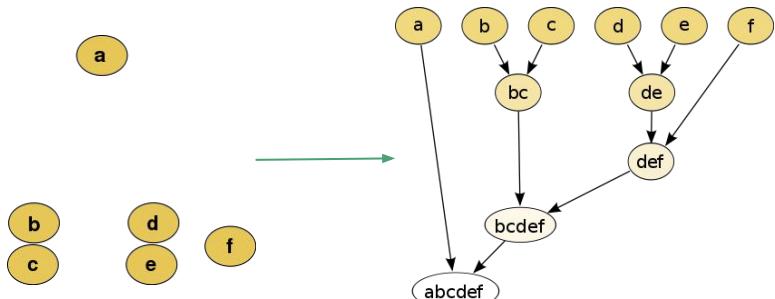


Unsupervised Learning

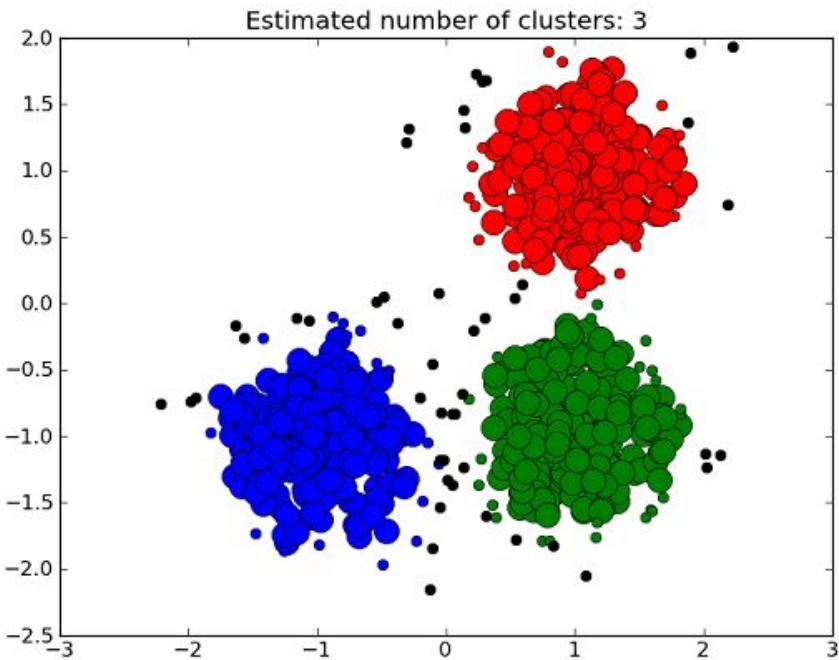
Unsupervised Learning

Bài toán - Clustering (Phân cụm)

- Nhóm những điểm dữ liệu có sự tương quan cao với nhau
- Ứng dụng:
 - Phân khúc khách hàng, ...
- Một số phương pháp thường dùng:
 - K-means, DBScan, Gaussian Mixture Model
 - Hierarchical clustering



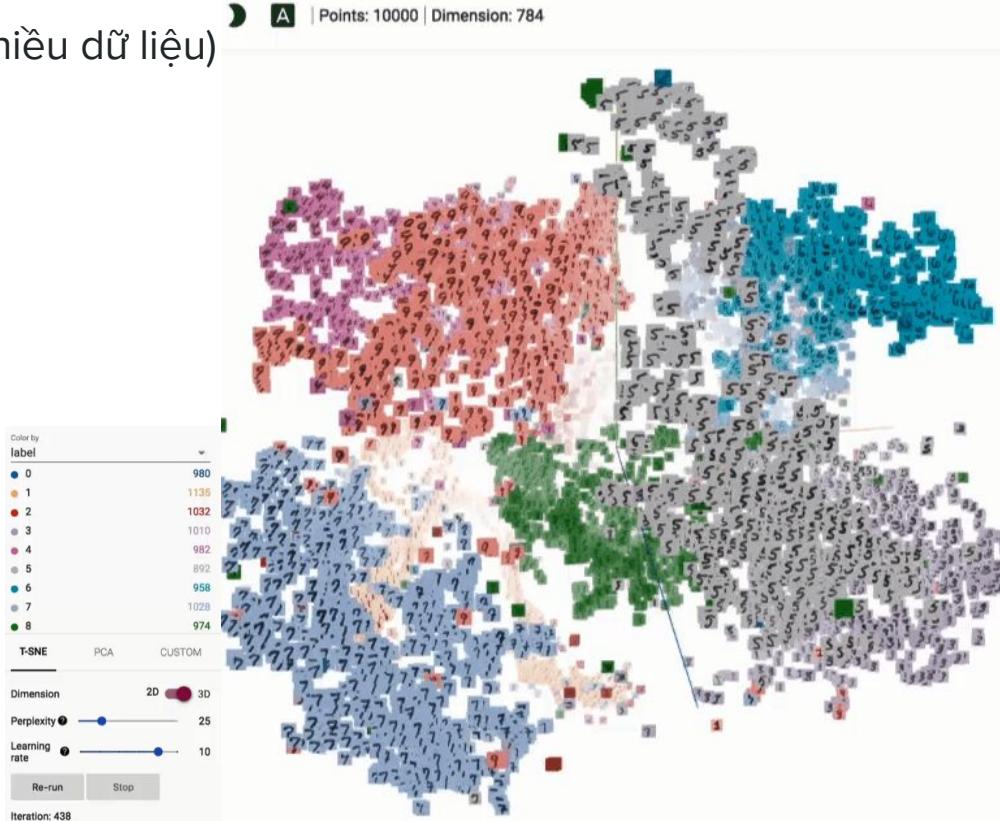
(image credits: [wikipedia](#))



Unsupervised Learning

Bài toán - Dimensionality Reduction (Giảm chiều dữ liệu)

- “Nén” dữ liệu mà vẫn giữ lại được lượng thông tin cần thiết
- Ứng dụng:
 - Data Compression, Data Visualization
 - Có thể là bước tiền xử lý để tăng tốc độ thuật toán ML chính được sử dụng sau đó (vd. kNN, SVM)
- Một số phương pháp thường dùng:
 - Principal Component Analysis (PCA)
 - t-distributed Stochastic Neighbor Embedding (t-SNE)
 - Auto-Encoder

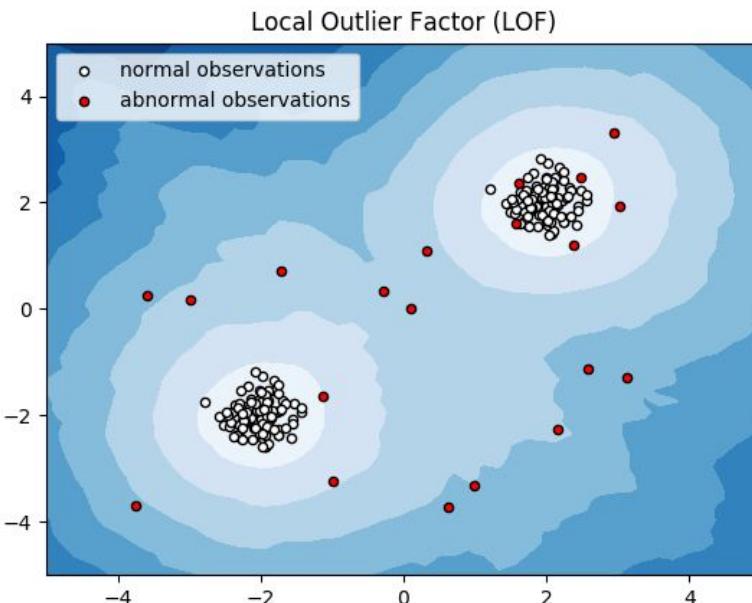


*t-SNE áp dụng trên bộ dữ liệu Chữ số viết tay MNIST
(visualize bằng TensorBoard)*

Unsupervised Learning

Bài toán - Anomaly Detection (Phát hiện bất thường)

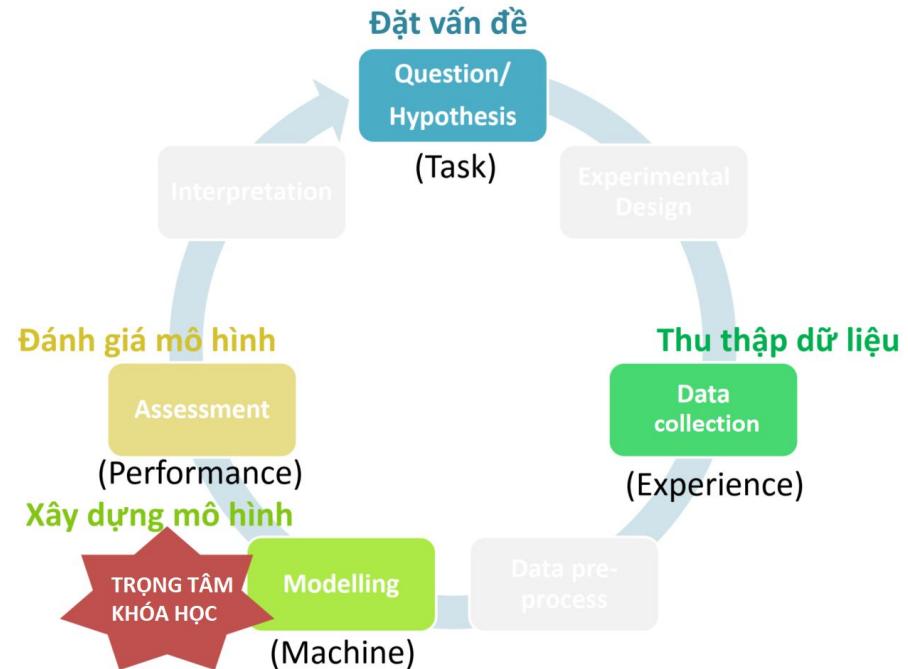
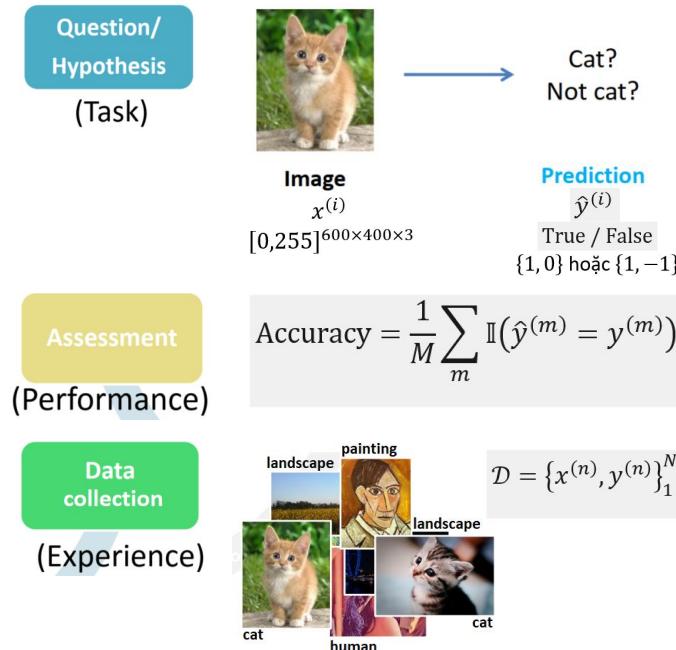
- Xác định những điểm dữ liệu bất thường, hiếm khi xảy ra
 - Outlier detection
 - Novelty detection
- Ứng dụng:
 - Dự đoán mức độ mạo hiểm tín dụng
 - Phát hiện gian lận trong thanh toán
 - Phát hiện lỗi của thiết bị
- Một số phương pháp thường dùng:
 - Density-based vd. Local Outlier Factor



Sơ lược các bước xây dựng mô hình / thuật toán Machine Learning

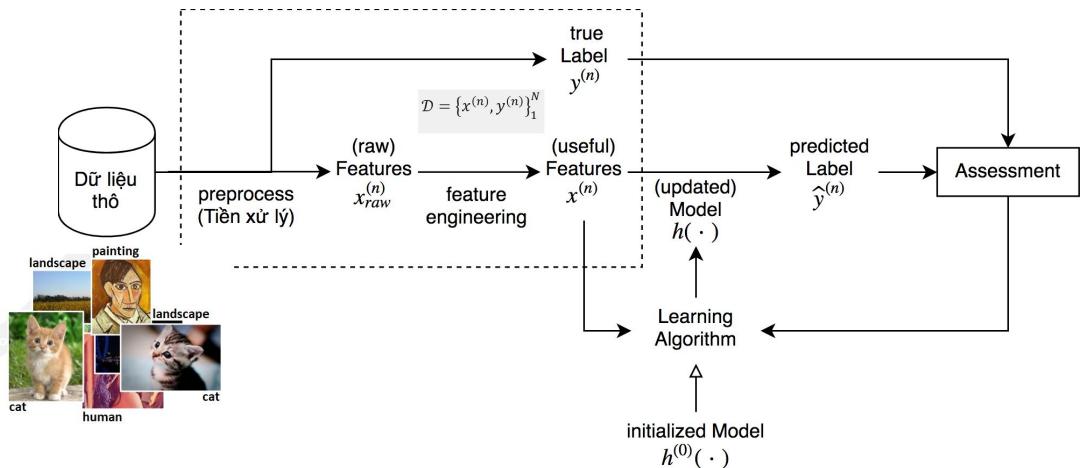
i. Tiếp cận bài toán

vd. Bài toán phân loại ảnh (phân loại nhị phân)



ii. Xây dựng mô hình

vd. Bài toán phân loại ảnh (phân loại nhị phân)



```

from your_ml_lib import DataReader, Featurizer, Model

# Đọc vào và clean dữ liệu thô
data = DataReader().read().clean()

# Biến đổi dữ liệu thành dictionary of Tensors. Cụ thể,
# Key 'X_raw' map với một Tensor 4D, chiều [số_ảnh x H x W x C]
# Key 'Y' map với một Tensor 1D, chiều [số_ảnh], là giá trị
# nhãn tương ứng
featurizer = Featurizer()
D = featurizer.preprocess(data)

# Biến đổi raw features thành features có ích cho dự đoán
D['X'] = featurizer.feature_engineer(D['X_raw'])

# Khởi tạo mô hình, constructor của Model class sẽ thiết
# lập cấu trúc/kiến trúc của mô hình
model = Model()

# Huấn luyện mô hình
model.fit(D['X'], D['Y'])
  
```

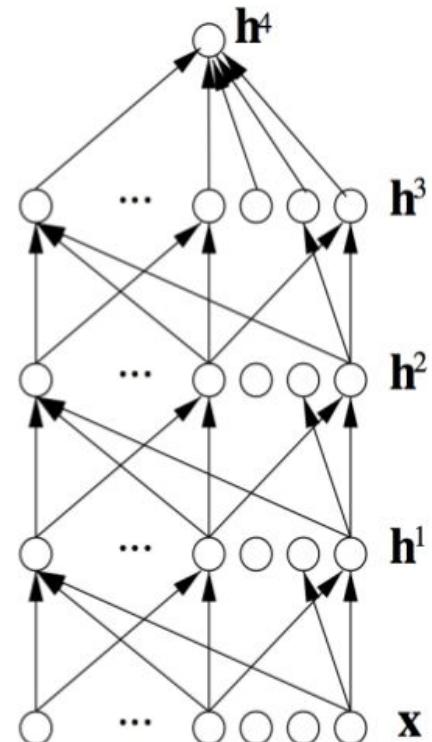
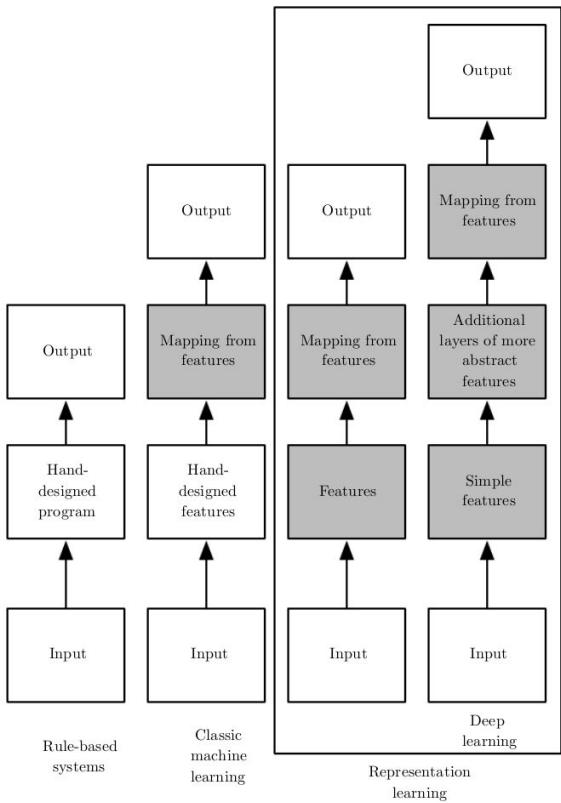
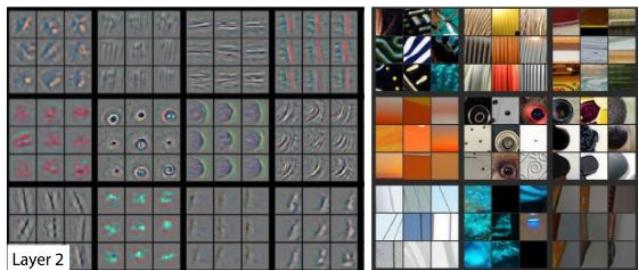
Chú ý: Những library ML bậc cao như [scikit-learn](#), [keras](#) thực tế cho phép code chỉ ngắn gọn như đoạn pseudo-code trên! Dù những thư viện này rất có ích trong thực tế, khóa học khuyến khích học viên chỉ sử dụng *sau khi* đã nắm được nền tảng lý thuyết nằm sau những methods trên, đặc biệt là các methods liên quan đến Thiết lập cấu trúc/kiến trúc mô hình và Huấn luyện mô hình.

iii. Đánh giá, phân tích ưu/nhược điểm
của Mô hình hiện tại
&

iv. Iterate: lặp lại i-iii để cải thiện chất lượng Mô hình

3. Một số hướng phát triển trong Machine Learning

Deep Learning



Các hướng phát triển khác

- Semi-supervised learning
- One-shot learning
- Transfer learning
- Multi-task learning
- Generative Model
- Các mô hình mang tính scalable và hiệu năng cao

Take-home messages

- Machine Learning (ML) là phương pháp dựa vào dữ liệu thu thập được để giải quyết một bài toán, mà không phải được lập trình / hướng dẫn cụ thể
- Với hướng tiếp cận dựa trên ML (data-driven) và dựa trên thành tựu mới của Deep Learning, nhiều ngành nghiên cứu và ứng dụng đã có những bước phát triển mạnh mẽ
- Mô hình ML được quy định bởi Tập giả thiết (Mô hình) H và Thuật toán học A
- *Nội dung tiếp theo:* Linear Regression - Mô hình tuyến tính cho bài toán Regression (Hồi quy) - và Thuật toán học Gradient Descent

Giới thiệu thư viện TensorFlow

Nội dung

- Khái quát về Deep Learning Frameworks và những ưu thế nổi trội của TensorFlow (TF)
- So sánh TF và NumPy
- Cách viết và chạy một chương trình đơn giản dựa trên TF

Tại sao cần Deep Learning Frameworks?

- Mở rộng mô hình cho production với hàng triệu người dùng, đặc biệt phần deployment và development.
- Hỗ trợ việc tự động tính toán đạo hàm (Automatic Differentiation).
- Cho phép việc xử lý song song (parallel processing) với card đồ họa (GPU).
- Đặt ra các tiêu chuẩn phát triển các ứng dụng Machine Learning.

Một số Deep Learning Frameworks



TensorFlow



PYTORCH



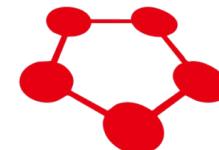
Keras

Caffe



mxnet

DEEPMLEARNING4J



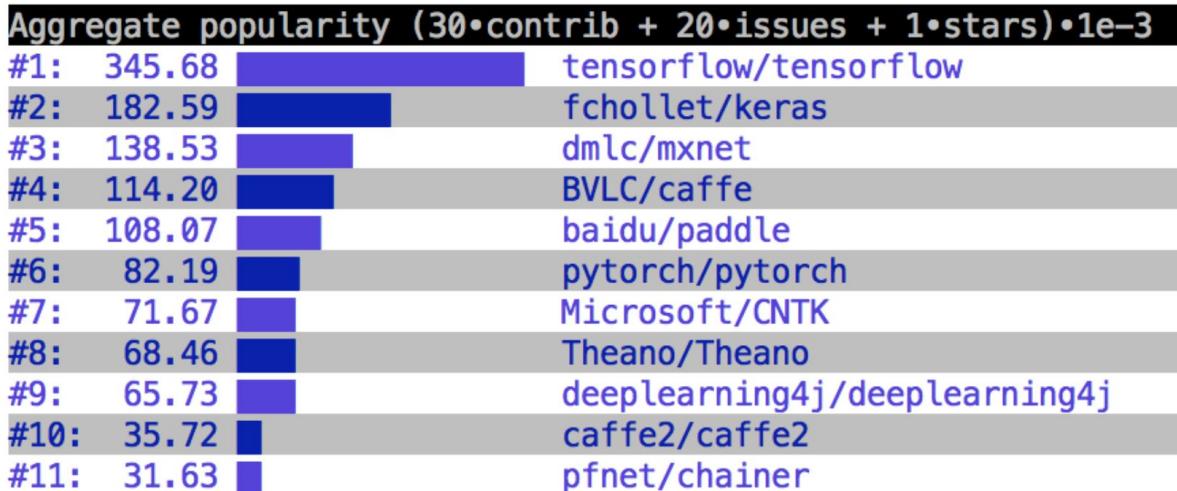
Chainer

Một số Deep Learning Frameworks



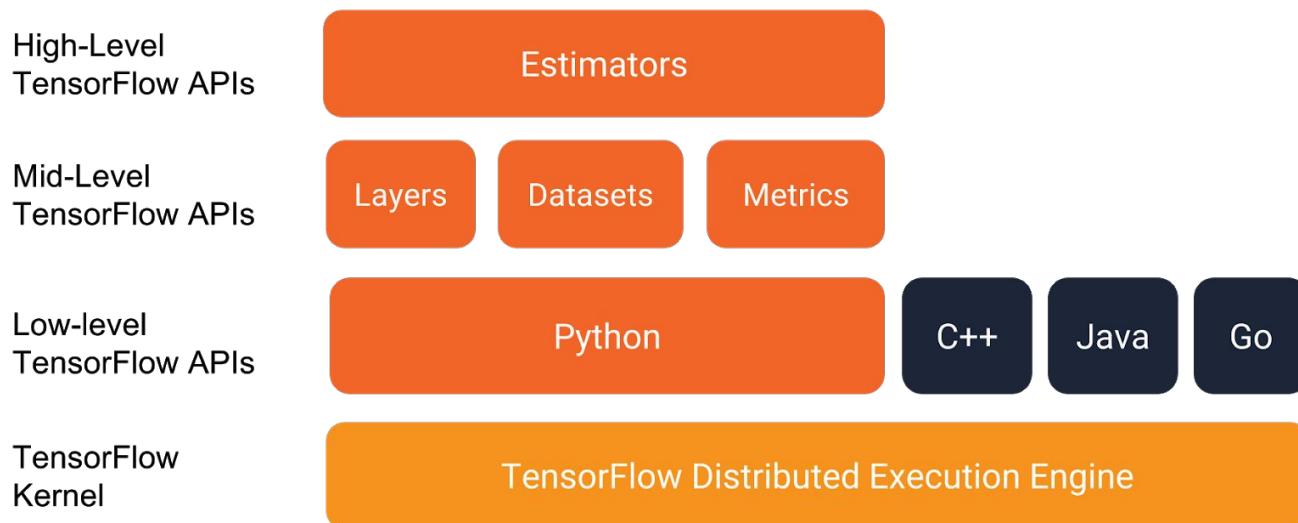
François Chollet  @fchollet · Mar 8

And here are the GitHub popularity metrics at this time.



TensorFlow

- Được phát hành và phát triển bởi Google dưới hình thức mã nguồn mở.
- Tương thích với Python, C++, Java, Go và JavaScript.



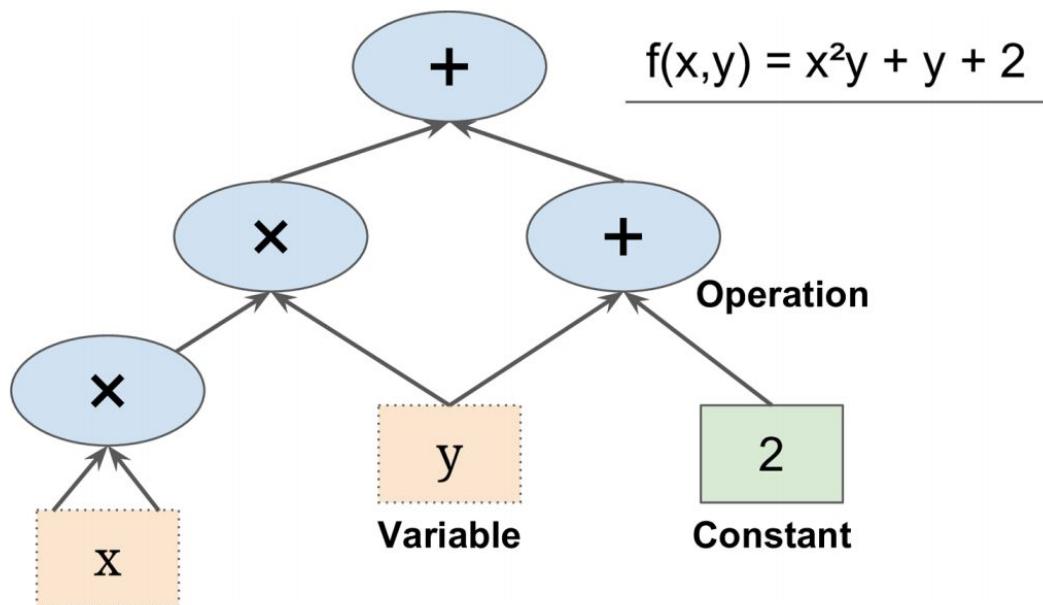
TensorFlow Core API

- Ý tưởng chính của TensorFlow là biểu diễn các phép tính số theo dạng đồ thị (Computational Graph)
- Trong đó:
 - Node là các toán tử
 - Cạnh là các Tensors
- Nhắc lại, Tensor là các mảng số thực có N chiều.



TensorFlow

Simple Computation Graph

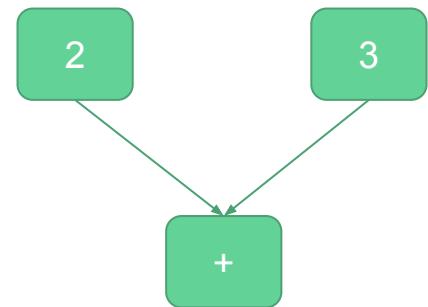


TensorFlow

The Computation Graph

- Một đồ thị có hướng, mô tả các chỉ dẫn làm thế nào để có thể tính các phép tính

```
import tensorflow as tf
two_node = tf.constant(2)
three_node = tf.constant(3)
sum_node = two_node + three_node ## equivalent to tf.add(two_node, three_node)
```



TensorFlow

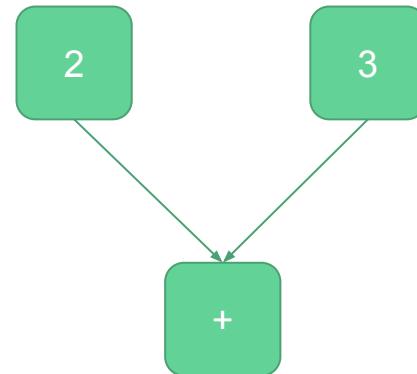
The Session

- Bắt đầu quá trình tính toán thực sự trên Computation Graph

```
import tensorflow as tf
node_two = tf.constant(2)
node_three = tf.constant(3)
sum_node = node_two + node_three
sess = tf.Session()
print(sess.run(sum_node))
```

Output:

5



TensorFlow

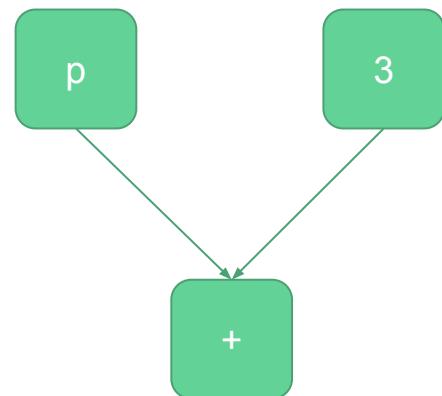
Placeholders & feed dict

- Giúp đưa dữ liệu đầu vào (input) vào chương trình

```
import tensorflow as tf
input_placeholder = tf.placeholder(tf.int32)
node_three = tf.constant(3)
sum_node = input_placeholder + node_three
sess = tf.Session()
print(sess.run(sum_node, feed_dict={input_placeholder: 2}))
```

Output:

```
5
```



So sánh giữa Numpy và TensorFlow

Các phép tính của TensorFlow được định nghĩa dựa trên **computational graph**, chỉ có giá trị cho đến lúc được tính.

```
a = np.zeros((2, 2)) # Numpy  
ta = tf.zeros((2, 2)) # Tensorflow
```

```
print(a)
```

```
[[ 0.  0.]  
 [ 0.  0.]]
```

```
print(ta)
```

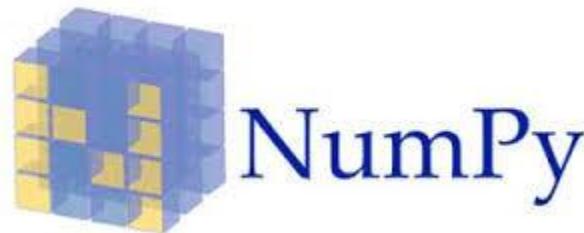
```
Tensor("zeros:0", shape=(2, 2), dtype=float32)
```

```
print(ta.eval())
```

```
[[ 0.  0.]  
 [ 0.  0.]]
```

So sánh giữa Numpy và TensorFlow

- Có nhiều điểm tương đồng
- Numpy có hỗ trợ mảng N chiều tuy nhiên lại không hỗ trợ tính toán trên GPU
- TF còn hỗ trợ thêm phần tạo ra tensor và tự tính đạo hàm.



So sánh giữa Numpy và TensorFlow

```
import numpy as np  
  
a = np.zeros((2,2)); b = np.ones((2,2))  
  
np.sum(b, axis=1)  
  
a.shape  
  
np.reshape(a, (1,4))
```

```
import tensorflow as tf  
  
tf.InteractiveSession()  
  
a = tf.zeros((2,2)); b = tf.ones((2,2))  
  
tf.reduce_sum(b, reduction_indices = 1).eval()  
  
a.get_shape()  
  
tf.reshape(a, (1,4)).eval()
```

So sánh giữa Numpy và TensorFlow

Numpy to TensorFlow Dictionary

Numpy	TensorFlow
<code>a = np.zeros((2,2)); b = np.ones((2,2))</code>	<code>a = tf.zeros((2,2)), b = tf.ones((2,2))</code>
<code>np.sum(b, axis=1)</code>	<code>tf.reduce_sum(a, reduction_indices=[1])</code>
<code>a.shape</code>	<code>a.get_shape()</code>
<code>np.reshape(a, (1,4))</code>	<code>tf.reshape(a, (1,4))</code>
<code>b * 5 + 1</code>	<code>b * 5 + 1</code>
<code>np.dot(a,b)</code>	<code>tf.matmul(a, b)</code>
<code>a[0,0], a[:,0], a[0,:]</code>	<code>a[0,0], a[:,0], a[0,:]</code>

TensorFlow

Lưu ý về Session

- `Tf.InteractiveSession()` là một cách viết tiện dụng cho phép tắt cả các mặc định session mở trong iPython. Để đóng lại session, ta sử dụng `session.close()`.
- `.eval()` là một cách viết khác của `session.run()`, đơn giản và theo kiểu Python.

Cách viết chương trình theo TensorFlow

Cài đặt chương trình theo hàm cost

Tìm giá trị nhỏ nhất của hàm số sau:

$$\mathbf{J}(\omega) = \omega^2 - 10\omega + 25$$

Thực hiện tuần tự theo các bước sau:

1. Khởi tạo biến
2. Bắt đầu session
3. Giải thuật
4. Close session

Cách viết chương trình theo TensorFlow

Code toàn bộ chương trình

```
coefficients = np.array([[1.0], [-10], [25.]])  
  
w = tf.Variable(initial_value=0, dtype=tf.float32)  
x = tf.placeholder(dtype=tf.float32, shape=(3, 1))  
  
cost = w ** 2 * x[0] + w * x[1] + x[2]  
train = tf.train.GradientDescentOptimizer(0.01).minimize(cost)  
  
init = tf.global_variables_initializer()  
session = tf.Session()  
session.run(init)  
  
num_iterations = 1000  
for _ in range(num_iterations):  
    session.run(train, feed_dict={x: coefficients})  
  
print(session.run(w))  
  
session.close()
```

Cách viết chương trình theo TensorFlow

Cách viết Session để dễ debug

```
init = tf.global_variables_initializer()
session = tf.Session()
session.run(init)
print(session.run(w))
session.close()
```



```
with tf.Session() as sess:
    sess.run(init)
    print(sess.run(w))
```

TensorFlow

Eager Execution

- Chế độ Eager execution của TF cho phép thực thi các toán tử ngay khi được khai báo mà không cần khởi tạo Graph

Eager Execution

```
import tensorflow as tf
tf.enable_eager_execution()
node_two = tf.constant(2)
node_three = tf.constant(3)
sum_node = node_two + node_three
print(sum_node)
```

Output

```
tf.Tensor(5, shape=(), dtype=int32)
```

Normal

```
import tensorflow as tf
node_two = tf.constant(2)
node_three = tf.constant(3)
sum_node = node_two + node_three
sess = tf.Session()
print(sess.run(sum_node))
```

Output

```
5
```

Tài liệu tham khảo

1. Numpy Reference Guide - <https://docs.scipy.org/doc/numpy-1.13.0/reference/>
2. TensorFlow documentation - https://www.tensorflow.org/api_docs/
3. Lecture 7 - CS224D - Stanford University
4. CS20SI - Stanford University
5. Deep Learning - Coursera - [Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization](#)
6. Programmer Guide, TensorFlow -
https://www.tensorflow.org/programmers_guide/faq#running_a_tensorflow_computation
7. <https://jacobbuckman.com/post/tensorflow-the-confusing-parts-1/>