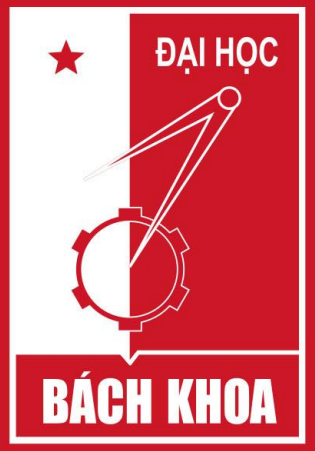


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

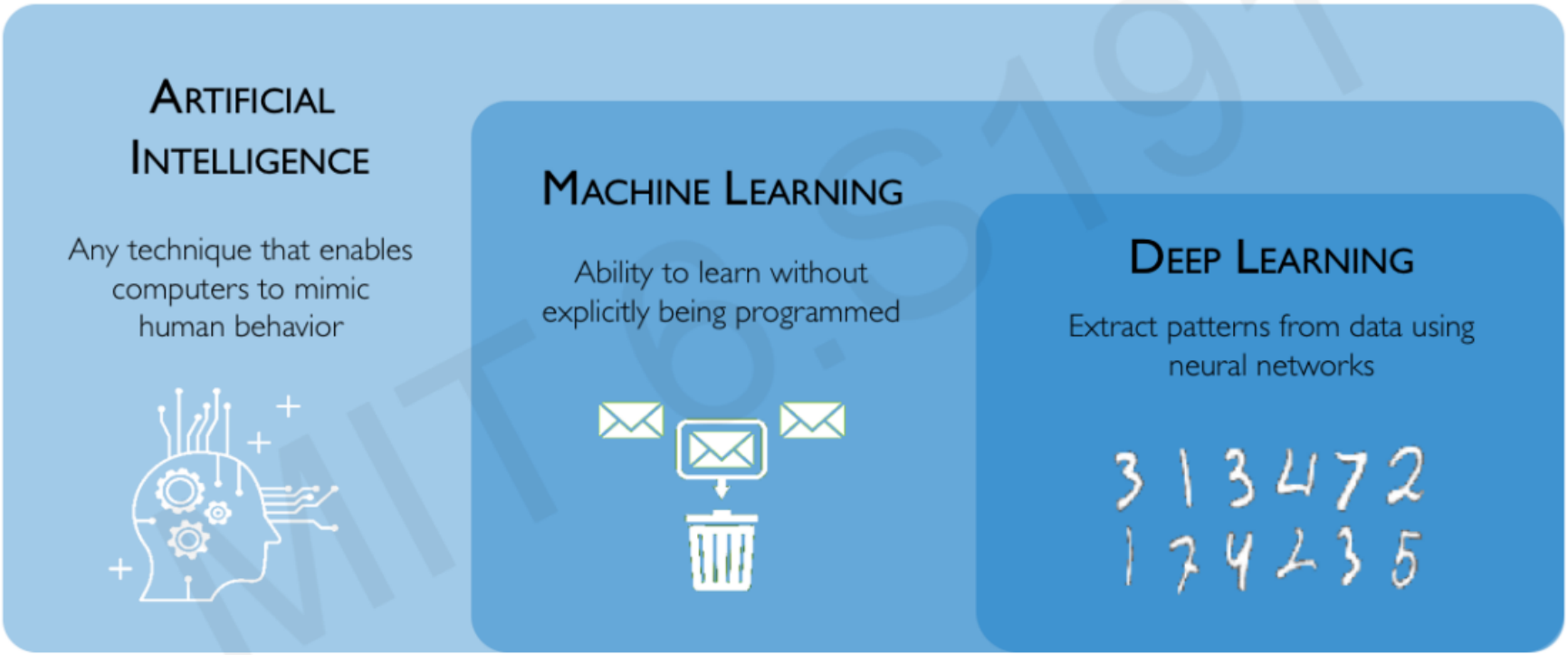
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Học sâu và ứng dụng (IT4653)

Bài 1:
Giới thiệu về học sâu

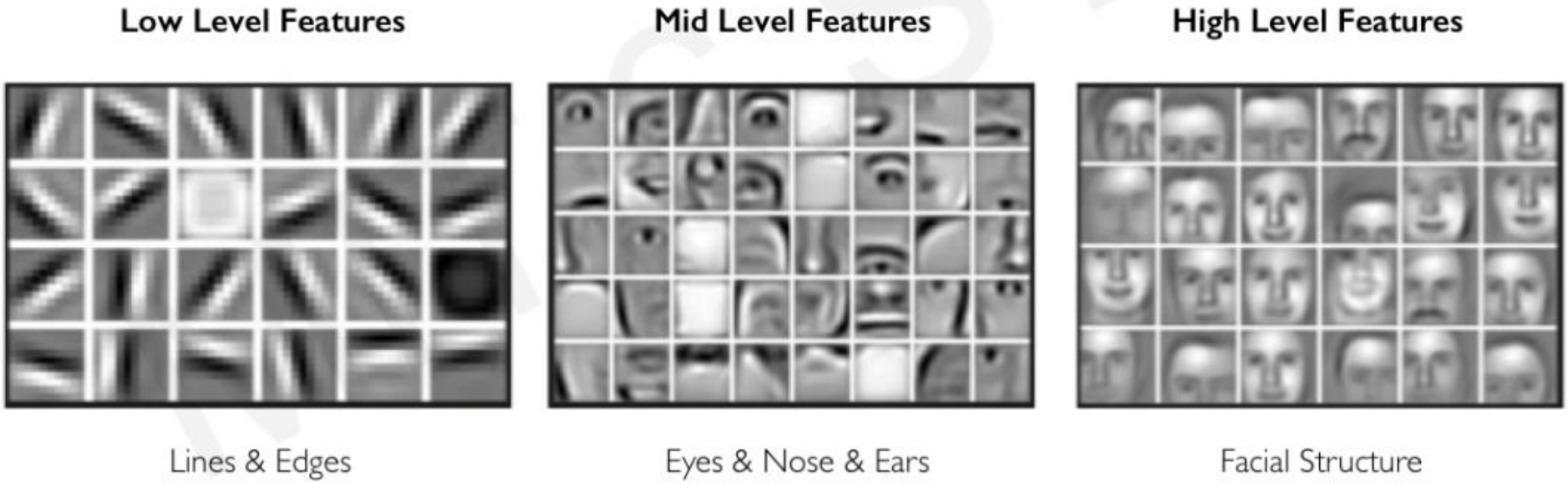
Thế nào là học sâu?

- Là phương pháp học máy sử dụng mạng nơ-ron nhân tạo để trích xuất đặc trưng tự động từ dữ liệu

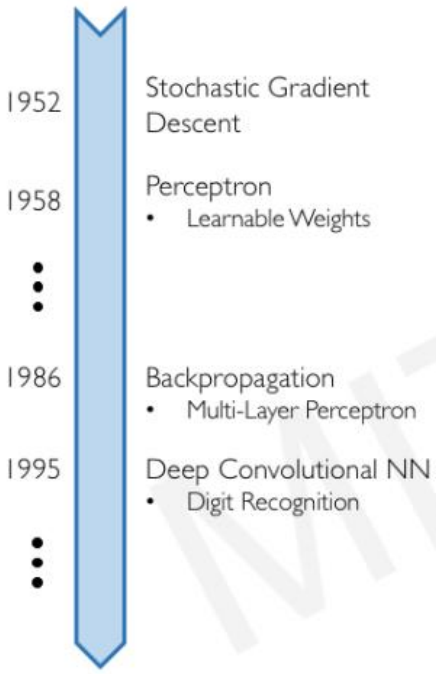


Tại sao cần học sâu?

- Phương pháp học máy truyền thống đòi hỏi trích xuất đặc trưng một cách thủ công, đòi hỏi kinh nghiệm và phụ thuộc từng bài toán cụ thể
- Học sâu cho phép trích chọn đặc trưng tự động từ dữ liệu



Tại sao giờ mới bùng nổ học sâu?



Neural Networks date back decades, so why the resurgence?

1. Big Data

- Larger Datasets
- Easier Collection & Storage

IMAGENET

WIKIPEDIA
The Free Encyclopedia

2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable

3. Software

- Improved Techniques
- New Models
- Toolboxes

TensorFlow

Học máy có giám sát

Functions $\mathcal{F}$

$f : \mathcal{X} \to \mathcal{Y}$

Training data

$\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$

LEARNING

find $\hat{f} \in \mathcal{F}$
s.t. $y_i \approx \hat{f}(x_i)$

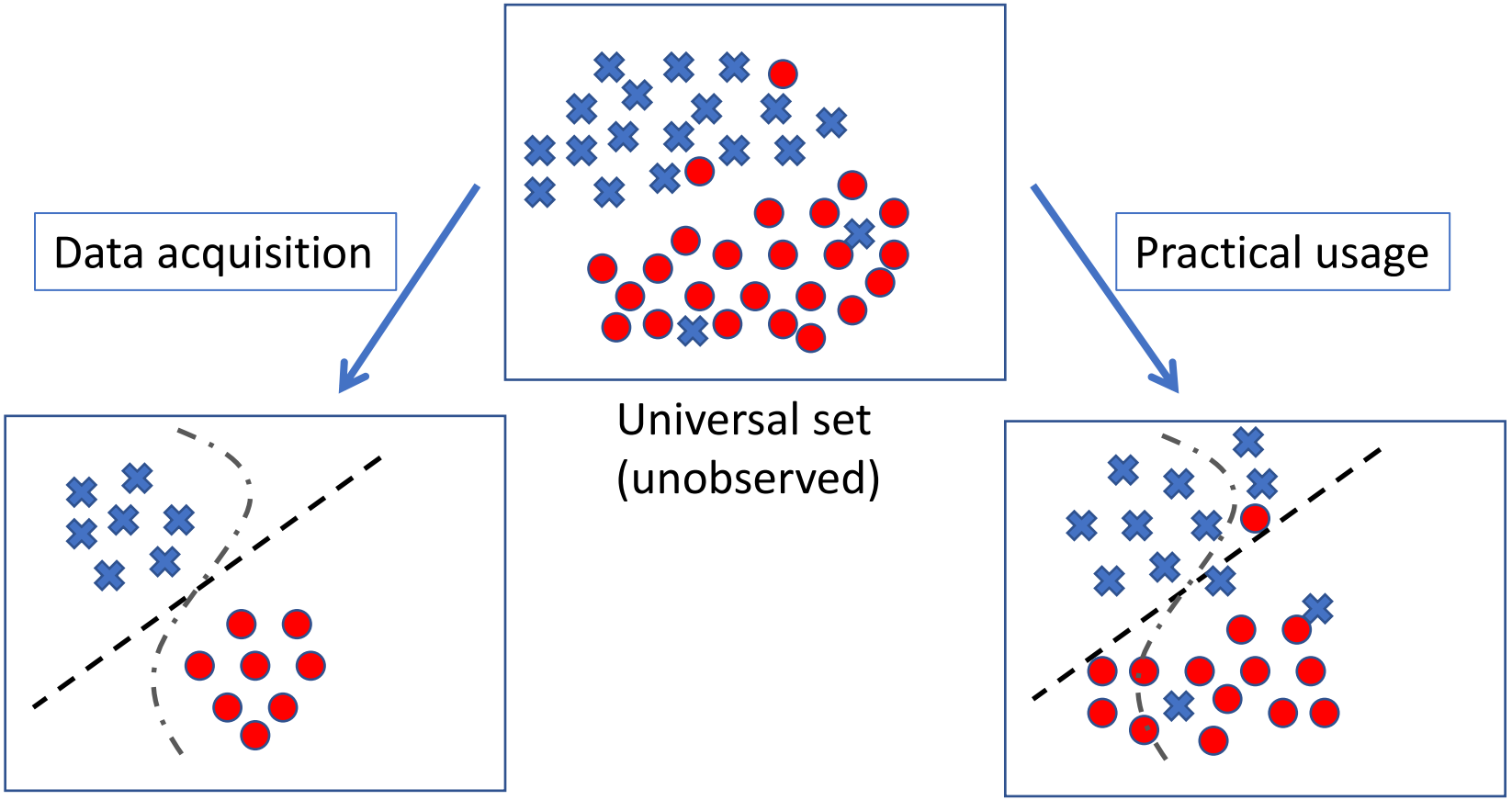
Learning machine

New data

PREDICTION

$y = \hat{f}(x)$ ⟸ x

Tập huấn luyện và tập kiểm tra

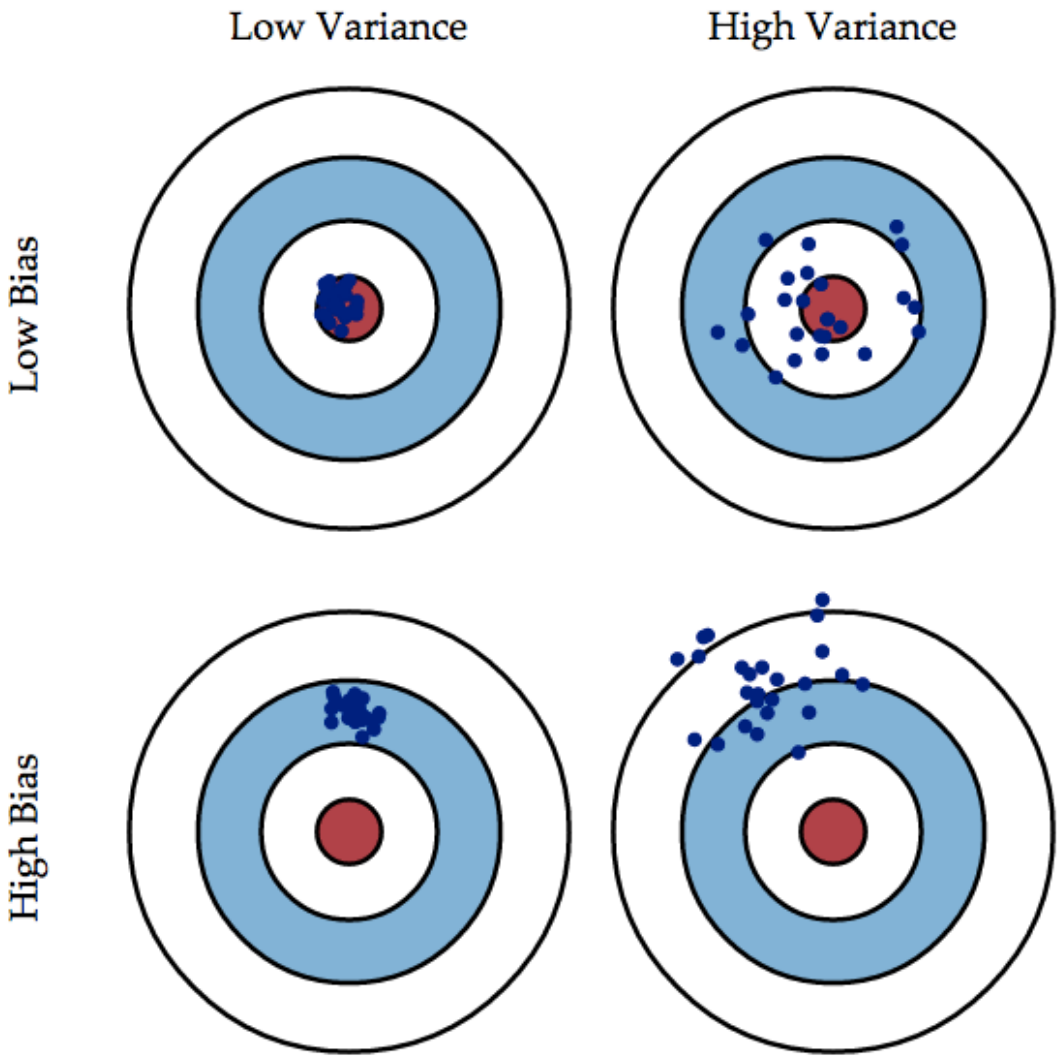


Hiện tượng overfit và underfit

- **Underfitting:** mô hình quá "đơn giản" để biểu diễn các tính chất của dữ liệu
  - Bias cao và variance thấp
  - Sai số cao trên tập huấn luyện và tập kiểm tra
- **Overfitting:** mô hình quá "phức tạp" dẫn tới học cả nhiễu trong dữ liệu
  - Bias thấp và variance cao
  - Sai số thấp trên tập huấn luyện và sai số cao trên tập kiểm tra

Adapted from L. Lazebnik

Minh họa Bias-Variance



Phân lớp tuyến tính

50,000 training images
each image is **32x32x3**

10,000 test images.

Phân lớp tuyến tính

$$f(x,W) = Wx$$

Image

Array of **32x32x3** numbers
(3072 numbers total)

f(**x**,**W**)

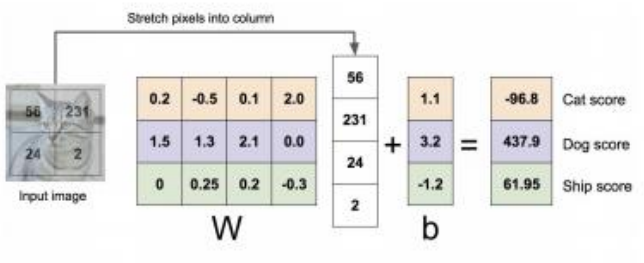
10 numbers giving
class scores

W
parameters
or weights

Phân lớp tuyến tính: 3 góc nhìn

Algebraic Viewpoint

$$f(x,W) = Wx$$

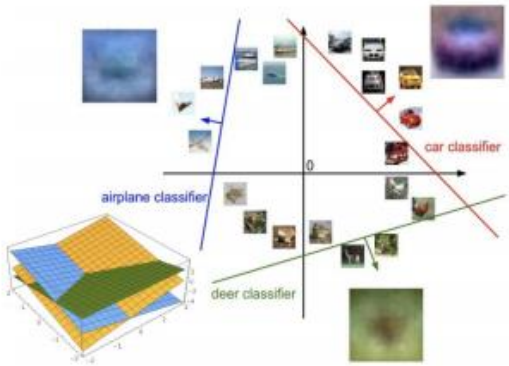


Visual Viewpoint

One template per class

Geometric Viewpoint

Hyperplanes cutting up space



Hàm mục tiêu

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

A **loss function** tells how good our current classifier is

Given a dataset of examples

$$\{(x_i, y_i)\}_{i=1}^{N}$$

Where x_i s image and
y_i s (integer) label

Loss over the dataset is a average of loss over examples:

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

Hàm mục tiêu

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Hiệu chỉnh

λ = regularization strength (hyperparameter)

$$L(W) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i)}_{\text{Data loss}} + \underbrace{\lambda R(W)}_{\text{Regularization}}$$

Data loss: Model predictions should match training data

Regularization: Prevent the model from doing *too* well on training data

Simple examples

L2 regularization: $R(W) = \sum_k \sum_l W_{k,l}^2$

L1 regularization: $R(W) = \sum_k \sum_l |W_{k,l}|$

Elastic net (L1 + L2): $R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$

More complex:

Dropout

Batch normalization

Stochastic depth, fractional pooling, etc

Bộ phân loại softmax

Want to interpret raw classifier scores as **probabilities**

$$s = f(x_i; W)$$

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax Function

$$L_i = -\log P(Y = y_i|X = x_i)$$

Probabilities must be >= 0

Probabilities must sum to 1

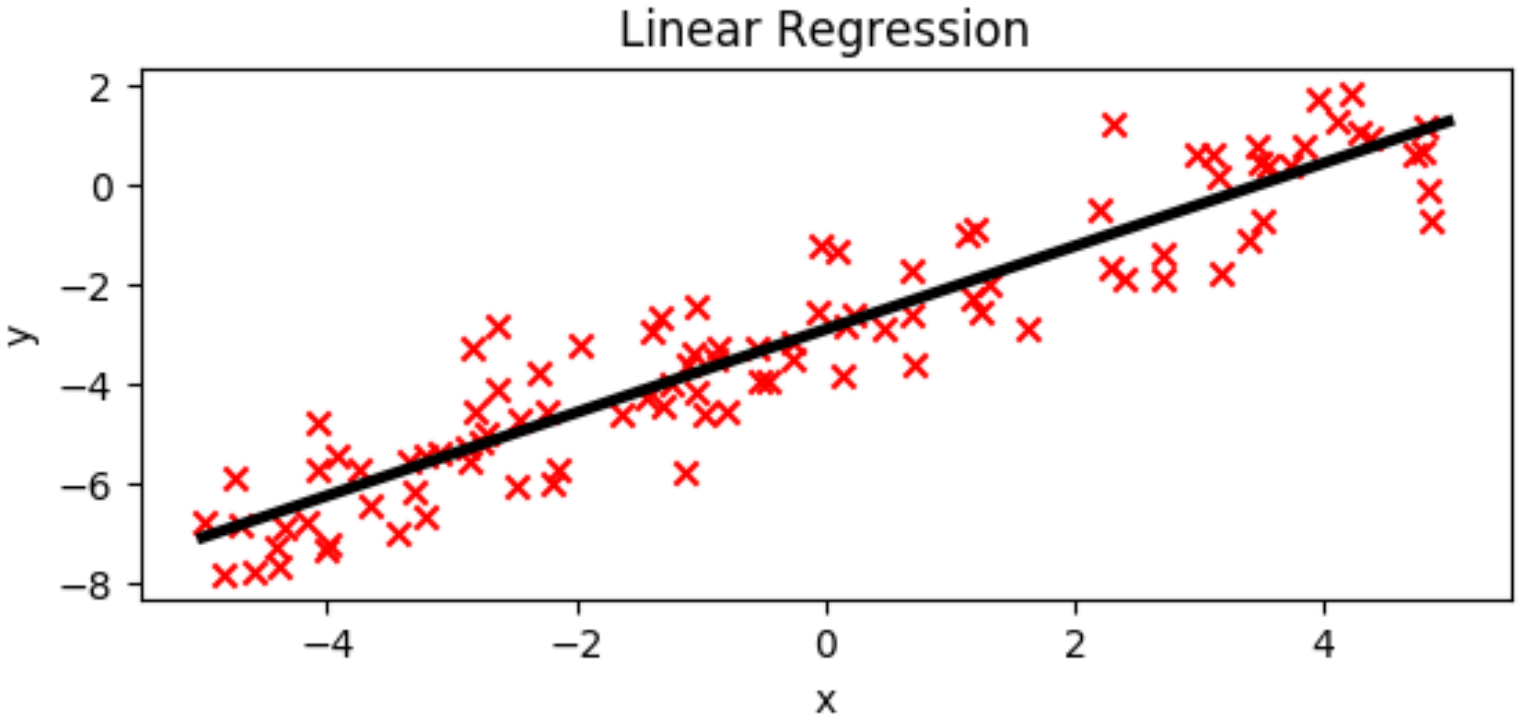
	Unnormalized log-probabilities / logits	exp	unnormalized probabilities	normalize	probabilities	compare	Correct probs
cat	**3.2**		**24.5**		**0.13**		**1.00**
car	5.1		164.0		0.87		0.00
frog	-1.7		0.18		0.00		0.00

Kullback–Leibler divergence

$$D_{KL}(P||Q) = \sum_y P(y) \log \frac{P(y)}{Q(y)}$$

Hồi quy tuyến tính

- $f(x; w) = w_0 + \sum_{i=1}^{d} w_i x_i = w^T x'$



Hồi quy tuyến tính

- Nên chọn hàm mục tiêu nào?
  - Mỗi $y^{(i)}$ là một số thực
  - Bình phương tối thiểu là một lựa chọn tốt ☺

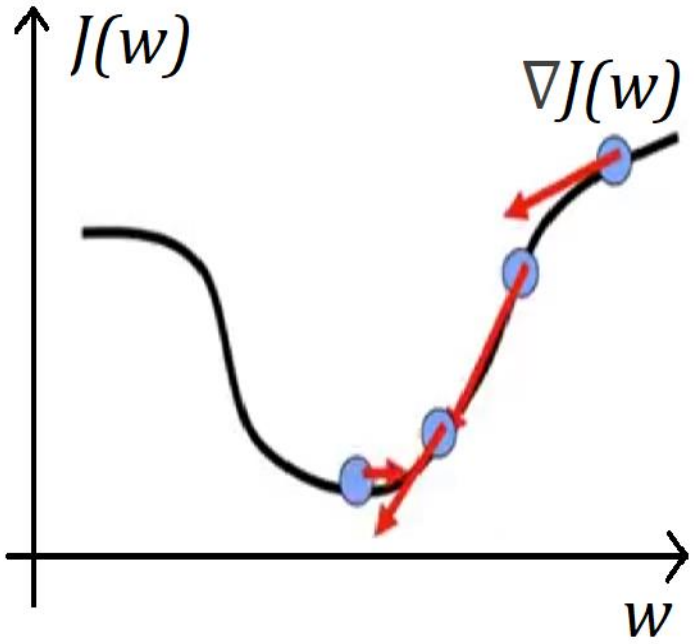
$$J(w; \mathbf{X}, \mathbf{Y}) = \frac{1}{N}\sum_{i=1}^{N}\left[f\left(x^{(i)}; w\right) - y^{(i)}\right]^2$$

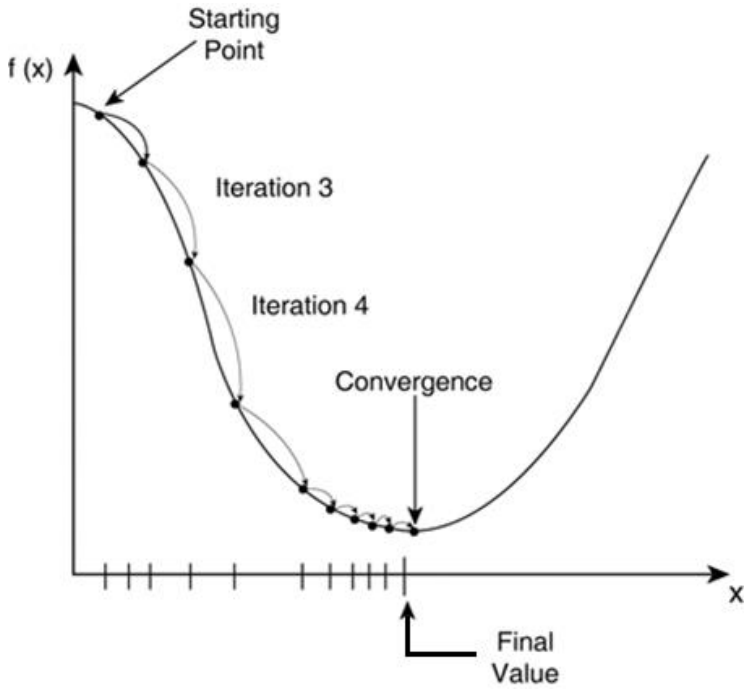
$$= \frac{1}{N}\sum_{i=1}^{N}\left[w^T x^{(i)\prime} - y^{(i)}\right]^2$$

$$= \frac{1}{N}(w^T\mathbf{X}' - \mathbf{Y})^T(w^T\mathbf{X}' - \mathbf{Y})$$

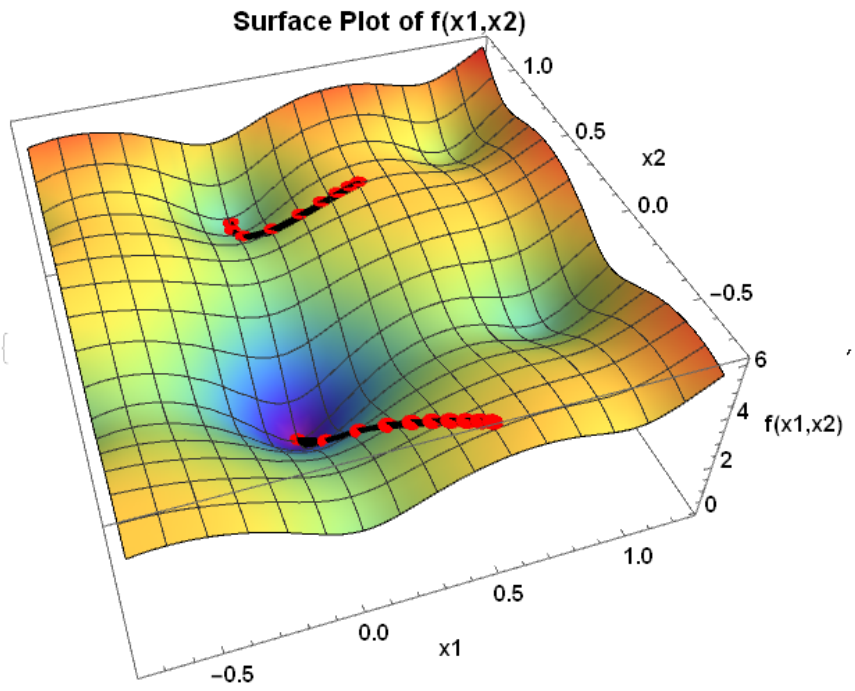
Tối ưu hàm mục tiêu

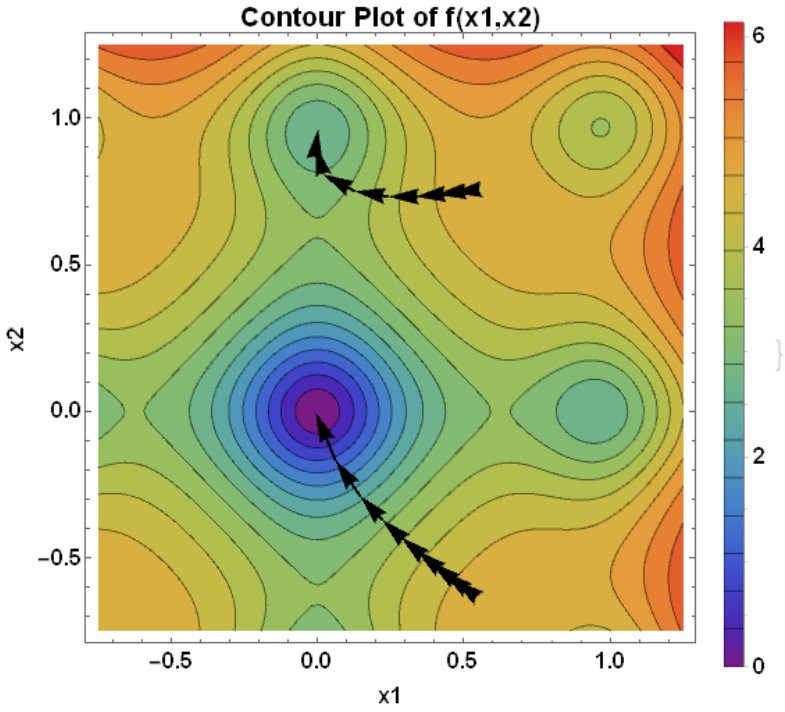
Gradient Descent





Gradient Descent





Gradient Descent

- Chọn tốc độ học learning rate η
- Khởi tạo w ngẫu nhiên
  - Khởi tạo w từ các phân bố thường gặp như phân bố đều hoặc phân bố chuẩn (gauss)
- Chừng nào w vẫn chưa hội tụ
  - Cập nhật $w \leftarrow w - \eta \nabla J(w; \mathbf{X}, \mathbf{Y})$

Stochastic Gradient Descent (SGD)

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(x_i, y_i, W) + \lambda R(W)$$

$$\nabla_W L(W) = \frac{1}{N}\sum_{i=1}^{N} \nabla_W L_i(x_i, y_i, W) + \lambda \nabla_W R(W)$$

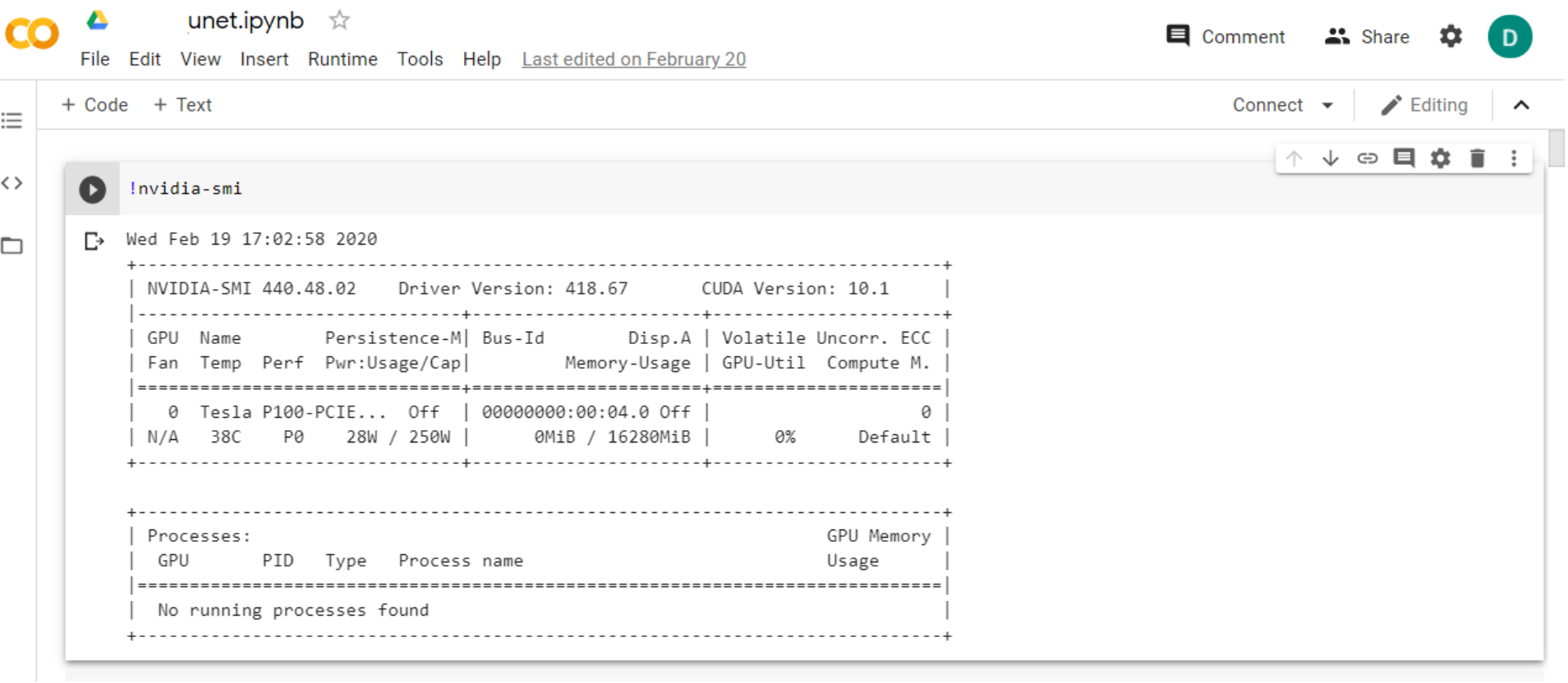
Full sum expensive when N is large!

Approximate sum using a **minibatch** of examples
32 / 64 / 128 common

```
# Vanilla Minibatch Gradient Descent

while True:
  data_batch = sample_training_data(data, 256) # sample 256 examples
  weights_grad = evaluate_gradient(loss_fun, data_batch, weights)
  weights += - step_size * weights_grad # perform parameter update
```

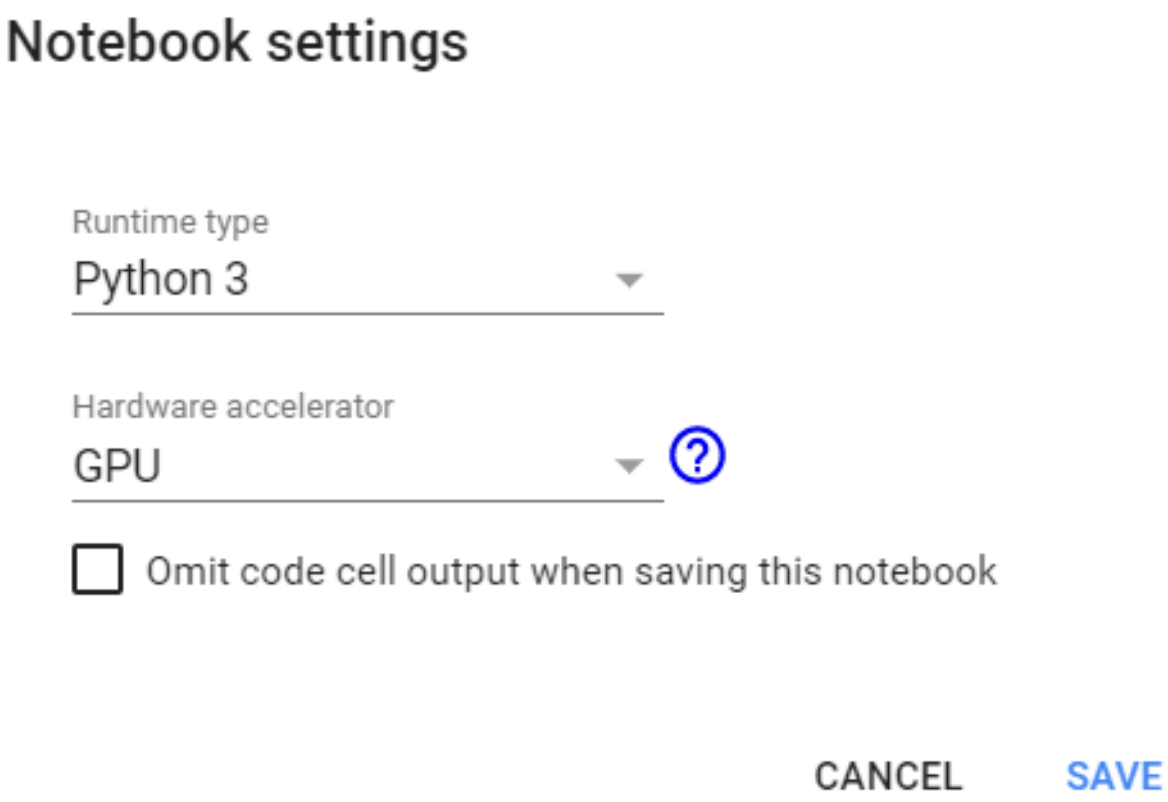
Giới thiệu công cụ và môi trường

- Google Colab: https://colab.research.google.com/
- Miễn phí GPU (Tesla T4/P100)
- Dùng liên tục 10 tiếng mỗi session



Google Colab

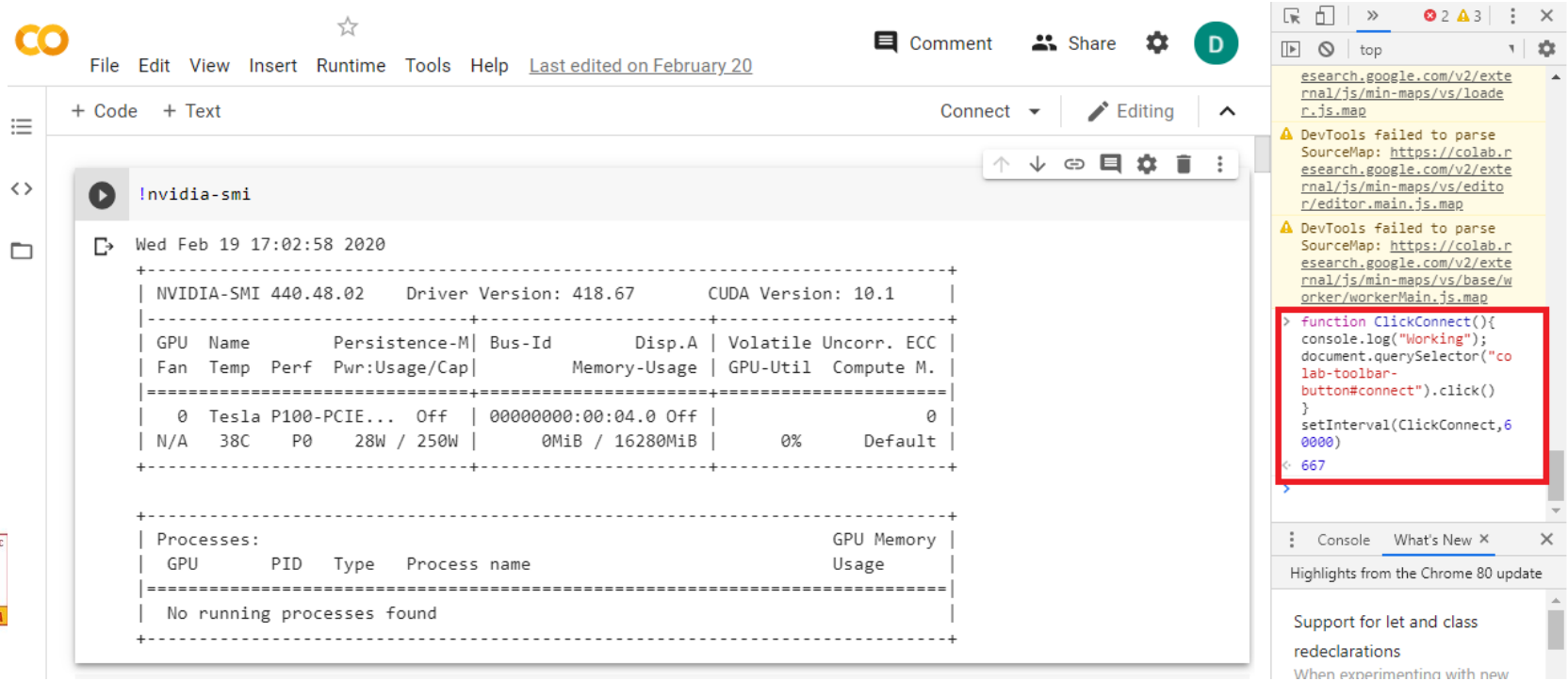
- Thiết lập GPU: Edit / Notebook settings



Google Colab

- Trick giúp colab chạy liên tục: Ấn F12, chọn Console

```
function ClickConnect(){
console.log("Working");
document.querySelector("colab-toolbar-button#connect").click()
}
setInterval(ClickConnect,60000)
```



Google Colab

- Tạo nhiều tài khoản google
- Share dữ liệu cho nhiều tài khoản google cùng dùng
- Mount dữ liệu:

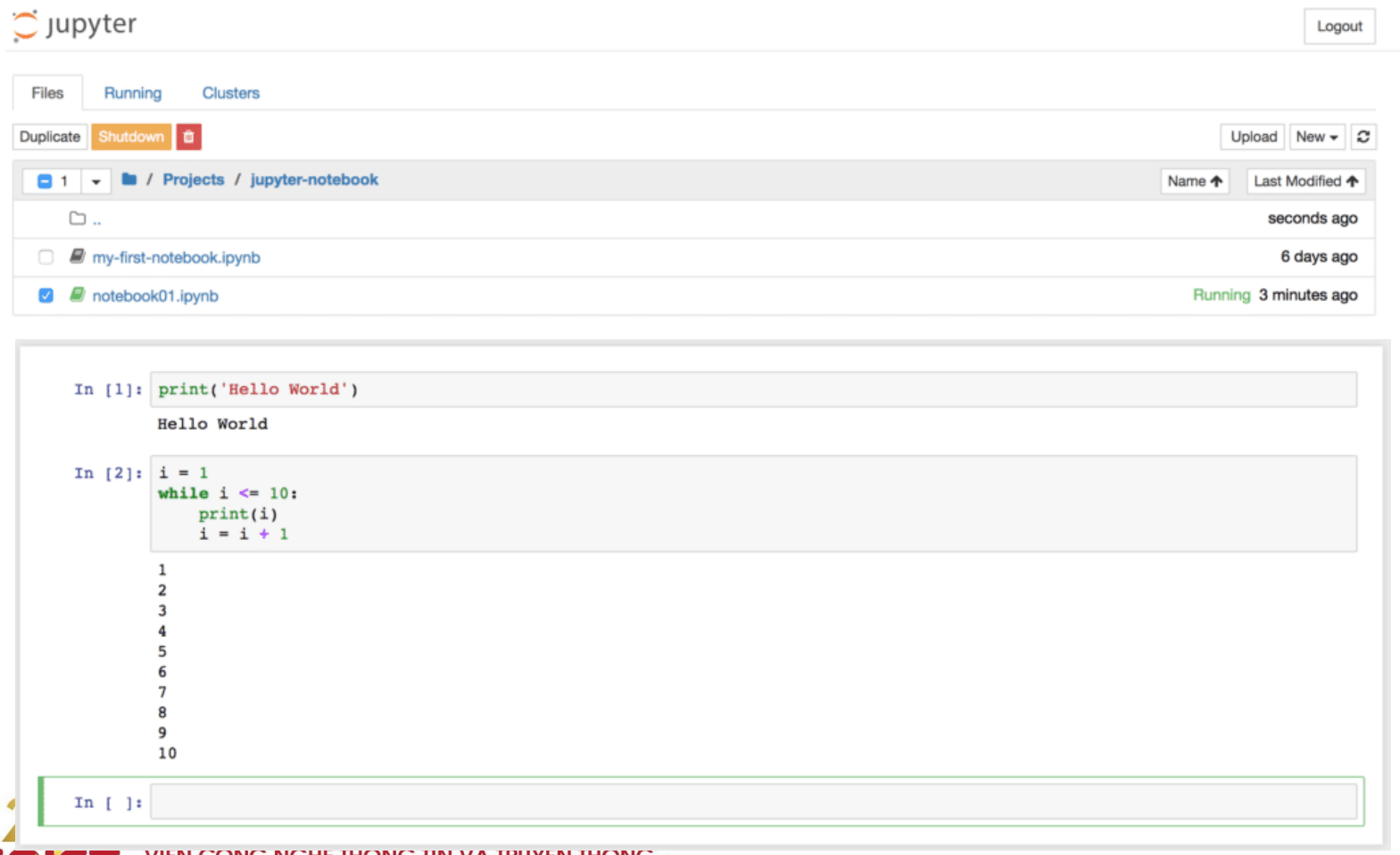
```
from google.colab import drive

drive.mount('/content/drive’)
```

- Lưu dữ liệu dưới dạng zip và unzip vào ổ cứng máy ảo Colab để tăng tốc độ xử lý dữ liệu:

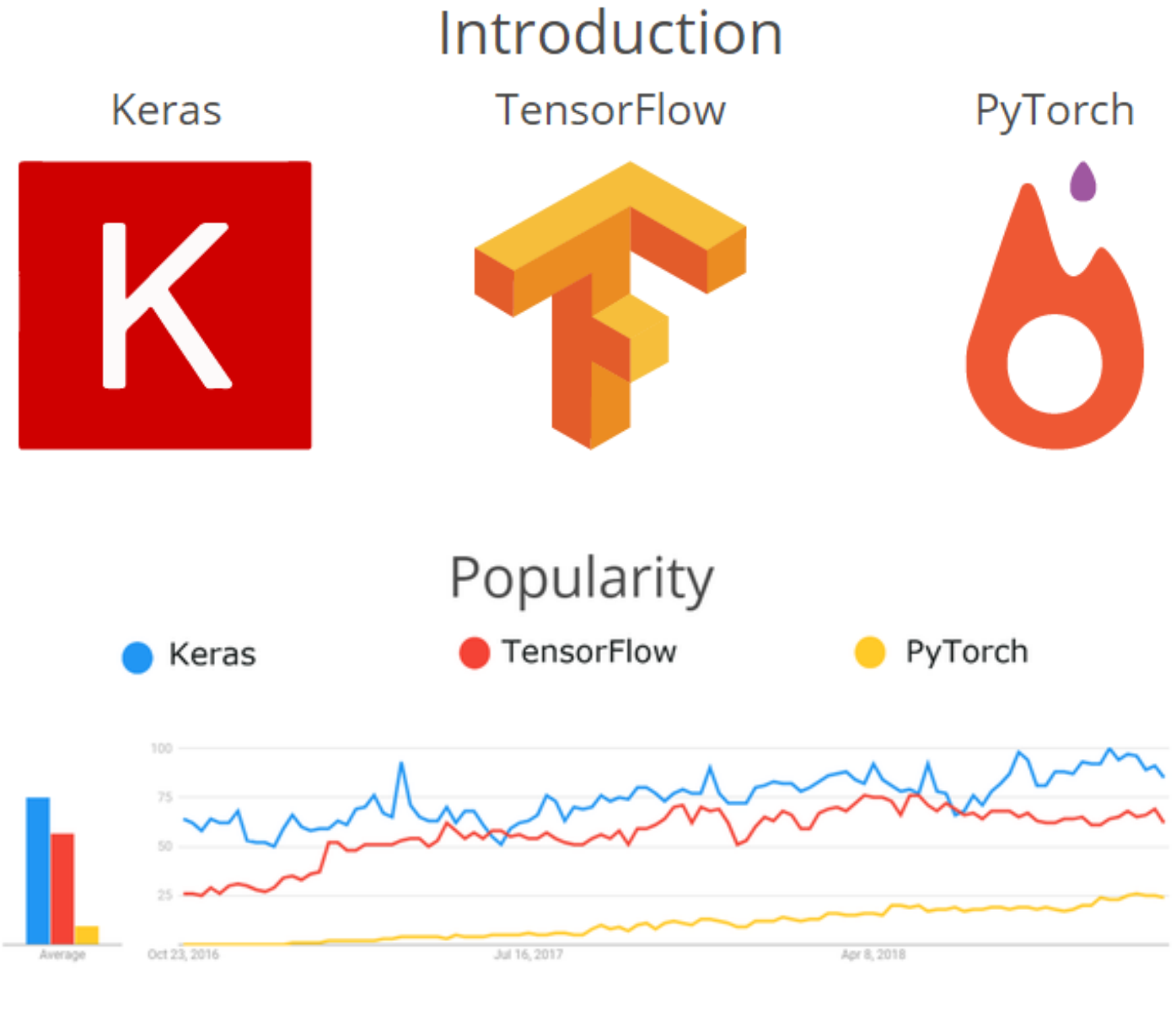
```
!unzip -uq "/content/drive/My Drive/Colab Notebooks/data.zip" -d "/content/"
```

Jupyter Notebook

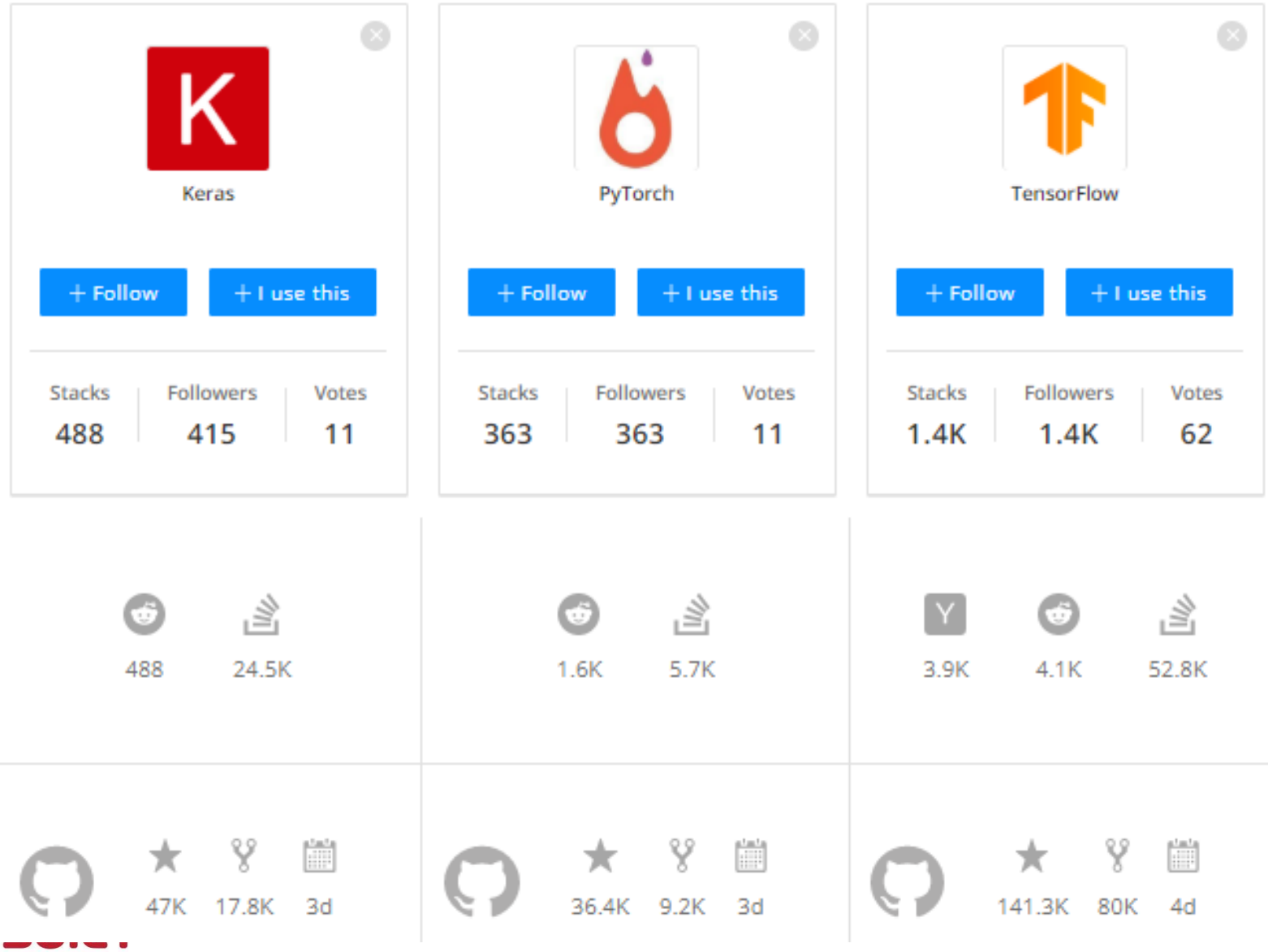
- Hướng dẫn sử dụng và cài đặt: https://www.dataquest.io/blog/jupyter-notebook-tutorial/



Tensorflow/Keras/PyTorch



Tensorflow/Keras/PyTorch



Tài liệu tham khảo

1. Khóa cs231n của Stanford: http://cs231n.stanford.edu

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

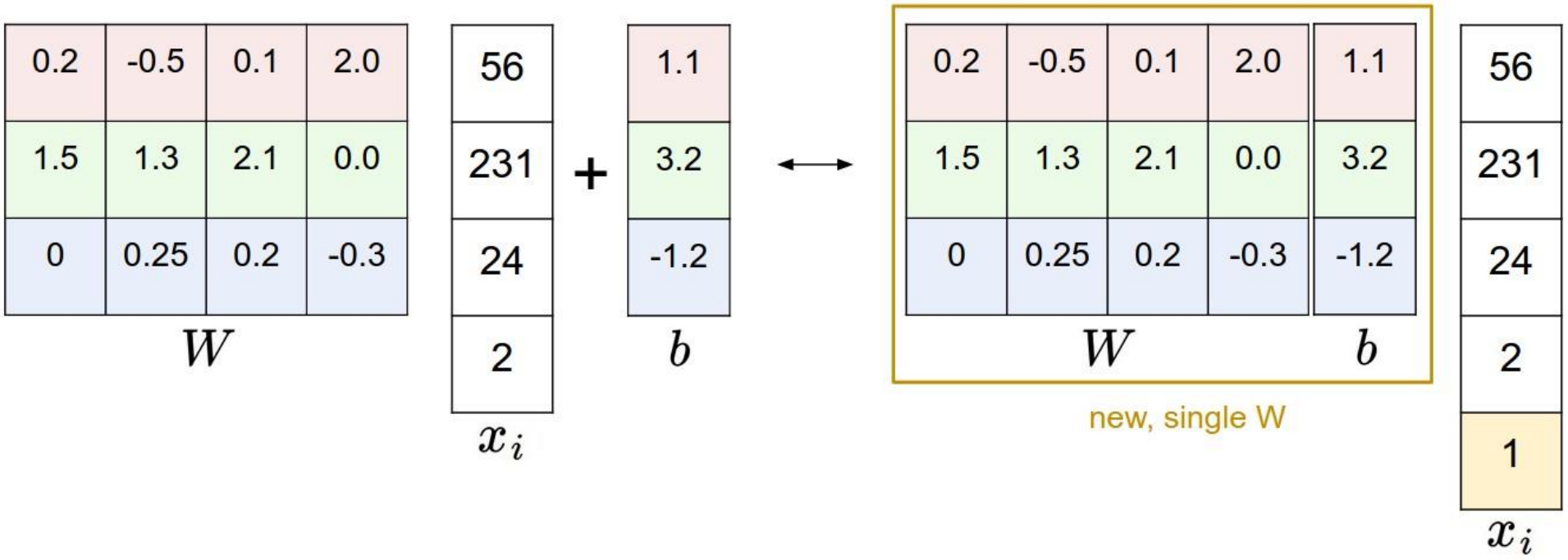
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict

Bias trick

- Before: $f(x_i,W,b)=Wx_i+b$
- After: $f(x_i,W)=Wx_i$



SVM vs. Softmax