



25
SOICT

YEARS ANNIVERSARY

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

ARTIFICIAL INTELLIGENCE

Any technique that enables computers to mimic human behavior



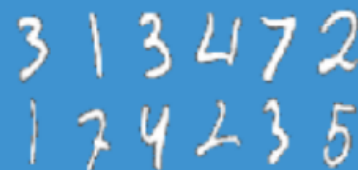
MACHINE LEARNING

Ability to learn without explicitly being programmed



DEEP LEARNING

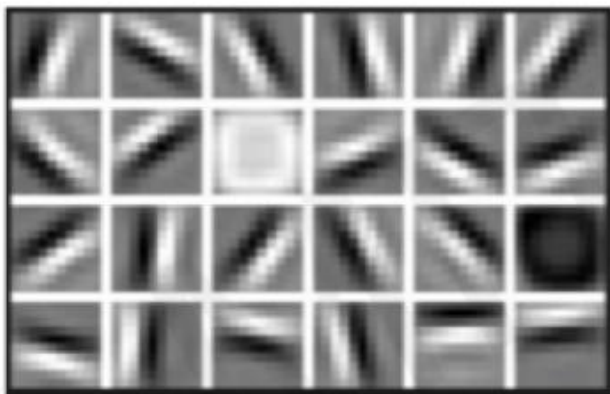
Extract patterns from data using neural networks



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

Low Level Features



Lines & Edges

Mid Level Features



Eyes & Nose & Ears

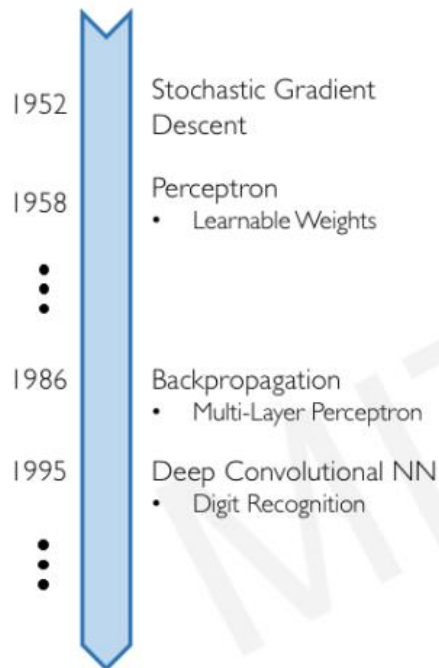
High Level Features



Facial Structure

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



2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable



3. Software

- Improved Techniques
- New Models
- Toolboxes



Học máy có giám sát

Functions \mathcal{F}

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

Training data

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

LEARNING

$$\begin{array}{l} \text{find } \hat{f} \in \mathcal{F} \\ \text{s.t. } y_i \approx \hat{f}(x_i) \end{array}$$



Learning machine

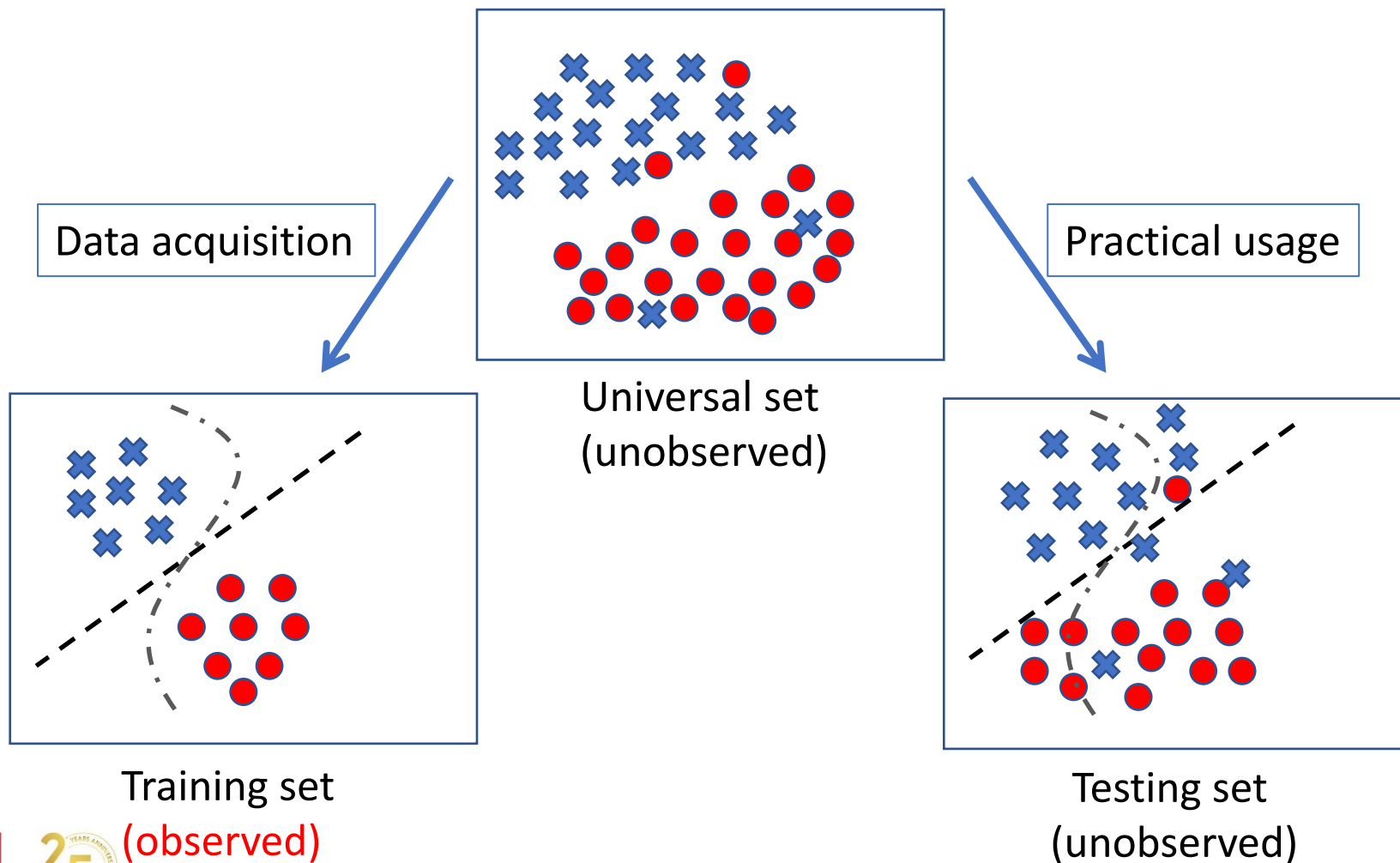
PREDICTION

$$y = \hat{f}(x)$$

New data

$$x$$

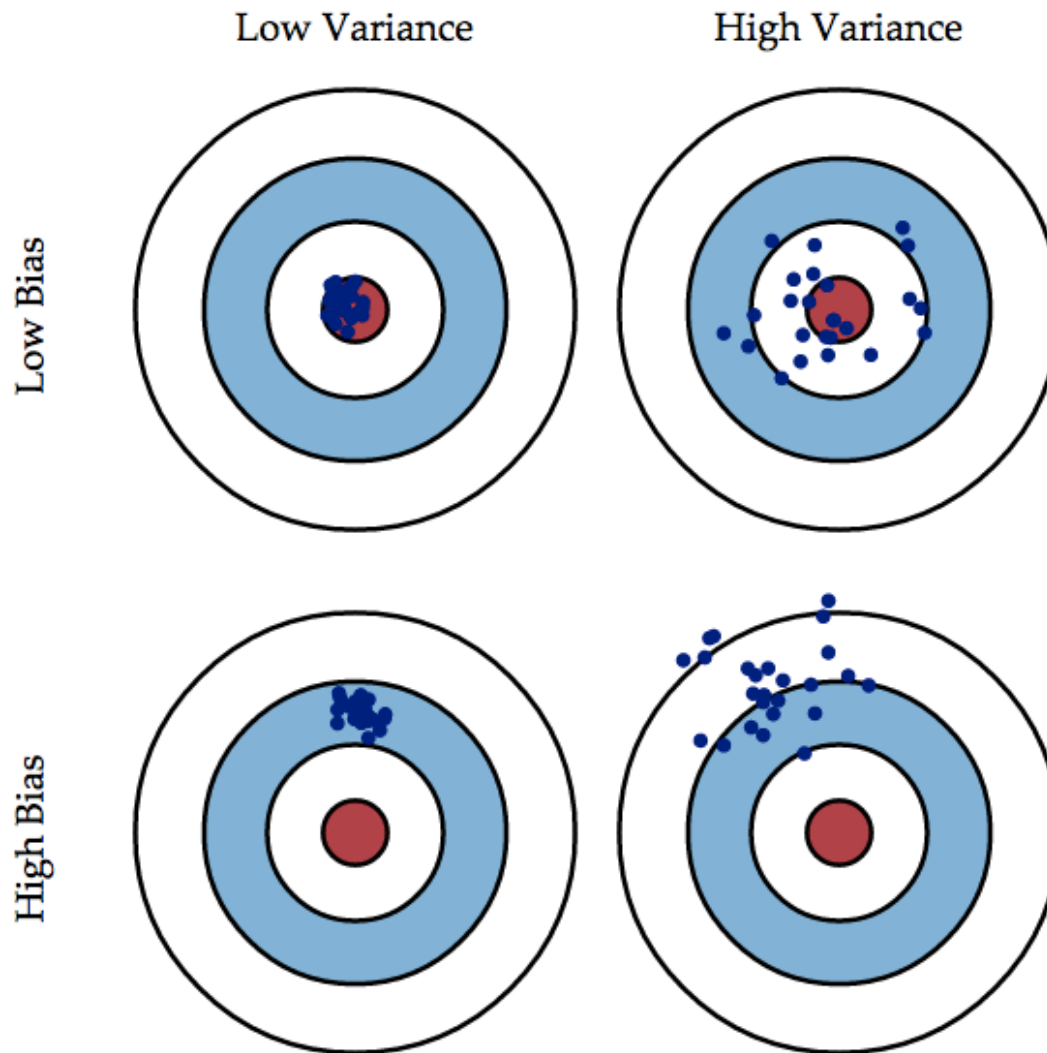
Tập huấn luyện và tập kiểm tra



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

Minh họa Bias-Variance



Phân lớp tuyến tính

airplane

automobile

bird

cat

deer

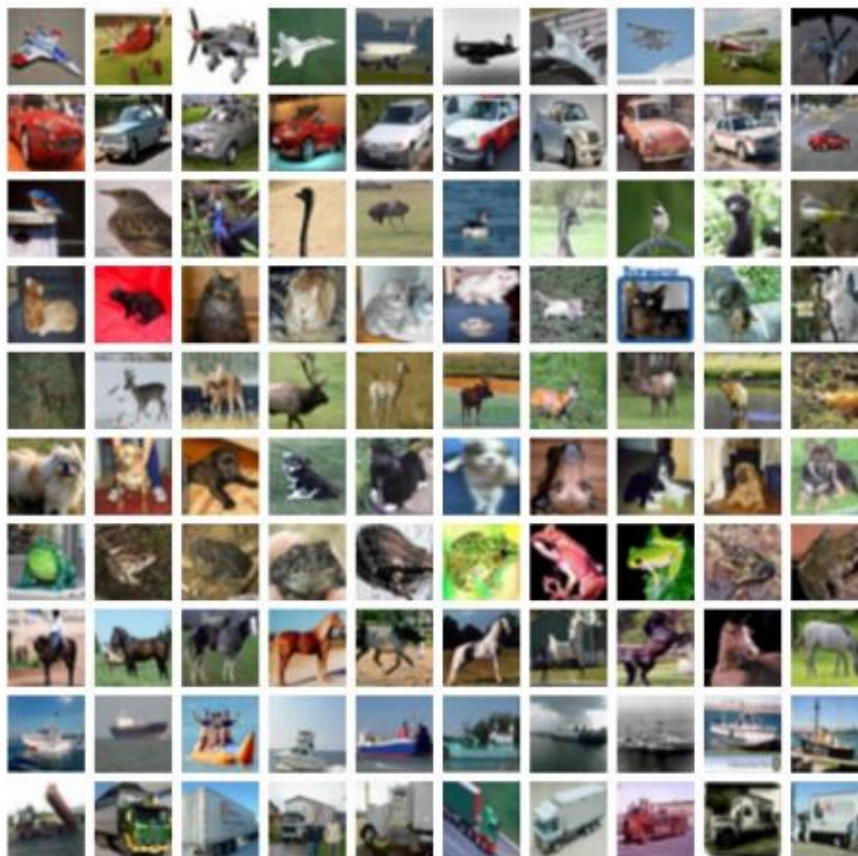
dog

frog

horse

ship

truck



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

10,000 test images.

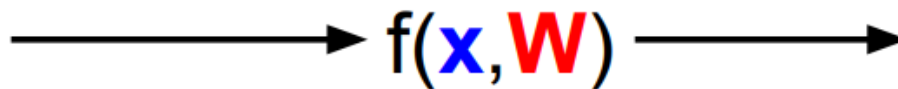
Phân lớp tuyến tính

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

Image



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



$f(x, W)$

10 numbers giving
class scores

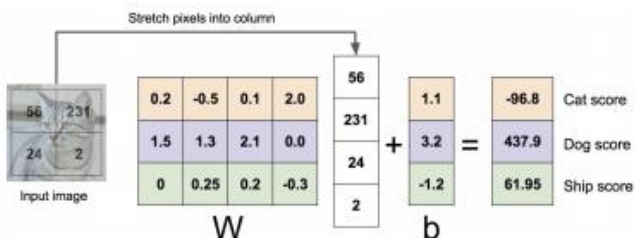
W

parameters
or weights

Phân lớp tuyến tính: 3 góc nhìn

Algebraic Viewpoint

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



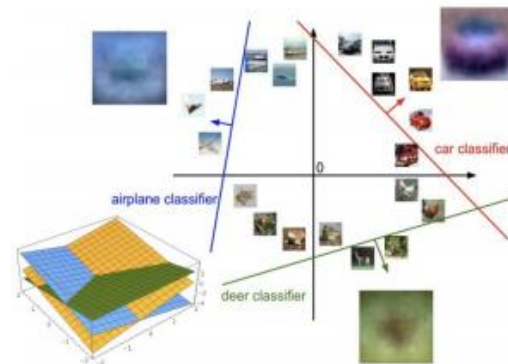
Visual Viewpoint

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:



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:

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

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

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**

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

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

Probabilities must be ≥ 0

Probabilities must sum to 1

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

cat
car
frog

3.2
5.1
-1.7

Unnormalized
log-probabilities / logits

exp

24.5
164.0
0.18

unnormalized
probabilities

normalize

0.13
0.87
0.00

probabilities

compare

Kullback–Leibler
divergence

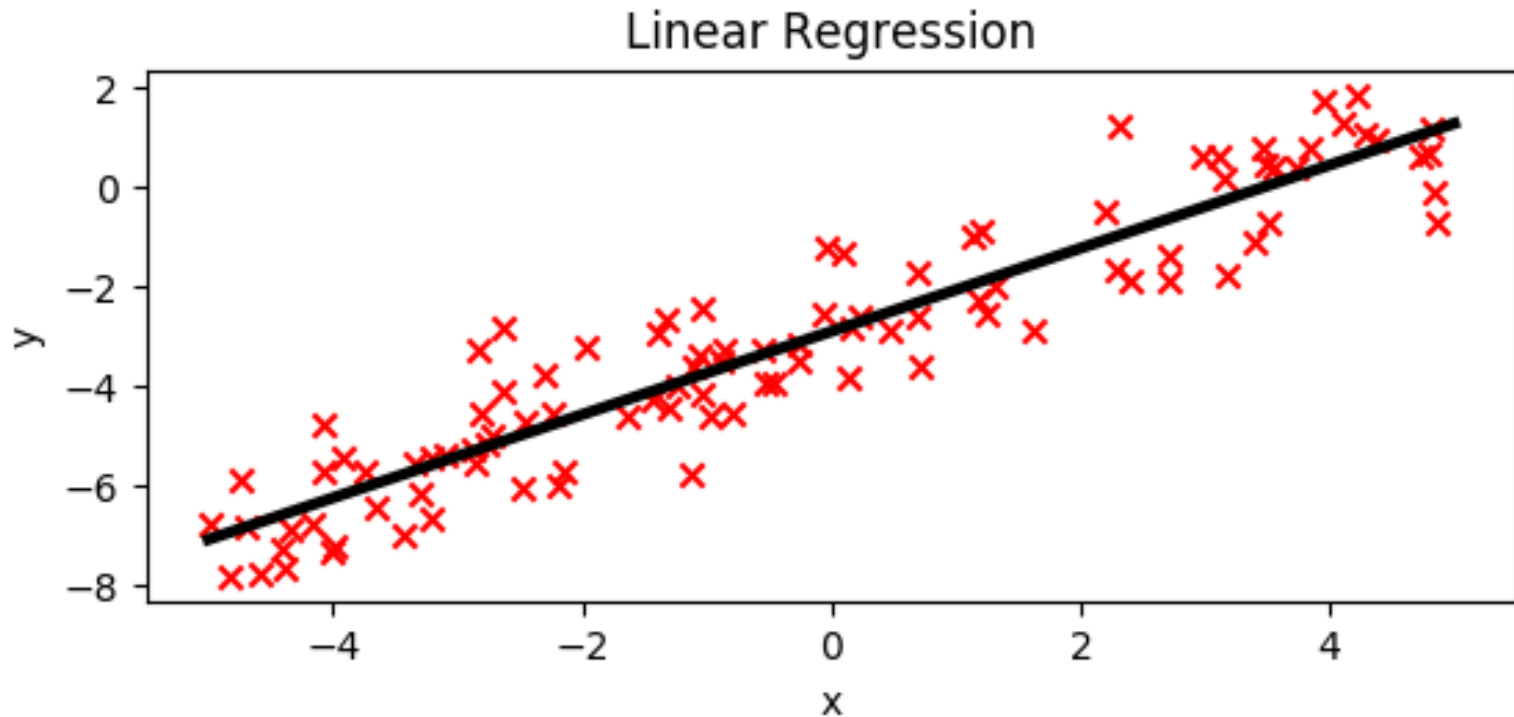
$$D_{KL}(P||Q) = \sum_y P(y) \log \frac{P(y)}{Q(y)}$$

1.00
0.00
0.00

Correct
probs

Hồi quy tuyến tính

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



Hồi quy tuyến tính

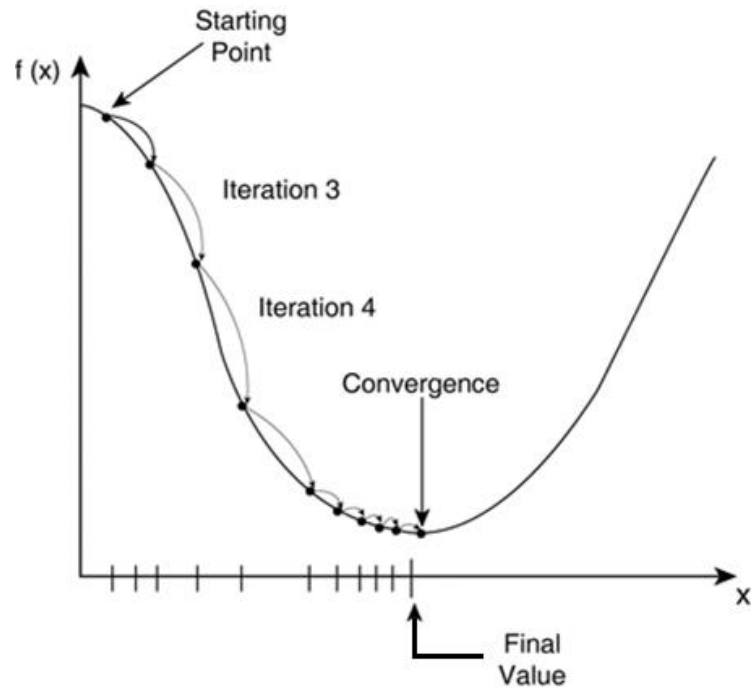
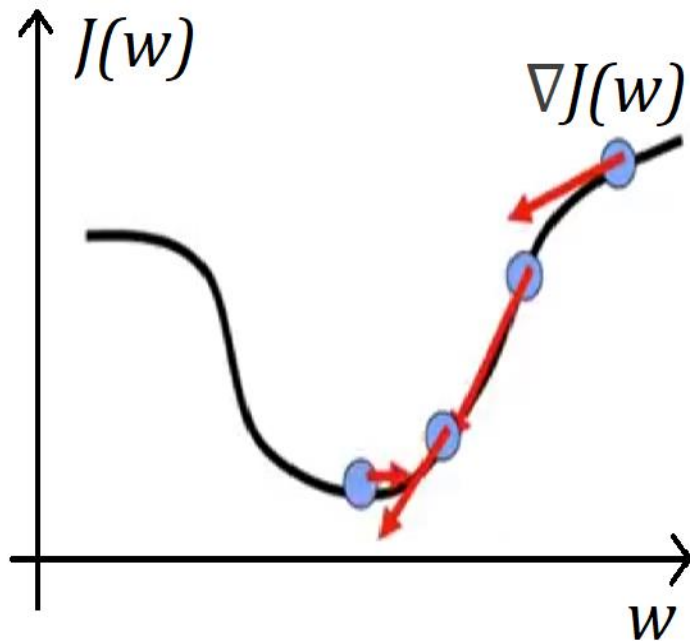
- Nên chọn hàm mục tiêu nào?
 - Mỗi $y^{(i)}$ là một số thực
 - Bình phương tối thiểu là một lựa chọn tốt ☺

$$\begin{aligned} J(w; \mathbf{X}, \mathbf{Y}) &= \frac{1}{N} \sum_{i=1}^N [f(x^{(i)}; w) - y^{(i)}]^2 \\ &= \frac{1}{N} \sum_{i=1}^N [w^T x^{(i)'} - y^{(i)}]^2 \\ &= \frac{1}{N} (w^T \mathbf{X}' - \mathbf{Y})^T (w^T \mathbf{X}' - \mathbf{Y}) \end{aligned}$$

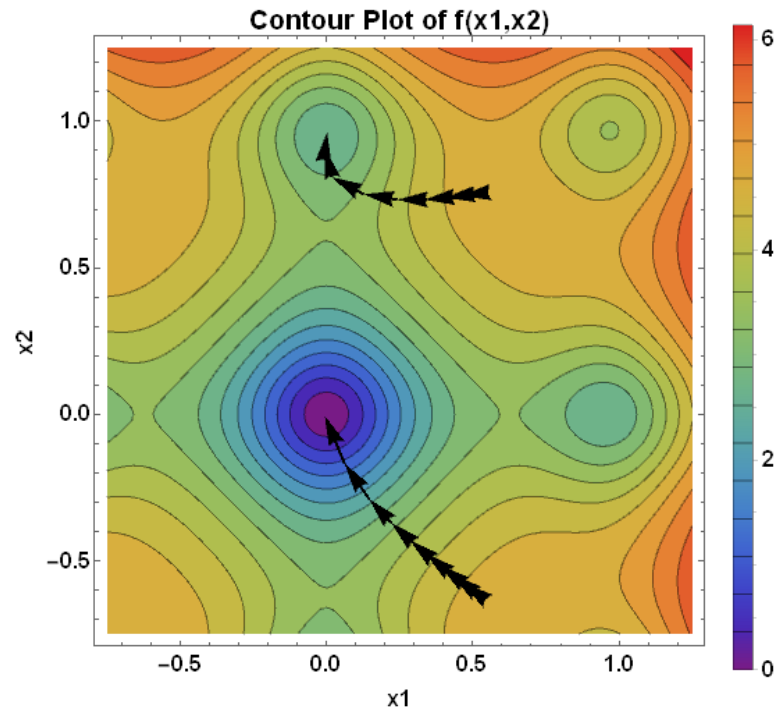
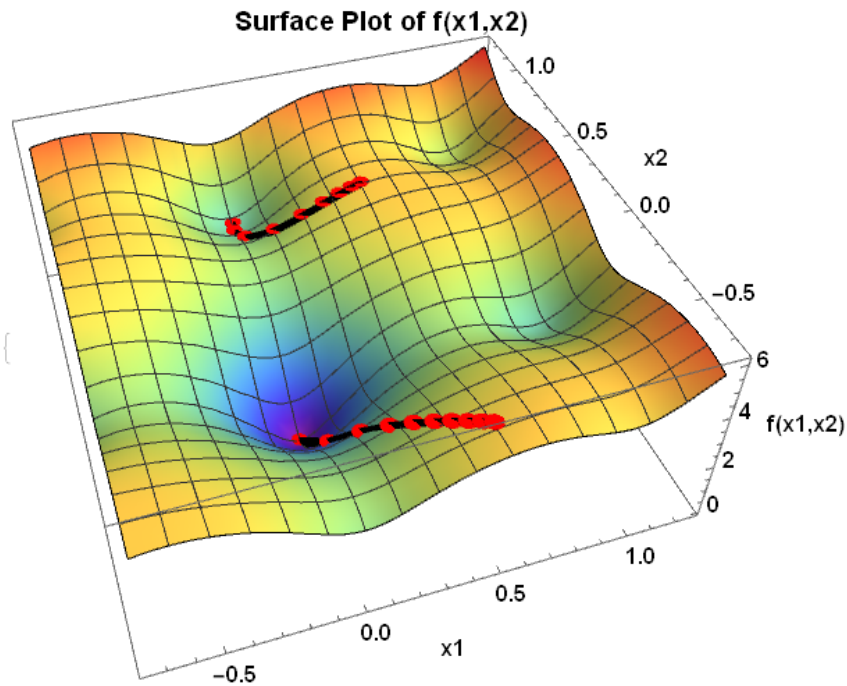
Tối ưu hàm mục tiêu



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



The screenshot shows the Google Colab interface. At the top, there's a header with the Colab logo, the notebook name 'unet.ipynb', and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are buttons for 'Comment', 'Share', a settings gear, and a user profile icon labeled 'D'. The main area shows a code cell with the command 'nvidia-smi' executed. The output is a timestamp 'Wed Feb 19 17:02:58 2020' followed by a table of GPU information.

```
!nvidia-smi
```

```
Wed Feb 19 17:02:58 2020
```

NVIDIA-SMI 440.48.02 Driver Version: 418.67 CUDA Version: 10.1							
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC	GPU-Util	Compute M.
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage			
0	Tesla P100-PCIE...	Off	00000000:00:04.0	Off		0	
N/A	38C	P0	28W / 250W	0MiB / 16280MiB		0%	Default

Processes:				GPU Memory Usage
GPU	PID	Type	Process name	
No running processes found				

Google Colab

- Thiết lập GPU: Edit / Notebook settings

Notebook settings

Runtime type

Python 3



Hardware accelerator

GPU

☐

Omit code cell output when saving this notebook

CANCEL

SAVE

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)
```

The screenshot shows the Google Colab web interface. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu, there's a toolbar with 'Code' and 'Text' tabs, and a 'Connect' button. The main area contains a terminal window with the command 'nvidia-smi' executed. The output shows GPU information for a Tesla P100-PCIE... card. Below the terminal, there's a section for 'Processes' which shows 'No running processes found'. On the right side, the Chrome DevTools Console is open, displaying a JavaScript function 'ClickConnect()' that logs 'Working' and clicks the 'connect' button in the Colab toolbar. This function is wrapped in a 'setInterval' call to run every 60,000 milliseconds. The console also shows some DevTools error messages related to SourceMap parsing.

File Edit View Insert Runtime Tools Help [Last edited on February 20](#)

+ Code + Text Connect Editing

!nvidia-smi

Wed Feb 19 17:02:58 2020

NVIDIA-SMI 440.48.02 Driver Version: 418.67 CUDA Version: 10.1									
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC				
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.			
0	Tesla P100-PCIE...	Off	00000000:00:04:0	Off	0				
N/A	38C	P0	28W / 250W	0MiB / 16280MiB	0%	Default			

Processes:

GPU	PID	Type	Process name	GPU Memory Usage
No running processes found				

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')
```

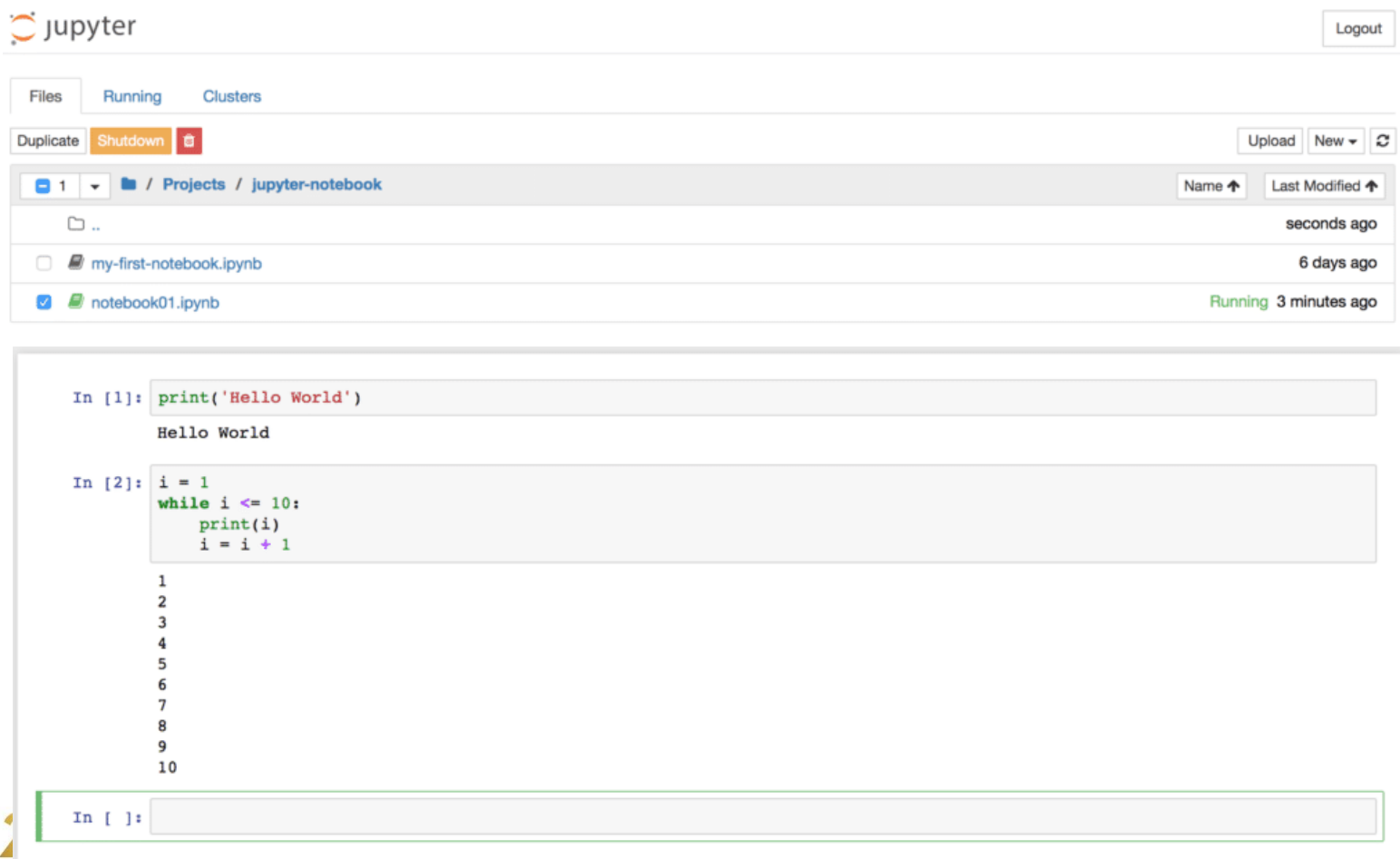
- Lưu dữ liệu dưới dạng zip và unzip vào ổ cứng máy ảo Colab để tăng tốc độ xử lý dữ liệu:

```
!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/>



The screenshot displays the Jupyter Notebook web interface. At the top, there is a 'jupyter' logo and a 'Logout' button. Below this, a navigation bar includes 'Files', 'Running', and 'Clusters' tabs. A secondary bar contains 'Duplicate', 'Shutdown', and a trash icon, along with 'Upload', 'New', and a refresh icon. The main area is a file browser for the 'Projects / jupyter-notebook' directory. It lists three items: '..' (parent directory), 'my-first-notebook.ipynb' (6 days ago), and 'notebook01.ipynb' (Running 3 minutes ago). The 'notebook01.ipynb' file is selected, and its content is displayed in a code editor. The code consists of two input cells: the first prints 'Hello World', and the second is a while loop that prints numbers from 1 to 10. The output of the first cell is 'Hello World', and the output of the second cell is the numbers 1 through 10, each on a new line. At the bottom, there is an empty input cell labeled 'In []:'.

jupyter Logout

Files Running Clusters

Duplicate Shutdown

Upload New

1 / Projects / jupyter-notebook

Name Last Modified

.. seconds ago

my-first-notebook.ipynb 6 days ago

notebook01.ipynb Running 3 minutes ago

```
In [1]: print('Hello World')
Hello World

In [2]: i = 1
while i <= 10:
    print(i)
    i = i + 1

1
2
3
4
5
6
7
8
9
10

In [ ]:
```

Tensorflow/Keras/PyTorch

Introduction

Keras



TensorFlow



PyTorch

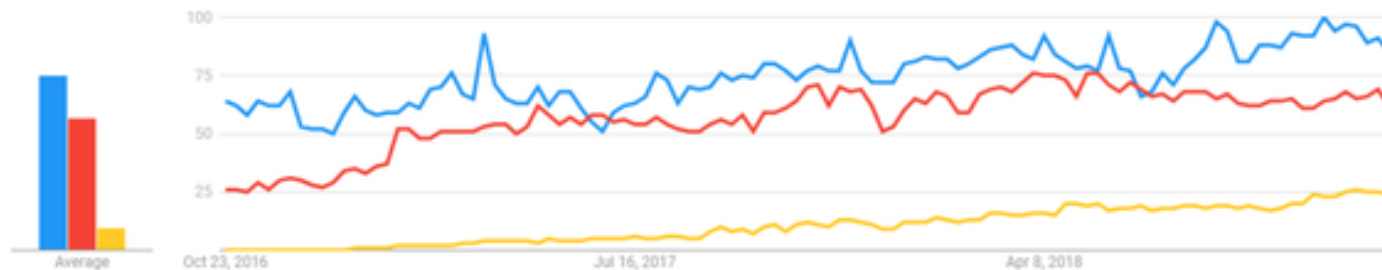


Popularity



● Keras

● TensorFlow

● PyTorch





Tensorflow/Keras/PyTorch



Keras

[+ Follow](#) [+ I use this](#)



Stacks	Followers	Votes
488	415	11



PyTorch

[+ Follow](#) [+ I use this](#)

Stacks	Followers	Votes
363	363	11



TensorFlow

[+ Follow](#) [+ I use this](#)

Stacks	Followers	Votes
1.4K	1.4K	62

 
488 24.5K

 
1.6K 5.7K

  
3.9K 4.1K 52.8K

Tài liệu tham khảo

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



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/

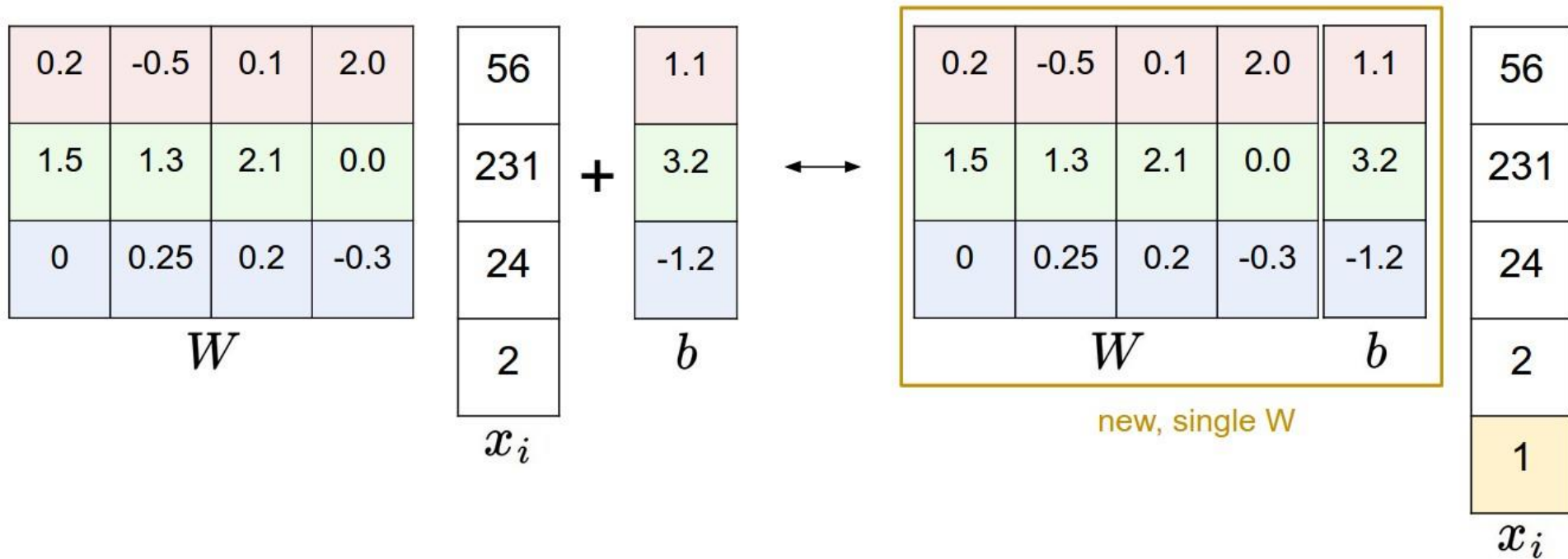


fb.com/groups/soict



Bias trick

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



SVM vs. Softmax

matrix multiply + bias offset

0.01	-0.05	0.1	0.05
0.7	0.2	0.05	0.16
0.0	-0.45	-0.2	0.03

W

-15
22
-44
56

x_i

+

0.0
0.2
-0.3

b

y_i 2

hinge loss (SVM)

-2.85
0.86
0.28

$$\begin{aligned} &\max(0, -2.85 - 0.28 + 1) + \\ &\max(0, 0.86 - 0.28 + 1) \\ &= \\ &\mathbf{1.58} \end{aligned}$$

cross-entropy loss (Softmax)

-2.85
0.86
0.28

\exp

0.058
2.36
1.32

normalize
(to sum to one)

0.016
0.631
0.353

$$\begin{aligned} &-\log(0.353) \\ &= \\ &\mathbf{1.04} \end{aligned}$$