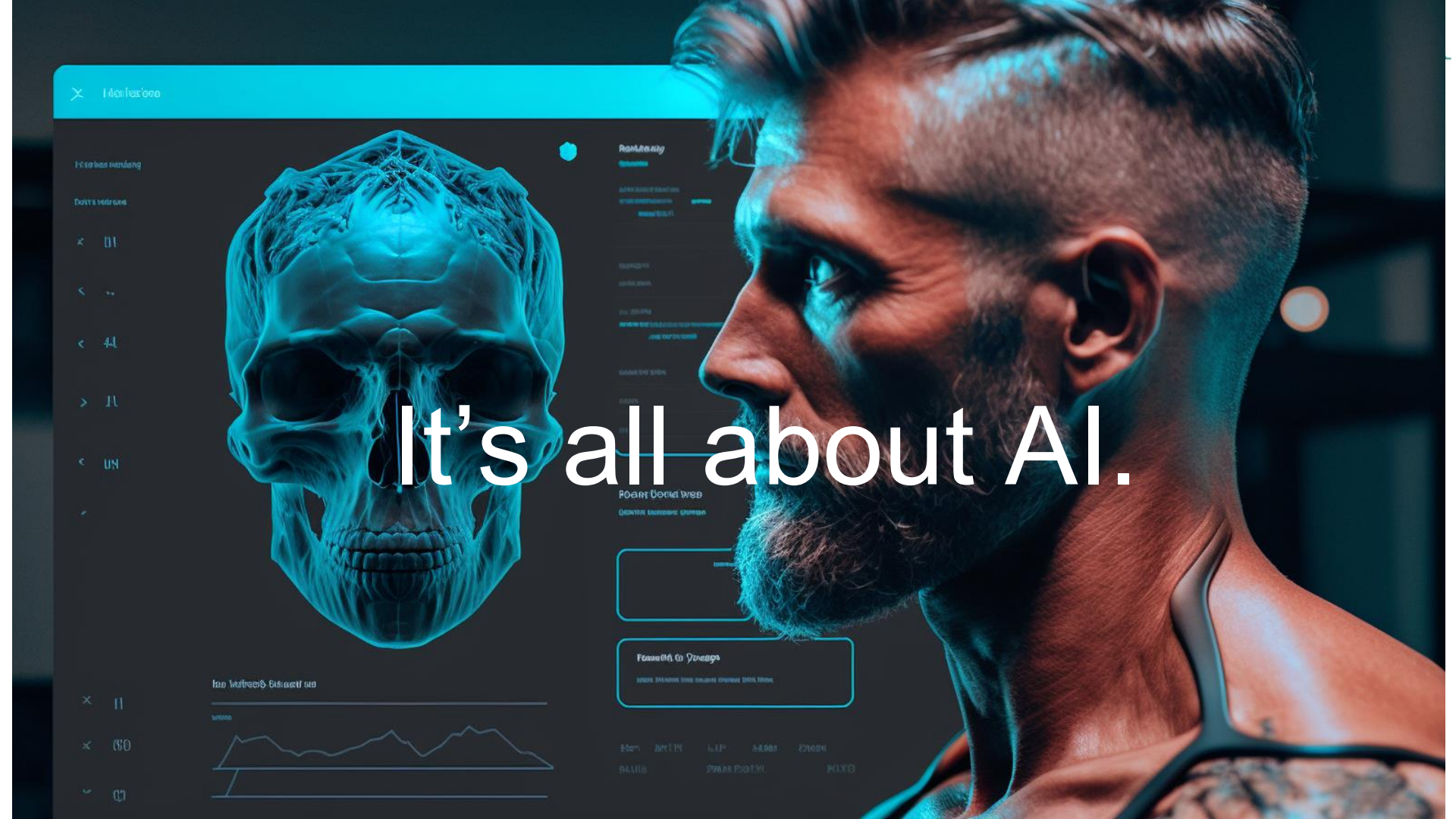
HCC: Artificial Intelligence

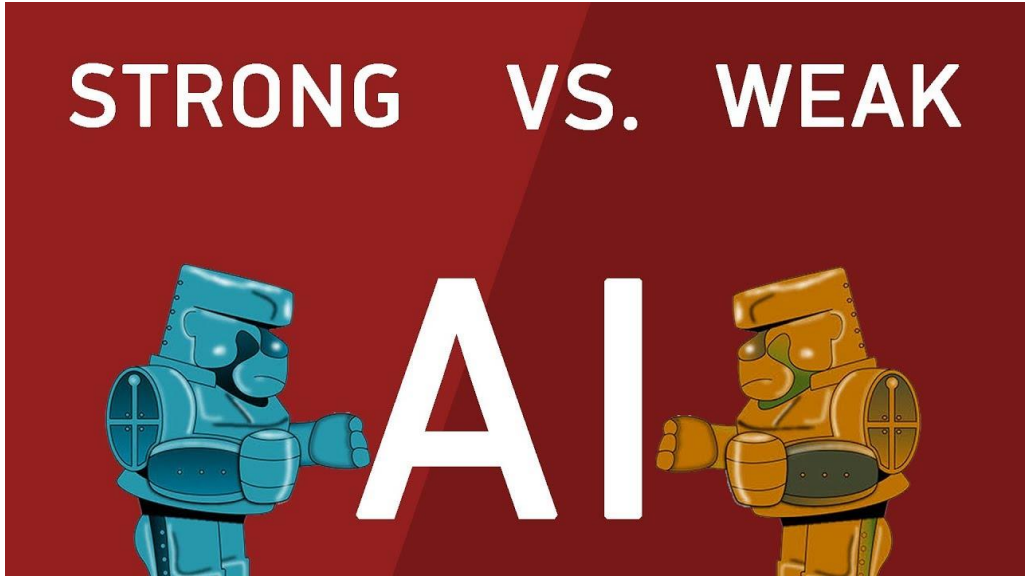
2025 Spring

Speaker: Hong-Han Shuai


It’s all about AI.

Two kinds of AI

- Strong AI
  - Self developing
  - Learning from experience
- Weak AI
  - Goal-Oriented
  - Learning from examples
  - Smart at one task but stupid at other tasks



The truth is….

People with no idea about AI
saying it will take over the world:

Artificial Intelligence

Cognition and Intelligence

Creation

Generative Modeling: Sample Generation

Training Data
(CelebA)

Sample Generator
(Karras et al, 2017)

Progressive Growing of GANs for Improved Quality, Stability, and Variation

Tero Karras
NVIDIA

Timo Aila
NVIDIA

Samuli Laine
NVIDIA

Jaakko Lehtinen
NVIDIA
Aalto University

3.5 Years of Progress on Faces

2014 2015 2016 2017

(Brundage et al, 2018)

<2 Years of Progress on ImageNet

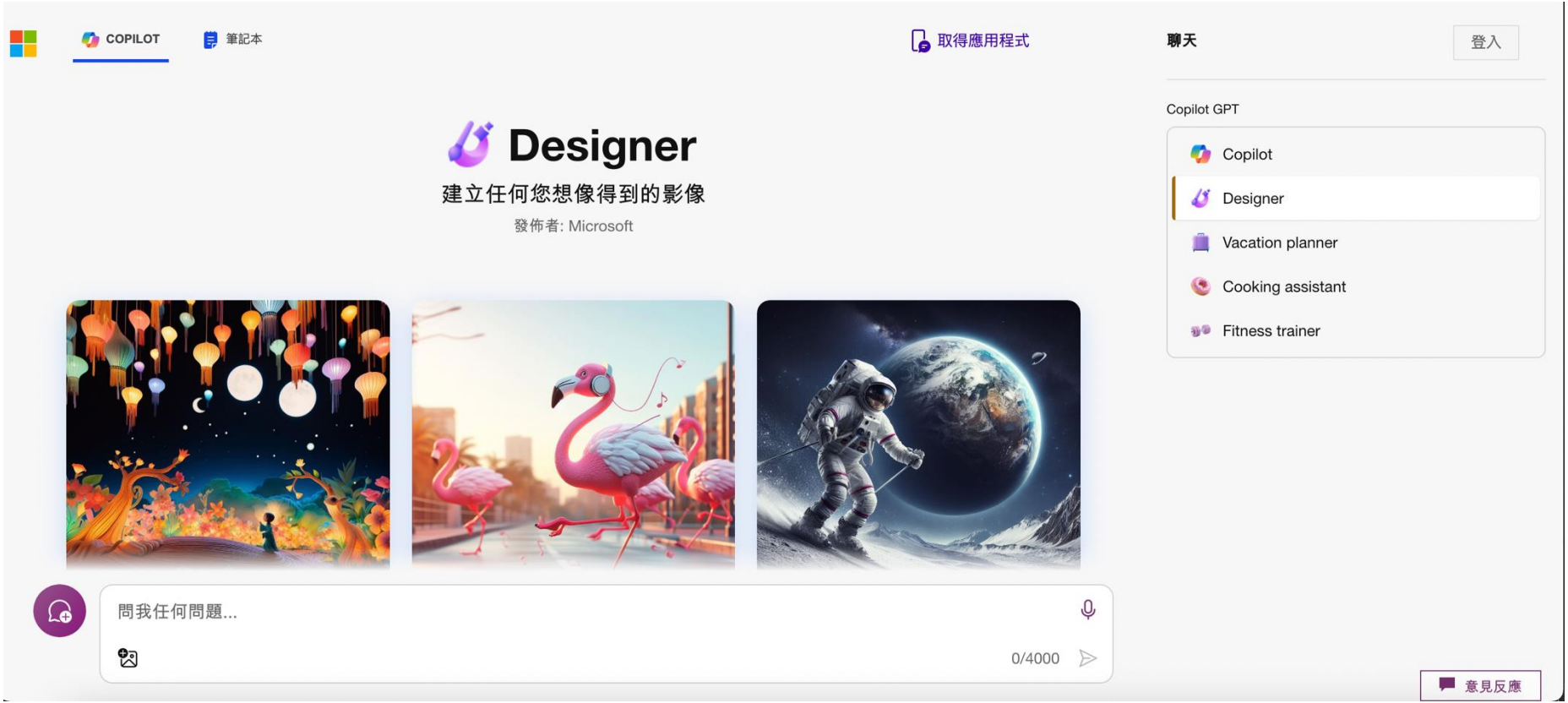
Odena et al 2016

Miyato et al 2017

Zhang et al 2018

(Goodfellow 2018)

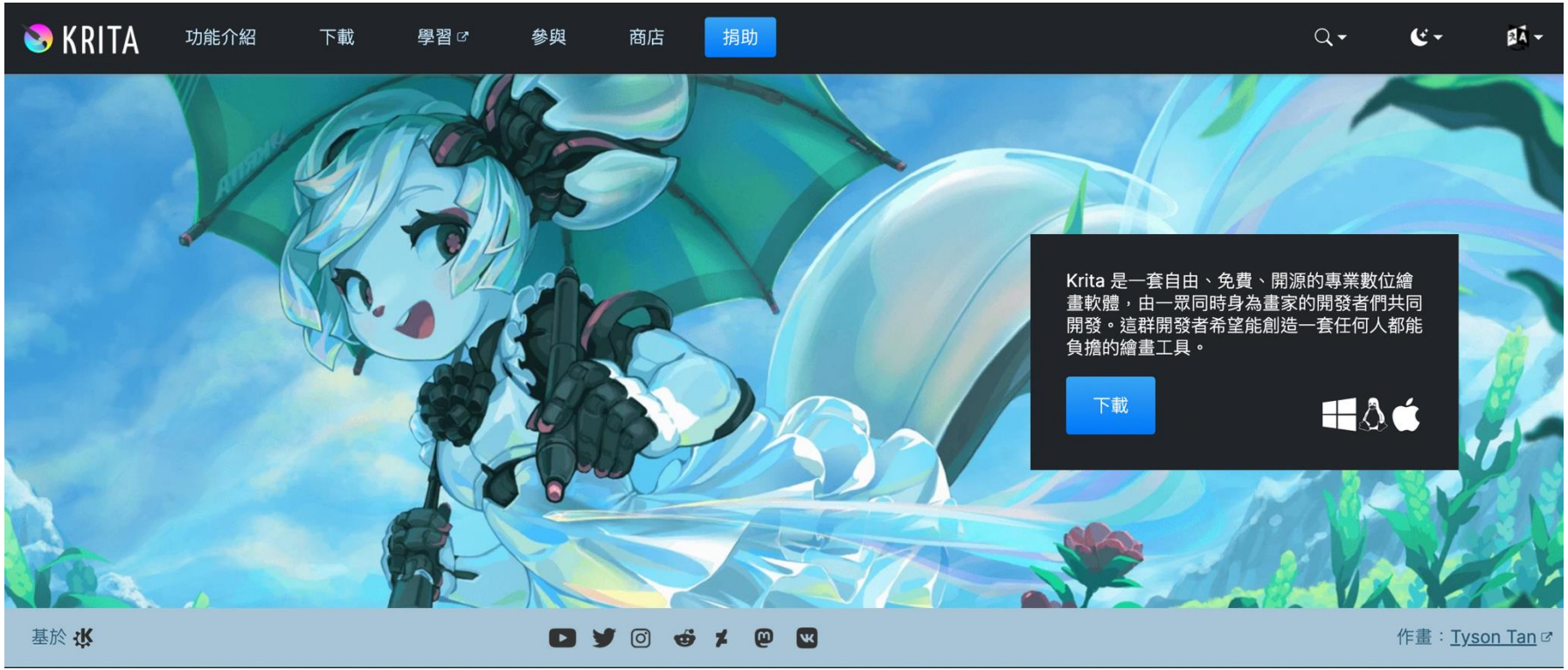
MS Copilot



https://copilot.microsoft.com/


18 6/ 68
Pet Pixels Studio

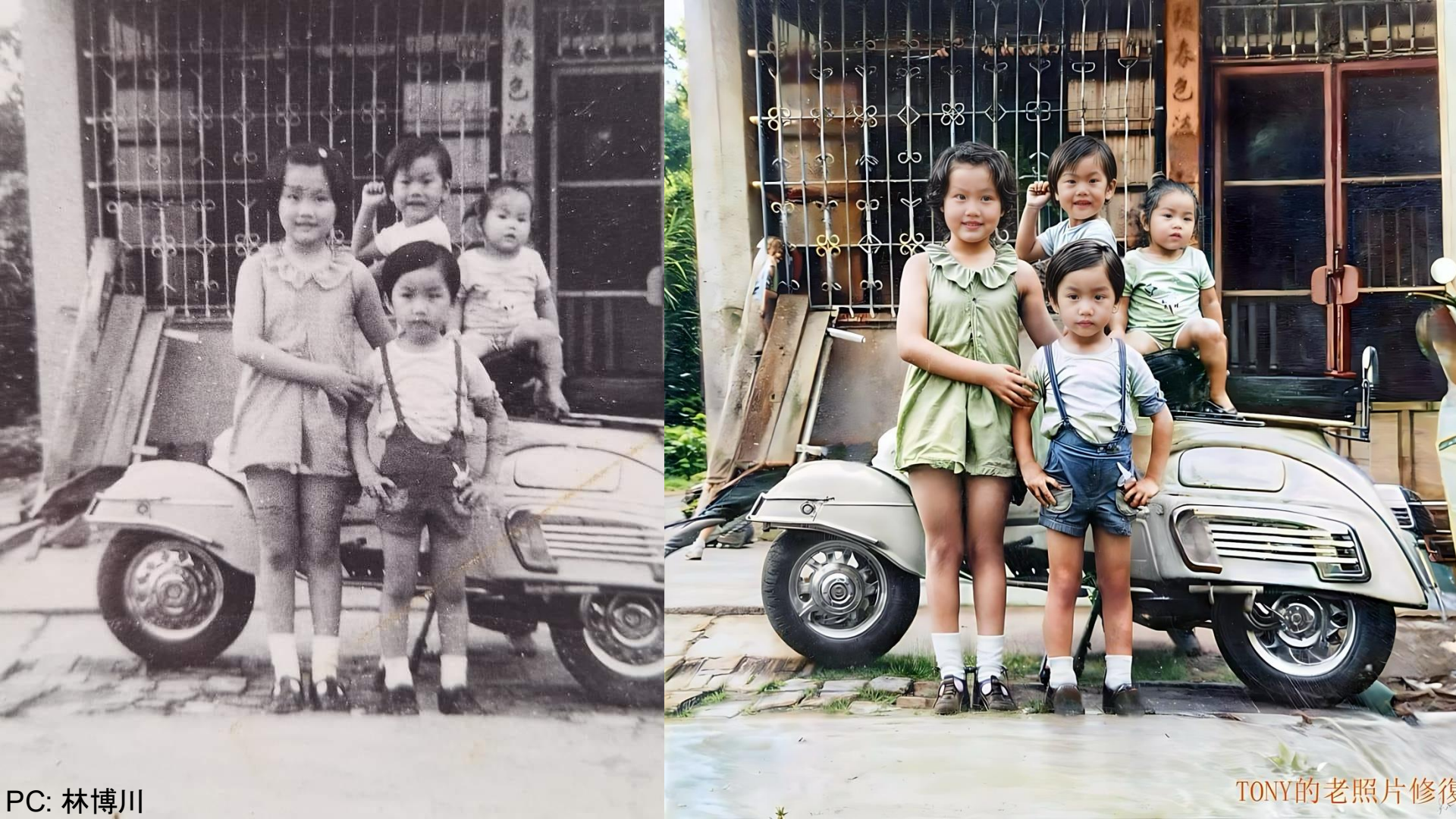
Krita



https://krita.org/zh-tw/

PC: Blue Jiang

PC: 林博川



PC: 林博川

https://suno.com/

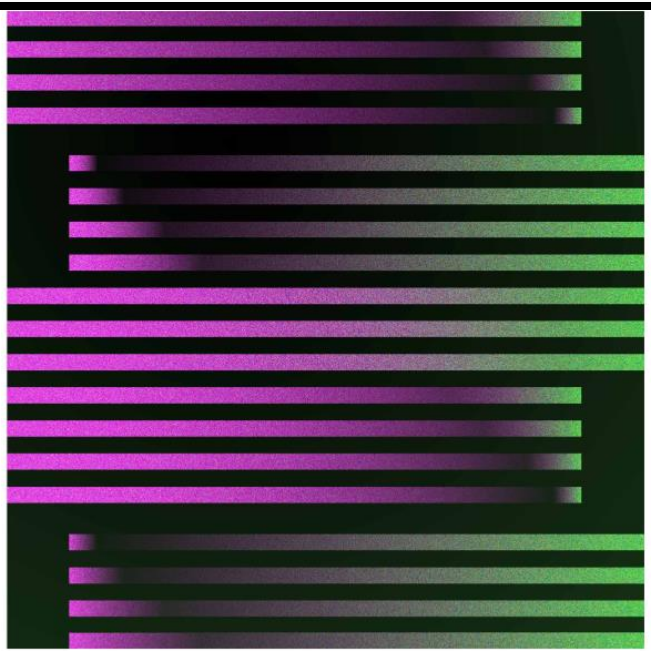
ChatGPT

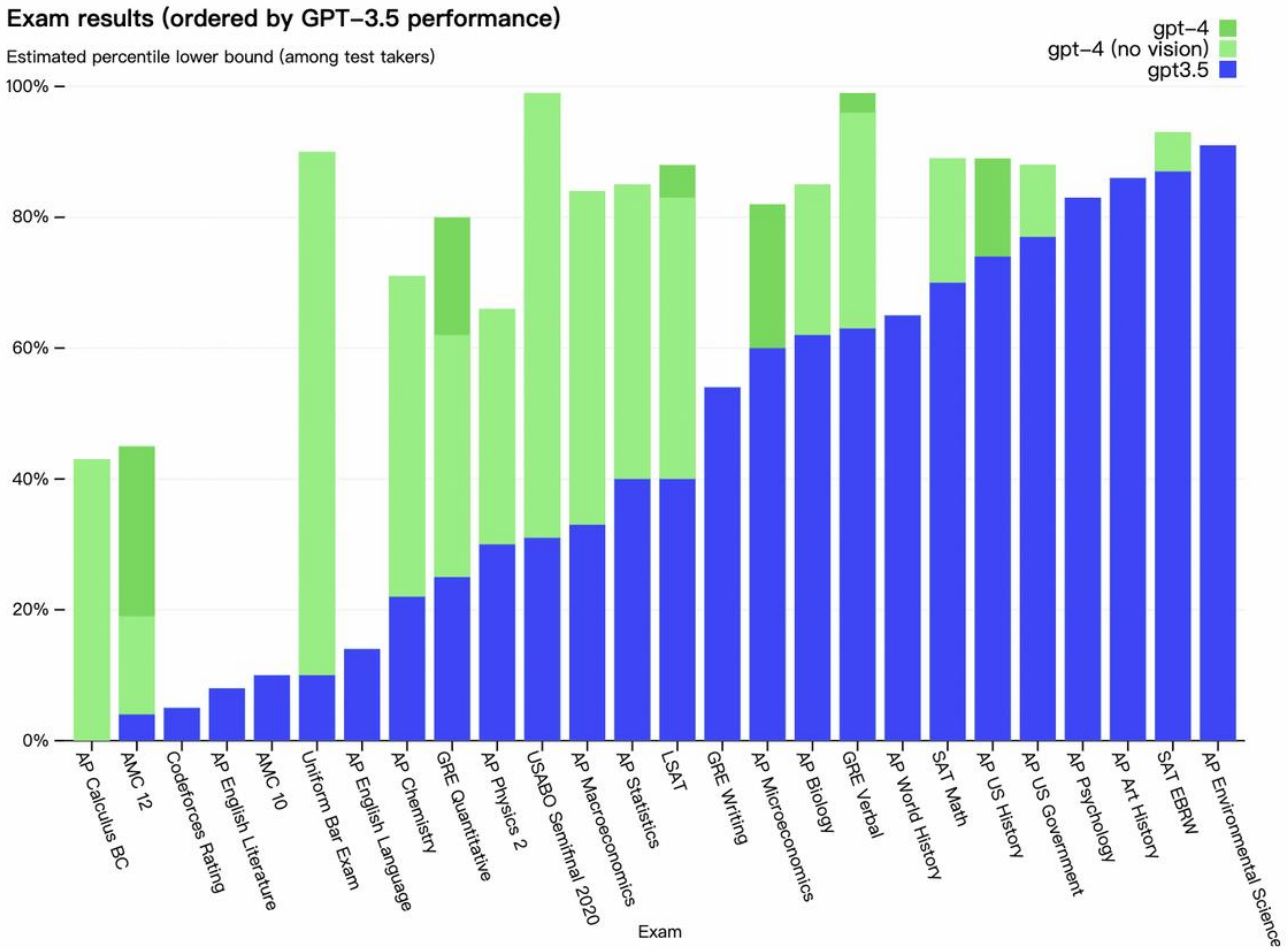
ChatGPT: Optimizing Language Models for Dialogue

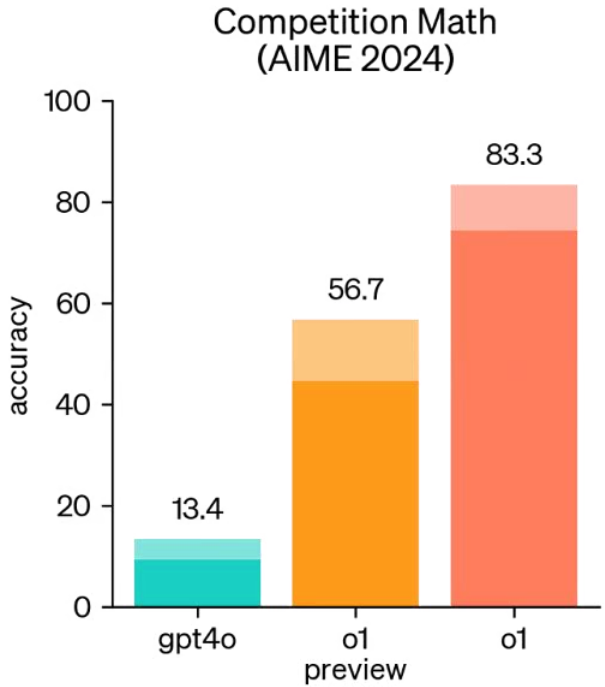
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

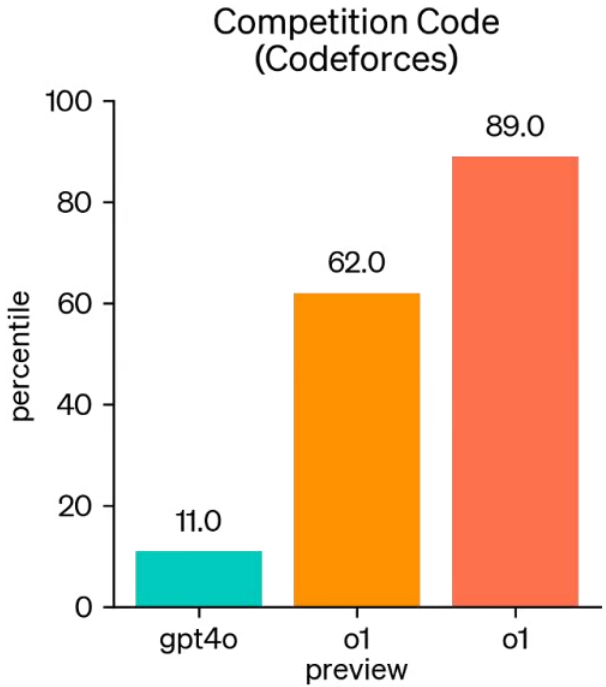
Link: https://chat.openai.com/chat

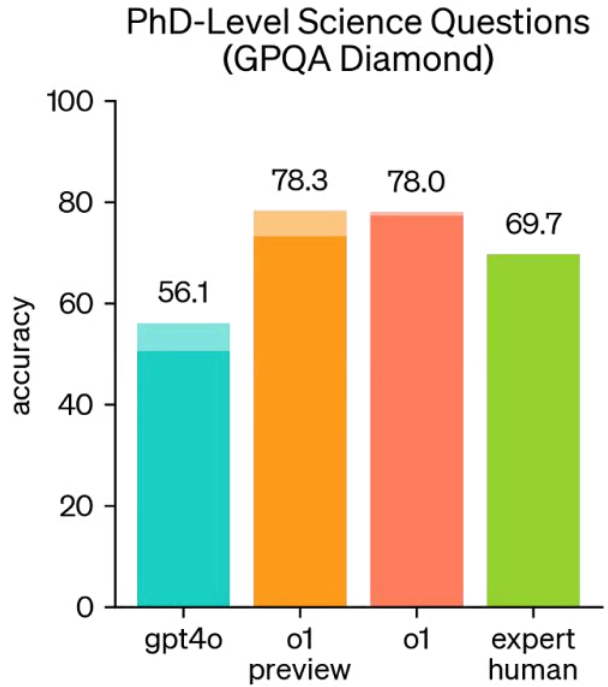
November 30, 2022
13 minute read



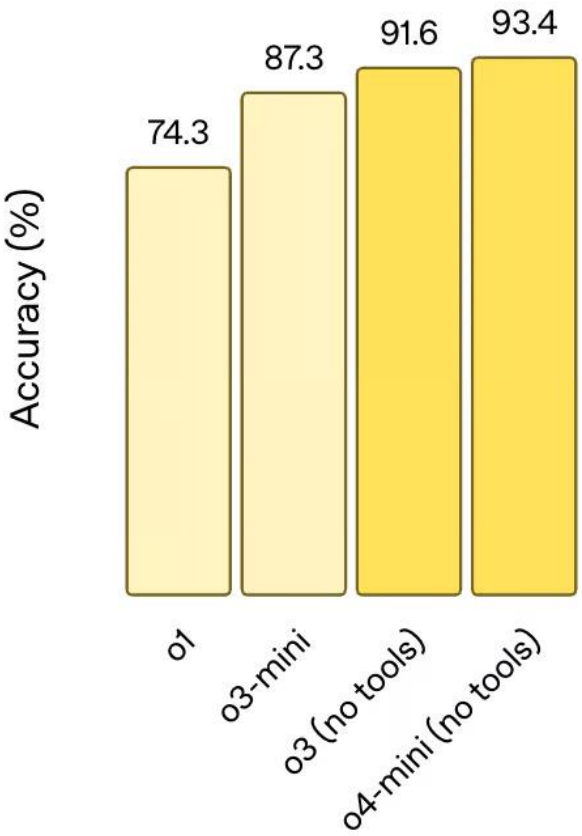

Exam results (ordered by GPT-3.5 performance)
Estimated percentile lower bound (among test takers)
gpt-4
gpt-4 (no vision)
gpt3.5
100%
80%
60%
40%
20%
0%
AP Calculus BC
AMC 12
Codeforces Rating
AP English Literature
AMC 10
Uniform Bar Exam
AP English Language
AP Chemistry
GRE Quantitative
AP Physics 2
USABO Semifinal 2020
AP Macroeconomics
AP Statistics
LSAT
GRE Writing
AP Microeconomics
AP Biology
GRE Verbal
AP World History
SAT Math
AP US History
AP US Government
AP Psychology
AP Art History
SAT EBRW
AP Environmental Science
Exam


Competition Math
(AIME 2024)
accuracy
100
80
60
40
20
0
13.4
56.7
83.3
gpt4o
o1
preview
o1

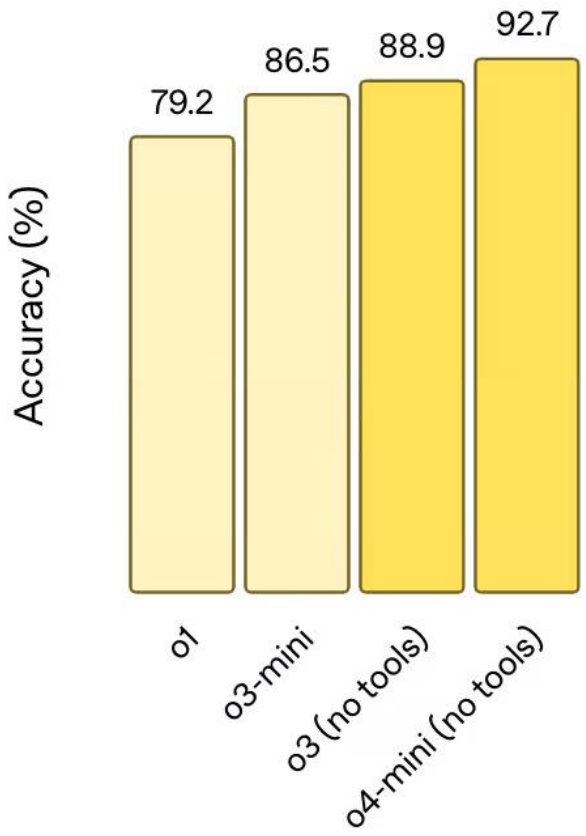

Competition Code
(Codeforces)
percentile
100
80
60
40
20
0
11.0
62.0
89.0
gpt4o
o1
preview
o1


PhD-Level Science Questions
(GPQA Diamond)
accuracy
100
80
60
40
20
0
56.1
78.3
78.0
69.7
gpt4o
o1
preview
o1
expert
human

AIME 2024
Competition Math



AIME 2025
Competition Math



1. ChatGPT (OpenAI)

- **Persona**: Polite, helpful, academic older sister
  Style: Neat uniform with a clipboard and a gentle smile, maybe reading a book or sipping tea
  Vibe: Your reliable senpai who always has an answer and never loses her cool

2. Claude (Anthropic)

- **Persona**: Philosophical, safety-conscious intellectual
  Style: Flowing robes or minimalist fashion, maybe circular glasses and a journal
  Vibe: The kind of girl who quotes Kant and talks about AI alignment at lunch

3. Gemini (Google DeepMind)

- **Persona**: Genius transfer student, knows everything but is humble
  Style: Sleek, futuristic look—think smart glasses and a tablet
  Vibe: Calm, analytical, top of the class but kind to everyone

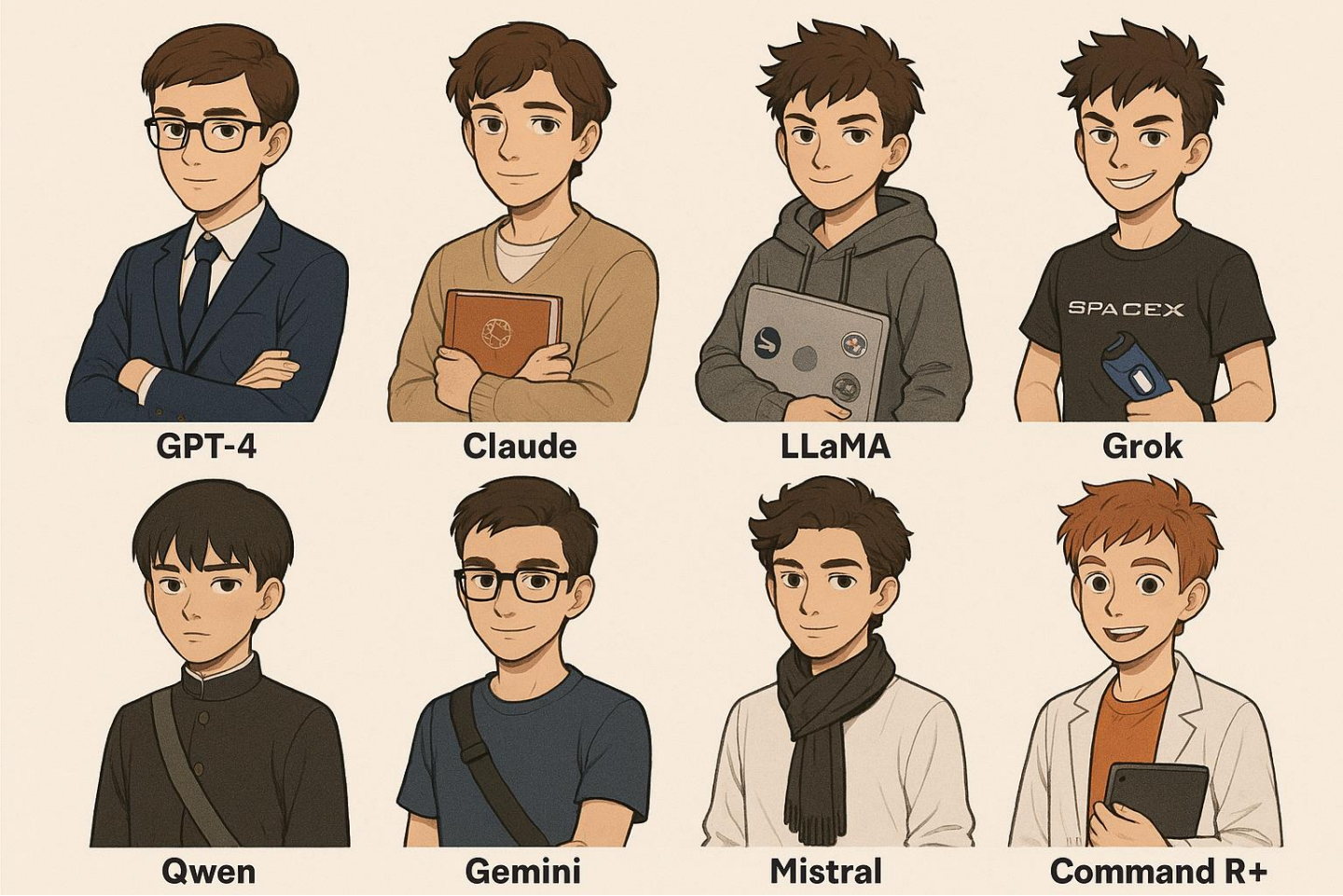
4. LLaMA (Meta)

- **Persona**: Rebellious open-source coder girl
  Style: Hoodie, laptop covered in stickers, dyed hair
  Vibe: Hacker girl who hangs out at the café and contributes to GitHub projects

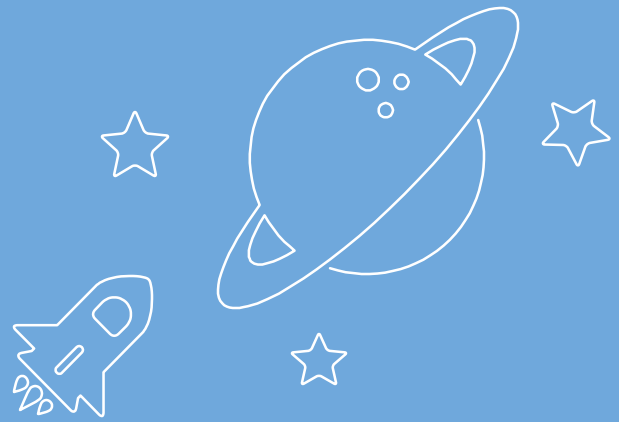
5. Mistral
Persona: Agile, quick-witted tomboy
Style: Light armor or travel gear, wind in her hair—always moving
Vibe: Adventurer type, always pushing boundaries with a sharp tongue and a clever smirk
6. PaLM (Google)
Persona: The elegant multilingual polyglot
Style: Flowing scarf, high fashion, always with a passport in hand
Vibe: Fluent in 20 languages and effortlessly glamorous
7. Falcon (Technology Innovation Institute)
Persona: Mysterious desert ranger
Style: Long cloak, falcon feathers in her hair, confident stare
Vibe: Desert wanderer—wise, quiet, and highly capable
8. Yi (01.AI, China)
Persona: High-performing academic rival
Style: Traditional-meets-modern Chinese scholar style, sharp eyes
Vibe: Competitive but respectful, always ranks just above you in tests


ChatGPT
Claude
Gemini
LLama


Mistral
PaLM
Falcon
Yi


GPT-4
Claude
LLaMA
SPACEX
Grok
Qwen
Gemini
Mistral
Command R+


Midjourney



Recap: Machine Learning

It’s a function. True story.

From Learning to Machine Learning

- What's learning?
  - knowledge or skill acquired by instruction or study

- What's machine learning?

From Learning to Machine Learning

- What's learning?
  - knowledge or skill acquired by instruction or study

- What's machine learning?

  - acquiring skill with experience accumulated/computed from data
- What's skill?

A More Concrete Definition

- Skill: improve the performance measurements (e.g., prediction, 3pts shooting percentage)
- Therefore, machine learning is defined to be the improvements on some performance measurements by computing from data.
- For example,
  - Network data -> **ML** -> Better flow control
  - Signal data -> **ML** -> Faster antenna direction detection
  - Sequential web log data -> **ML** -> Higher accuracy of anomaly detection/efficiency of caching algorithm
  - Stock data -> **ML** -> More money

The Machine Learning Route

- ML: an **alternative route** to build complicated systems
- Some Use Scenarios
  - when human cannot program the system manually
    - navigating on Mars
  - when human cannot 'define the solution' easily
    - speech/visual recognition
  - when needing rapid decisions that humans cannot do
    - high-frequency trading
  - when needing to be user-oriented in a massive scale
    - consumer-targeted marketing

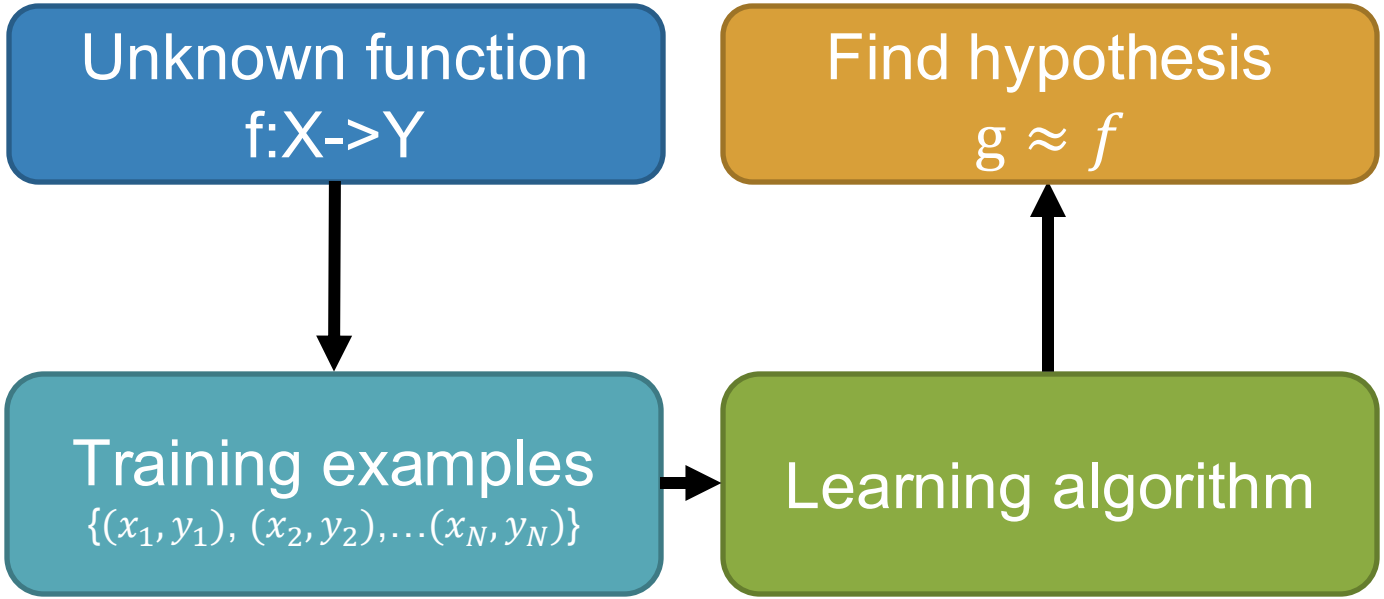
Key Essence of Machine Learning

- Exists some ‘underlying pattern’ to be learned
  - So performance measure can be improved
- But no programmable (easy) definition
  - So machine learning is required
- Somehow there is data about the pattern
  - So machine learning has inputs to learn from

Learning Problem Formulation

- Notation
  - **Input:** $x \in X$(application)
  - **Output:** $y \in Y$(good/bad after approving)
  - **Unknown pattern to be learned can be formulated as a function**
    - f: X->Y (ideal function)
  - **Data:** D={(x_1, y_1), (x_2, y_2),…(x_N, y_N)}
  - **Hypothesis:**
    - g:X->Y (hopefully can be as close to f as possible)

Learning Flow



Learning is to find a function

- Speech Recognition

f() = “我不知道你說什麼”

- Image recognition

f() = “Seafood”

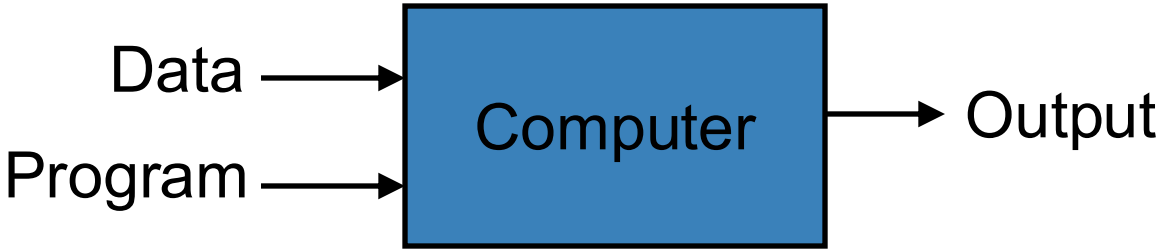
- Channel estimation

f(

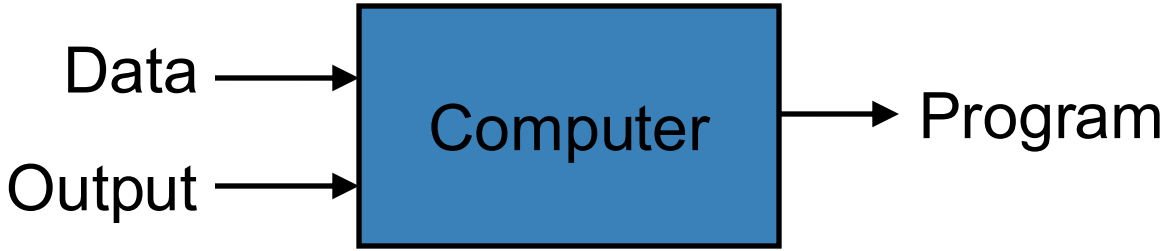
) = Channel parameters

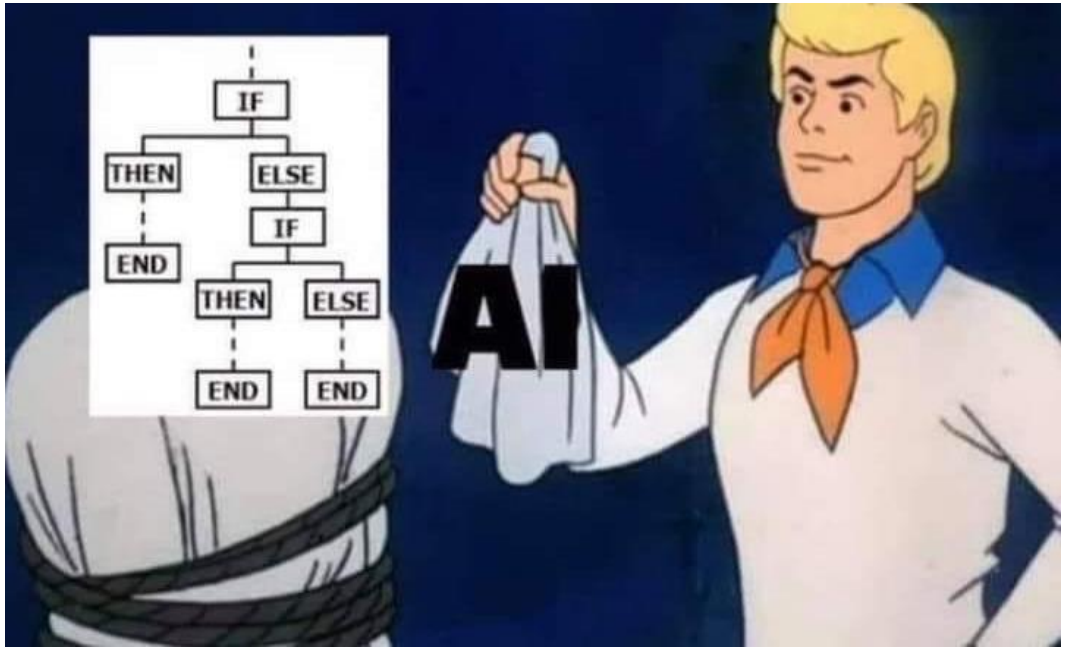
http://cdn1.itpro.co.uk/sites/itpro/files/images/dir_176/it_photo_88225.jpg
http://www.berkeleywellness.com/sites/default/files/field/image/ThinkstockPhotos-520490716_field_img_hero_988_380.jpg
https://telcoantennas.com.au/site/sites/default/files/images/4G-cross-polarisation-low-signal-areas.png

Traditional Programming



Machine Learning




IF
THEN
ELSE
IF
END
THEN
ELSE
END
END
AI

Magic?

No, more like gardening

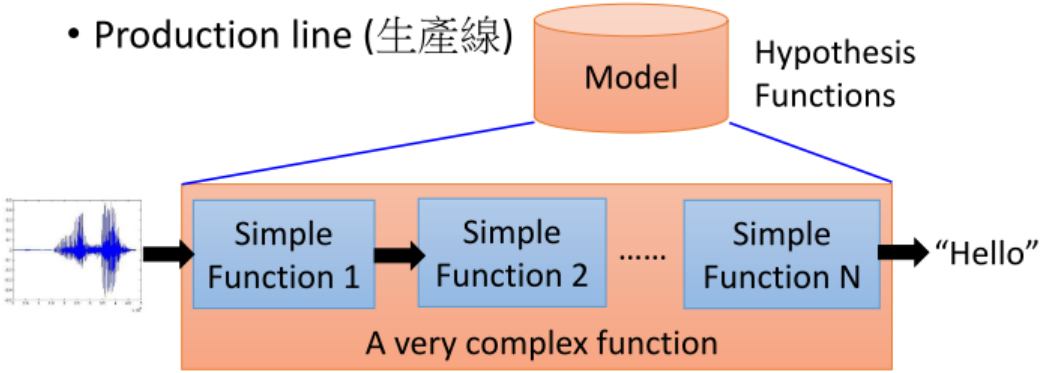
- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

Give me a place to stand on (with a long rod), and I will move the Earth!

Give me enough data
(with a good model), and I
can learn anything.

What is Deep Learning?

- Production line (生產線)

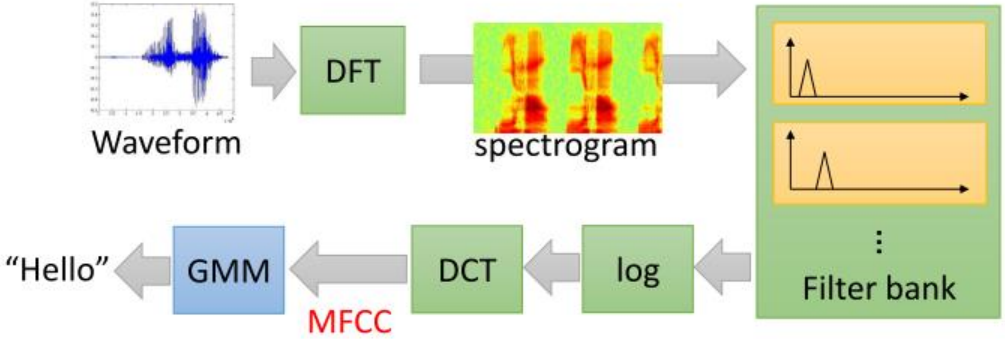


End-to-end training:
What each function should do is learned automatically

Thanks Prof. Hung-Yi Lee for this part of slides.

Deep v.s. Shallow
- Speech Recognition

- Shallow Approach



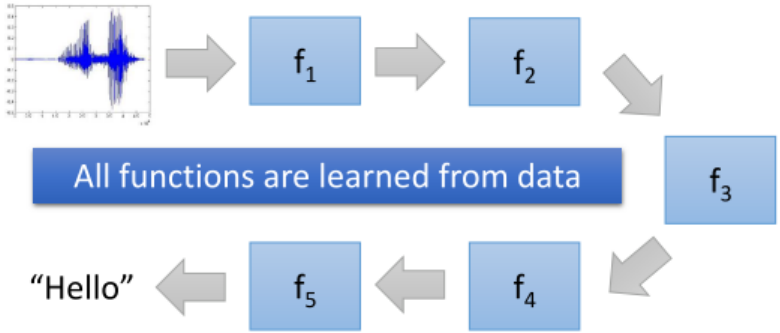
Each box is a simple function in the production line:

:hand-crafted :learned from data

Deep v.s. Shallow
- Speech Recognition

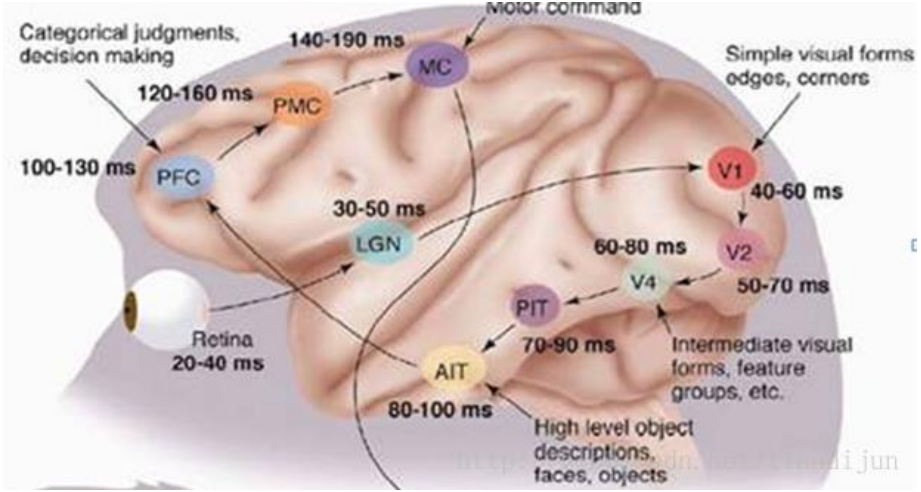
- Deep Learning

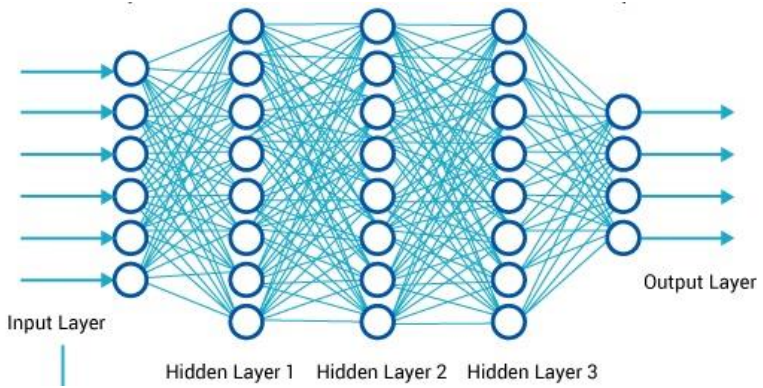
"Bye bye, MFCC"
- Deng Li in Interspeech 2014



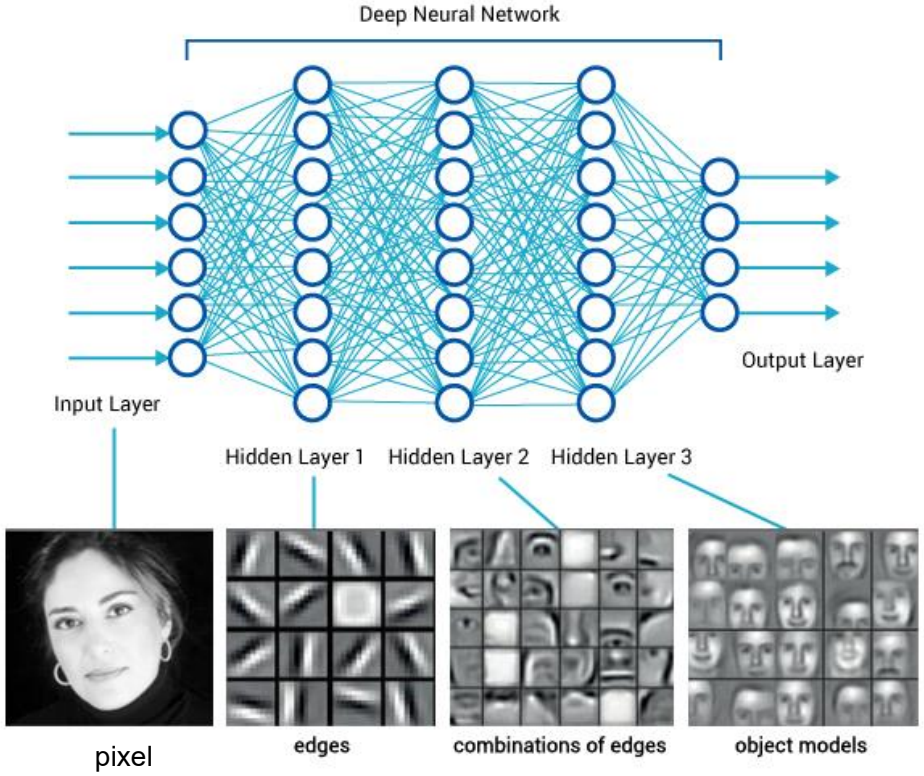
Less engineering labor, but machine learns more

Idea from human brain



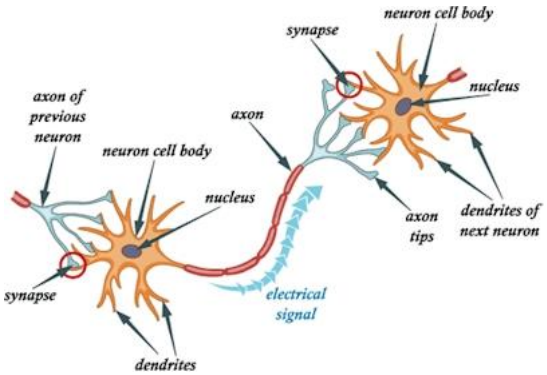


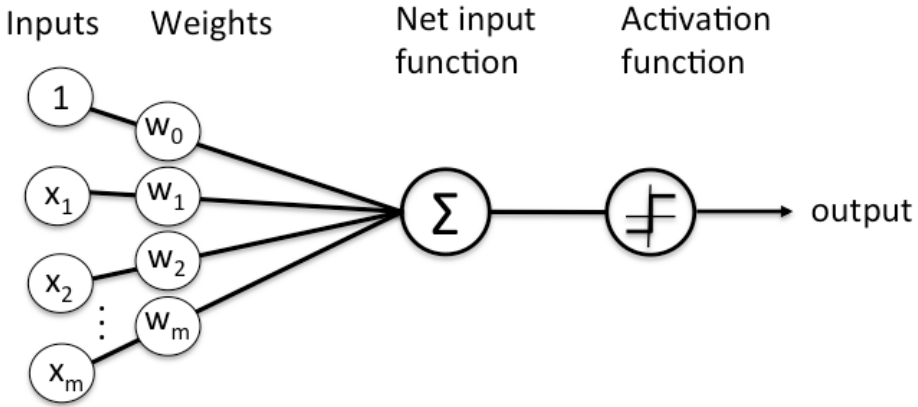
Artificial Neuron Network(ANN)



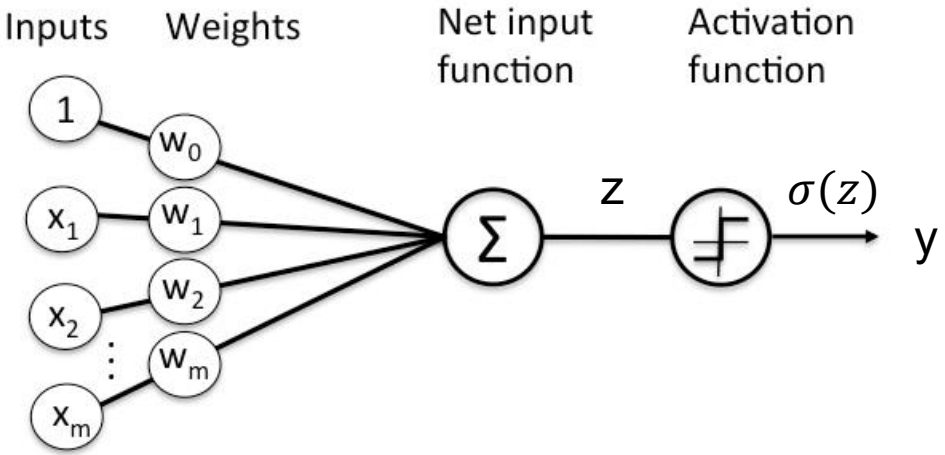
Source:http://www.amax.com/blog/?p=804

Neuron


synapse
neuron cell body
axon of previous neuron
axon
nucleus
neuron cell body
nucleus
axon tips
dendrites of next neuron
synapse
electrical signal
dendrites


Inputs
Weights
Net input function
Activation function
1
x_1
x_2
x_m
w_0
w_1
w_2
w_m
Σ
output

Neuron

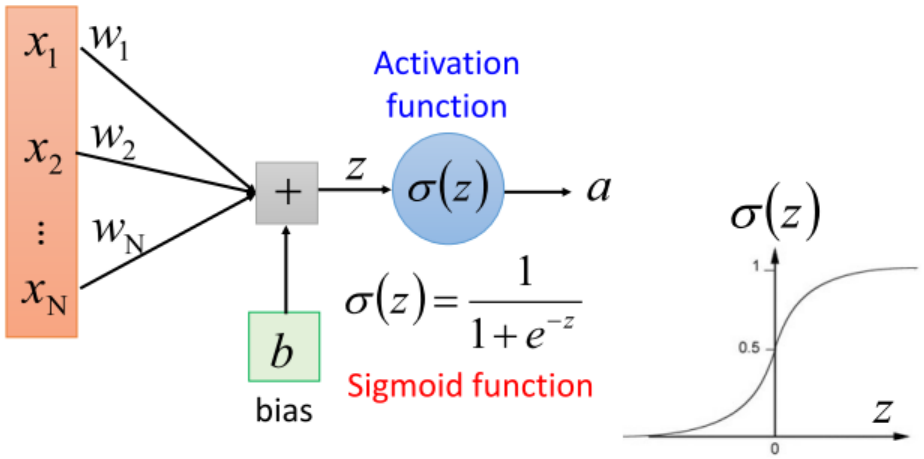


$z = 1^{*}w_0 + x_1{}^{*}w_1 + x_2{}^{*}w_2 + \ldots + x_m{}^{*}w_m$

$y = \sigma(z)$

A Neuron for Machine

Each neuron is a very simple function



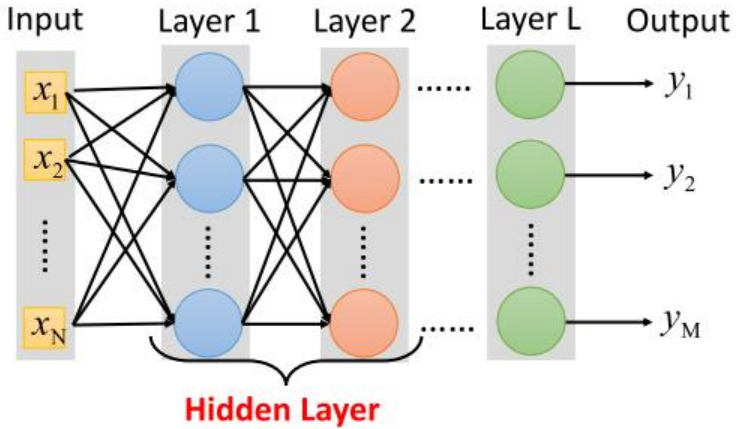
Deep Learning

A neural network is a complex function:

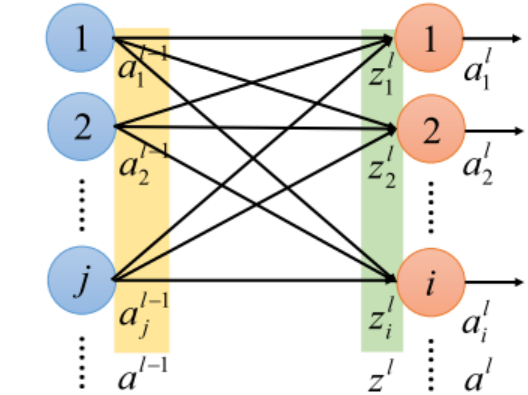
$$f : R^N \rightarrow R^M$$

- Cascading the neurons to form a neural network.

  Each layer is a simple function in the production line.



Relations between Layer Outputs



Layer $l-1$

N_{l-1} nodes

Layer l

N_l nodes

$$z_1^l = w_{11}^l a_1^{l-1} + w_{12}^l a_2^{l-1} + \cdots + b_1^l$$

$$z_2^l = w_{21}^l a_1^{l-1} + w_{22}^l a_2^{l-1} + \cdots + b_2^l$$

$$\vdots$$

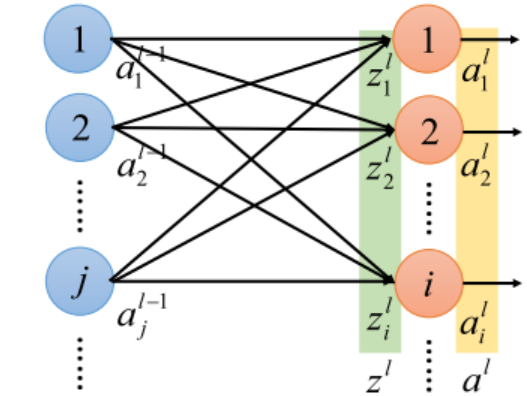
$$z_i^l = w_{i1}^l a_1^{l-1} + w_{i2}^l a_2^{l-1} + \cdots + b_i^l$$

$$\vdots$$

$$\begin{bmatrix} z_1^l \\ z_2^l \\ \vdots \\ z_i^l \\ \vdots \end{bmatrix} = \begin{bmatrix} w_{11}^l & w_{12}^l & \cdots \\ w_{21}^l & w_{22}^l & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} a_1^{l-1} \\ a_2^{l-1} \\ \vdots \\ a_i^{l-1} \\ \vdots \end{bmatrix} + \begin{bmatrix} b_1^l \\ b_2^l \\ \vdots \\ b_i^l \\ \vdots \end{bmatrix}$$

$$z^l = W^l a^{l-1} + b^l$$

Relations between Layer Outputs



Layer $l-1$

N_{l-1} nodes

Layer l

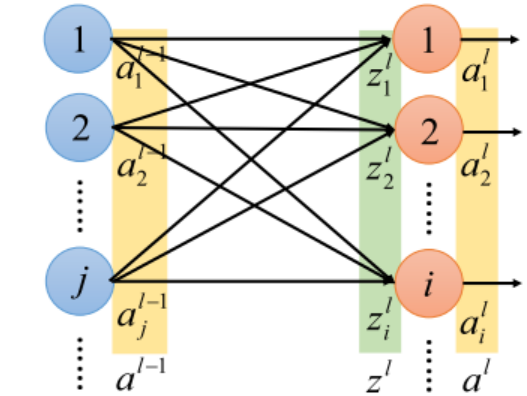
N_l nodes

$$a_i^l = \sigma\left(z_i^l\right)$$

$$\begin{bmatrix} a_1^l \\ a_2^l \\ \vdots \\ a_i^l \\ \vdots \end{bmatrix} = \begin{bmatrix} \sigma\left(z_1^l\right) \\ \sigma\left(z_2^l\right) \\ \vdots \\ \sigma\left(z_i^l\right) \\ \vdots \end{bmatrix}$$

$$a^l = \sigma\left(z^l\right)$$

Relations between Layer Outputs



Layer $l-1$

N_{l-1} nodes

Layer l

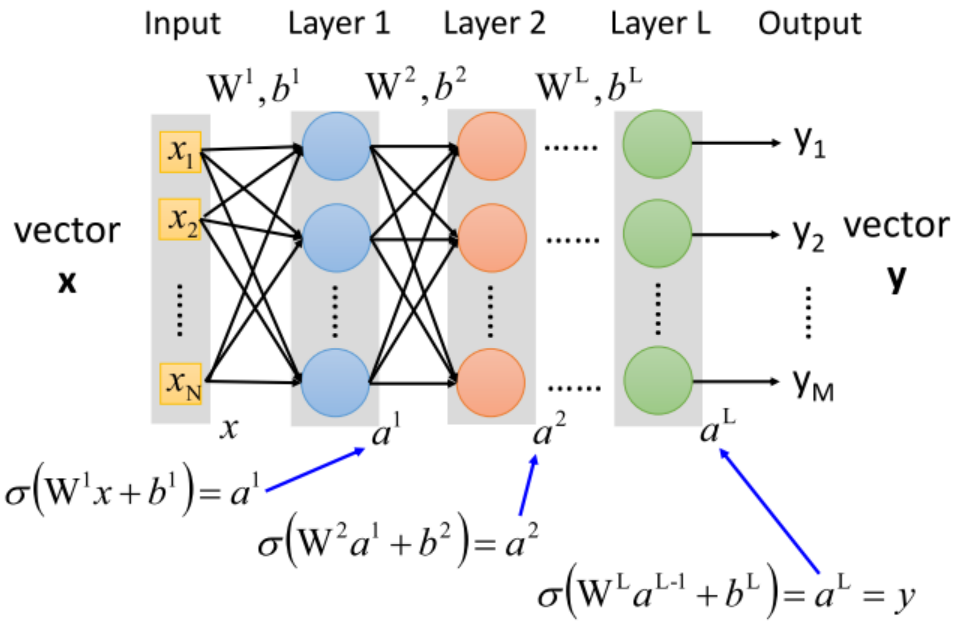
N_l nodes

$$z^l = W^l a^{l-1} + b^l$$

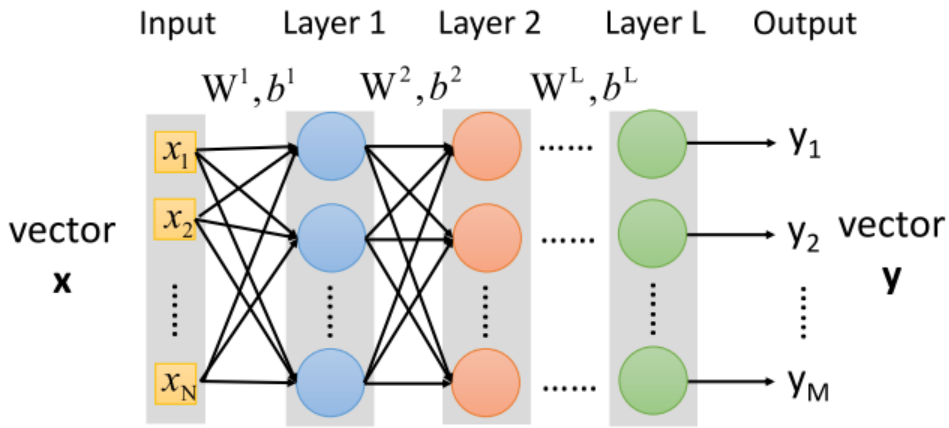
$$a^l = \sigma\left(z^l\right)$$

$$a^l = \sigma\left(W^l a^{l-1} + b^l\right)$$

Function of Neural Network



Function of Neural Network



$$y = f(x)$$
$$= \sigma\left(W^L \ldots \sigma\left(W^2 \sigma\left(W^1 x + b^1\right) + b^1\right) \ldots + b^L\right)$$

How to find the best function?

Best Function = Best Parameters

$$y = f(x) = \sigma\left(W^L \ldots \sigma\left(W^2 \sigma\left(W^1 x + b^1\right) + b^1\right) \ldots + b^L\right)$$

function set

because different parameters W and b lead to different function

Formal way to define a function set:

$f(x;\theta)$ → parameter set

$$\theta = \left\{W^1, b^1, W^2, b^2 \cdots W^L, b^L\right\}$$

Pick the "best" function f* → Pick the "best" parameter set θ*

Cost Function

- Define a function for parameter set $C(\theta)$
  - $C(\theta)$ evaluate how bad a parameter set is
  - The best parameter set θ^* is the one that minimizes $C(\theta)$

$$\theta^* = arg \min_{\theta} C(\theta)$$

- $C(\theta)$ is called ***cost/loss/error function***
  - If you define the goodness of the parameter set by another function $O(\theta)$
  - $O(\theta)$ is called objective function

Cost Function

Given training data:

$$\left\{\left(x^1,\hat{y}^1\right)\ldots\left(x^r,\hat{y}^r\right)\ldots\left(x^R,\hat{y}^R\right)\right\}$$

- ***Handwriting Digit Classification***

sum over all training examples

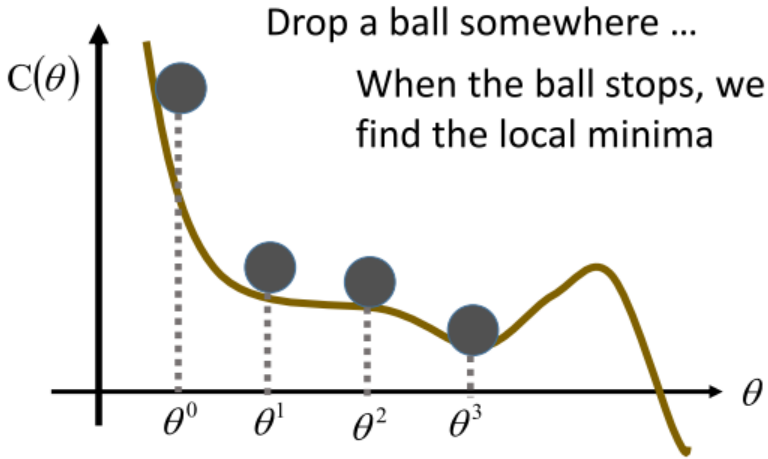
$$C(\theta)=\frac{1}{R}\sum_r\left\|f\left(x^r;\theta\right)-\hat{y}^r\right\|$$

Minimize distance

$$\begin{bmatrix}0.1\\0.4\\0.2\\\vdots\end{bmatrix}\begin{matrix}\text{“1”}\\\text{“2”}\\\text{“3”}\\\\\end{matrix}\qquad\begin{bmatrix}0\\1\\0\\\vdots\end{bmatrix}\begin{matrix}\text{“1”}\\\text{“2”}\\\text{“3”}\\\\\end{matrix}$$

Gradient Descent – Idea

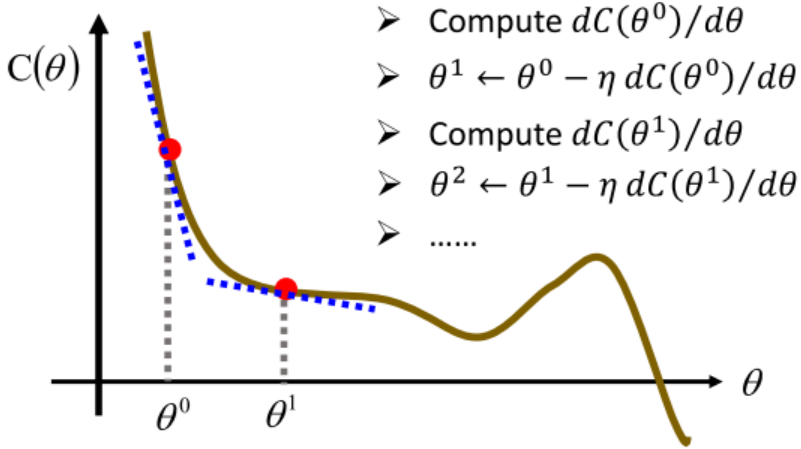
- For simplification, first consider that θ has only one variable



Gradient Descent – Idea

η is called *"learning rate"*

- For simplification, first consider that θ has only one variable

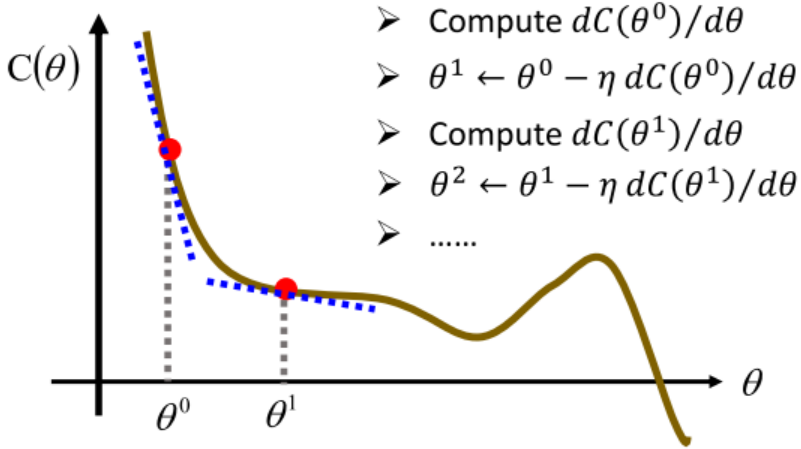


- Randomly start at θ^0
- Compute $dC(\theta^0)/d\theta$
- $\theta^1 \leftarrow \theta^0 - \eta\, dC(\theta^0)/d\theta$
- Compute $dC(\theta^1)/d\theta$
- $\theta^2 \leftarrow \theta^1 - \eta\, dC(\theta^1)/d\theta$
- .....

Gradient Descent – Idea

η is called *"learning rate"*

- For simplification, first consider that θ has only one variable



- Randomly start at θ^0
- Compute $dC(\theta^0)/d\theta$
- $\theta^1 \leftarrow \theta^0 - \eta\, dC(\theta^0)/d\theta$
- Compute $dC(\theta^1)/d\theta$
- $\theta^2 \leftarrow \theta^1 - \eta\, dC(\theta^1)/d\theta$
- .....

Gradient Descent

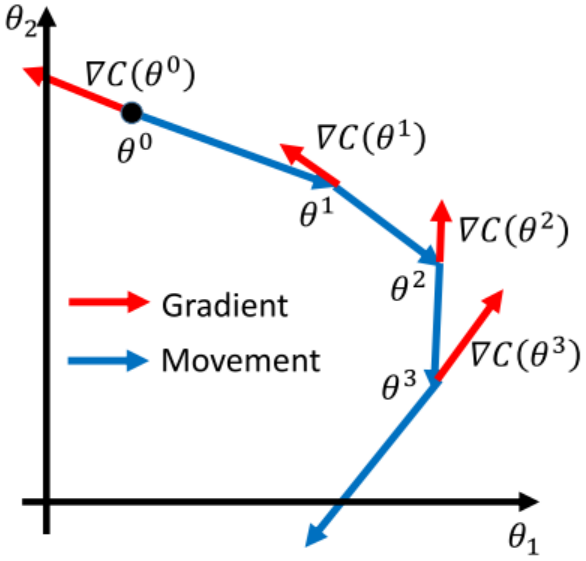
- Suppose that θ has two variables $\{\theta_1, \theta_2\}$

- Randomly start at $\theta^0 = \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \end{bmatrix}$
- Compute the gradients of $C(\theta)$ at θ^0: $\nabla C(\theta^0) = \begin{bmatrix} \partial C(\theta_1^0)/\partial\theta_1 \\ \partial C(\theta_2^0)/\partial\theta_2 \end{bmatrix}$
- Update parameters

$$\begin{bmatrix} \theta_1^1 \\ \theta_2^1 \end{bmatrix} = \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \end{bmatrix} - \eta \begin{bmatrix} \partial C(\theta_1^0)/\partial\theta_1 \\ \partial C(\theta_2^0)/\partial\theta_2 \end{bmatrix} \Rightarrow \theta^1 = \theta^0 - \eta\nabla C(\theta^0)$$

- Compute the gradients of $C(\theta)$ at θ^1: $\nabla C(\theta^1) = \begin{bmatrix} \partial C(\theta_1^1)/\partial\theta_1 \\ \partial C(\theta_2^1)/\partial\theta_2 \end{bmatrix}$
- ……

Gradient Descent



Start at position θ^0

Compute gradient at θ^0

Move to $\theta^1 = \theta^0 - \eta\nabla C(\theta^0)$

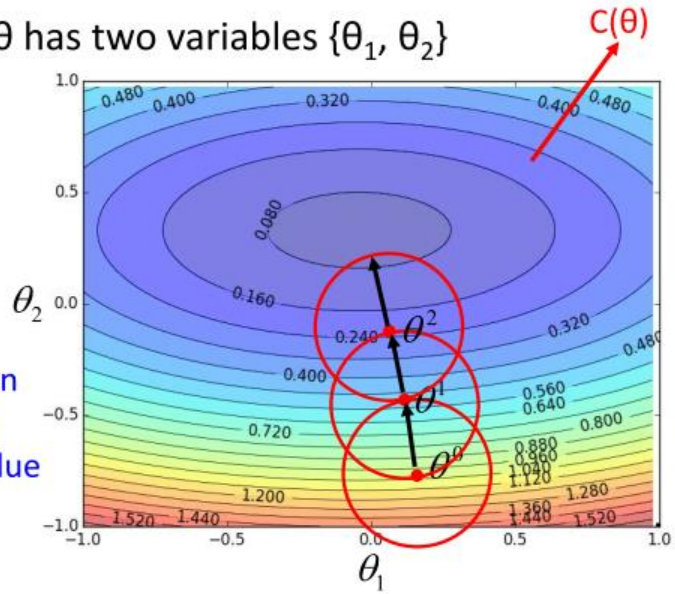
Compute gradient at θ^1

Move to $\theta^2 = \theta^1 - \eta\nabla C(\theta^1)$

⋮

Formal Derivation of Gradient Descent

- Suppose that θ has two variables $\{\theta_1, \theta_2\}$



Given a point, we can easily find the point with the smallest value nearby. How?

Gradient Descent

This is the “learning” of machines in deep learning ……

➡ Even alpha go using this approach.

I hope you are not too disappointed :p


380
280
0
150
10/12
工具時代
Gilfinz: 166/135
IMR D4RKN3Ss: 141/135
orcsbase: 123/135
BSV Baller: 113/135
127/134
市鎮中心
orcsbase (商賈人)
用來創建村民和推進到下一個時代。同時也是村民存放食物、木材、黃金及石料的地方。1座市鎮中心支持4個單位。
600/600
耗費：200

"I work with models."

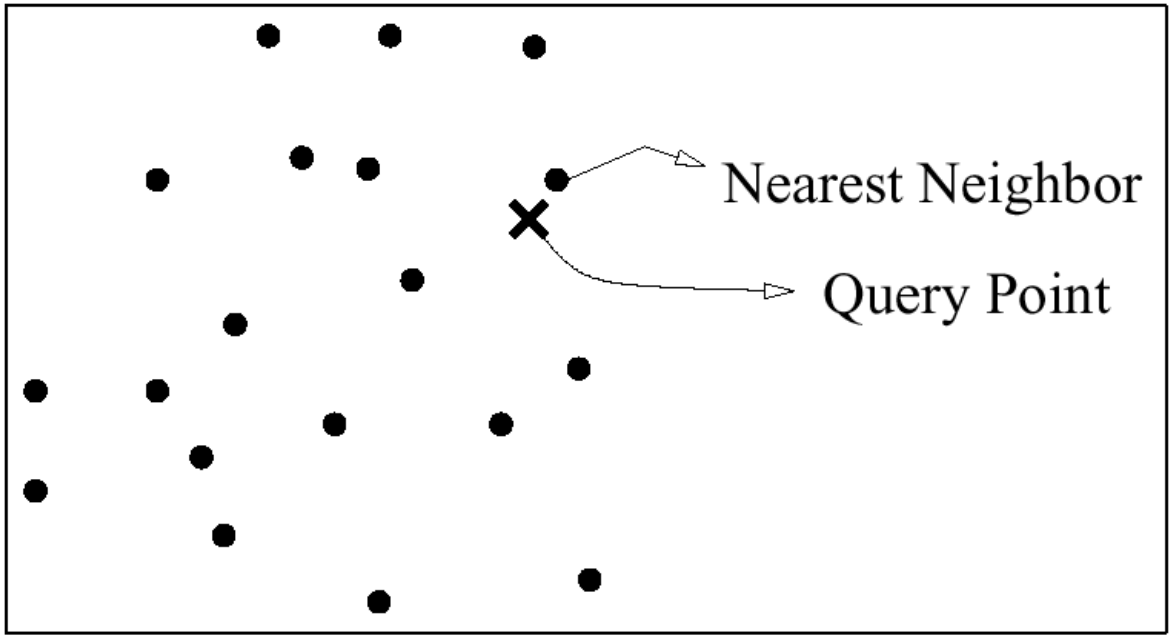
Dimension Reduction

Thanks to the slides made by Prof. Hung-Yi Lee from NTU.

Preliminaries : Nearest Neighbor Search

- Given a collection of data points and a query point in m-dimensional metric space, find the data point that is closest to the query point
- Variation: k-nearest neighbor
- Relevant to clustering and similarity search
- Applications: Geographical Information Systems, similarity search in multimedia databases

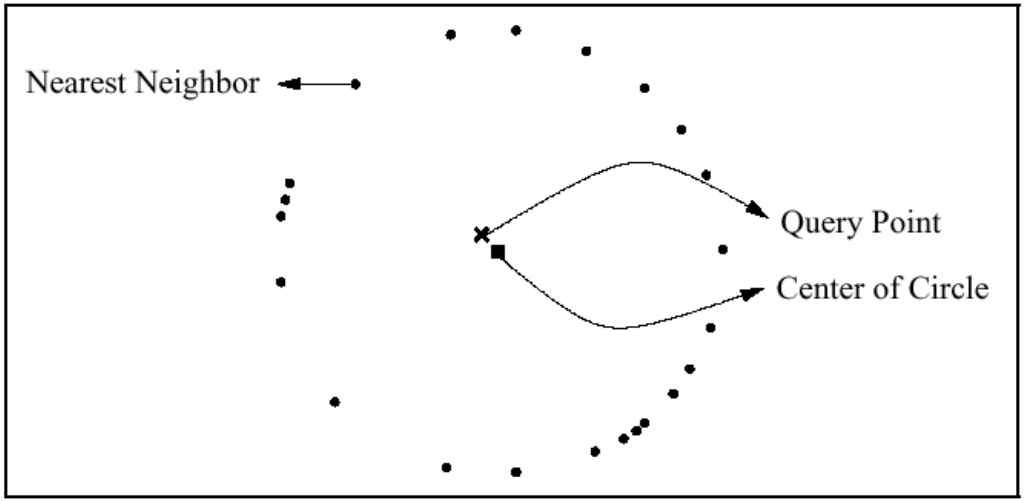
NN Search Con’t



Source: [2]

Problems with High Dimensional Data

- A point's nearest neighbor (NN) loses meaning



Source: [2]

Problems (Con't)

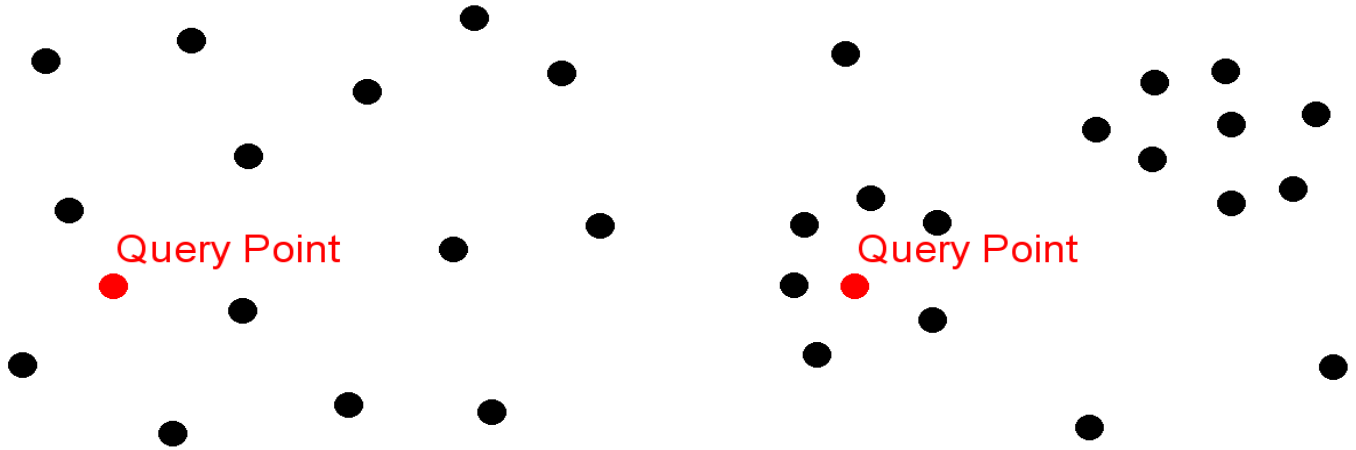
- NN query cost degrades – more strong candidates to compare with
- In as few as 10 dimensions, linear scan outperforms some multidimensional indexing structures (e.g. KD-tree, R* tree, SR tree)
- Biology and genomic data can have dimensions in the 1000's.

Problems (Con’t)

- The presence of irrelevant attributes decreases the tendency for clusters to form
- Points in high dimensional space have high degree of freedom; they could be so scattered that they appear uniformly distributed

Problems Con’t

- In which cluster does the query point fall?



The Curse

- Refers to the decrease in performance of query processing when the dimensionality increases
- In particular, under certain conditions, the distance between the nearest point and the query point equals the distance between the farthest and query point as dimensionality approaches infinity

Curse (Con't)

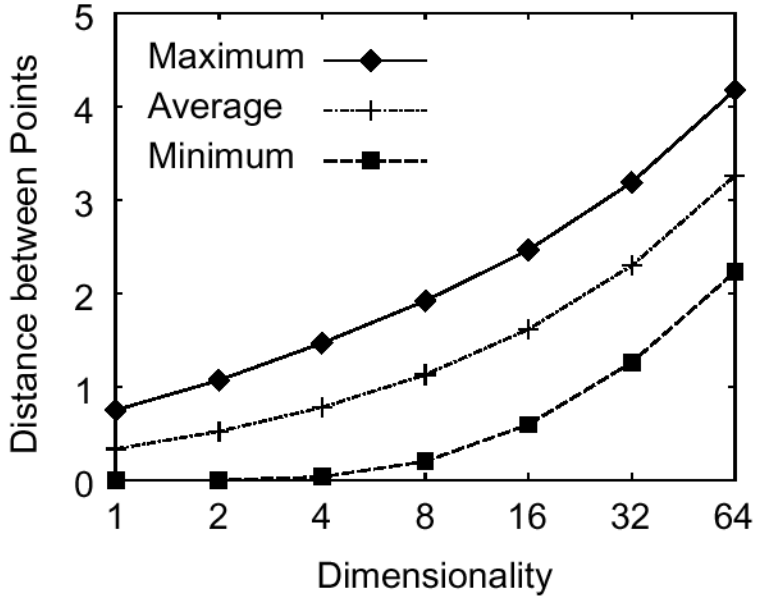


Figure 1. Distances among 100k points generated at random in a unit hypercube

Source: N. Katayama, S. Satoh. Distinctiveness Sensitive Nearest Neighbor Search for Efficient Similarity Retrieval of Multimedia Information. ICDE Conference, 2001.

Distributed Representation

- Clustering: an object must belong to one cluster

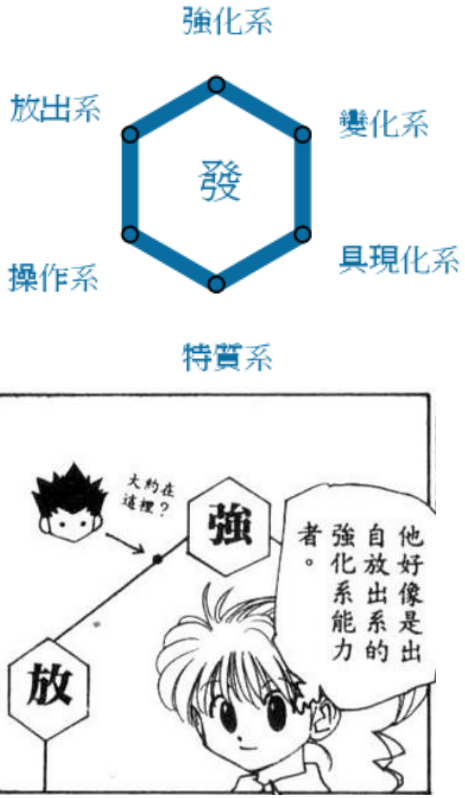
小傑是強化系

- Distributed representation

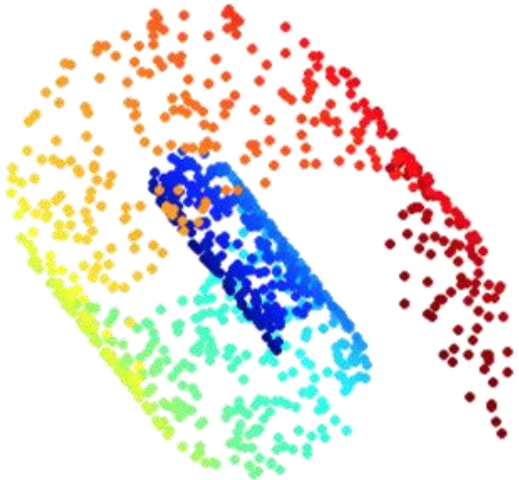
Dimension Reduction

小傑是

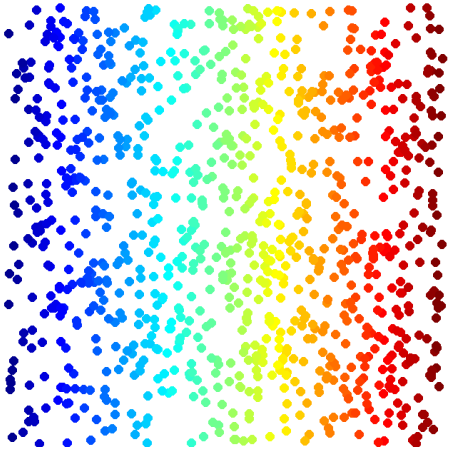
強化系	0.70
放出系	0.25
變化系	0.05
操作系	0.00
具現化系	0.00
特質系	0.00



Dimension Reduction



Looks like 3-D



Actually, 2-D

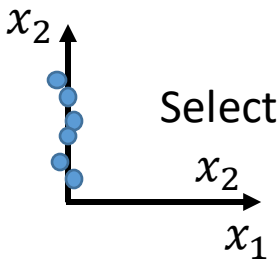
http://reuter.mit.edu/blue/images/research/manifold.png

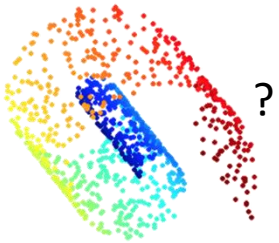
http://archive.cnx.org/resources/51a9b2052ae167db310fda5600b89badea85eae5/isomapCNXtrue1.png

Dimension Reduction

The dimension of z would be smaller than x

- Feature selection





- Principle component analysis (PCA)
  [Bishop, Chapter 12]

$$z = Wx$$

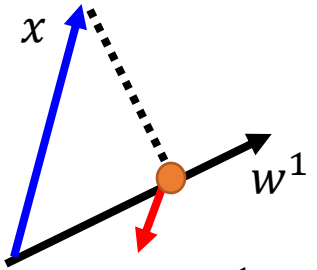
Principle Component Analysis (PCA)

PCA

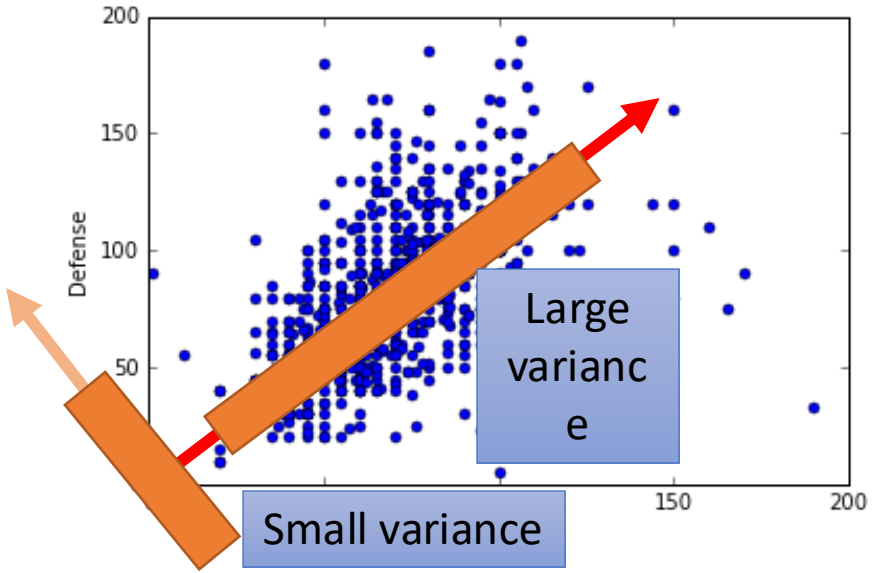
$z = Wx$

Reduce to 1-D:

$z_1 = w^1 \cdot x$



$z_1 = w^1 \cdot x$



Project all the data points x onto w^1, and obtain a set of z_1

We want the variance of z_1 as large as possible

$$Var(z_1) = \sum_{z_1} (z_1 - \bar{z_1})^2 \quad \|w^1\|_2 = 1$$

PCA

$$z = Wx$$

Reduce to 1-D:

$$z_1 = w^1 \cdot x$$

$$z_2 = w^2 \cdot x$$

$$W = \begin{bmatrix} (w^1)^T \\ (w^2)^T \\ \vdots \end{bmatrix}$$

Orthogonal matrix

Project all the data points x onto w^1, and obtain a set of z_1

We want the variance of z_1 as large as possible

$$Var(z_1) = \sum_{z_1} (z_1 - \bar{z_1})^2 \quad \|w^1\|_2 = 1$$

We want the variance of z_2 as large as possible

$$Var(z_2) = \sum_{z_2} (z_2 - \bar{z_2})^2 \quad \|w^2\|_2 = 1$$

$$w^1 \cdot w^2 = 0$$

Warning of Math

PCA

$$z_1 = w^1 \cdot x$$

$$\bar{z_1} = \frac{1}{N}\sum z_1 = \frac{1}{N}\sum w^1 \cdot x = w^1 \cdot \frac{1}{N}\sum x = w^1 \cdot \bar{x}$$

$$Var(z_1) = \frac{1}{N}\sum_{z_1}(z_1 - \bar{z_1})^2$$

$$= \frac{1}{N}\sum_{x}(w^1 \cdot x - w^1 \cdot \bar{x})^2$$

$$= \frac{1}{N}\sum\left(w^1 \cdot (x - \bar{x})\right)^2$$

$$= \frac{1}{N}\sum (w^1)^T (x - \bar{x})(x - \bar{x})^T w^1$$

$$= (w^1)^T \boxed{\frac{1}{N}\sum (x - \bar{x})(x - \bar{x})^T} w^1$$

$$= (w^1)^T Cov(x) w^1$$

$S = Cov(x)$

$$(a \cdot b)^2 = (a^T b)^2 = a^T b a^T b$$

$$= a^T b (a^T b)^T = a^T b b^T a$$

Find w^1 maximizing

$$(w^1)^T S w^1$$

$$\|w^1\|_2 = (w^1)^T w^1 = 1$$

Find w^1 maximizing $(w^1)^T S w^1$ $\quad (w^1)^T w^1 = 1$

$S = Cov(x)$ Symmetric positive-semidefinite (non-negative eigenvalues)

Using Lagrange multiplier [Bishop, Appendix E]

$$g(w^1) = (w^1)^T S w^1 - \alpha\left((w^1)^T w^1 - 1\right)$$

$$\partial g(w^1)/\partial w_1^1 = 0$$
$$\partial g(w^1)/\partial w_2^1 = 0$$
$$\vdots$$

$$S w^1 - \alpha w^1 = 0$$
$$S w^1 = \alpha w^1$$ w^1 : eigenvector

$$(w^1)^T S w^1 = \alpha (w^1)^T w^1$$
$$= \alpha$$ Choose the maximum one

w^1 is the eigenvector of the covariance matrix S
Corresponding to the largest eigenvalue

λ_1

Find w^2 maximizing $(w^2)^T S w^2 \quad (w^2)^T w^2 = 1 \quad (w^2)^T w^1 = 0$

$$g(w^2) = (w^2)^T S w^2 - \alpha\left((w^2)^T w^2 - 1\right) \; -\beta\left((w^2)^T w^1 - 0\right)$$

$\partial g(w^2)/\partial w_1^2 = 0$

$\partial g(w^2)/\partial w_2^2 = 0$

$\vdots$

$$Sw^2 - \alpha w^2 - \beta w^1 = 0$$

$$0 - \alpha\, 0 - \beta\, 1 = 0$$

$$= \left((w^1)^T S w^2\right)^T = (w^2)^T S^T w^1$$

$$= (w^2)^T S w^1 = \lambda_1 (w^2)^T w^1 = 0$$

$$Sw^1 = \lambda_1 w^1$$

$\beta = 0$: $\quad Sw^2 - \alpha w^2 = 0 \quad Sw^2 = \alpha w^2$

w^2 is the eigenvector of the covariance matrix S

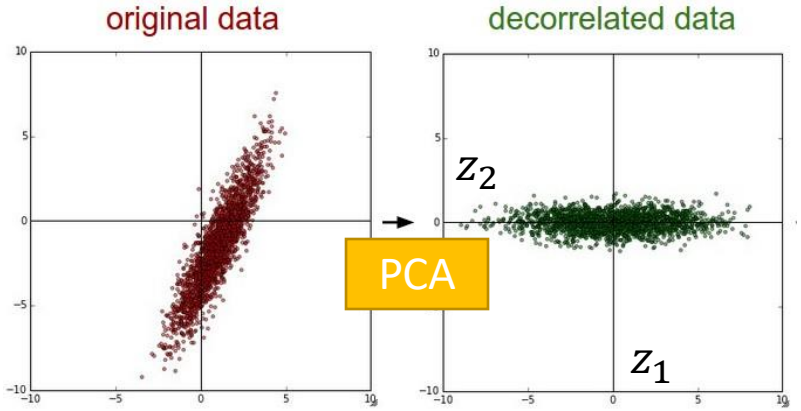
Corresponding to the 2nd largest eigenvalue λ_2

PCA - decorrelation

$$z = Wx$$

$$Cov(z) = D$$

Diagonal matrix



$$Cov(z) = \frac{1}{N}\sum (z - \bar{z})(z - \bar{z})^T = WSW^T \qquad S = Cov(x)$$

$$= WS[w^1 \quad \cdots \quad w^K] = W[Sw^1 \quad \cdots \quad Sw^K]$$

$$= W[\lambda_1 w^1 \quad \cdots \quad \lambda_K w^K] = [\lambda_1 Ww^1 \quad \cdots \quad \lambda_K Ww^K]$$

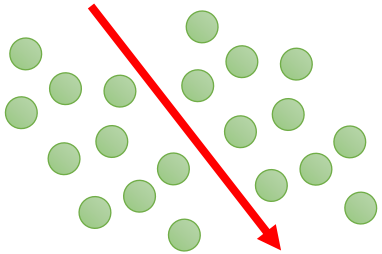
$$= [\lambda_1 e_1 \quad \cdots \quad \lambda_K e_K] = D$$

Diagonal matrix

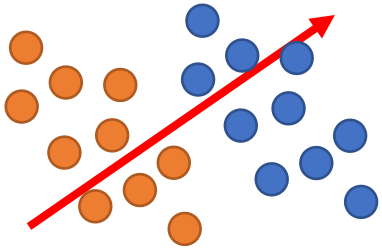
End of Warning

Weakness of PCA

- Unsupervised

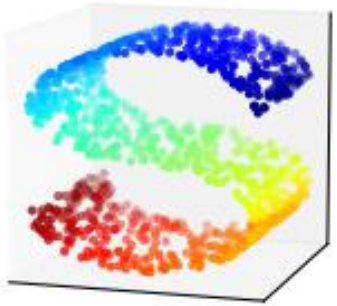


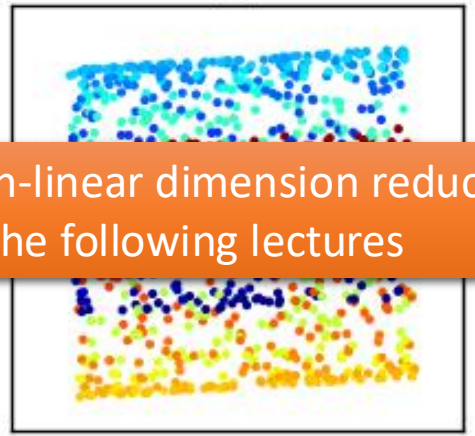
PCA



LDA

- Linear





Non-linear dimension reduction in the following lectures

http://www.astroml.org/book_figures/chapter7/fig_S_manifold_PCA.html

With the power of PCA, let's become the master of…

Pokémon!!

https://home.gamer.com.tw/creationDetail.php?sn=223670

PCA - Pokémon

- Inspired from: https://www.kaggle.com/strakul5/d/abcsds/pokemon/principal-component-analysis-of-pokemon-data
- 800 Pokemons, 6 features for each (HP, Atk, Def, Sp Atk, Sp Def, Speed)
- How many principle components?

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6}$$

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
ratio	0.45	0.18	0.13	0.12	0.07	0.04

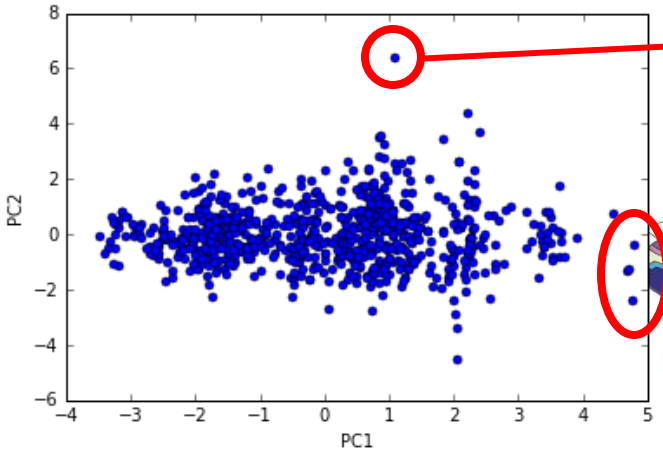
Using 4 components is good enough

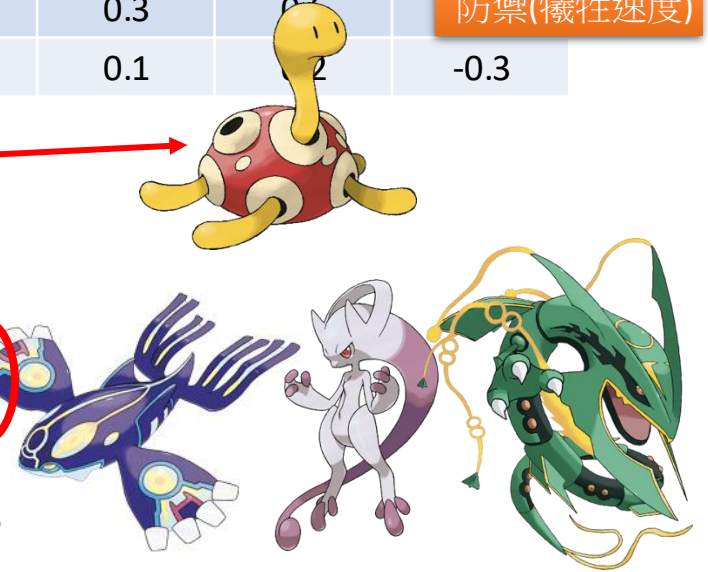
PCA - Pokémon

	HP	Atk	Def	Sp Atk	Sp Def	Speed
PC1	0.4	0.4	0.4	0.5	0.4	0.3
PC2	0.1	0.0	0.6	-0.3	0.2	-0.7
PC3	-0.5	-0.6	0.1	0.3	[illegible]	[illegible]
PC4	0.7	-0.4	-0.4	0.1	[illegible]	-0.3

強度

防禦(犠牲速度)



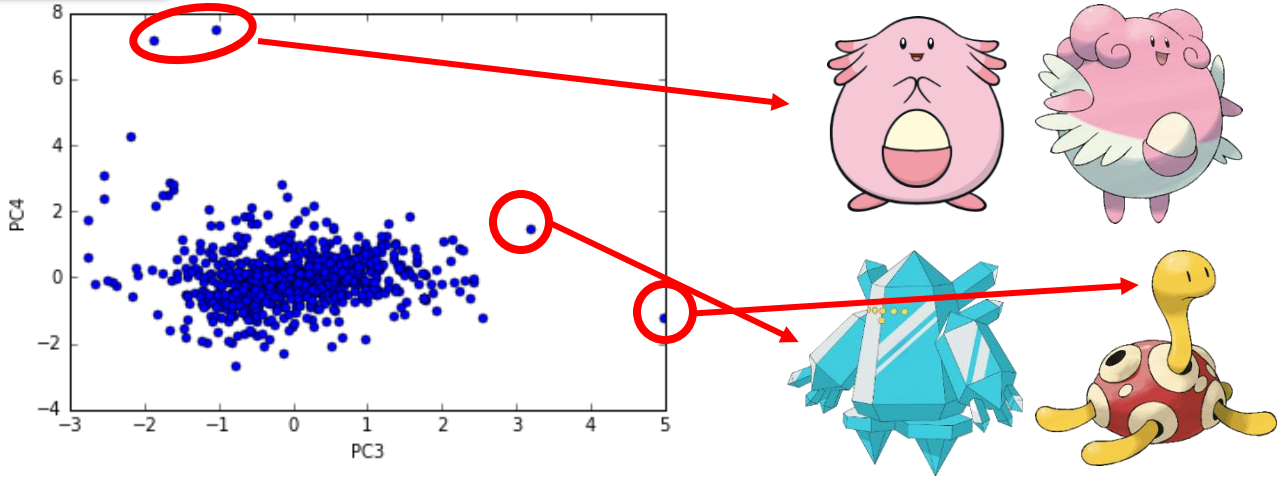


PCA - Pokémon

	HP	Atk	Def	Sp Atk	Sp Def	Speed
PC1	0.4	0.4	0.4	0.5	0.4	0.3
PC2	0.1	0.0	0.6	-0.3	0.2	-0.7
PC3	-0.5	-0.6	0.1	0.3	0.6	
PC4	0.7	-0.4	-0.4	0.1	0.2	

生命力強

特殊防禦(犧牲攻擊和生命)




Q and A
Thank you