多媒體數據分析與應用

張家瑋 博士
助理教授
國立臺中科技大學資訊工程系

FOUNDATIONS OF NATURAL LANGUAGE PROCESSING

自然語言處理的原理與應用

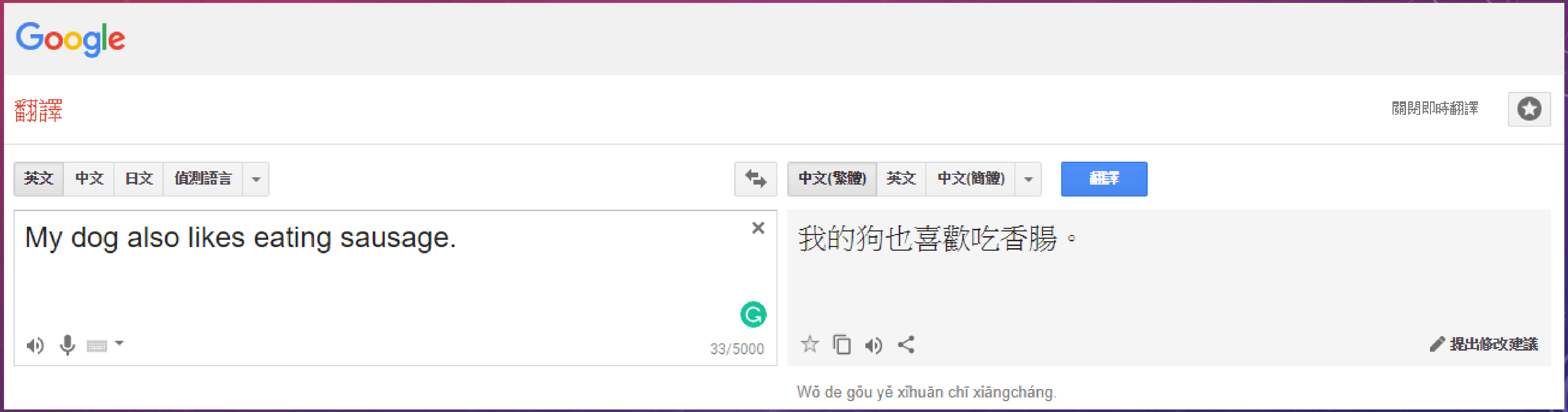
自然語言處理的主要範疇

- 機器翻譯 (Machine Translation)
- 自然語言理解/語意分析 (Natural Language Understanding / Semantic Analysis)
  1. 問答系統 (Question Answering)
  2. 萃取式摘要 (Extractive Summarization)
  3. 文件分類 (Text Categorization)
- 自然語言生成 (Natural Language Generation)
  1. 進階問答系統 (Advanced Question Answering)
  2. 抽象式摘要 (Abstractive Summarization)
  3. 聊天機器人 (Chatbot)
- 語法分析 (Syntactic Parsing)
  1. 中文斷詞 (Chinese word segmentation)
  2. 詞性標註 (Part-of-speech Tagging)
  3. 實體辨識 (Named Entity Recognition)
  4. 詞彙依存 (Typed Dependencies)
  5. 文法樹 (Parse Tree)
- 語音辨識 (Speech Recognition)
- 文字轉語音 (Text to Speech)
- 語音轉文字 (Speech to Text)

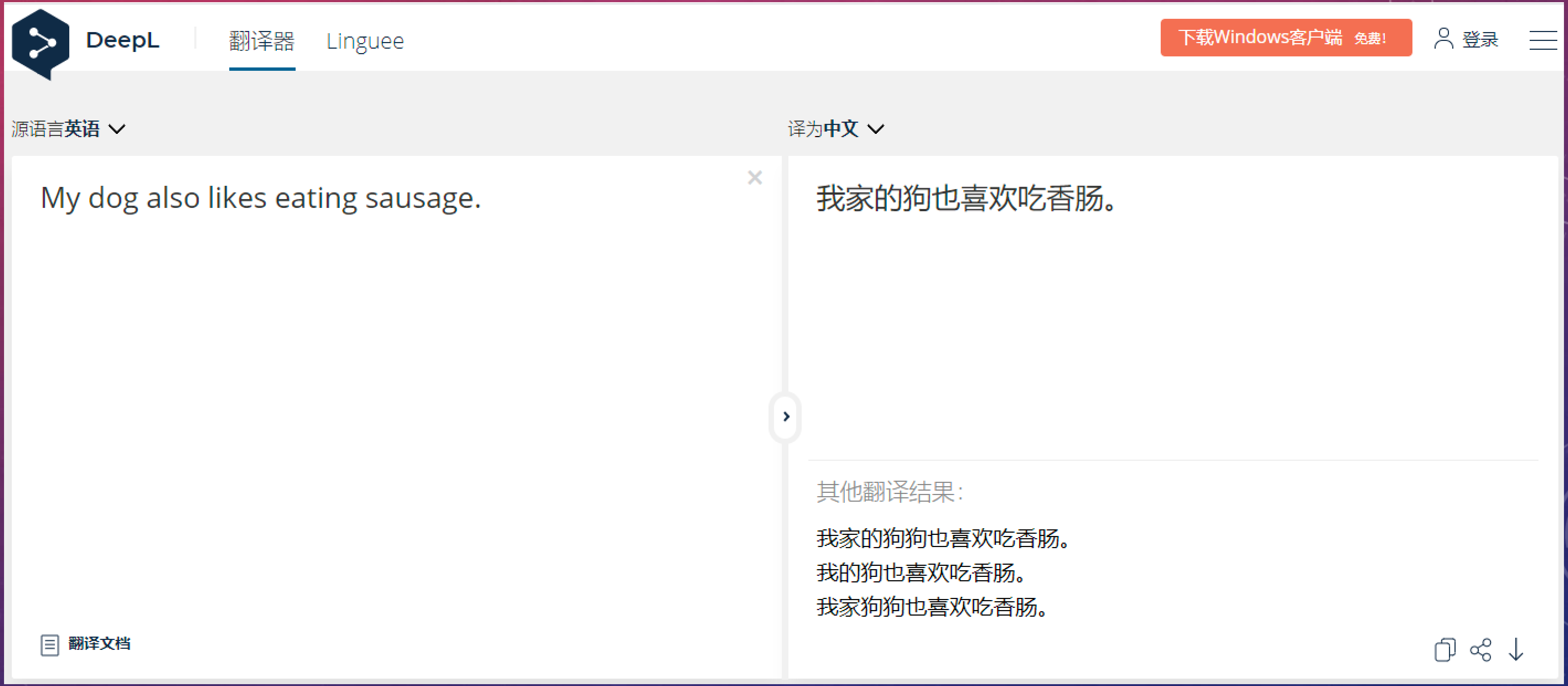
機器翻譯

MACHINE TRANSLATION

GOOGLE 翻譯


翻譯
關閉即時翻譯
英文 中文 日文 偵測語言
中文(繁體) 英文 中文(簡體)
翻譯
My dog also likes eating sausage.
33/5000
我的狗也喜歡吃香腸。
提出修改建議
Wǒ de gǒu yě xǐhuān chī xiāngcháng.

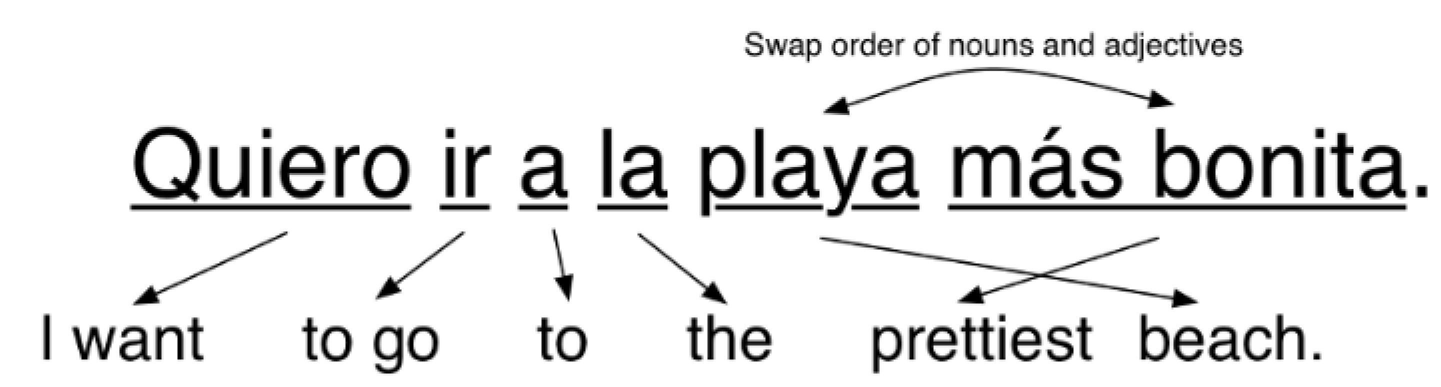
DEEP L



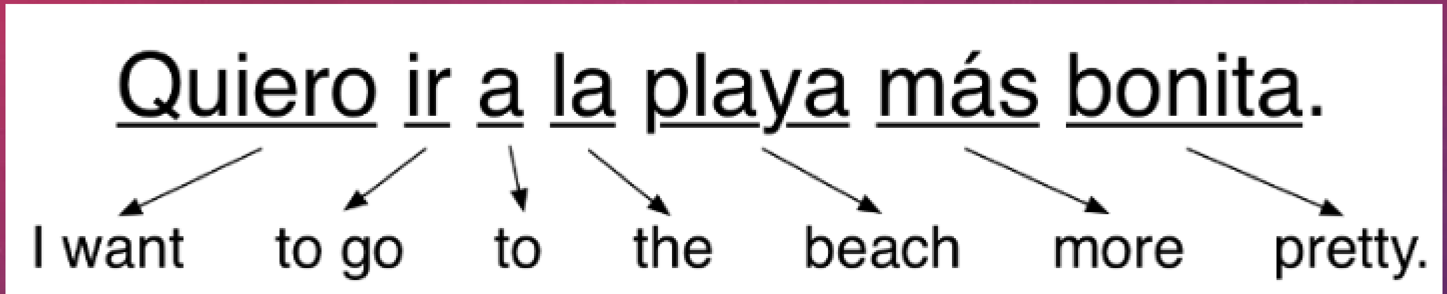
DeepL
翻译器
Linguee
下载Windows客户端 免费!
登录
源语言英语
译为中文
My dog also likes eating sausage.
我家的狗也喜欢吃香肠。
其他翻译结果:
我家的狗狗也喜欢吃香肠。
我的狗也喜欢吃香肠。
我家狗狗也喜欢吃香肠。
翻译文档

平行語料


Quiero ir a la playa más bonita.
I want
to go
to
the
beach
more
pretty.
We just replace each Spanish word with the matching English word.

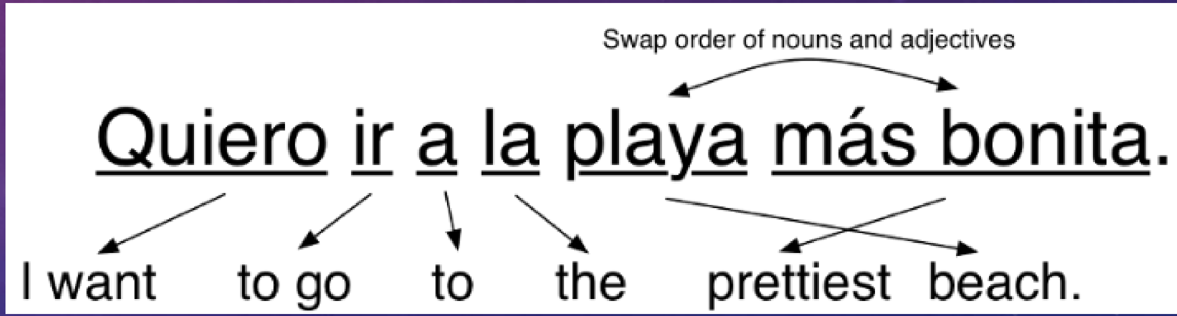

Swap order of nouns and adjectives
Quiero ir a la playa más bonita.
I want
to go
to
the
prettiest
beach.

統計式機器翻譯之原理

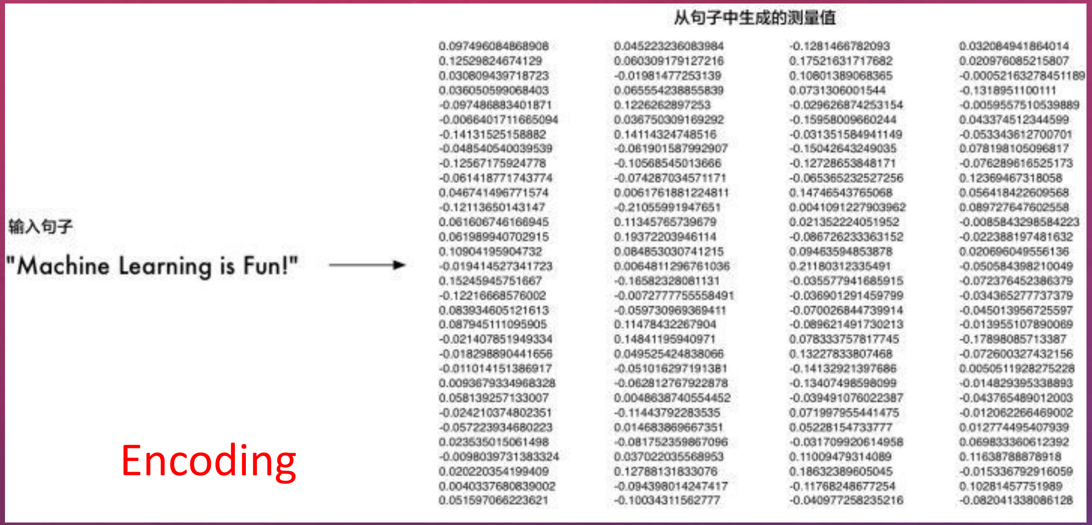


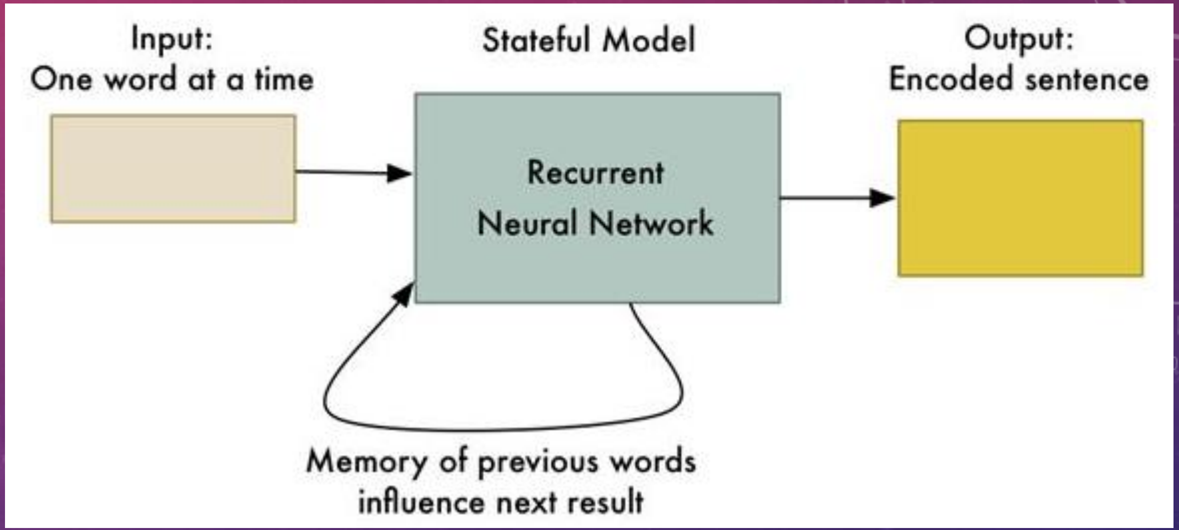
We just replace each Spanish word with the matching English word.

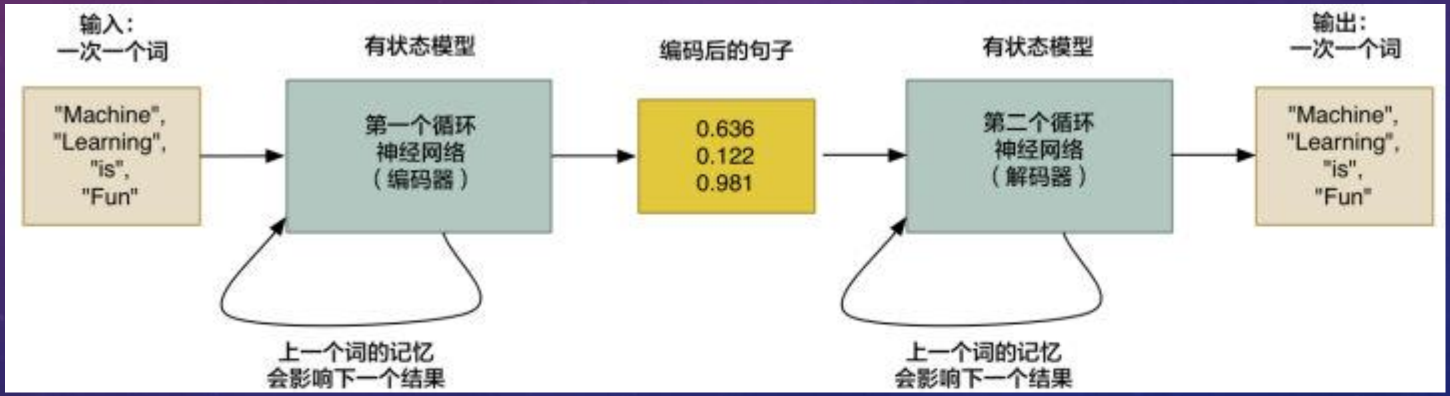
Even the most common phrases have lots of possible translations.



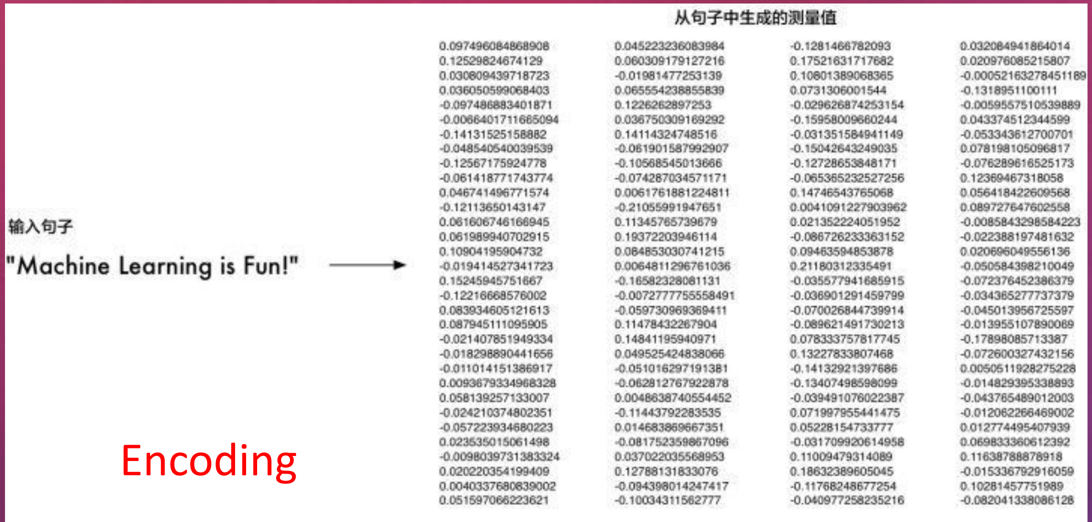
深度學習於機器翻譯之原理

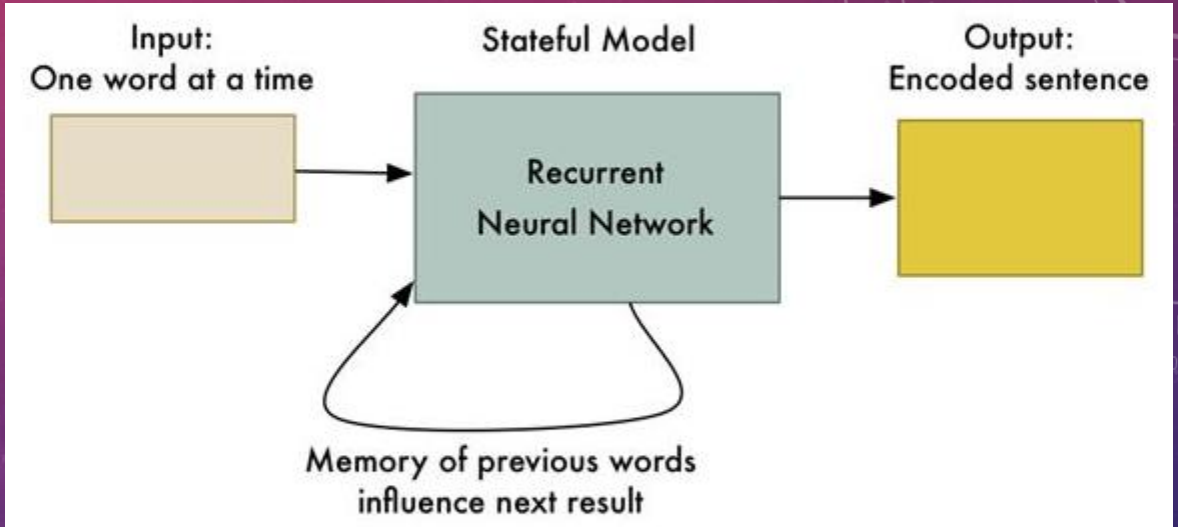


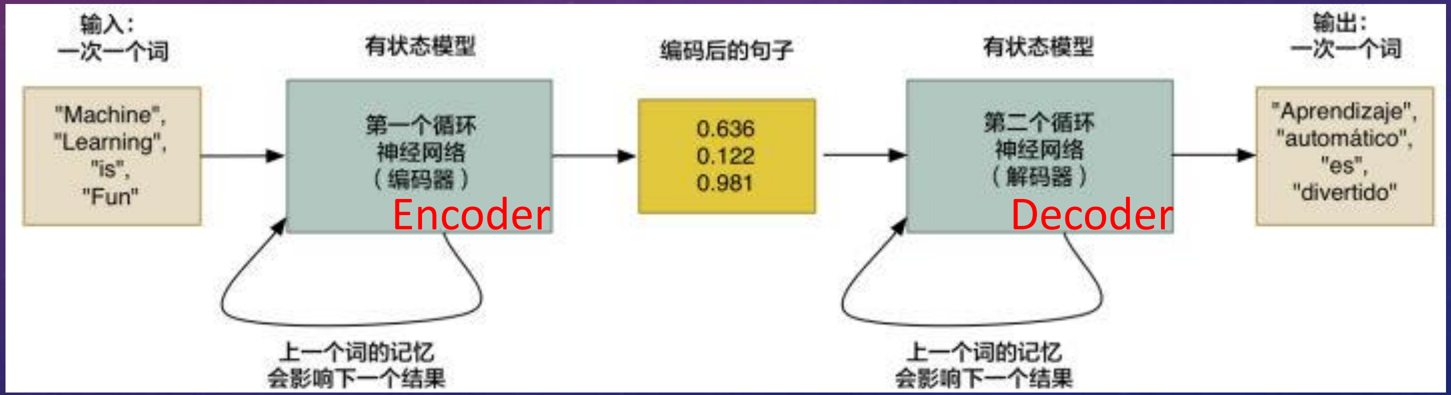




深度學習於機器翻譯之原理







自然語言理解

NATURAL LANGUAGE UNDERSTANDING

VECTOR REPRESENTATION

	w_1	w_2	w_3	..	..	..	w_{n-1}	w_n	label
D_1	0.11	0.23	0	..	..	..	0.57	0	0
D_2	0	0	0	..	..	..	0.29	0.7	1
D_3	0	0.81	0.44	..	..	..	0	0	0
D_4	0	0.37	0	..	..	..	0	0.16	1
..	..	..	..	..	..	..	..	..	..
D_k	..	..	..	..	..	..	..	..	1

TF-IDF

- TF: term frequency: $\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$
- IDF: inverse document frequency: $\mathrm{idf_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$

where:

- |D|: total number of documents in the corpus
- $|\{j : t_i \in d_j\}|$: number of documents where term t_i appears

Then:

- $\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$

Document 1	
Term	**Term Count**
this	1
is	1
a	2
sample	1

Document 2	
Term	**Term Count**
this	1
is	1
another	2
example	3

- The calculation of tf–idf for the term "this" is performed as follows:
  -

$$\mathrm{tf}(''\text{this}'', d_1) = \frac{1}{5} = 0.2$$
$$\mathrm{tf}(''\text{this}'', d_2) = \frac{1}{7} \approx 0.14$$

$$\mathrm{idf}(''\text{this}'', D) = \log\left(\frac{2}{2}\right) = 0$$

  - So tf–idf is zero for the word "this", which implies that the word is not very informative as it appears in all documents.

$$\mathrm{tfidf}(''\text{this}'', d_1) = 0.2 \times 0 = 0$$
$$\mathrm{tfidf}(''\text{this}'', d_2) = 0.14 \times 0 = 0$$

Document 1	
Term	**Term Count**
this	1
is	1
a	2
sample	1

Document 2	
Term	**Term Count**
this	1
is	1
another	2
example	3

- A slightly more interesting example arises from the word "example", which occurs three times only in the second document:
  - $$\mathrm{tf}(''\text{example}'', d_1) = \frac{0}{5} = 0$$
  - $$\mathrm{tf}(''\text{example}'', d_2) = \frac{3}{7} \approx 0.429$$

$$\mathrm{idf}(''\text{example}'', D) = \log\left(\frac{2}{1}\right) = 0.301$$

$$\mathrm{tfidf}(''\text{example}'', d_1) = \mathrm{tf}(''\text{example}'', d_1) \times \mathrm{idf}(''\text{example}'', D) = 0 \times 0.301 = 0$$
$$\mathrm{tfidf}(''\text{example}'', d_2) = \mathrm{tf}(''\text{example}'', d_2) \times \mathrm{idf}(''\text{example}'', D) = 0.429 \times 0.301 \approx 0.13$$

潛藏語意分析(LSA)

- 奇異值分解
  - Singular Value Decomposition (SVD)

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

=

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

*

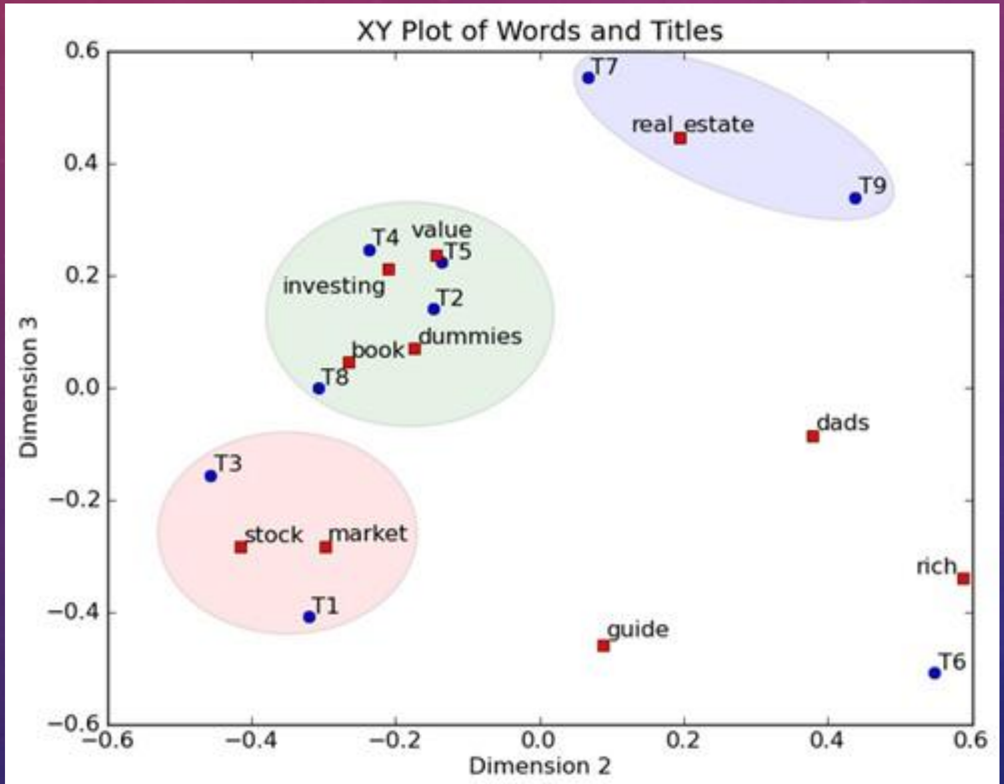
3.91	0	0
0	2.61	0
0	0	2.00

*

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

潛藏語意分析(LSA)

- 文件分類/主題探勘
- 語意分析



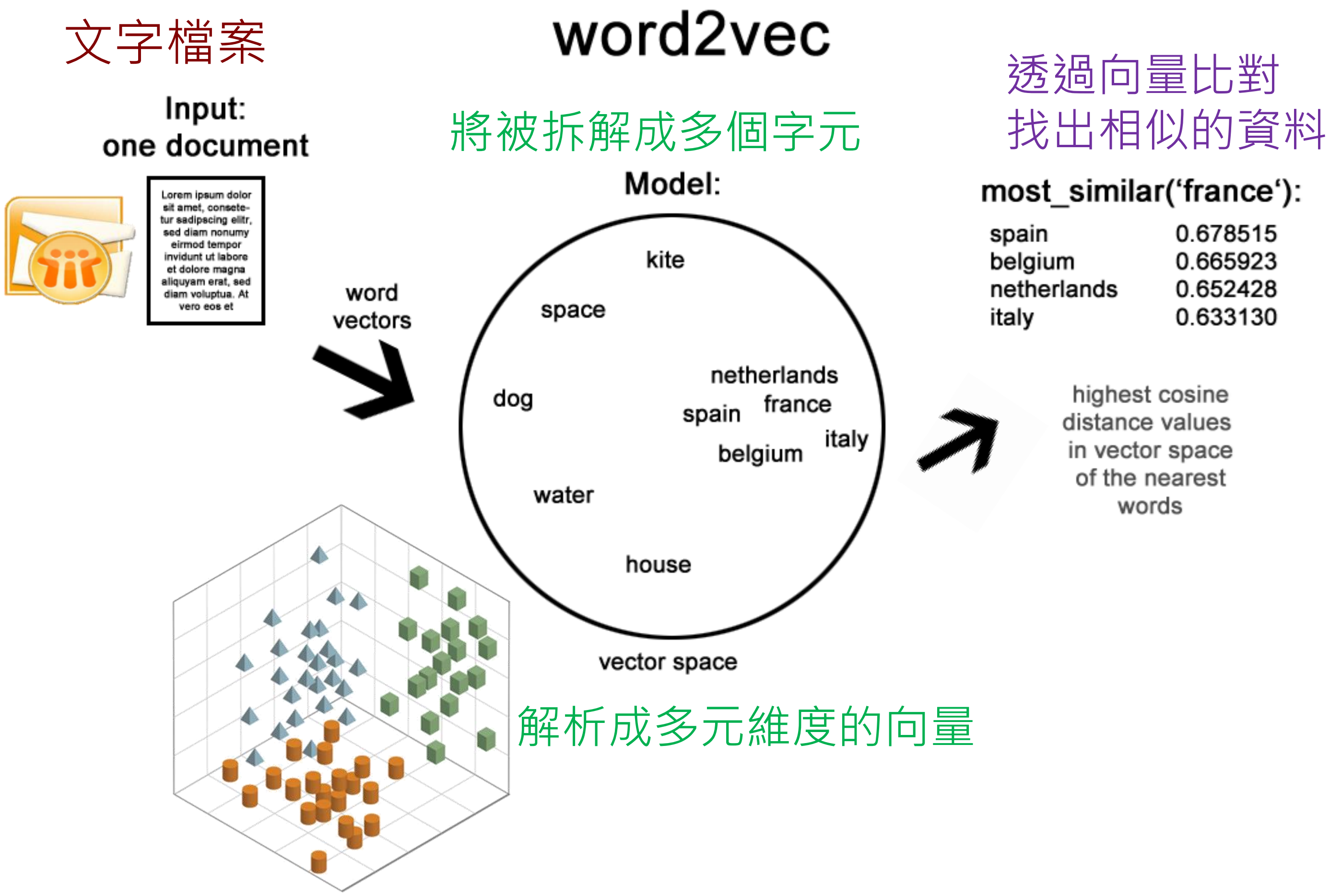
Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

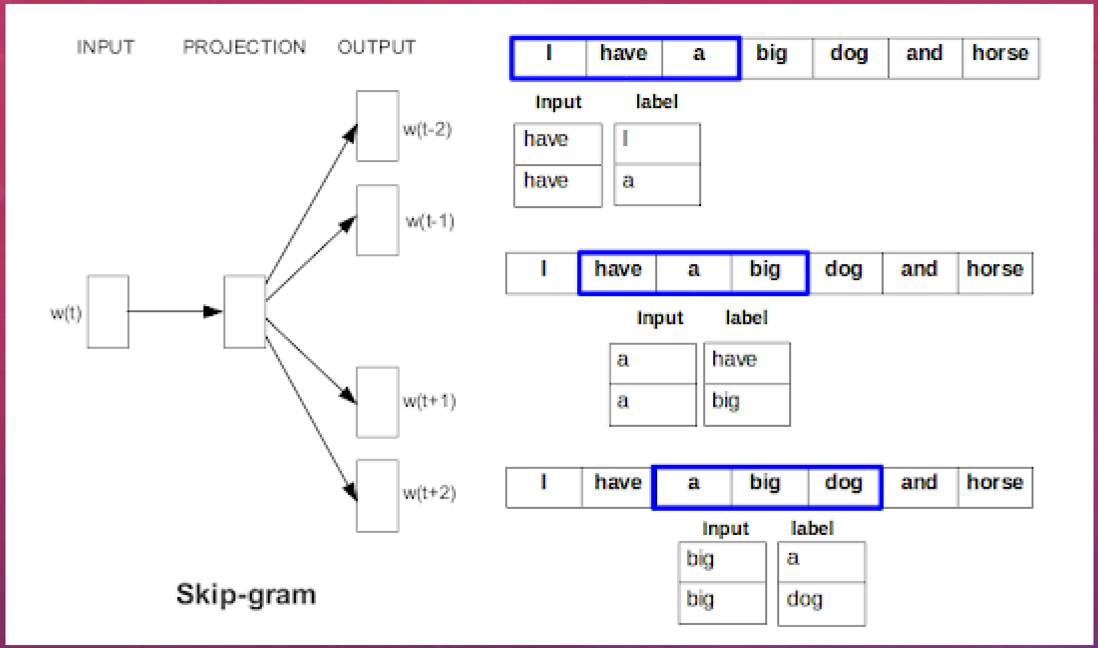
文字檔案

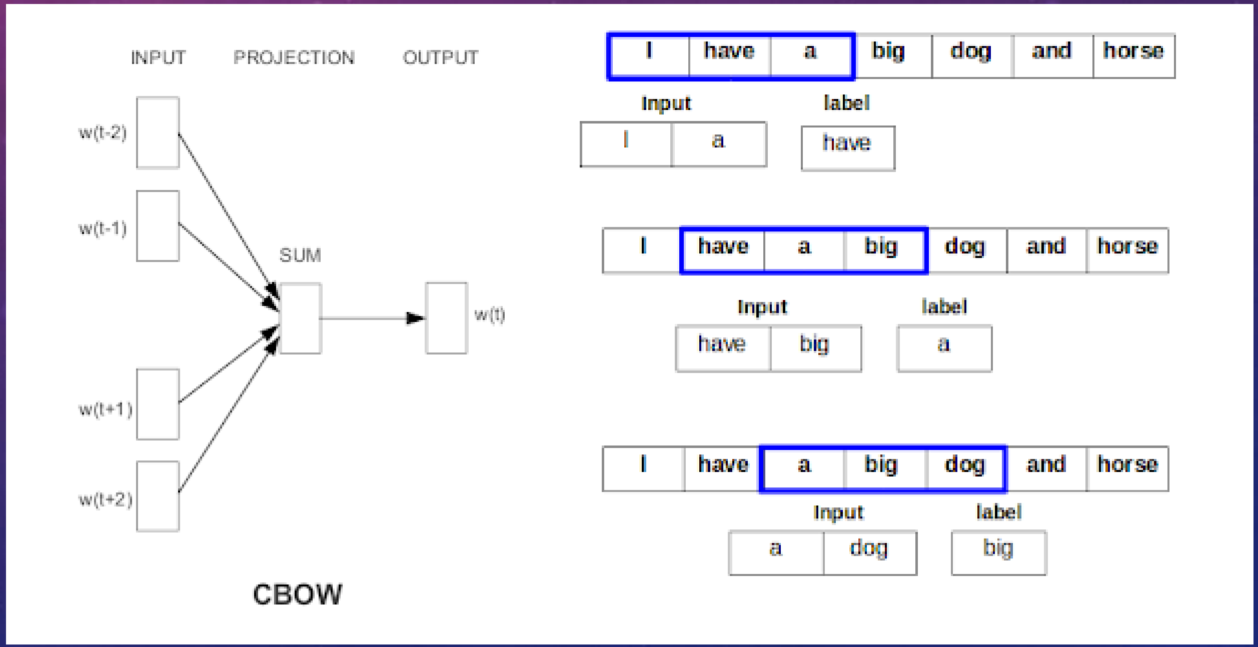
將被拆解成多個字元

透過向量比對
找出相似的資料

解析成多元維度的向量





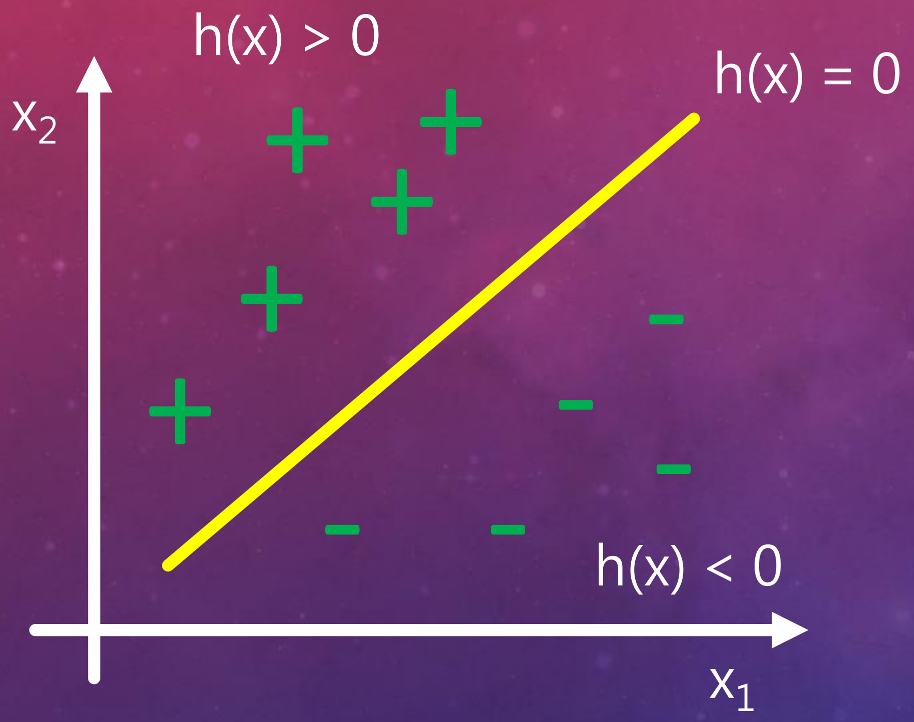


http://zongsoftwarenote.blogspot.com/2017/04/word2vec-model-introduction-skip-gram.html

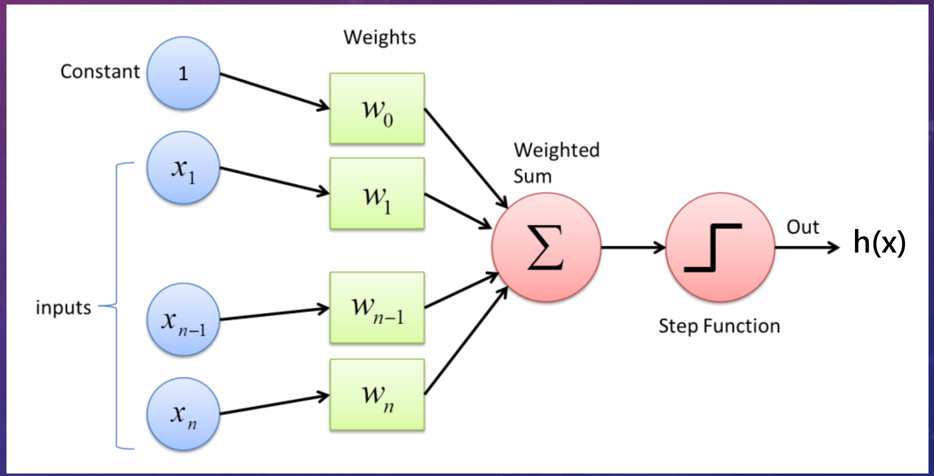
One Hot Encoding

```
The -> [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
cat -> [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
jump -> [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
over -> [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
the -> [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
dog -> [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
The -> [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
dog -> [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
ate -> [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
my -> [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
homework -> [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
```

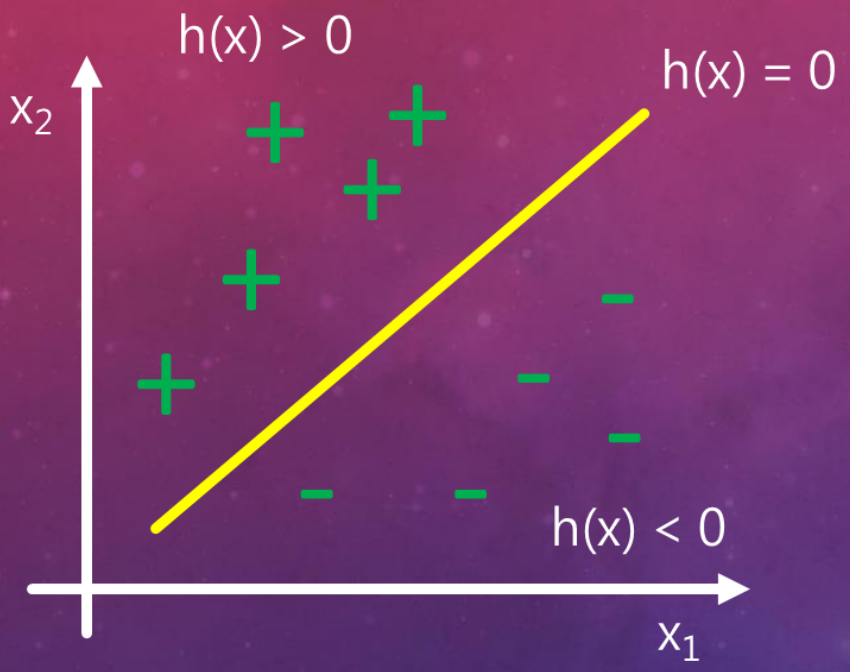
Perceptron Linear Algorithm



- Features: $x = (x_1, x_2)$
- Target: $y = +1$ or -1
- $h(x) = w_0 + w_1x_1 + w_2x_2$



Perceptron Linear Algorithm



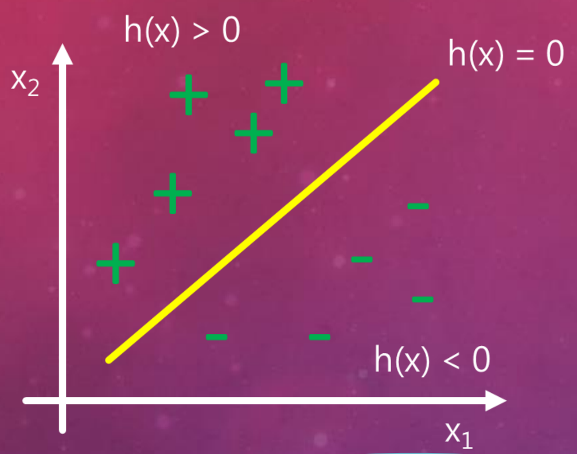
$$h(x) = w_0 + w_1x_1 + w_2x_2$$

$$scores = \sum_{i}^{N} w_i x_i + b$$

$$scores = \sum_{i}^{N+1} w_i x_i$$

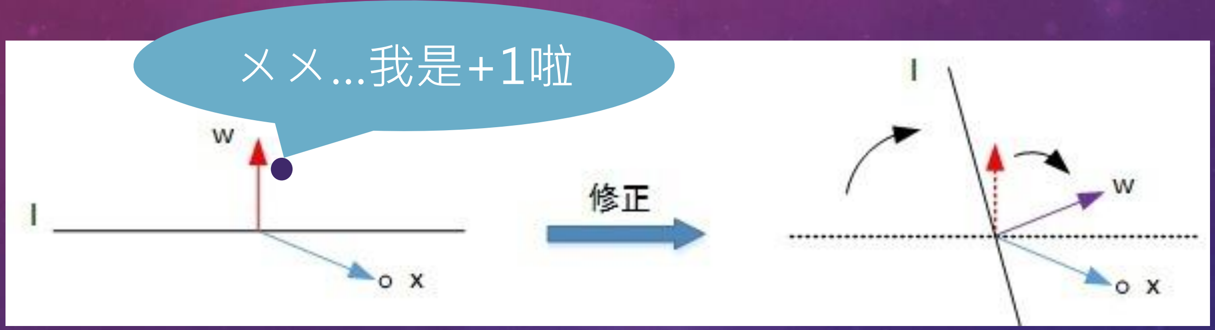
- 若 $scores \geq 0$ ，则 $\hat{y} = 1$
- 若 $scores < 0$ ，则 $\hat{y} = -1$

Perceptron Linear Algorithm



- 若 $scores \geq 0$ ，则 $\hat{y} = 1$
- 若 $scores < 0$ ，则 $\hat{y} = -1$

$$\overset{+}{w_{t+1}} = w_t + \overset{-}{y_t} x_t$$

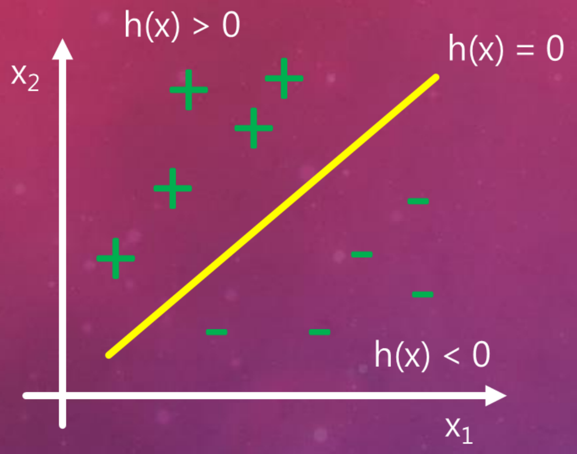


[Case 1]
y = 1 錯分成 y = -1

$$\overset{-}{w_{t+1}} = w_t + \overset{+}{y_t} x_t$$

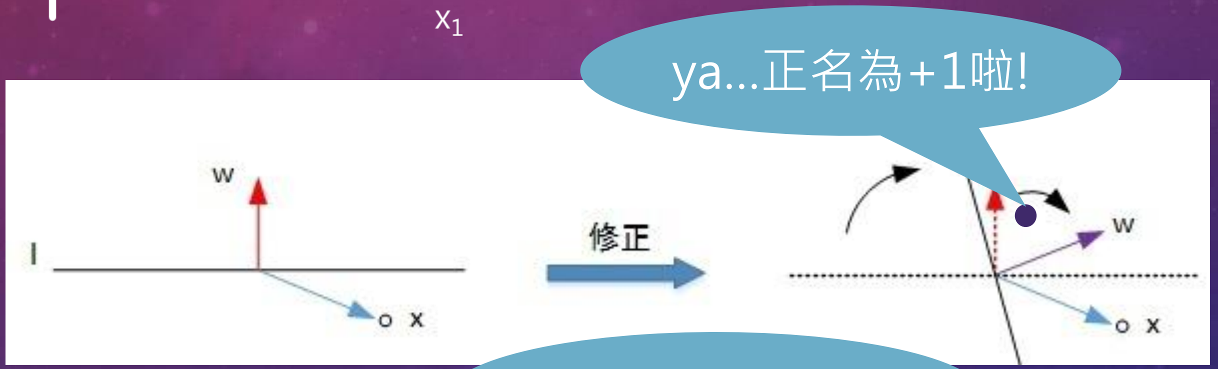
[Case 2]
y = -1 錯分成 y = 1

Perceptron Linear Algorithm



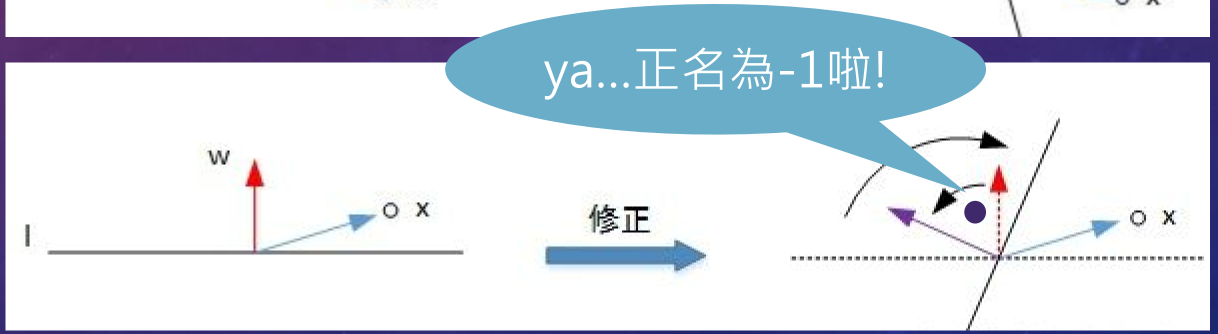
- 若 $scores \geq 0$ ，则 $\hat{y} = 1$
- 若 $scores < 0$ ，则 $\hat{y} = -1$

$$\overset{+}{w_{t+1}} = \overset{-}{w_t} + \overset{+}{y_t x_t}$$

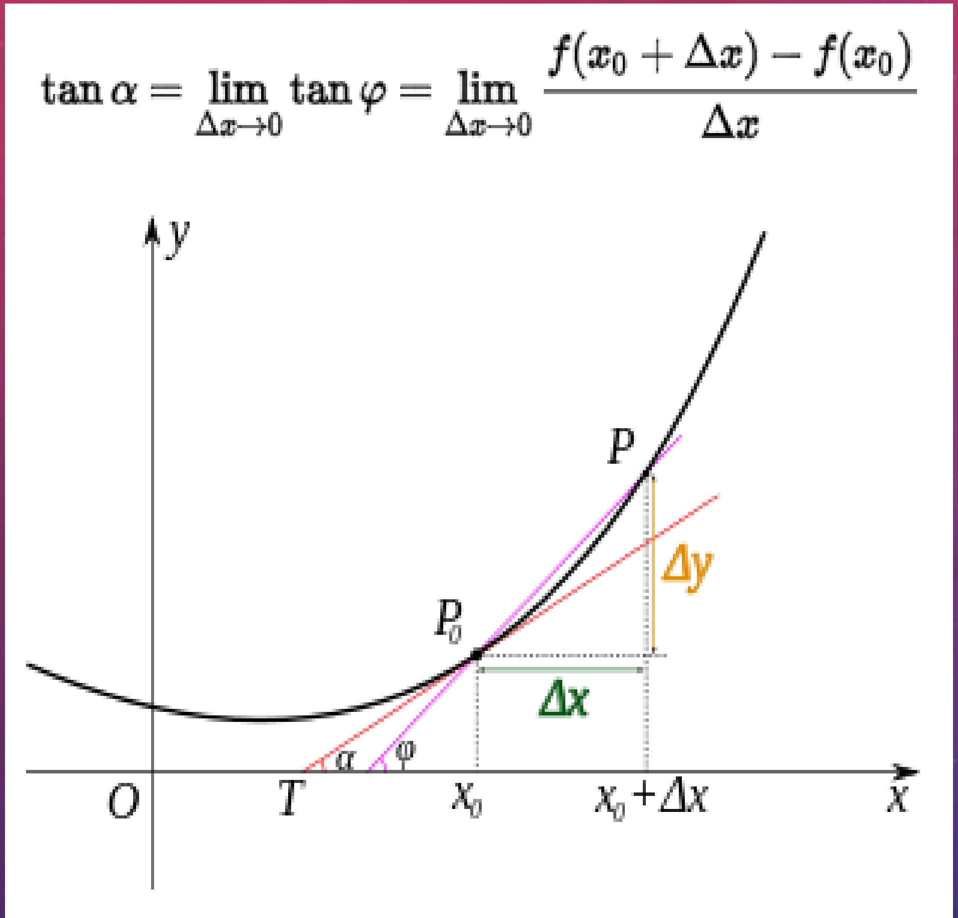


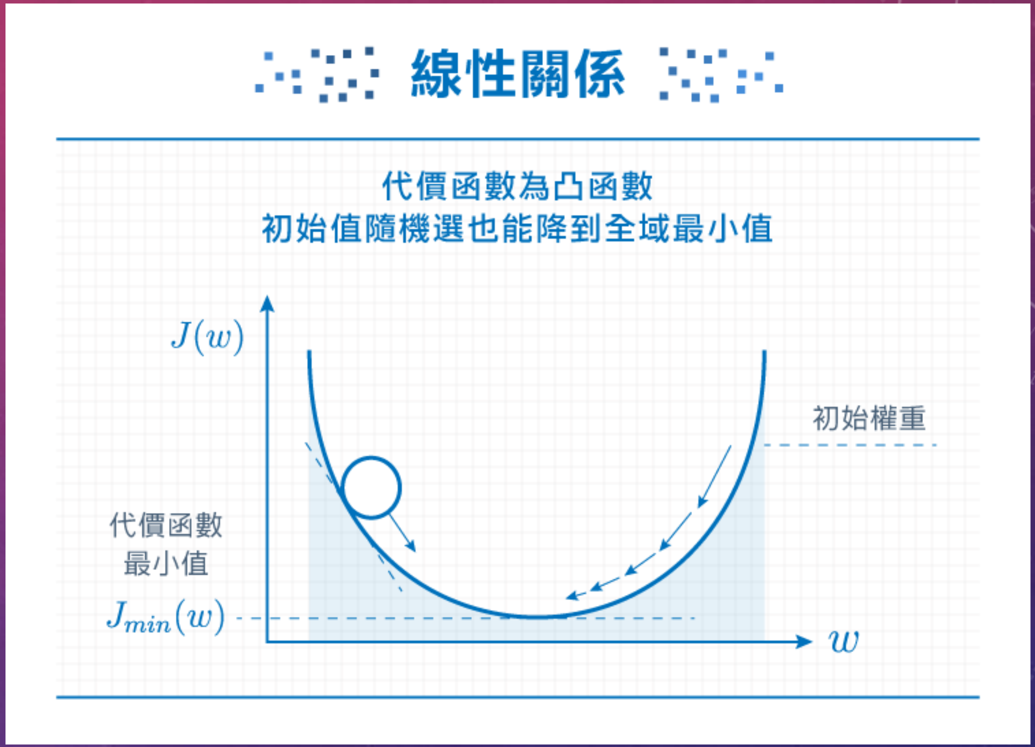
[Case 1]
y = 1 錯分成 y = -1

$$\overset{-}{w_{t+1}} = \overset{+}{w_t} + \overset{-}{y_t x_t}$$

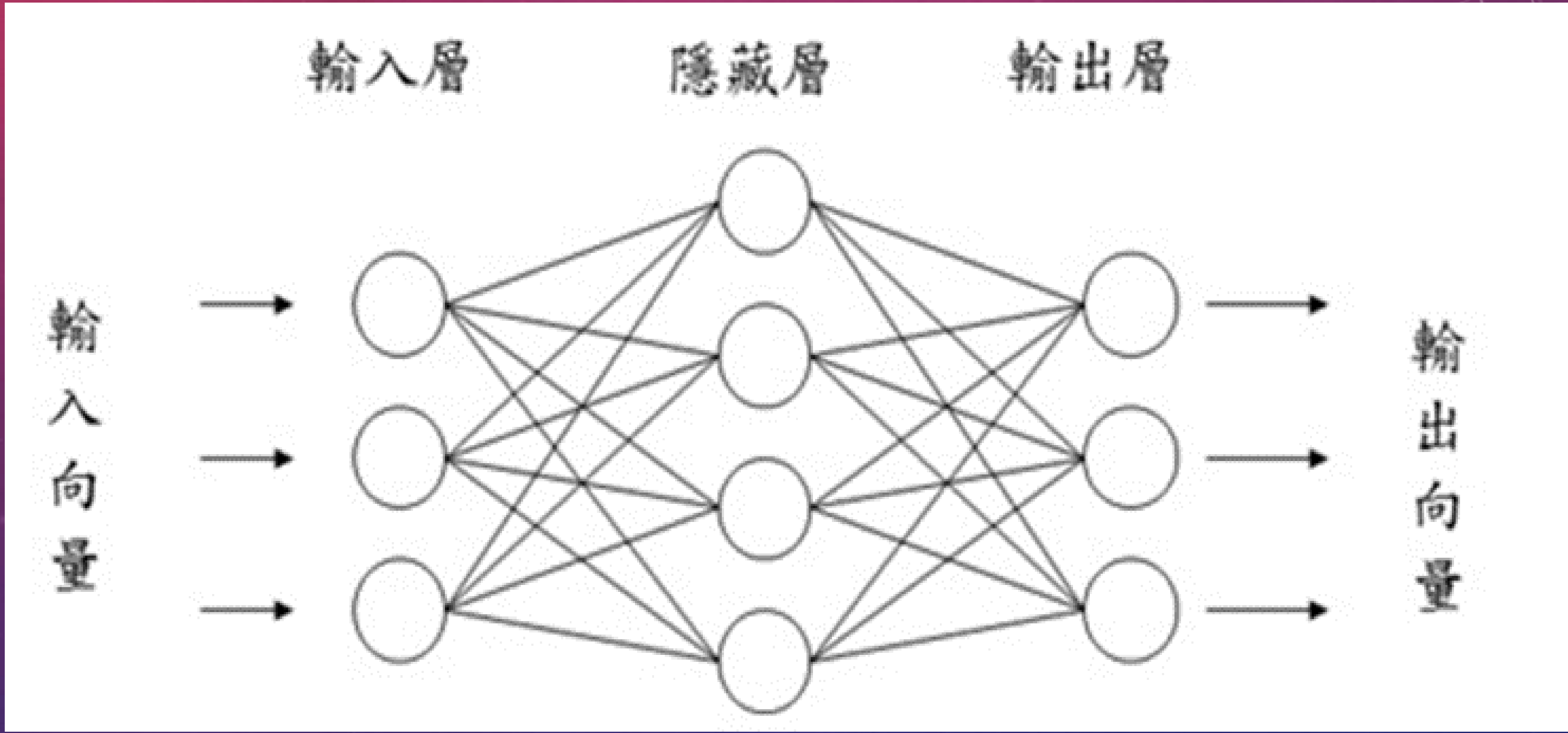


[Case 2]
y = -1 錯分成 y = 1


$\tan\alpha = \lim_{\Delta x \to 0} \tan\varphi = \lim_{\Delta x \to 0} \frac{f(x_0+\Delta x) - f(x_0)}{\Delta x}$
y
P
P_0
Δy
Δx
α
φ
O
T
x_0
$x_0 + \Delta x$
x

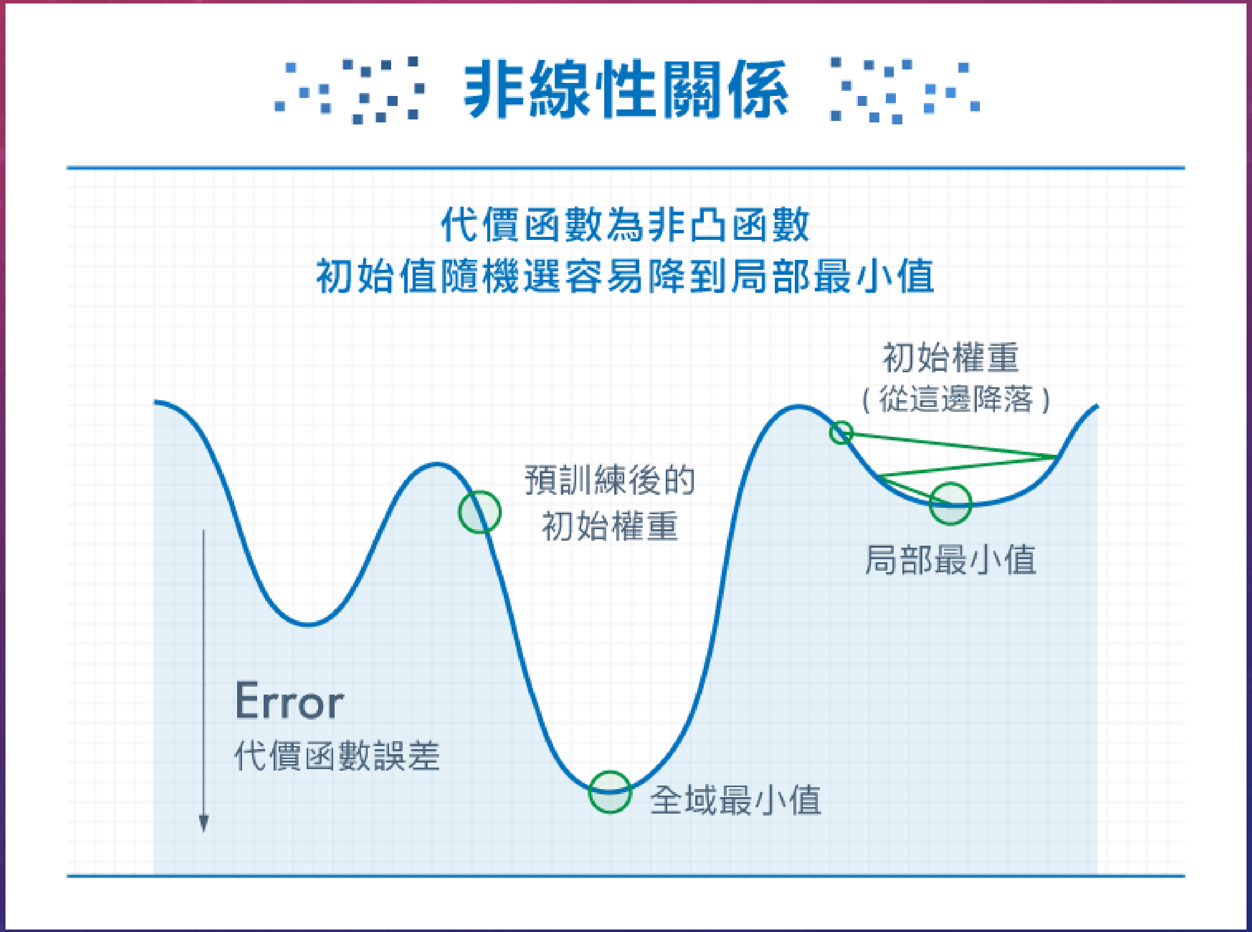

線性關係
代價函數為凸函數
初始值隨機選也能降到全域最小值
$J(w)$
初始權重
代價函數
最小值
$J_{min}(w)$
w

Multi-Layer Perceptron (MLP)



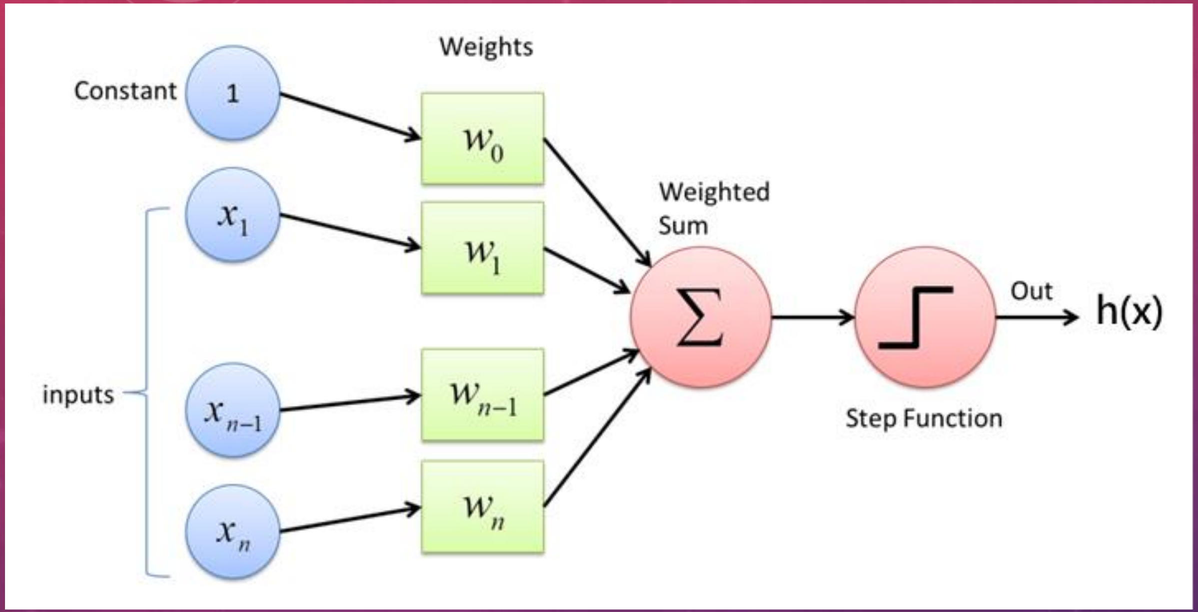
*線性組合 $w = a_1 v_1 + a_2 v_2 + a_3 v_3 + \cdots + a_n v_n$

梯度消失

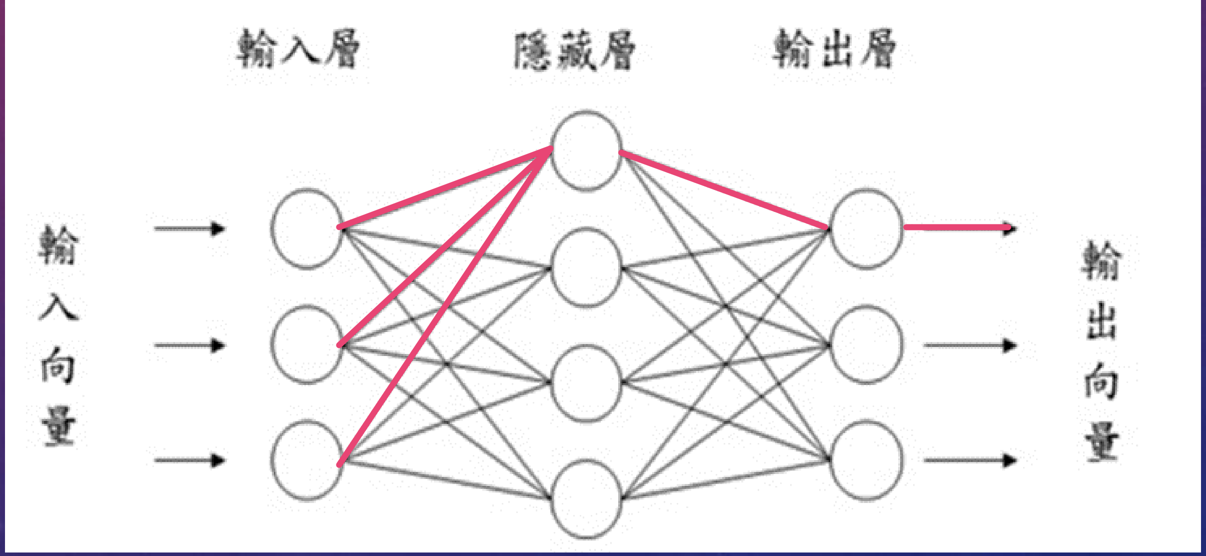


*現實世界的資料多為非線性，因此激活函數通常也是使用非線性函數(非凸函數)傳遞

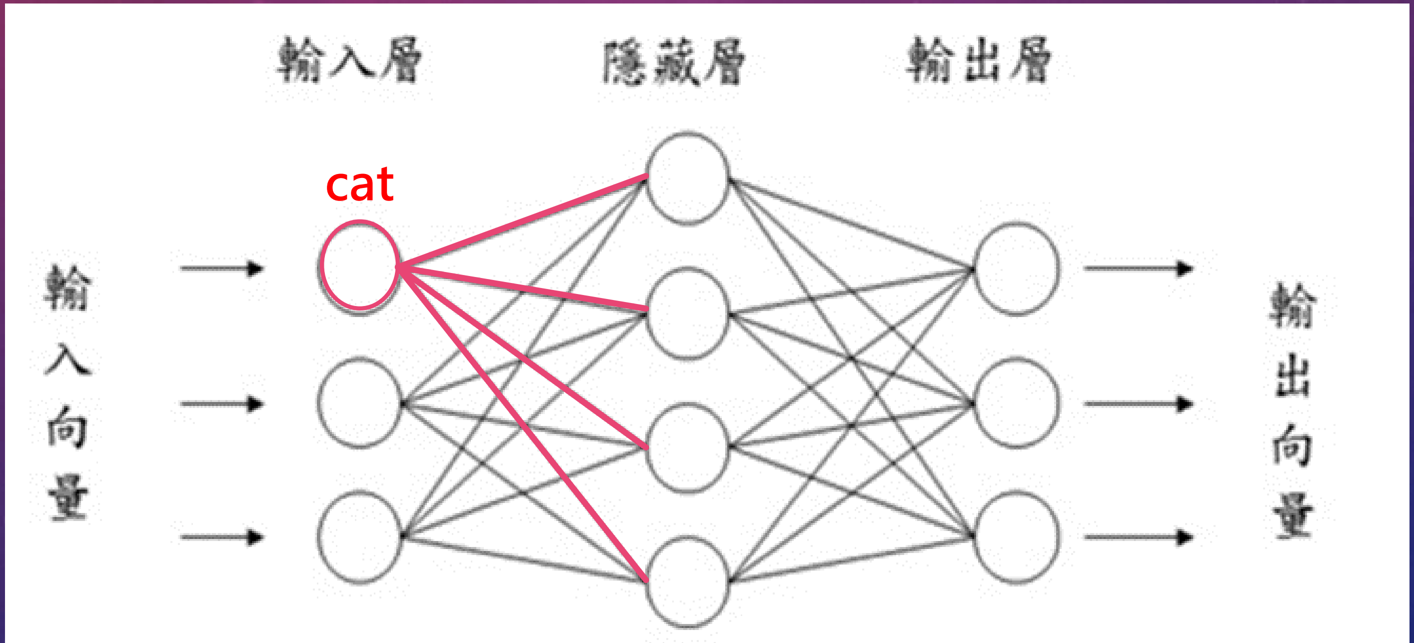
PLA

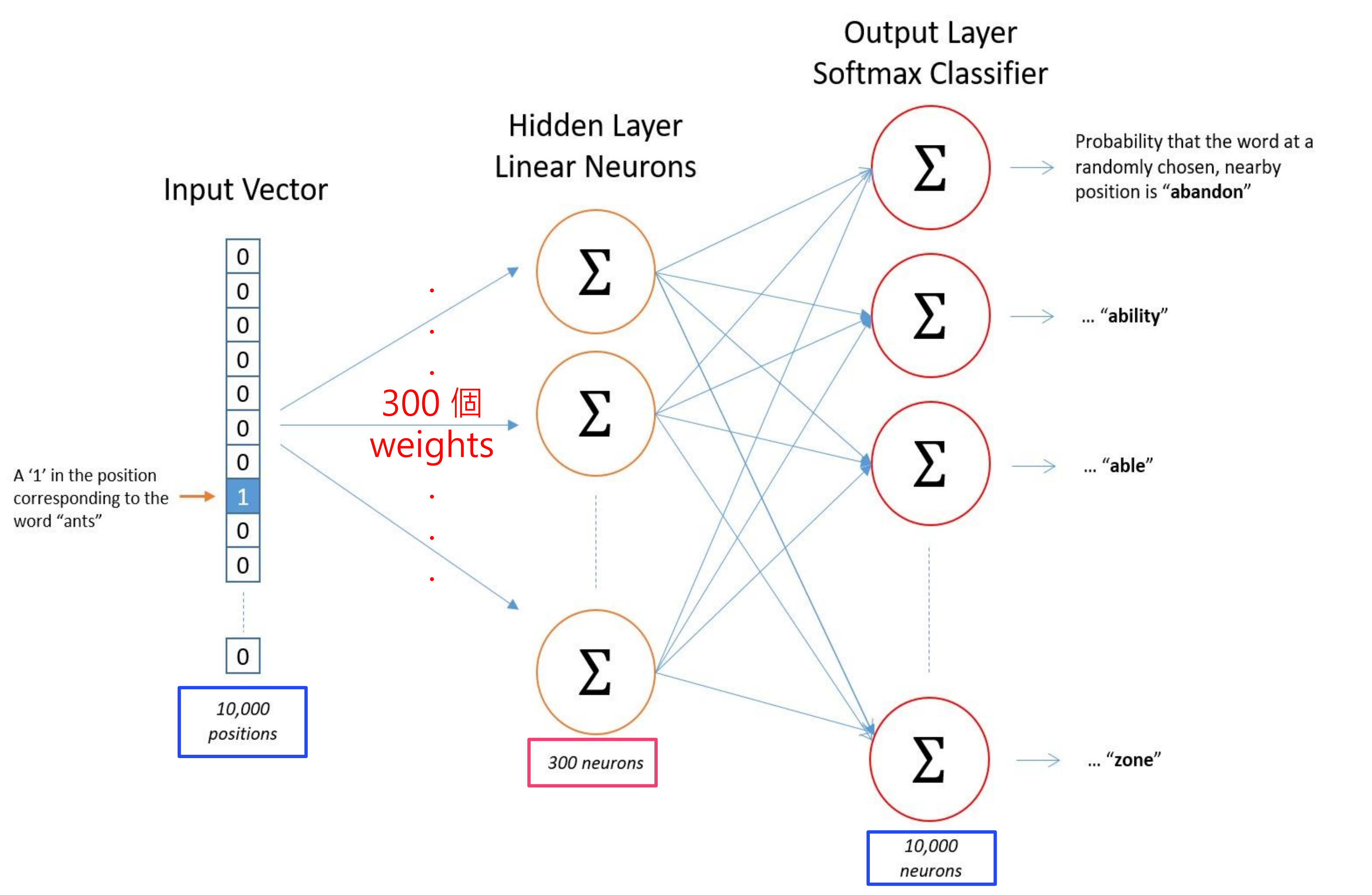

Weights
Constant
1
w_0
x_1
w_1
Weighted Sum
Σ
Out
h(x)
inputs
x_{n-1}
w_{n-1}
x_n
w_n
Step Function

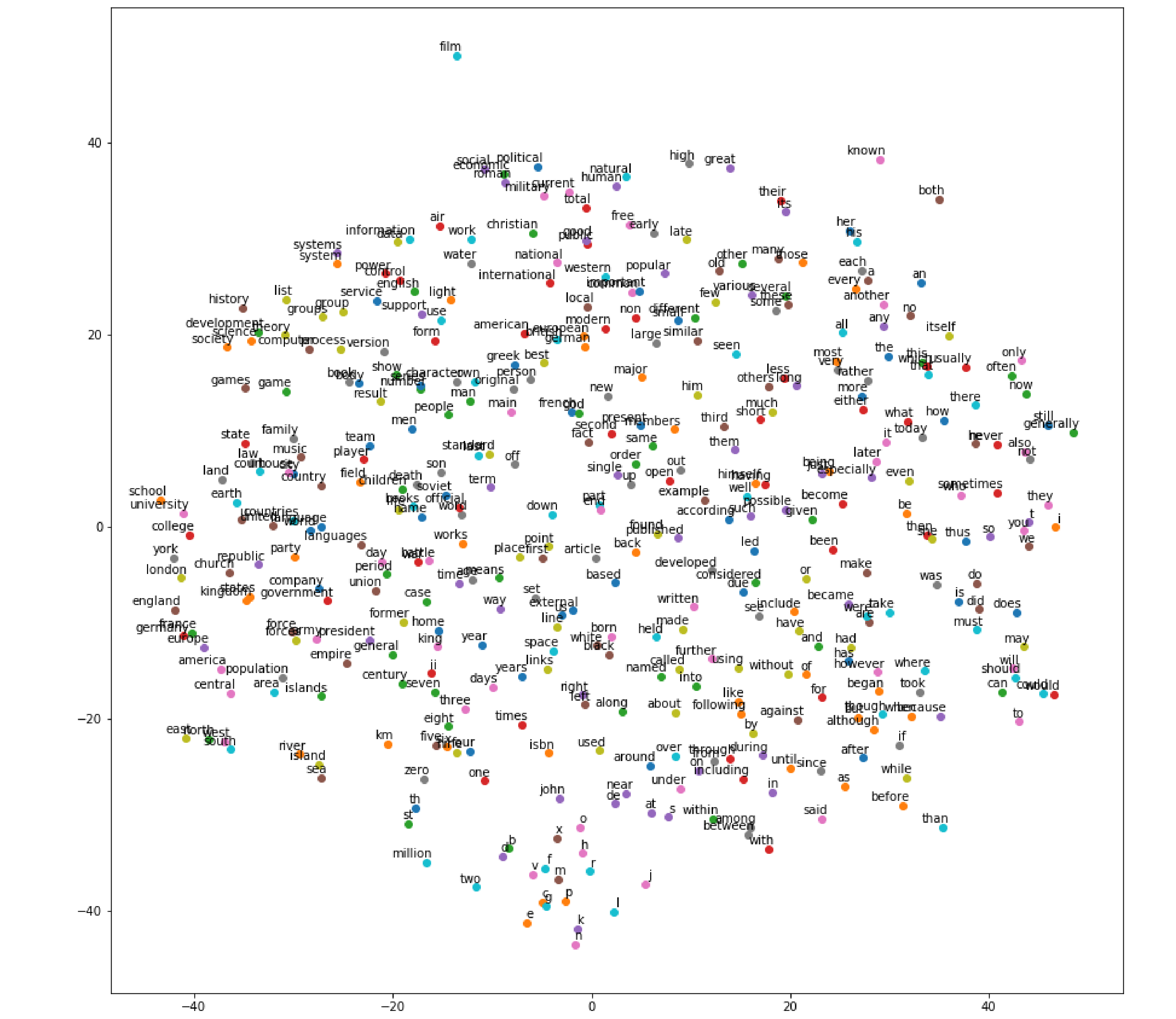
MLP


輸入層
隱藏層
輸出層
輸入向量
輸出向量

```
The -> [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
cat -> [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
jump -> [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
over -> [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
the -> [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
dog -> [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
The -> [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
dog -> [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
ate -> [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
my -> [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
homework -> [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
```




Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
300 個
weights
10,000 positions
300 neurons
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "**abandon**"
... "**ability**"
... "**able**"
... "**zone**"


40
20
0
−20
−40
−40
−20
0
20
40
film
known
social
political
economic
roman
military
current
natural
human
total
high
great
their
its
both
her
his
information
data
air
work
christian
free
early
late
systems
system
water
national
international
western
popular
old
other
many
those
each
every
another
an
history
list
service
group
groups
support
light
local
important
common
few
various
several
these
some
no
any
development
theory
science
society
computer
process
version
american
european
british
german
modern
non
different
small
large
similar
all
itself
form
greek
best
seen
most
very
the
this
which
that
usually
only
often
now
games
game
book
body
show
character
own
person
original
major
less
others
long
rather
more
either
number
result
man
main
people
new
him
there
french
god
present
members
much
short
what
how
still
generally
men
second
fact
third
it
today
never
also
not
state
family
team
player
standard
last
same
them
later
music
order
out
especially
being
even
law
land
death
son
off
single
up
open
himself
having
well
sometimes
they
country
field
children
soviet
term
example
become
school
university
earth
books
official
down
part
end
according
possible
given
be
countries
united
language
name
world
found
published
then
thus
you
so
college
languages
works
point
first
place
article
back
led
been
we
york
party
republic
day
battle
war
developed
make
london
church
period
means
time
based
considered
due
or
was
do
company
union
set
written
became
is
did
does
kingdom
states
government
case
external
include
take
were
are
must
england
former
home
line
made
see
have
held
france
germany
force
forces
army
president
born
white
black
had
has
and
may
europe
king
year
space
general
further
using
without
of
however
where
will
should
america
population
empire
links
years
named
called
into
began
took
can
could
would
central
area
islands
century
seven
days
right
left
like
following
for
three
along
about
against
though
when
because
to
east
north
west
south
eight
times
although
by
if
river
island
km
five
six
four
isbn
used
over
through
during
after
around
from
on
until
since
sea
zero
one
including
while
as
th
john
near
de
under
in
before
st
at
s
within
among
between
said
than
with
million
o
x
b
d
h
f
r
v
m
j
two
g
c
p
l
e
k
n

THINKING TIME

POTENTIAL APPLICATIONS

FOUNDATIONS OF COMPUTER VISION

電腦視覺的原理與應用

HOW WE TEACH COMPUTERS TO UNDERSTAND PICTURES

電腦視覺

- 利用攝影機和電腦代替人眼對目標進行識別、跟蹤和測量等機器視覺，並做圖像處理，用電腦處理為更適合人眼觀察或傳送給儀器檢測的圖像。
- 人工智慧的主要研究問題是：如何讓系統具備「計劃」和「決策能力」，使之完成特定的動作，如移動機器人通過特定環境。
  - 此問題中，電腦視覺可作為感知器，為決策提供資訊。其中研究方向包括模式識別和機器學習，因此電腦視覺被看作人工智慧的分支。

電腦視覺應用

- 作為一個工程學科，電腦視覺基於相關理論來建立電腦視覺系統。這類系統的組成部分包括：
  1. 過程控制(Process Control)（如工業機器人和無人駕駛車）
  2. 事件監測(Event Monitoring)（如圖像監測）
  3. 資訊組織(Information Organization)（如圖像資料庫和圖像序列的索引建立）
  4. 物體與環境建模（如工業檢查，醫學圖像分析和拓撲建模）
  5. 交感互動（如人機互動的輸入裝置）

卷積神經網路

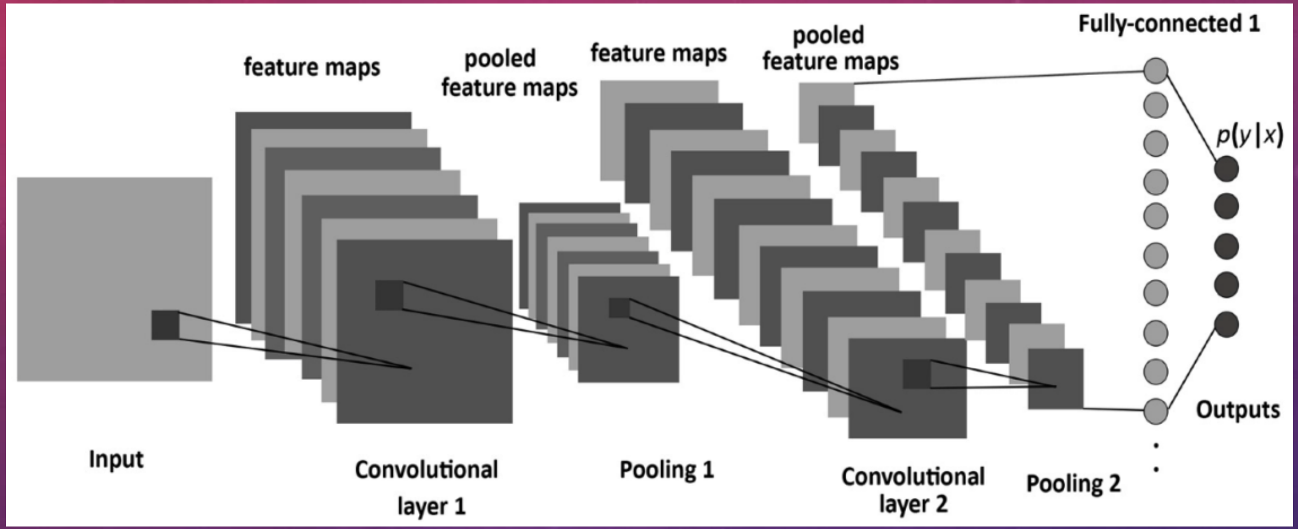
CONVOLUTIONAL NEURAL NETWORK

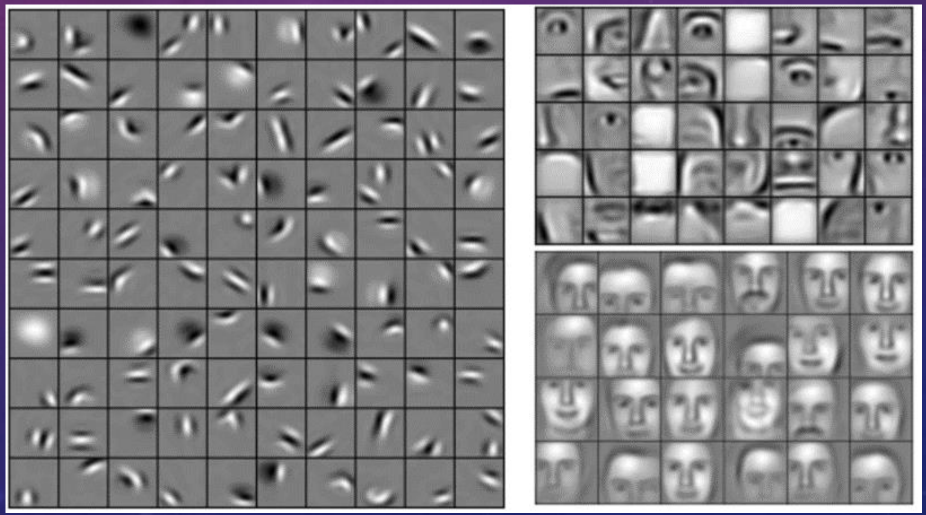
參考文獻

HTTPS://GOO.GL/FJVJG1

HTTPS://GOO.GL/Q5YKPK

CONVOLUTIONAL NEURAL NETWORK



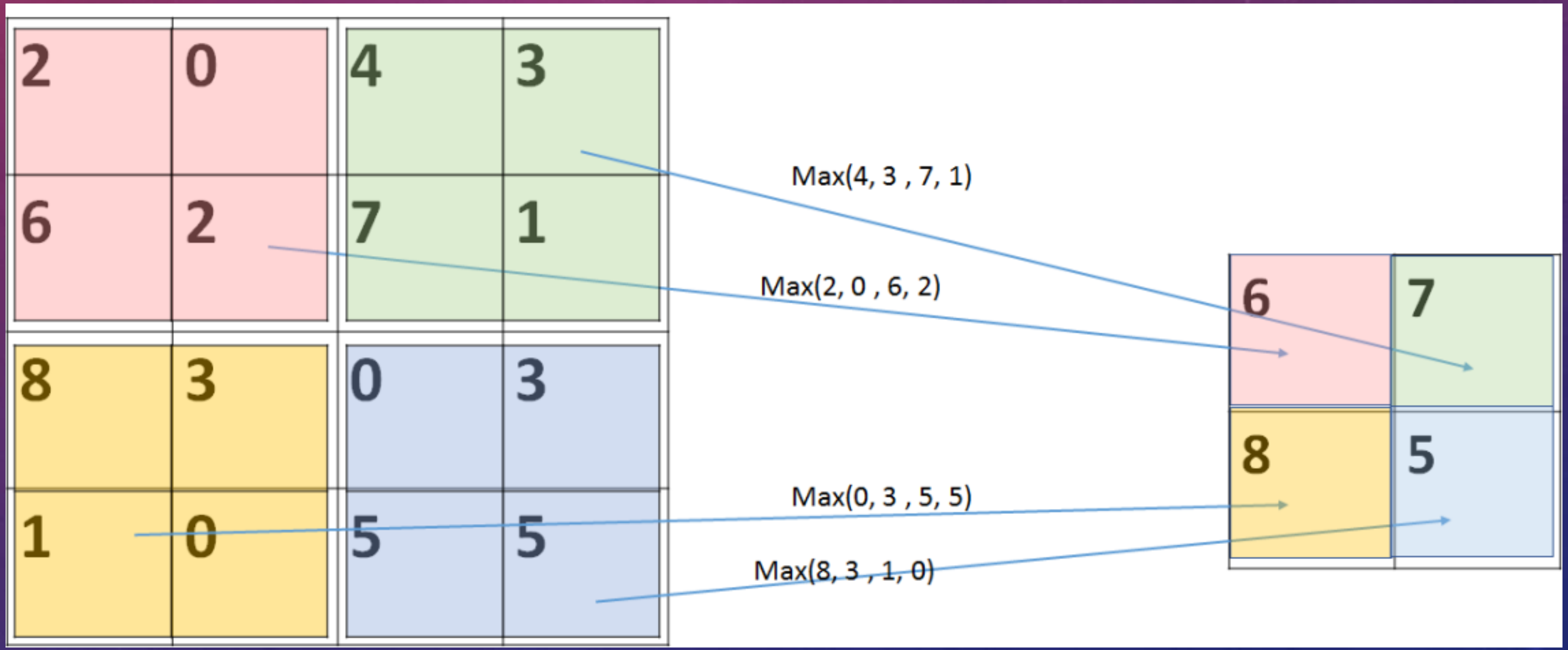


CONVOLUTIONAL NEURAL NETWORK

卷積層的運作

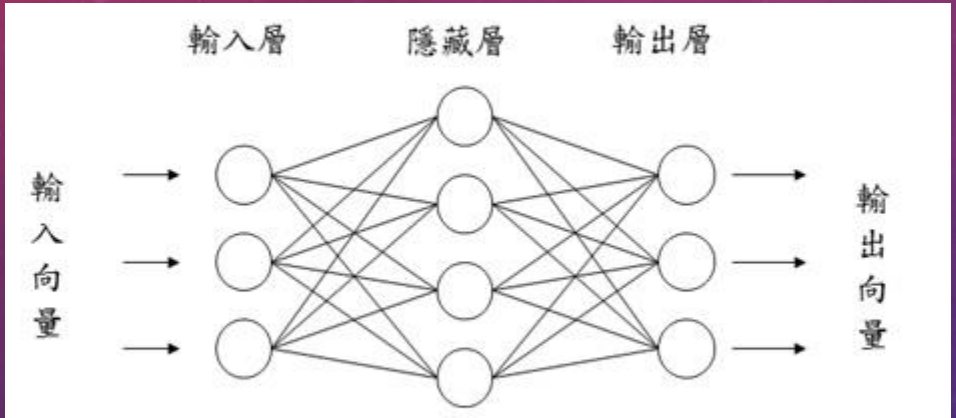
CONVOLUTIONAL NEURAL NETWORK

最大池化層的運作

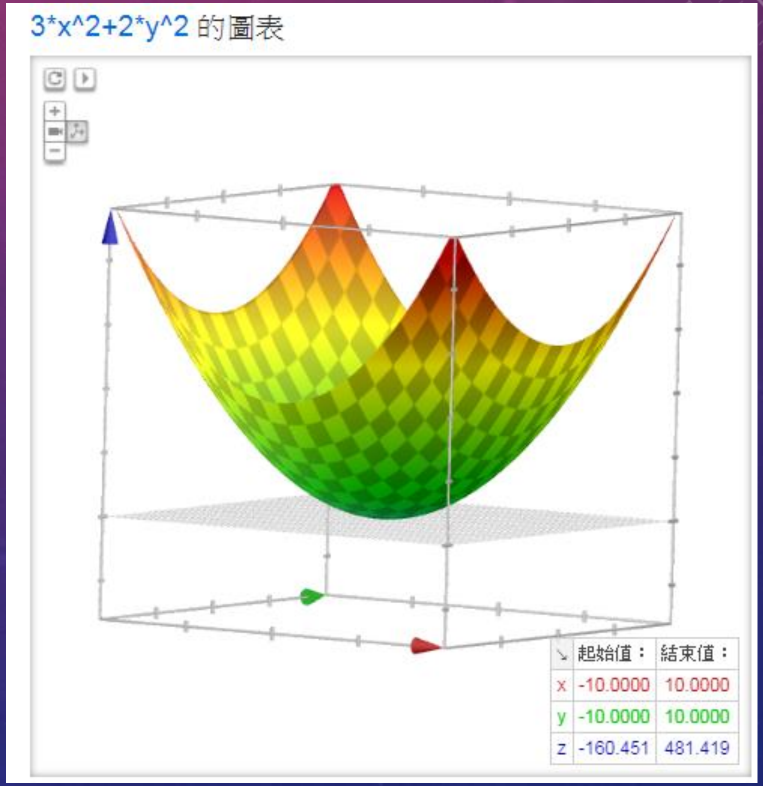


CONVOLUTIONAL NEURAL NETWORK

全連接層的運作



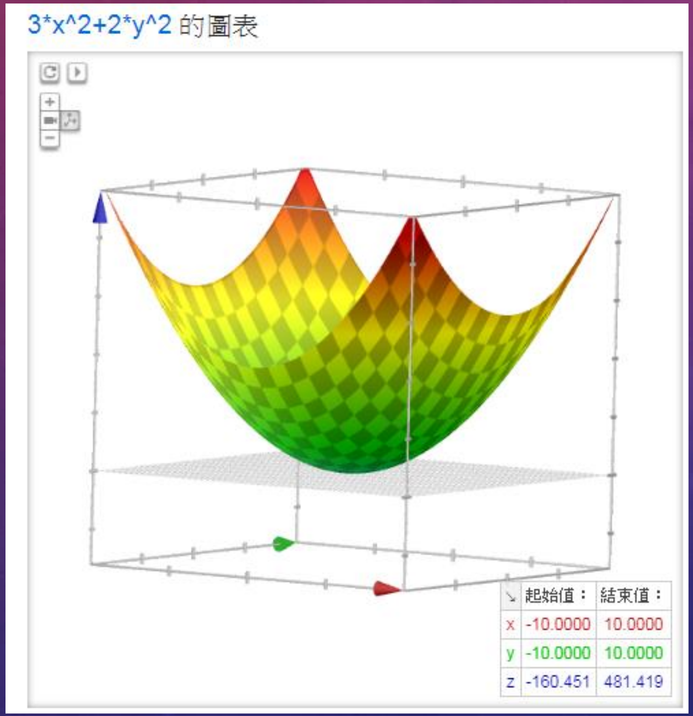
$3x^2+2y^2$

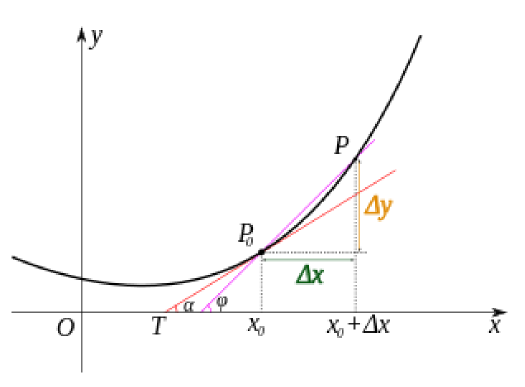


CONVOLUTIONAL NEURAL NETWORK

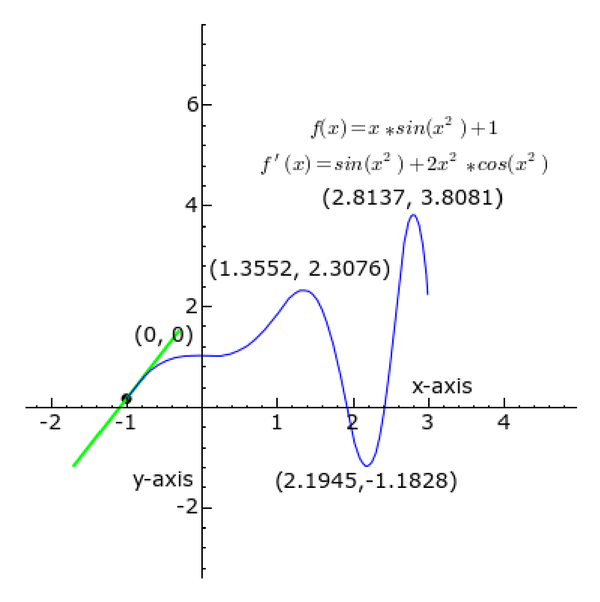
全連接層的運作

$3x^2+2y^2$

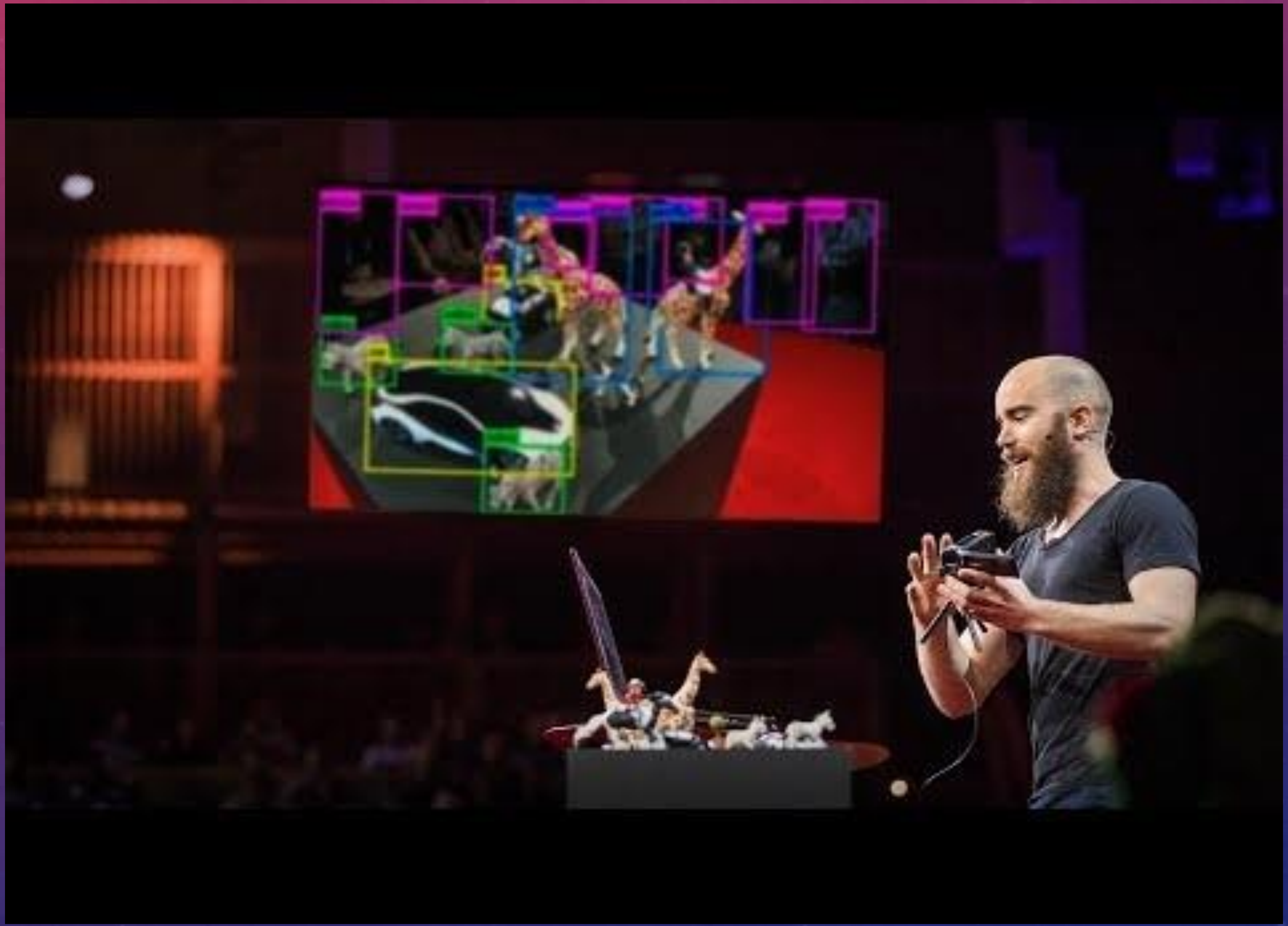




$$\tan\alpha = \lim_{\Delta x \to 0} \tan\varphi = \lim_{\Delta x \to 0} \frac{f(x_0+\Delta x) - f(x_0)}{\Delta x}$$



HOW COMPUTERS LEARN TO RECOGNIZE OBJECTS INSTANTLY



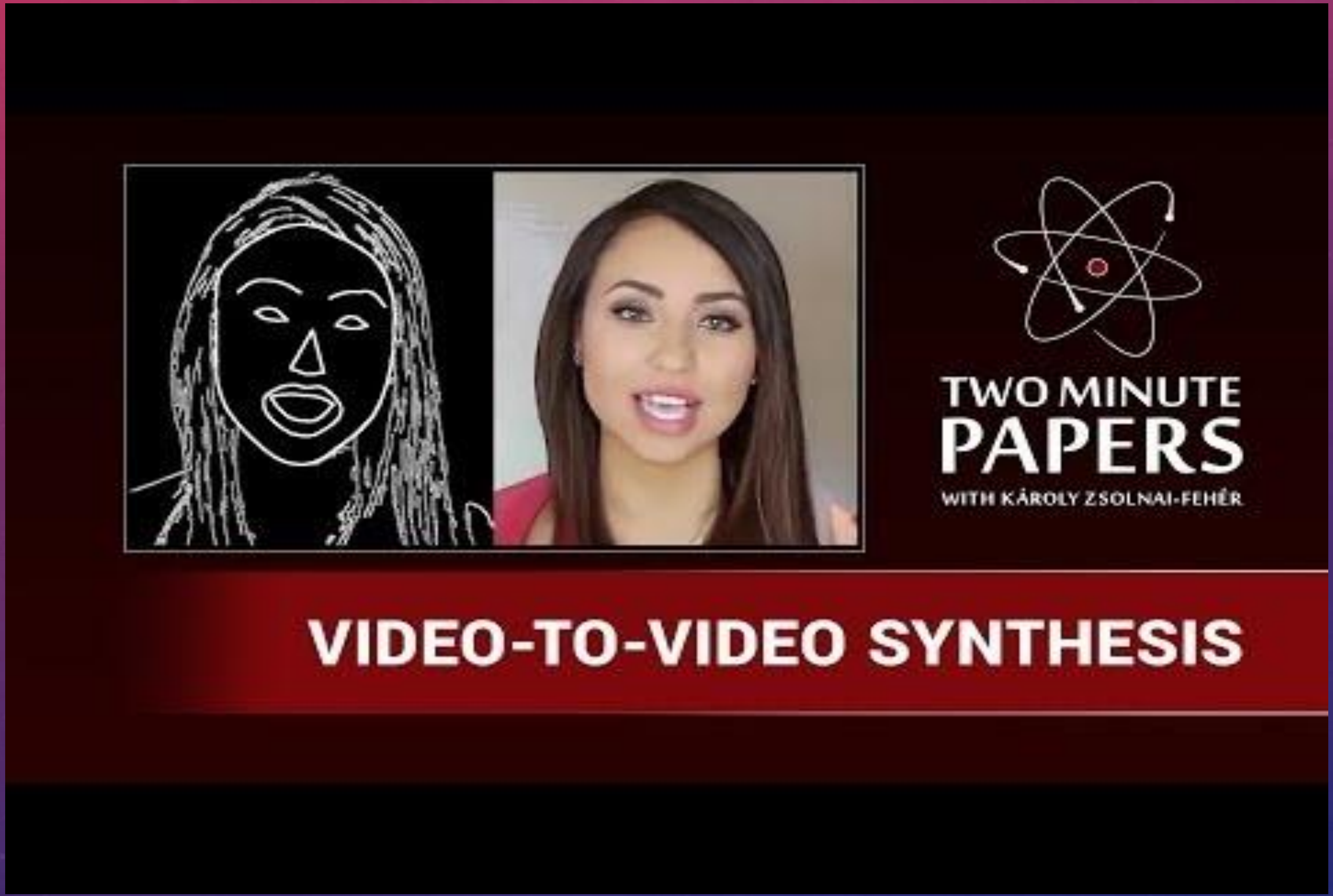
http://mropengate.blogspot.com/2018/06/yolo-yolov3.html

實際案例

貓狗辨識的結果

```
[0.14723705 0.98256    ]
[0.04469623 0.99915826]
[0.99998796 0.00834211]
[0.99993527 0.01661933]
[0.99877125 0.05415697]
[0.43036035 0.6431105  ]
[0.20522958 0.95668554]
[0.8880429  0.28058085]
[0.08643436 0.9956601  ]
[0.99908304 0.04826429]
[0.9999168  0.01841162]
[0.1397599  0.98483086]
[0.99558544 0.08876048]
[0.43750185 0.6269166  ]
[0.10178897 0.9934009  ]
[0.10858331 0.9922093  ]
```

VIDEO TO VIDEO



The paper "Video-to-Video Synthesis" and its source code is available here:
https://tcwang0509.github.io/vid2vid/
https://github.com/NVIDIA/vid2vid

智慧視覺系統機器人

視覺整合信號轉換的體感操控機械手臂

THANK YOU