FOUNDATIONS OF NATURAL LANGUAGE PROCESSING
自然語言處理的基礎

張家瑋 博士
國立臺中科技大學資訊工程系助理教授

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)

DATA PREPARATION

- Data preprocessing and cleaning
  - Preprocess data in order to reduce noise and handle missing values
  - 斷字, 斷詞
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
  - 移除stop words, 擷取有用資訊(TF-IDF)
- Data transformation
  - Generalize and/or normalize data
  - 轉成向量(Vector representation)

SEGMENTATION

- Segment by word/ sentence
- Segment in English
  - In English, we can directly segment the word by space “ ”
  - Ex: I love machine learning. → [I, love, machine, learning]
- Segment in Chinese
  - In Chinese, we segment the word by meaningful word rather than directly segment by characters.
  - Ex: 我喜歡機器學習 → [我, 喜歡, 機器, 學習]

    rather than [我, 喜, 歡, 機, 器, 學, 習]

根據字詞結構將一句話斷字

Dear 小明,
這是目前公司的最新技術，利用 apples 和 pens 的特性可以讓產能最佳化............

Dear, 小明, 這是, 目前, 公司, 的, 最新, 技術, 利用, apples, 和, pens, 的, 特性, 可以, 讓, 產能, 最佳化, ...

REMOVING STOP WORDS

- Remove the word which is meaningless.
- Usually do after segment.
- Remove stop words in Chinese
  - Example of stop words: 的, 了, 且, 個, 是
  - Ex: 今天的空氣品質不好 → [今天, 空氣, 品質, 不好]
- Remove stop words in English
  - Example of stop words: is, the, an, and, a
  - Ex: Today 's air quality is not good → [Today's, air, quality, not, good]

Dear, 小明, 這是, 目前, 公司, 的, 最新, 技術, 利用, apples, 和, pens, 的, 特性, 可以, 讓, 產能, 最佳化, …………

移除stop-word

小明, 目前, 最新, 技術, 利用, apples, pens, 特性, 產能, 最佳化, …………

STEMMING

- Stemming is to transform the word into its original type by removing word endings such as -s , -ed and -ing.
  - “bikes” is replaced with “bike” ,
  - “raining” is replaced with “rain”
  - “tried” is replaced with “try”

小明, 目前, 最新, 技術, 利用, apples, pens, 特性, 產能, 最佳化, …………

stemming

小明, 目前, 最新, 技術, 利用, apple, pen, 特性, 產能, 最佳化, …………

REPRESENTATION

- Select features from the data
- Transform data into vector model

- Ex)
  - WordNet
  - TF-IDF (Term Frequency - Inverse Document Frequency)
  - Word2Vec

自然語言理解

NATURAL LANGUAGE UNDERSTANDING

WORD-SENSE DISAMBIGUATION(1/2)

- Ambiguity: a word or phrase with multiple meanings.
  1. “procure” (I will get the drinks)
  2. “become” (she got scared)
  3. “have” (I have got three dollars)
  4. “understand” (I get it)

WORDNET

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: apple Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

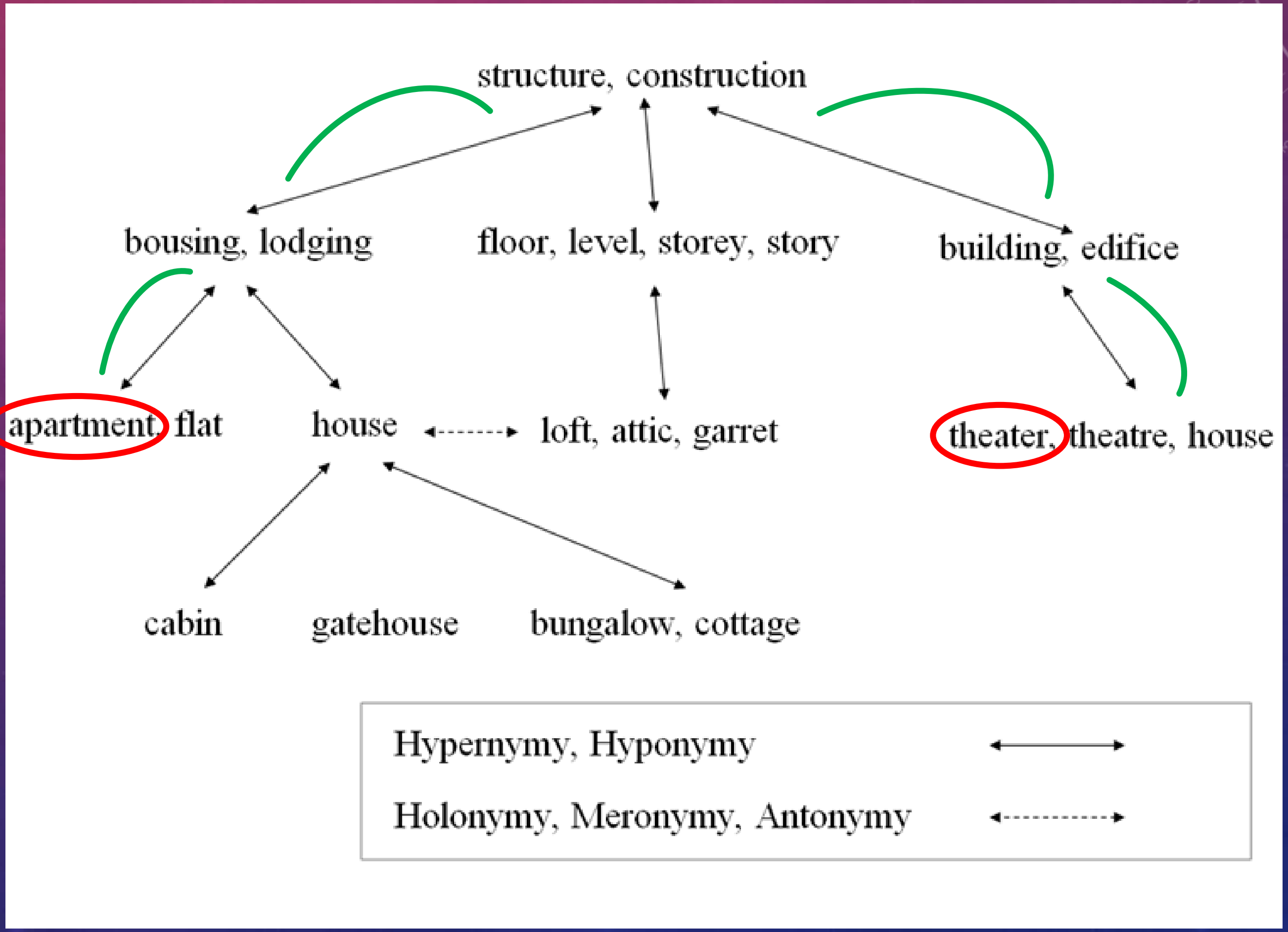
Noun

- S: (n) **apple** (fruit with red or yellow or green skin and sweet to tart crisp whitish flesh)
- S: (n) **apple**, orchard apple tree, Malus pumila (native Eurasian tree widely cultivated in many varieties for its firm rounded edible fruits)

http://wordnetweb.princeton.edu/perl/webwn

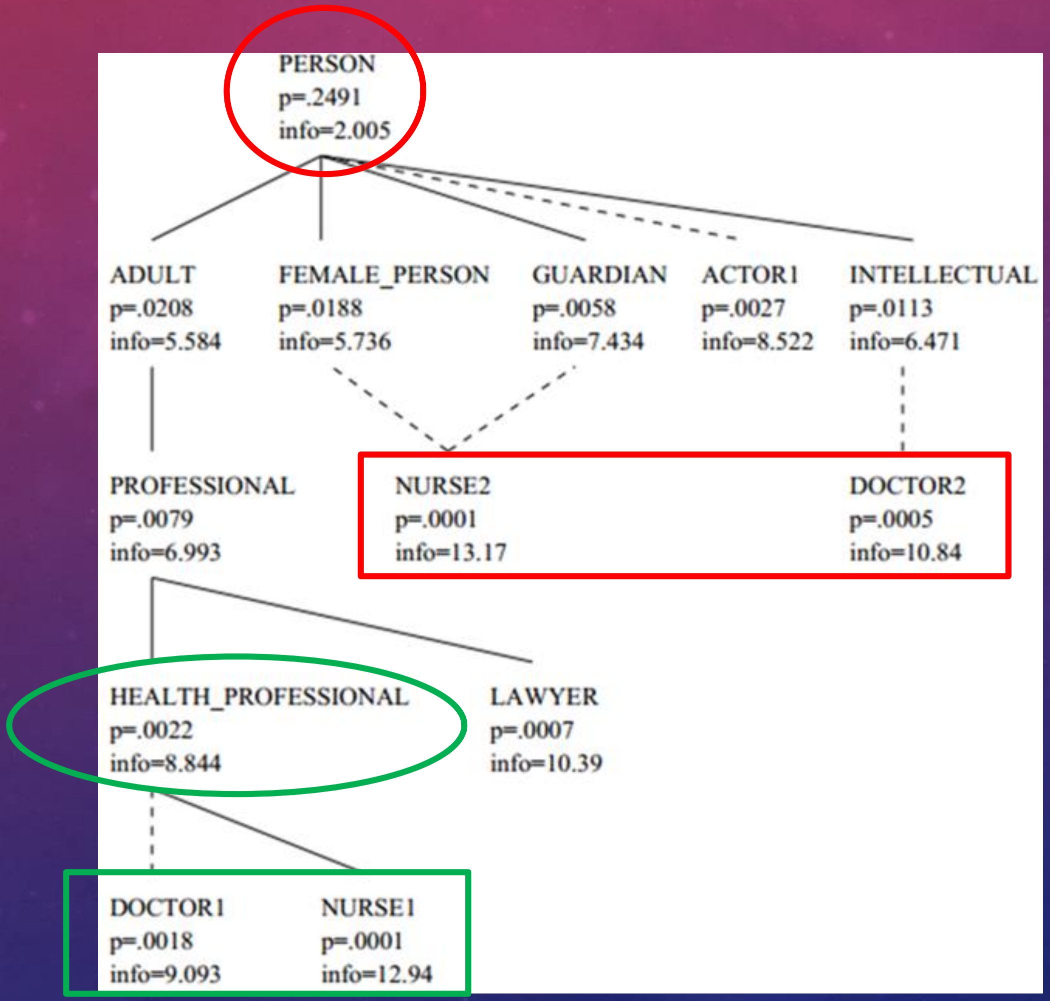
WORD-SENSE DISAMBIGUATION(2/2)

- Distance-based: PATH (Rada, Mili, Bicknell, & Blettner, 1989)

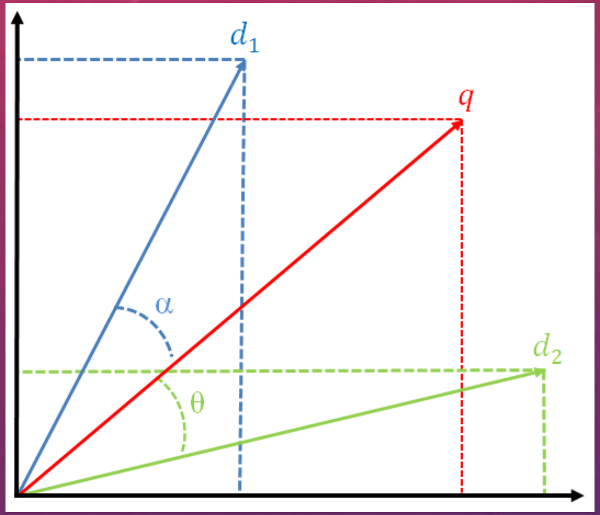


WORD-SENSE DISAMBIGUATION(2/2)

- Information Content-based: RES (Resnik, 1995)



WORD-SENSE DISAMBIGUATION(2/2)



- Gloss-based: VECTOR (Patwardhan, 2003)

Cute	Cunning
1. attractive especially by means of smallness or prettiness or quaintness	1. attractive especially by means of smallness or prettiness or quaintness
2. obviously contrived to charm	2. marked by skill in deception
	3. showing inventiveness and skill

STANFORD PARSER

Stanford Parser

Please enter a sentence to be parsed:

My dog also likes eating sausage.

Language: English   Sample Sentence   Parse

Your query

My dog also likes eating sausage.

Tagging

```
My/PRP$  dog/NN  also/RB  likes/VBZ  eating/VBG  sausage/NN  ./.
```

Parse

```
(ROOT
  (S
    (NP (PRP$ My) (NN dog))
    (ADVP (RB also))
    (VP (VBZ likes)
      (S
        (VP (VBG eating)
          (NP (NN sausage)))))
    (. .)))
```

Universal dependencies

```
nmod:poss(dog-2, My-1)
nsubj(likes-4, dog-2)
advmod(likes-4, also-3)
root(ROOT-0, likes-4)
xcomp(likes-4, eating-5)
dobj(eating-5, sausage-6)
```

The Stanford Natural Language Processing Group   people   publications   research blog   software   teaching   local

Software > Stanford Parser

The Stanford Parser: A statistical parser

About | Citing | Questions | Download | Included Tools | Extensions | Release history | Sample output | Online | FAQ

About

A natural language parser is a program that works out the grammatical **structure of sentences**, for instance, which groups of words go together (as "phrases") and which words are the **subject** or **object** of a verb. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the *most likely* analysis of new sentences. These statistical parsers still make some mistakes, but commonly work rather well. Their development was one of the biggest breakthroughs in natural language processing in the 1990s. You can try out our parser online.

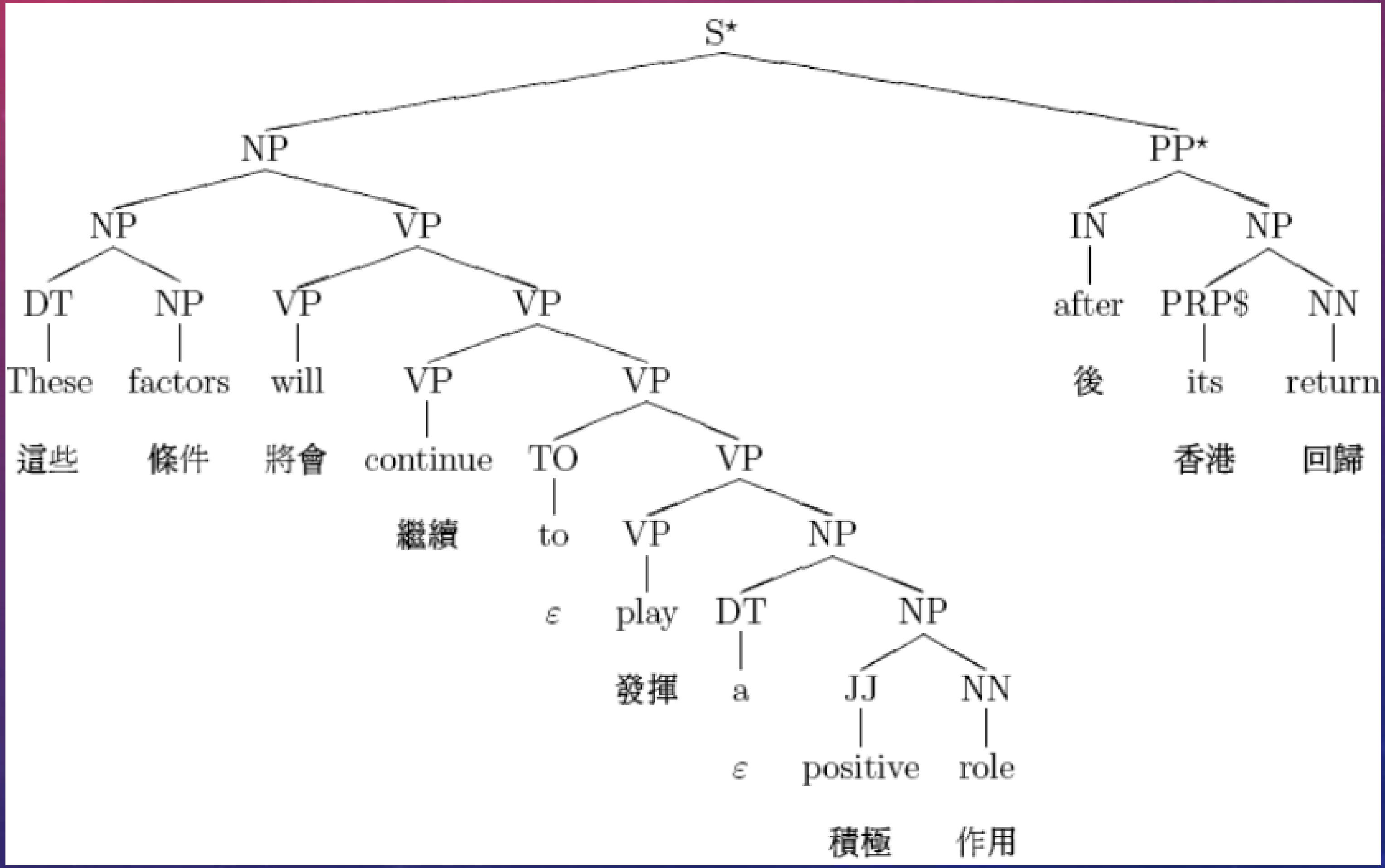
Package contents

This package is a Java implementation of probabilistic natural language parsers, both highly optimized PCFG and lexicalized dependency parsers, and a lexicalized PCFG parser. The original version of this parser was mainly written by Dan Klein, with support code and linguistic grammar development by Christopher Manning. Extensive additional work (internationalization and language-specific modeling, flexible input/output, grammar compaction, lattice parsing, *k*-best parsing, typed dependencies output, user support, etc.) has been done by Roger Levy, Christopher Manning, Teg Grenager, Galen Andrew, Marie-Catherine de Marneffe, Bill MacCartney, Anna Rafferty, Spence Green, Huihsin Tseng, Pi-Chuan Chang, Wolfgang Maier, and Jenny Finkel.

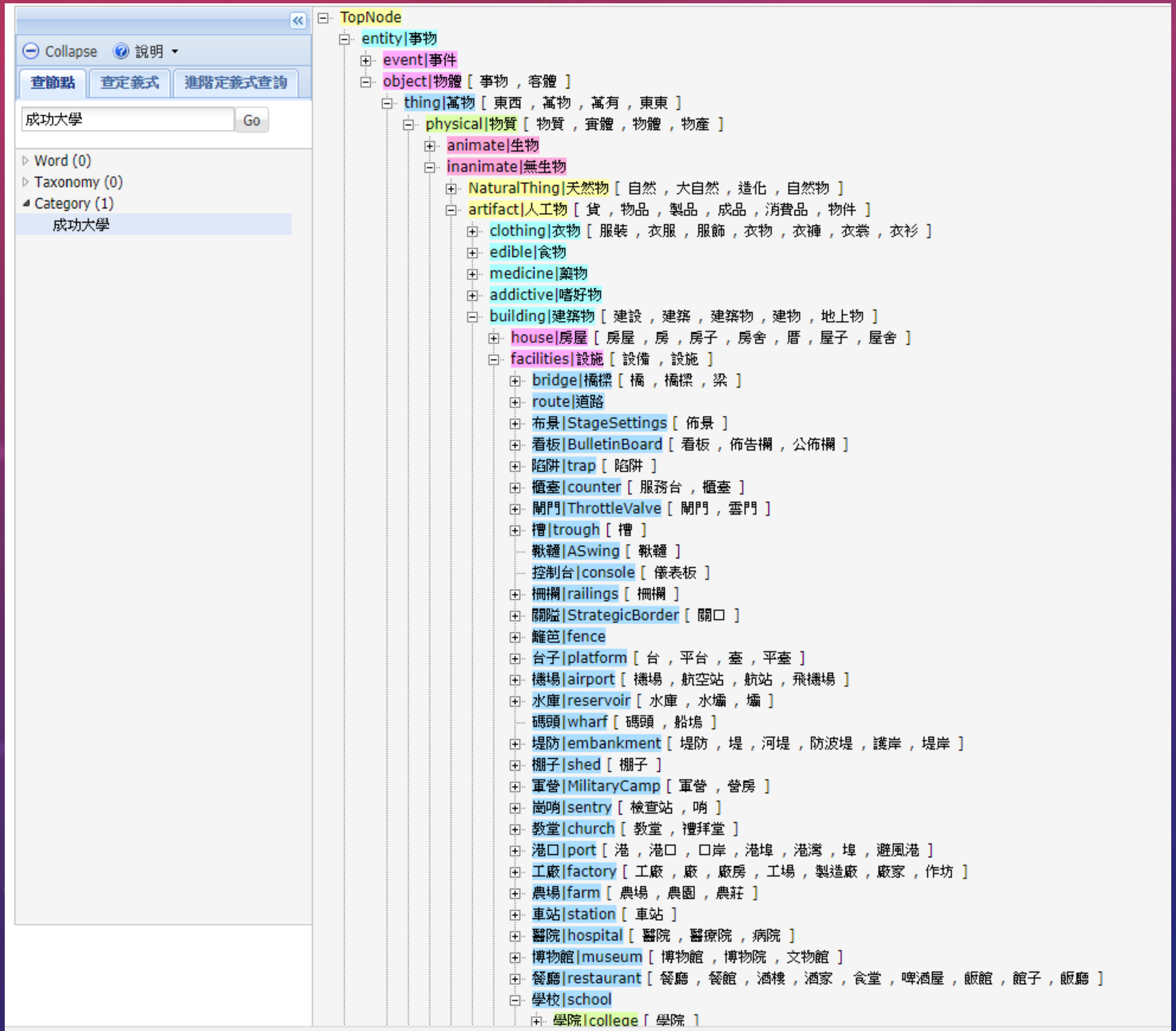
The lexicalized probabilistic parser implements a factored product model, with separate PCFG phrase structure and lexical dependency experts, whose preferences are combined by efficient exact inference, using an A* algorithm. Or the software can be used simply as an accurate unlexicalized stochastic context-free grammar parser. Either of these yields a good performance statistical parsing system. A GUI is provided for viewing the phrase structure tree output of the parser.

https://nlp.stanford.edu/software/lex-parser.shtml#Sample

PARSE TREE


S*
NP
PP*
NP
VP
IN
NP
DT
NP
VP
VP
after
PRP$
NN
These
factors
will
VP
VP
後
its
return
這些
條件
將會
continue
TO
VP
香港
回歸
繼續
to
VP
NP
ε
play
DT
NP
發揮
a
JJ
NN
ε
positive
role
積極
作用

廣義知網知識本體



http://ehownet.iis.sinica.edu.tw/ehownet.php

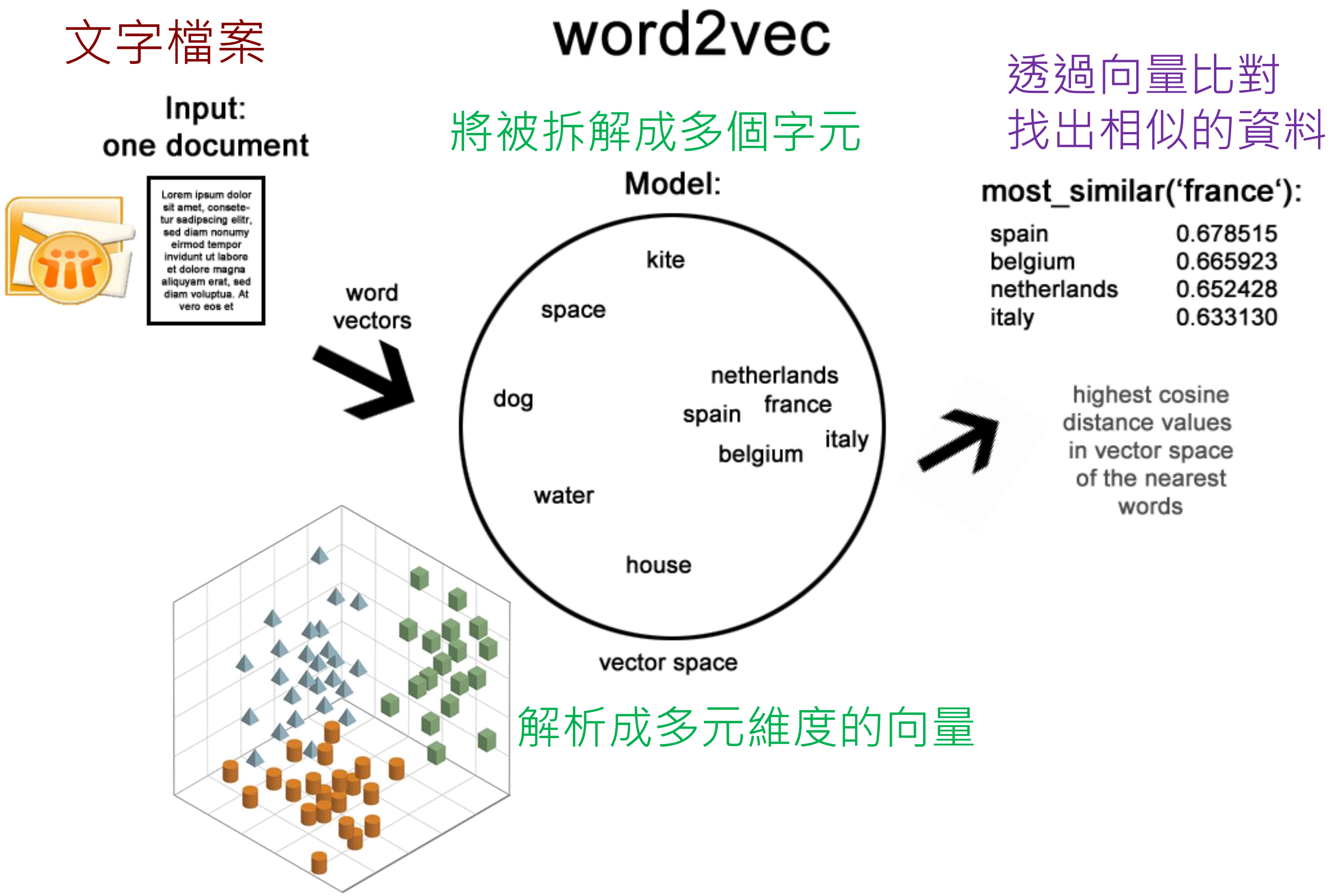
SEMANTIC SIMILARITY MEASURES

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



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

Machine learning

TF-IDF

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

*

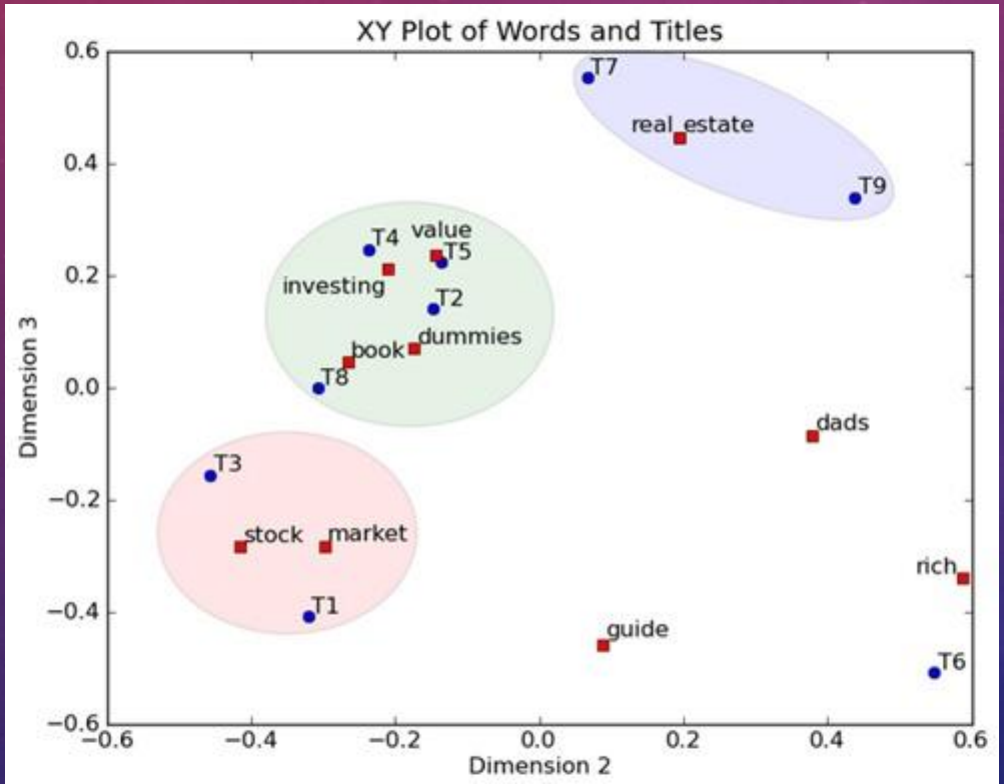
3.91	0	0
0	2.61	0
0	0	2.00

*

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				

以語言學習輔助工具為例

Collocation online suggestion v1.0

英語搭配詞線上檢索系統

介紹　常用搭配詞查詢　整句搭配詞查詢與推薦

整句搭配詞查詢與推薦

輸入句子： We commonly use a small cell for medical research.

清除

送出

輸入的句子為

We commonly use a small cell for medical research .

副詞修飾(V/Adv/Adj組合)

commonly + V/Adv/Adj

#	collocation	freq(%)
1	commonly use	46.5
2	commonly used	4.7
3	commonly find	4.4
4	commonly know	3.3
5	commonly employ	2.4
6	commonly refer	2.2
7	commonly observe	1.9
8	commonly report	1.9
9	commonly encounter	1.4
10	commonly available	1.3

commonly與use的搭配字同義組合

commonly + 搭配同義字

#	collocation	freq(%)
1	commonly use	46.5
2	commonly employ	2.4
3	commonly apply	0.5

同義詞搭配詞組搜尋結果

commonly的同義字 + use 的同義字

#	collocation	count
1	commonly use	296
2	often use	140
3	frequently use	68
4	commonly employ	15
5	frequently employ	9
6	often employ	6
7	frequently apply	5
8	repeatedly use	5
9	routinely use	5
10	frequently utilize	4
11	routinely employ	3
12	commonly apply	3

查詢總時間:0.52sec

以語言學習輔助工具為例

	computer	data	pinch	result	sugar
aprocot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

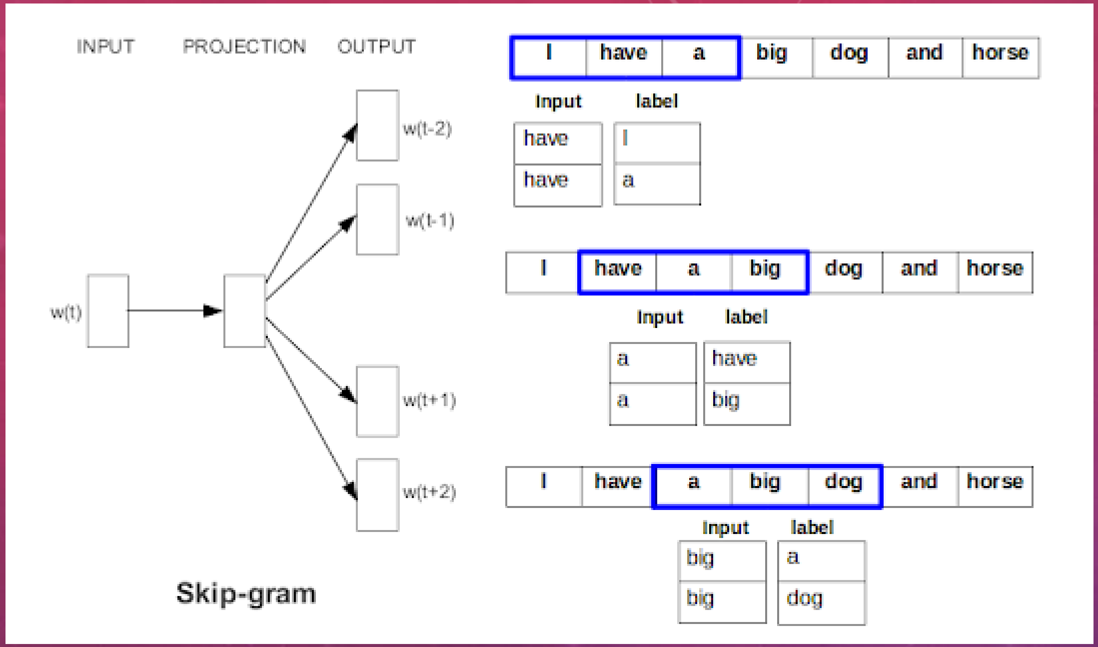
$$P(x = information, y = data) = \frac{6}{19} = 0.32$$

$$P(x = information) = \frac{6+4+1}{19} = \frac{11}{19} = 0.58$$

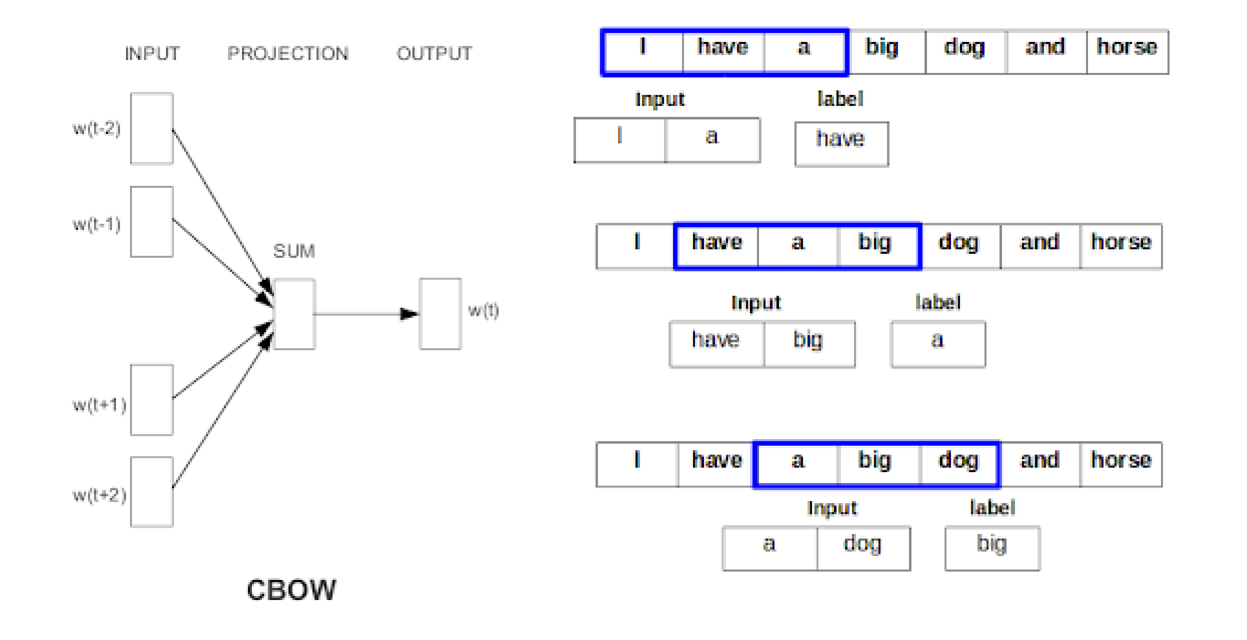
$$P(y = data) = \frac{6+1}{19} = \frac{7}{19} = 0.37$$

$$\begin{aligned} & pmi(x = information, y = data) \\ & = log \frac{P(x = information, y = data)}{P(x = information) \times P(y = data)} \\ & = log 1.49 \\ & = 0.57 \end{aligned}$$

WORD2VEC



INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
I have a big dog and horse
Input label
have I
have a
I have a big dog and horse
Input label
a have
a big
I have a big dog and horse
Input label
big a
big dog
Skip-gram

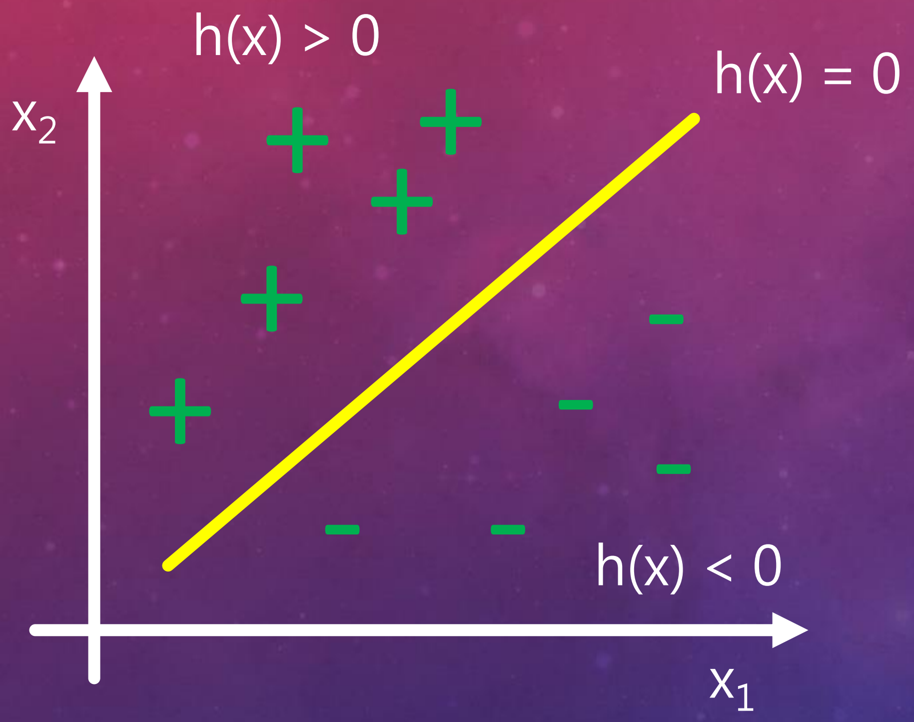


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
w(t+1)
w(t+2)
SUM
w(t)
I have a big dog and horse
Input label
I a have
I have a big dog and horse
Input label
have big a
I have a big dog and horse
Input label
a dog big
CBOW

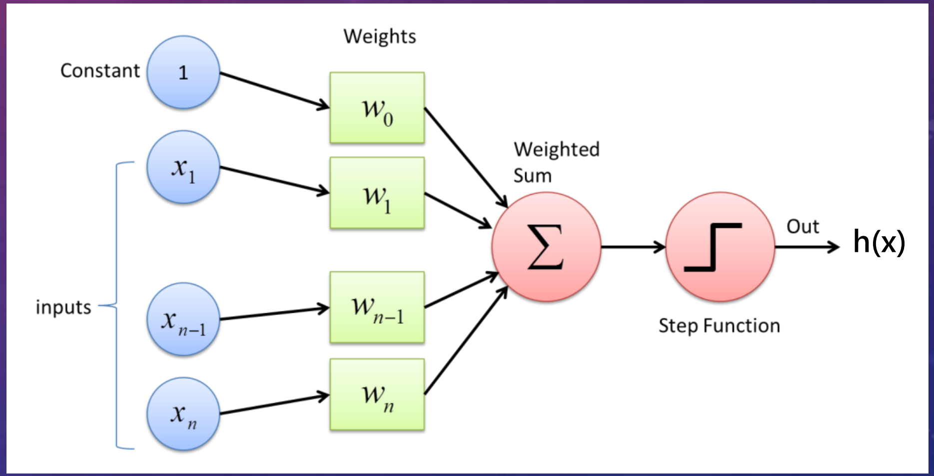
One Hot Encoding

```
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.]
```

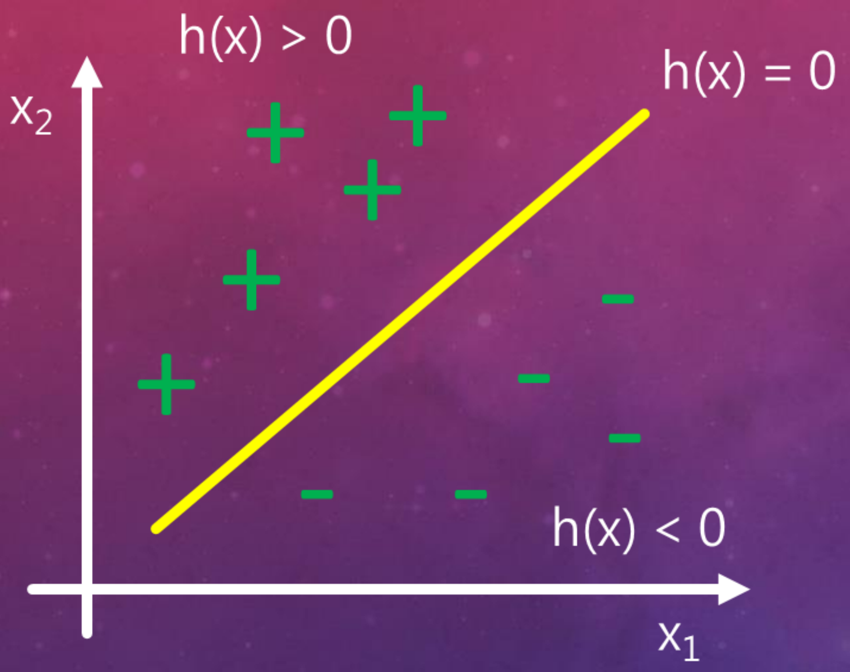
Perceptron Linear Algorithm



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



Perceptron Linear Algorithm



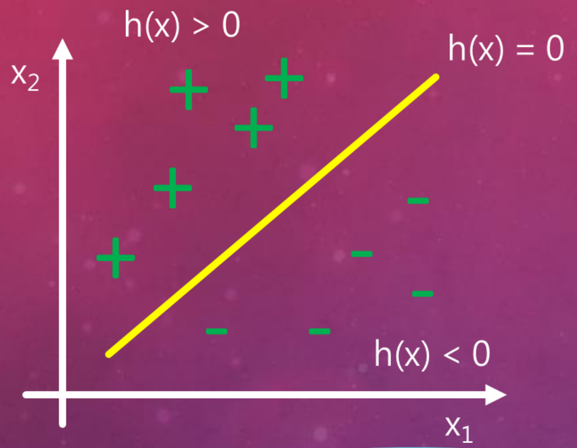
$$h(x) = w_0 + w_1x_1 + w_2x_2$$

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

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

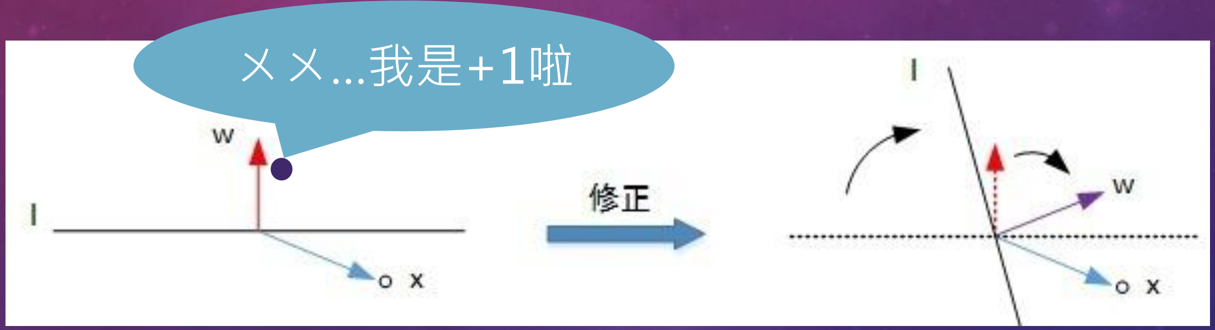
- 若 $scores \geq 0$ ，则 $\hat{y} = 1$
- 若 $scores < 0$ ，则 $\hat{y} = -1$

Perceptron Linear Algorithm



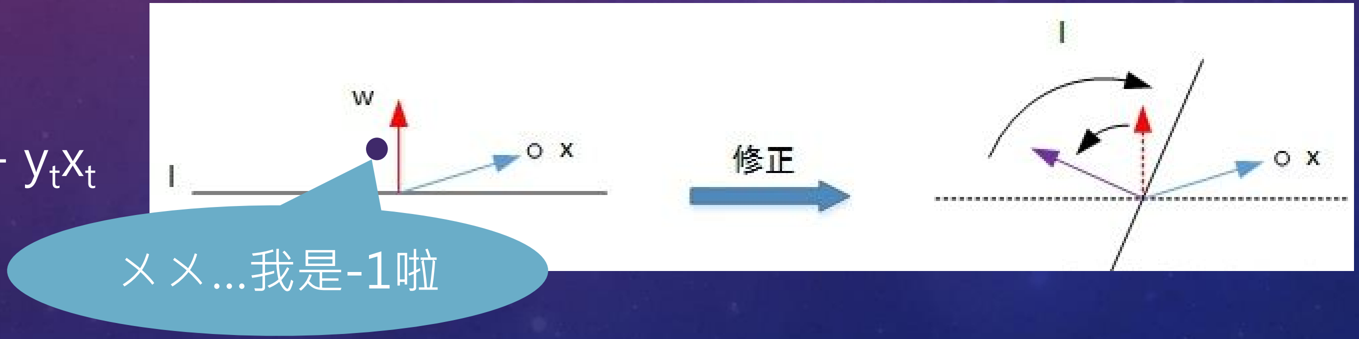
- 若 $scores \geq 0$ ，则 $\hat{y} = 1$
- 若 $scores < 0$ ，则 $\hat{y} = -1$

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



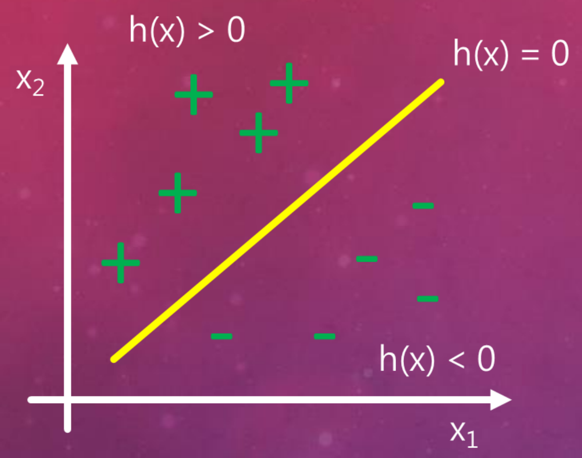
[Case 1]
y = 1 錯分成 y = -1

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



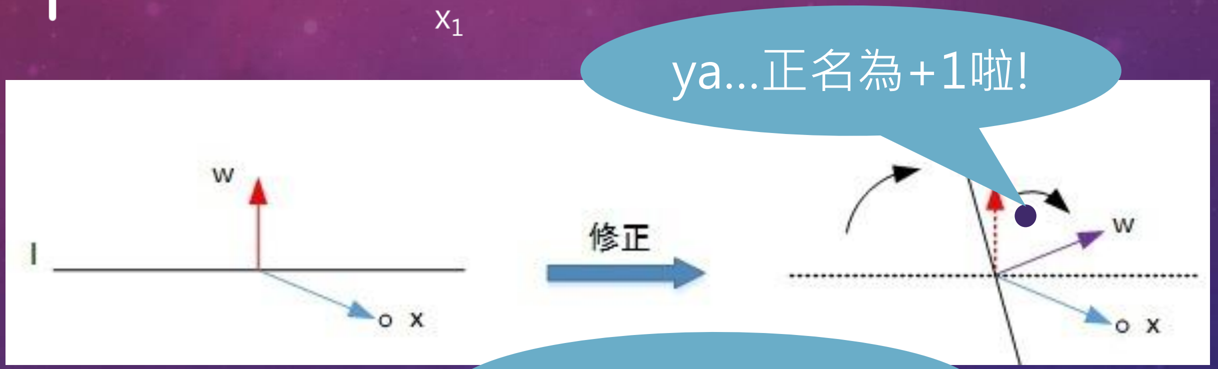
[Case 2]
y = -1 錯分成 y = 1

Perceptron Linear Algorithm



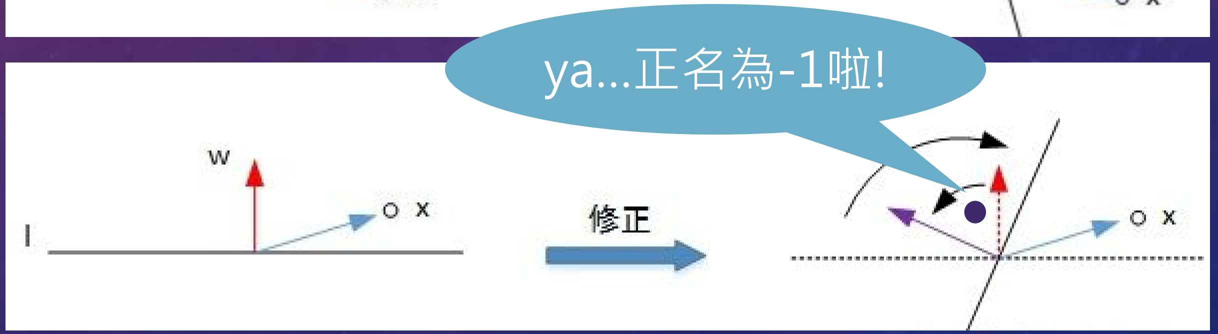
- 若 $scores \geq 0$ ，则 $\hat{y} = 1$
- 若 $scores < 0$ ，则 $\hat{y} = -1$

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

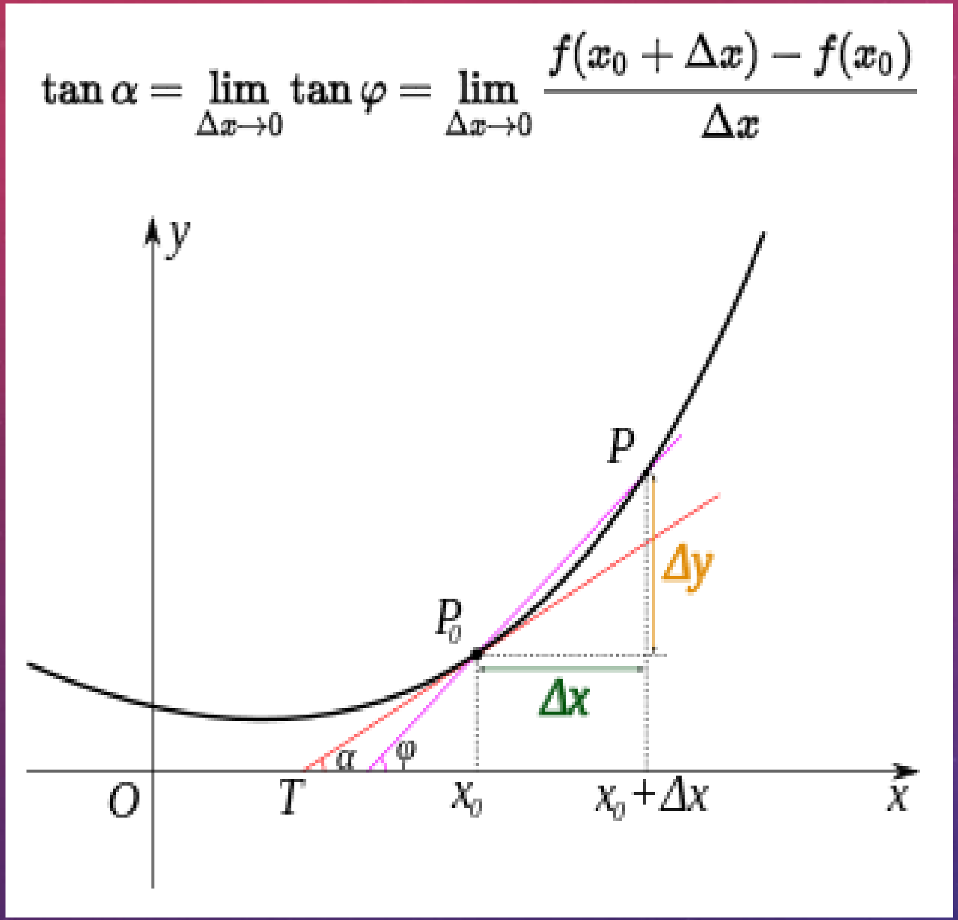


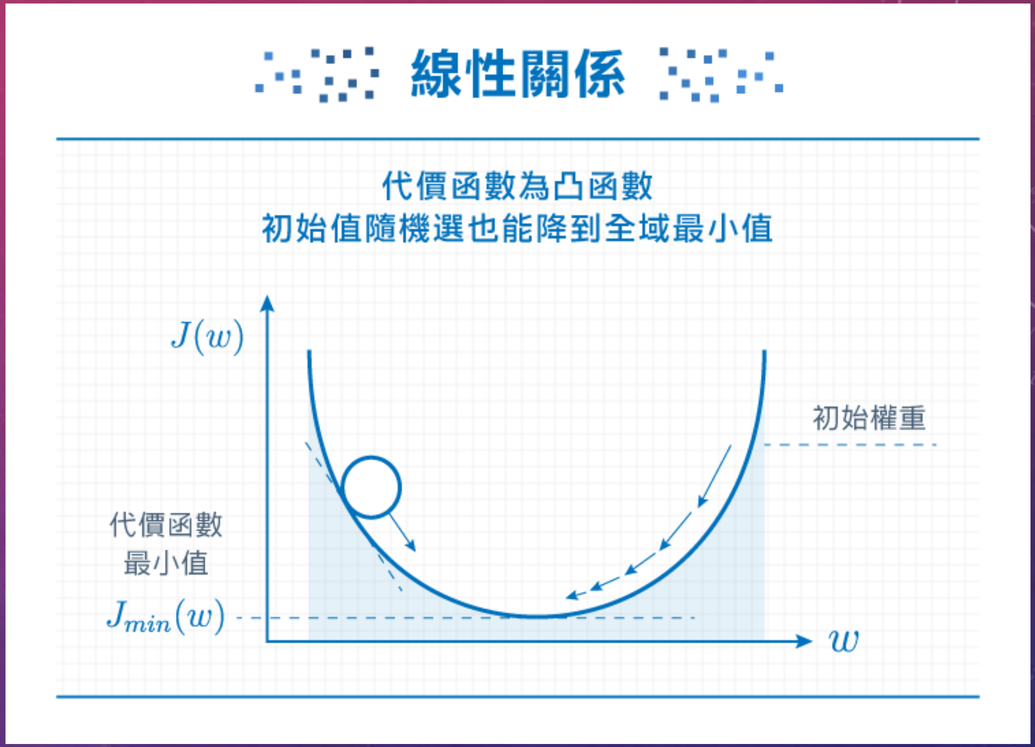
[Case 1]
y = 1 錯分成 y = -1

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

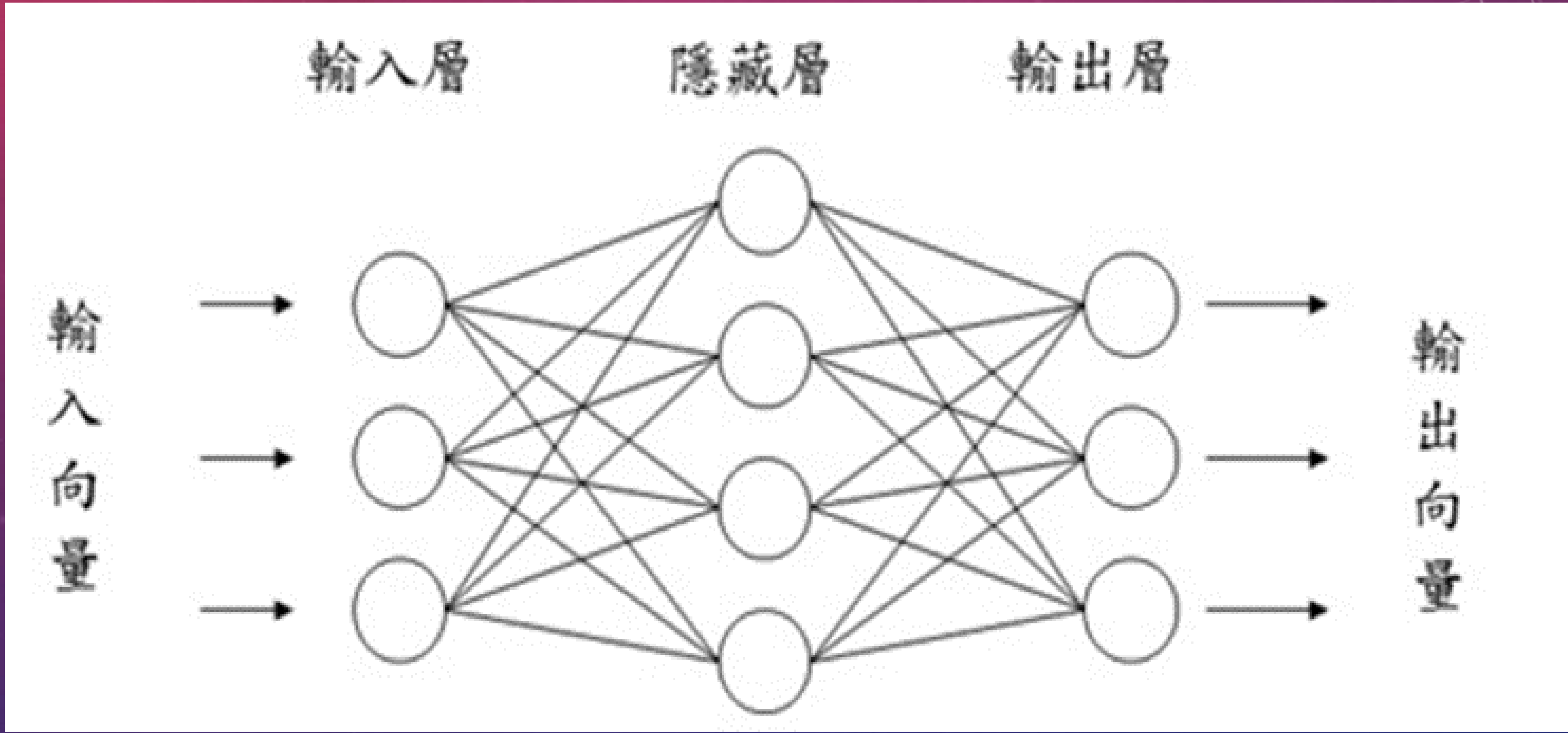


[Case 2]
y = -1 錯分成 y = 1


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

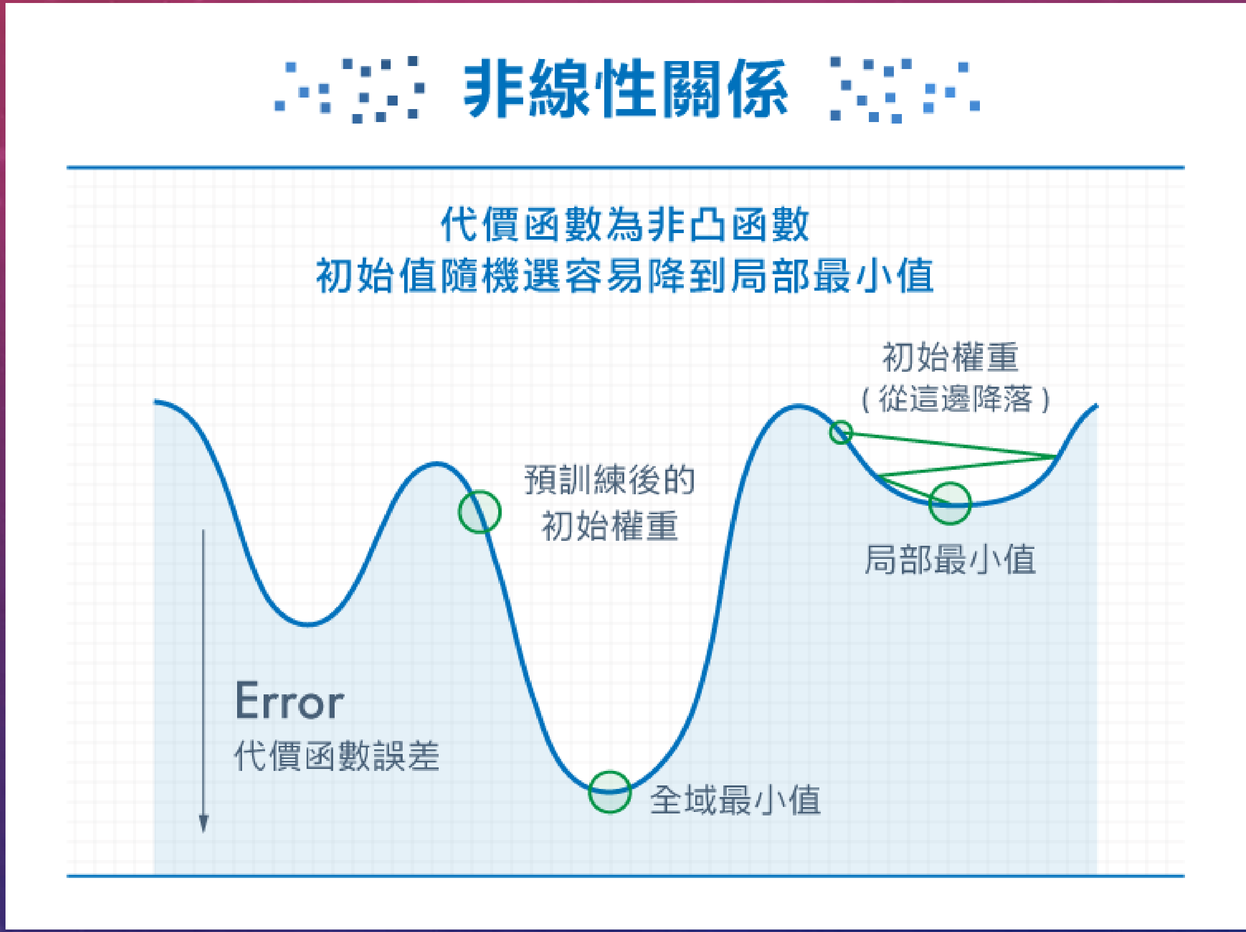

線性關係
代價函數為凸函數
初始值隨機選也能降到全域最小值
$J(w)$
初始權重
代價函數
最小值
$J_{min}(w)$
w

Multi-Layer Perceptron (MLP)



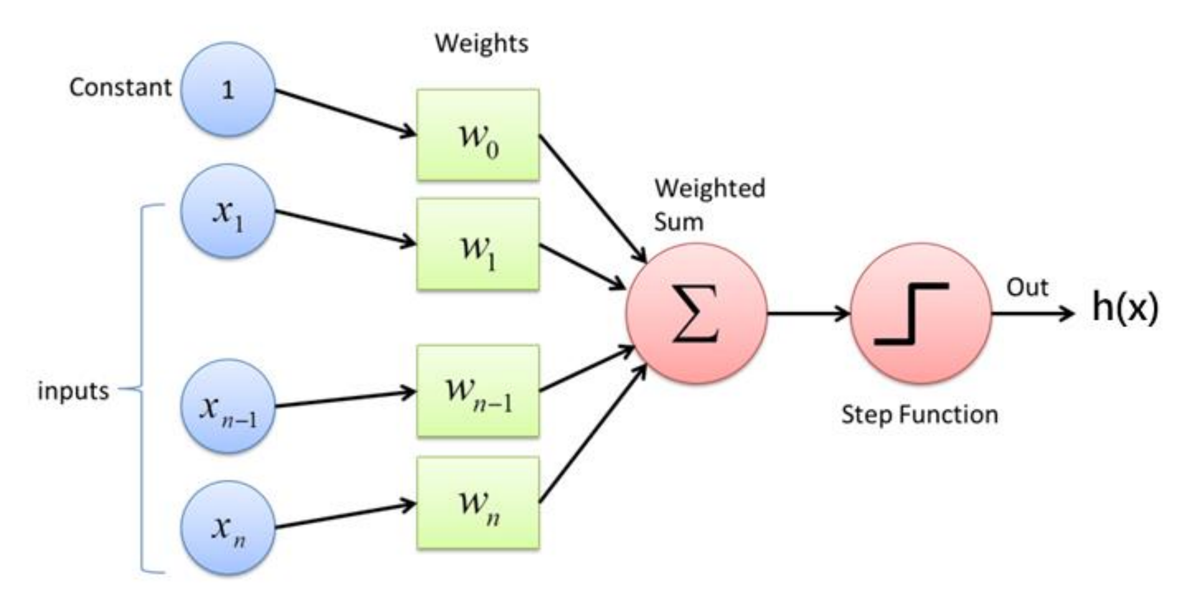
*線性組合 $w = a_1 v_1 + a_2 v_2 + a_3 v_3 + \cdots + a_n v_n$

梯度消失



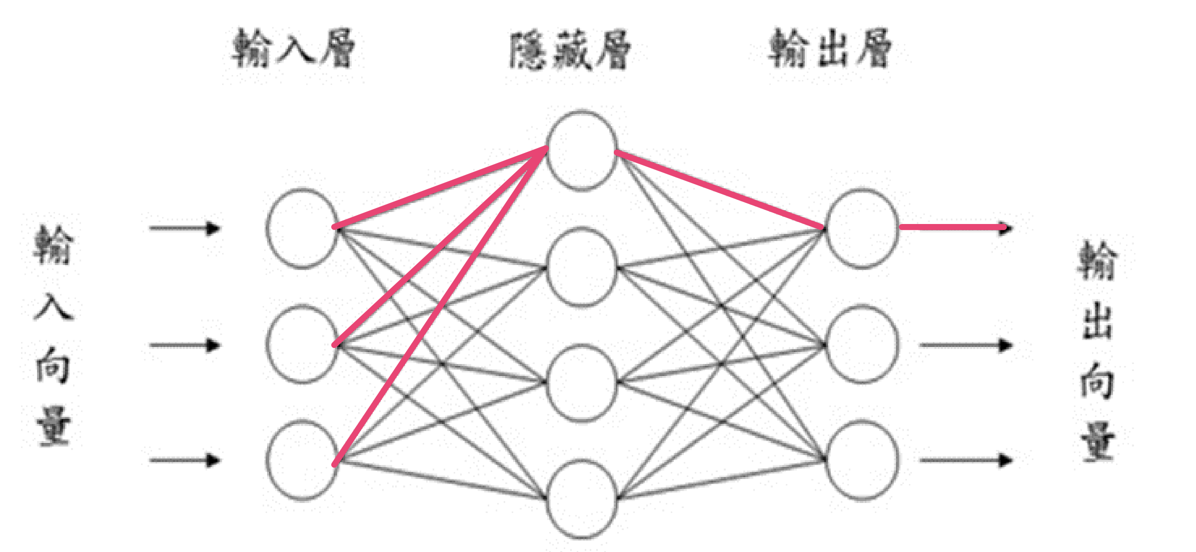
*現實世界的資料多為非線性，因此激活函數通常也是使用非線性函數(非凸函數)傳遞

PLA



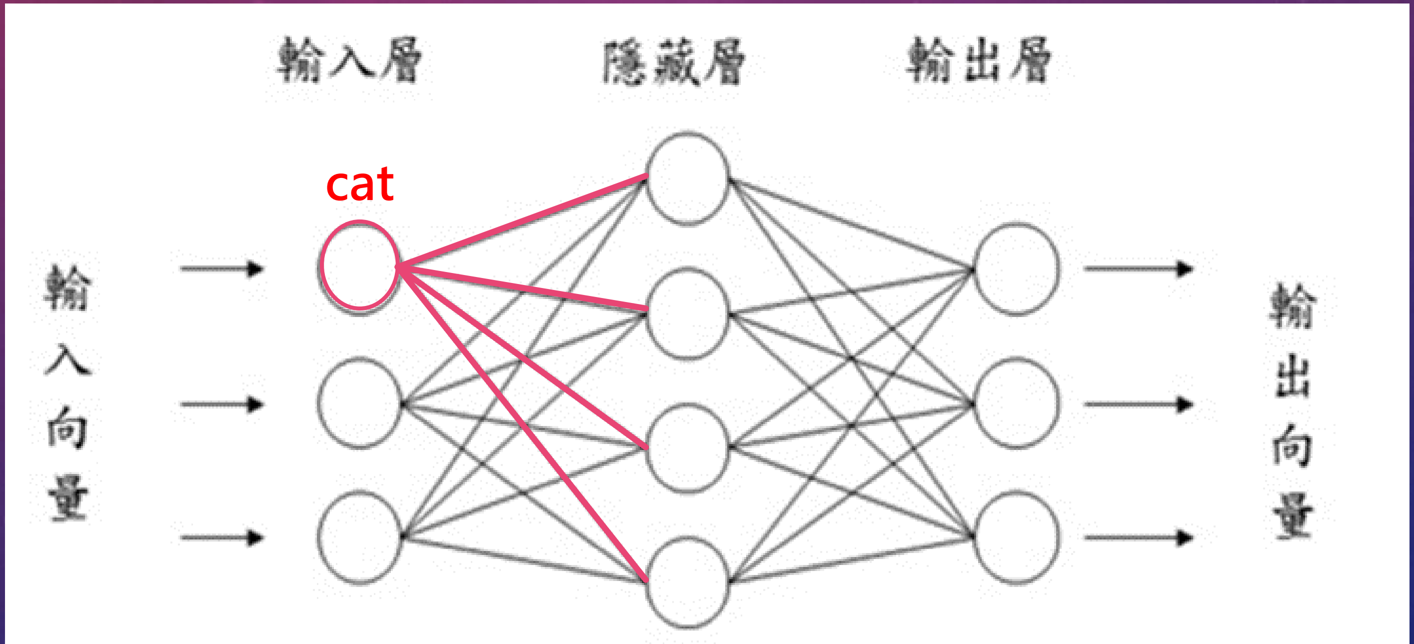
Weights
Constant
1
w_0
x_1
w_1
Weighted Sum
Σ
Out
h(x)
inputs
x_{n-1}
w_{n-1}
Step Function
x_n
w_n

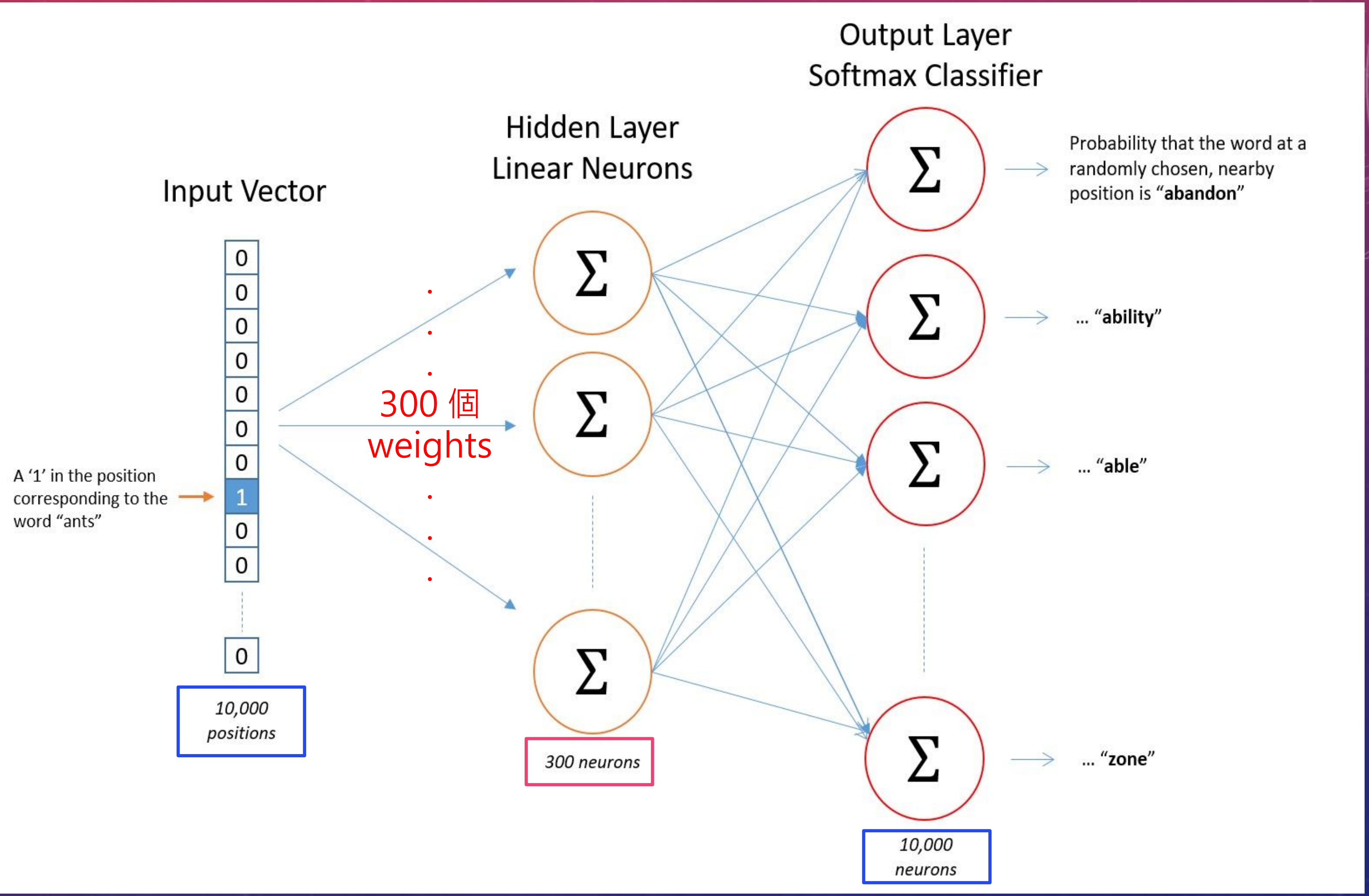
MLP

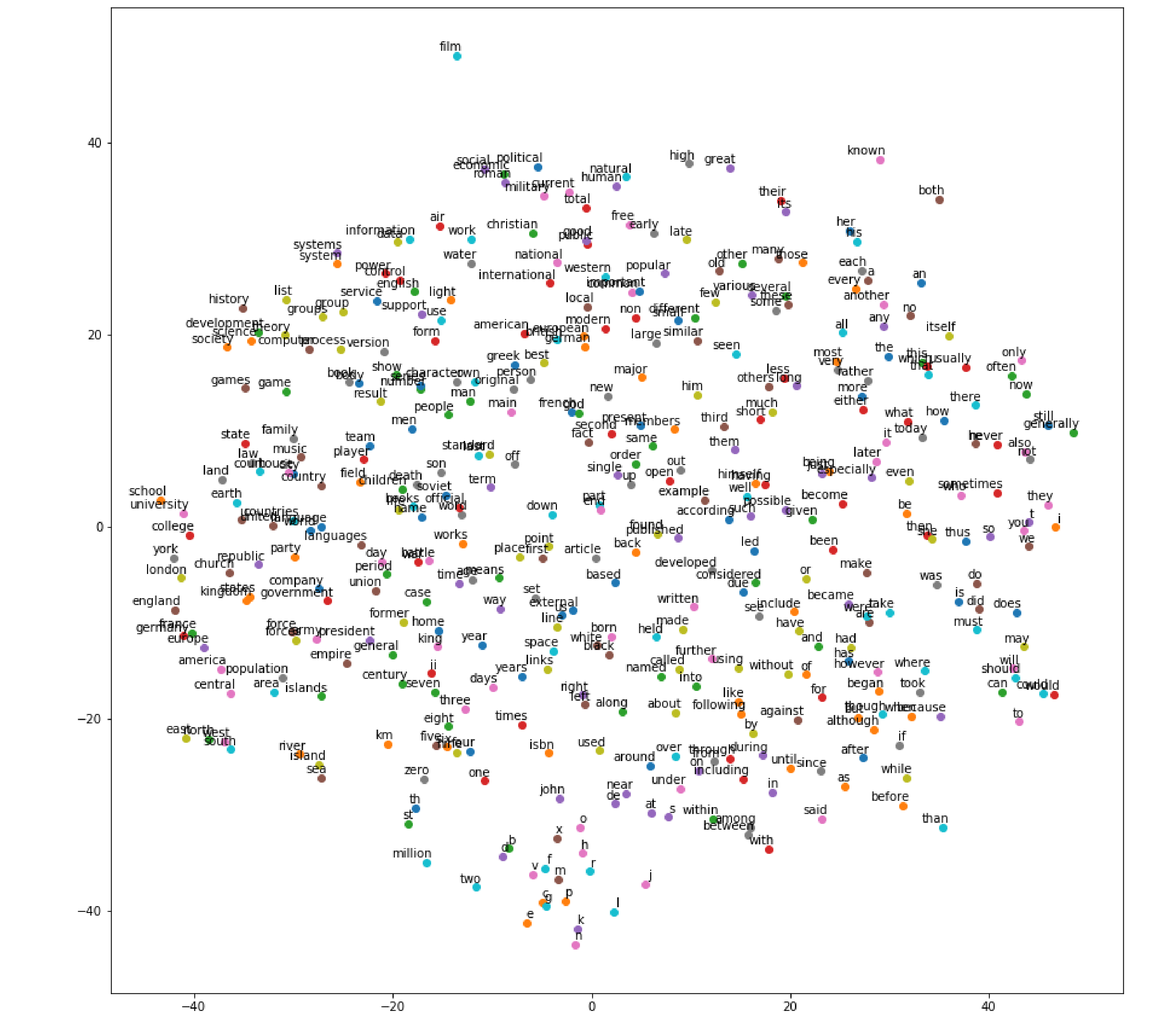


輸入層
隱藏層
輸出層
輸入向量
輸出向量

```
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
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
10,000 positions
300 neurons
10,000 neurons

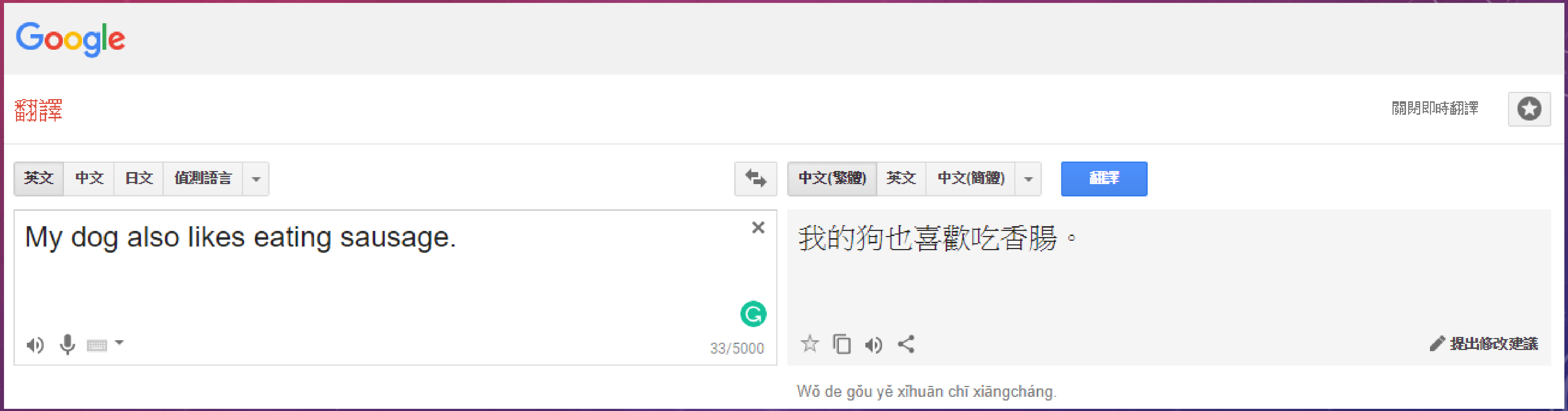


40
20
0
−20
−40
−40
−20
0
20
40
film
social
political
economic
roman
military
current
total
natural
human
high
great
known
both
their
its
her
his
air
work
christian
information
data
systems
system
power
control
english
water
national
international
good
public
free
early
late
other
many
those
old
western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

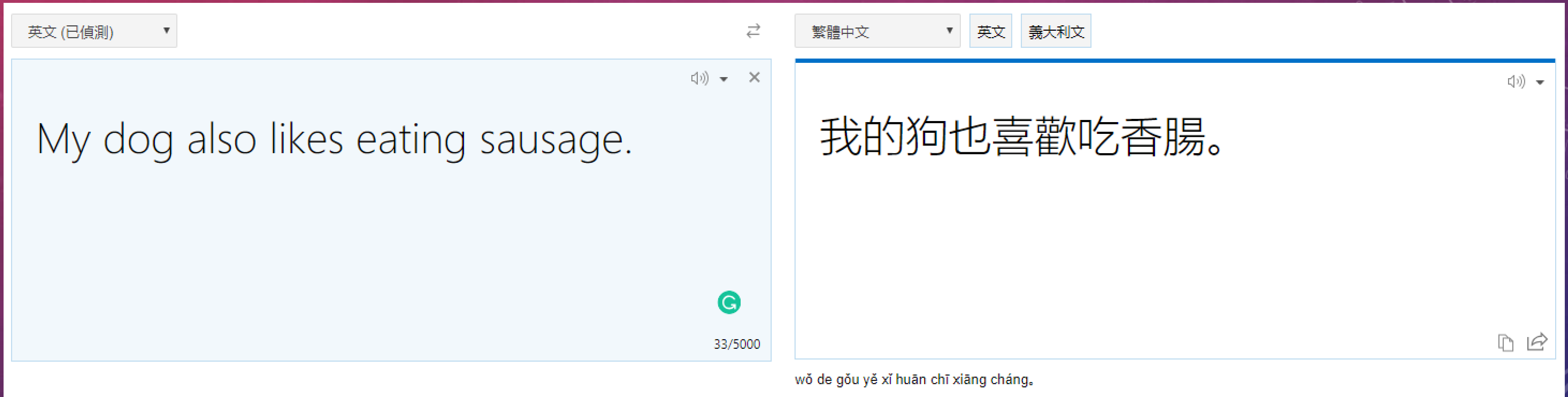
機器翻譯

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.

BING 翻譯



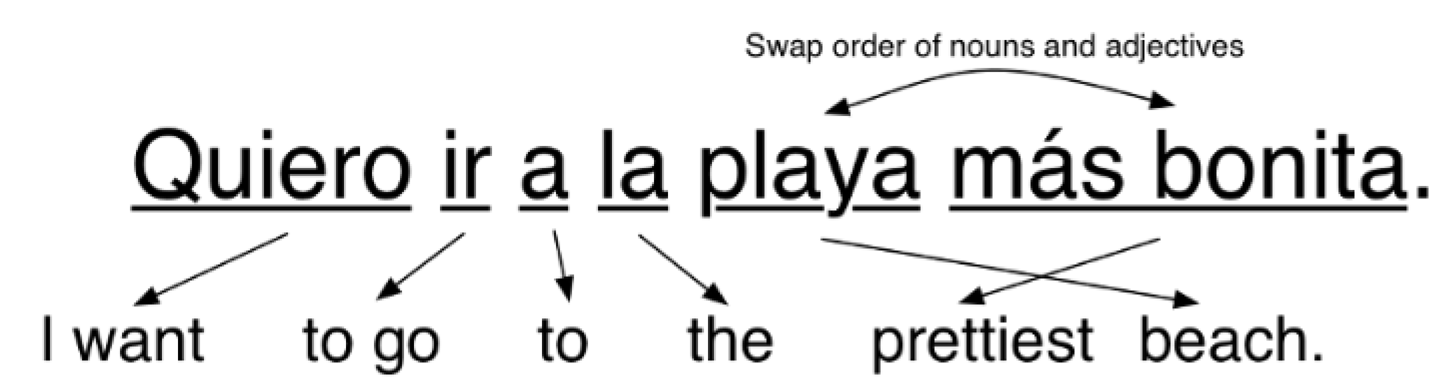
英文 (已偵測)
My dog also likes eating sausage.
33/5000
繁體中文
英文
義大利文
我的狗也喜歡吃香腸。
wǒ de gǒu yě xǐ huān chī xiāng cháng。

有道翻譯


检测到：英语 » 中文
翻译
人工翻译
划词
My dog also likes eating sausage.
33/5000
我的狗也喜欢吃香肠。
修改翻译结果

平行語料

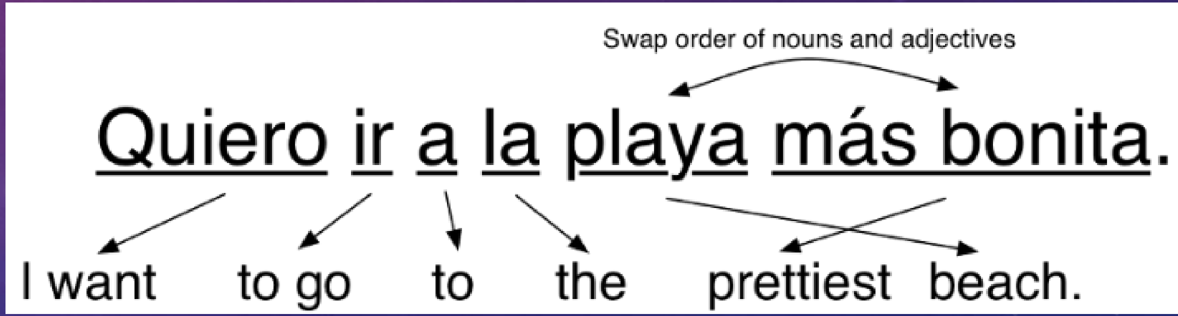

Quiero ir a la playa más bonita.
I want
to go
to
the
beach
more
pretty.
We just replace each Spanish word with the matching English word.


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

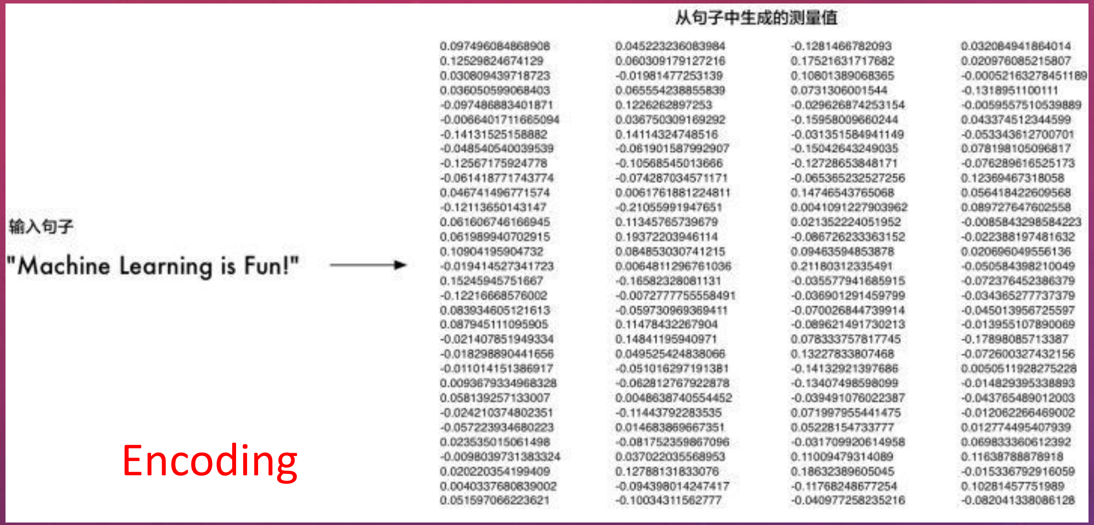
統計式機器翻譯之原理

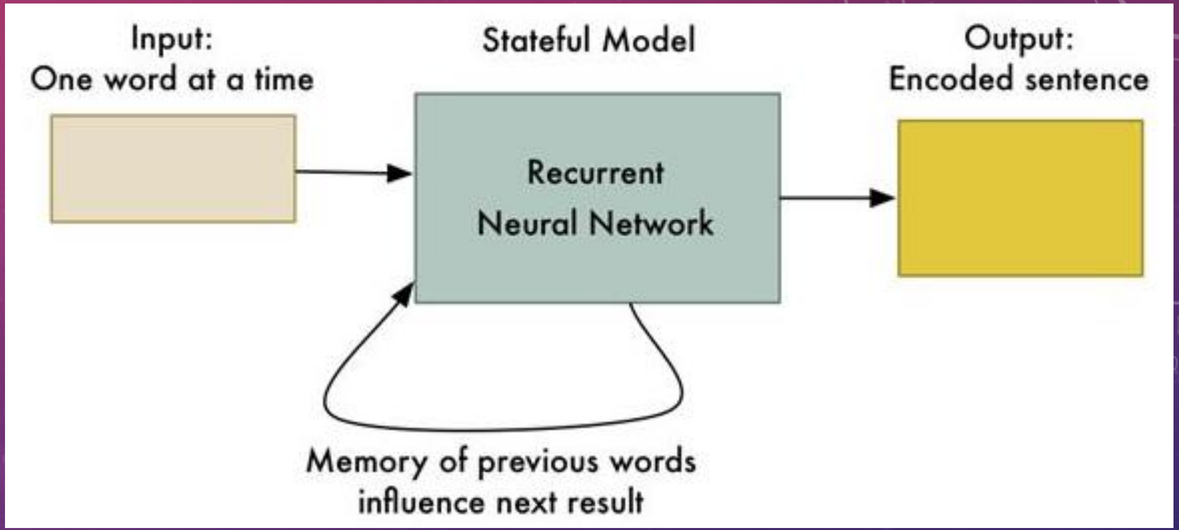
We just replace each Spanish word with the matching English word.

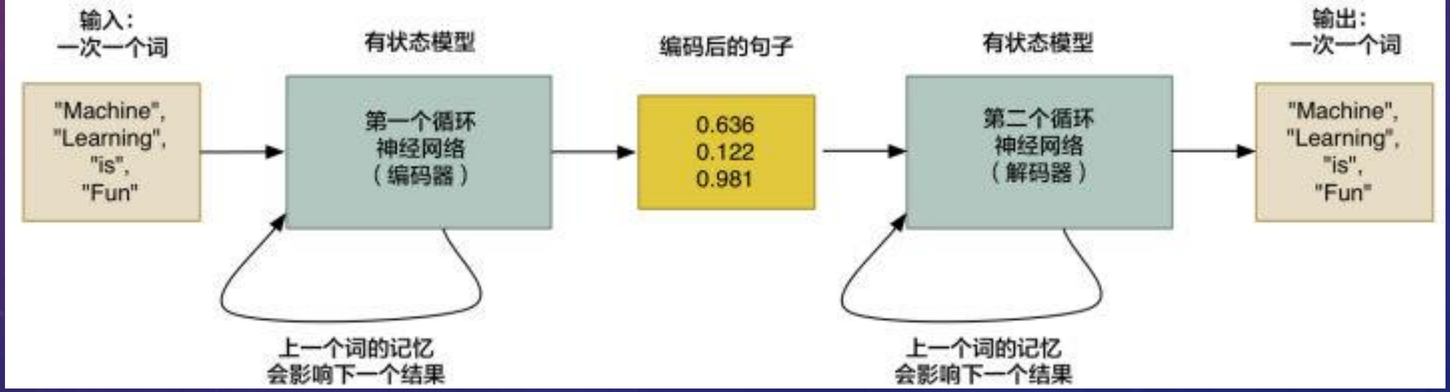
Even the most common phrases have lots of possible translations.



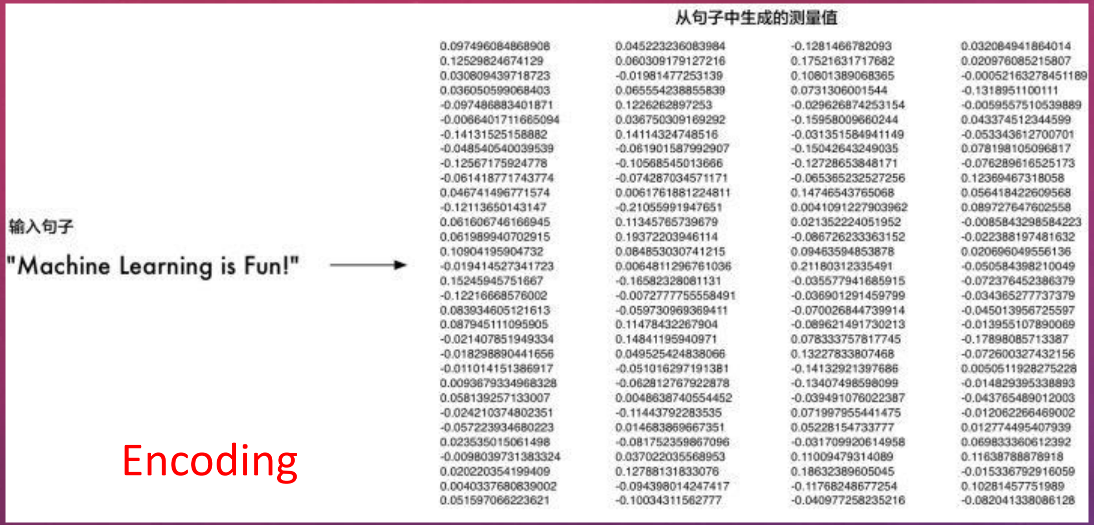
深度學習於機器翻譯之原理



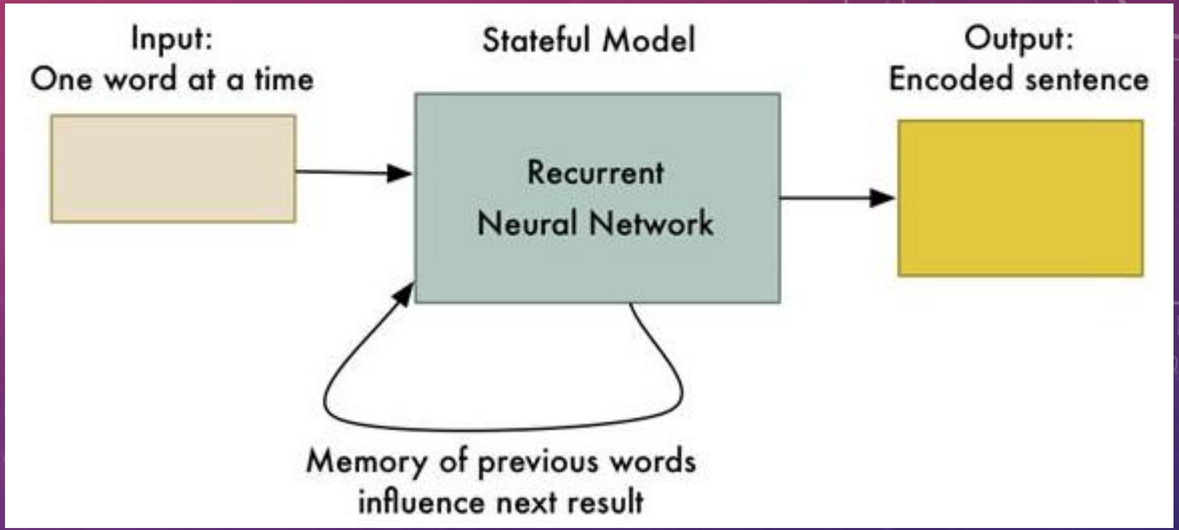


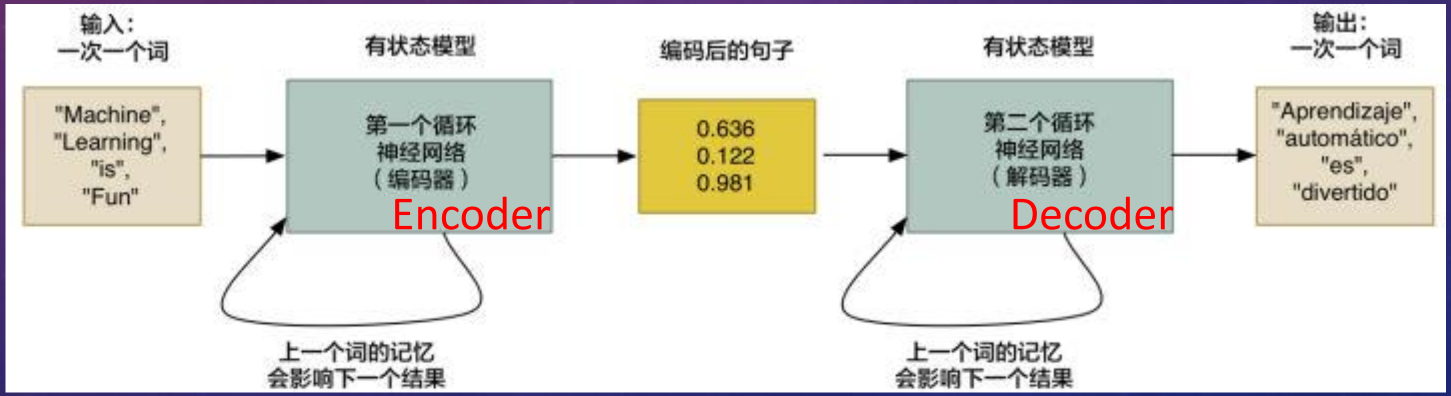


深度學習於機器翻譯之原理



0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128





THANK YOU