

生成式 AI：文字與圖像生成的原理與實務

04. 大型語言模型原來 這麼簡單

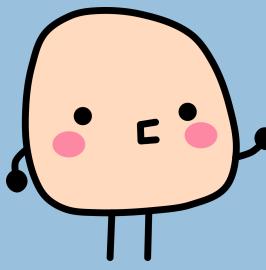


蔡炎龍
政治大學應用數學系



01.

文字生成 AI 的原理

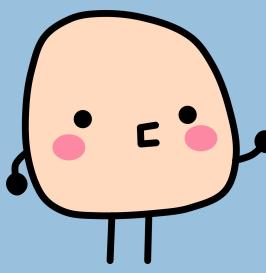


自然語言處理 (NLP)

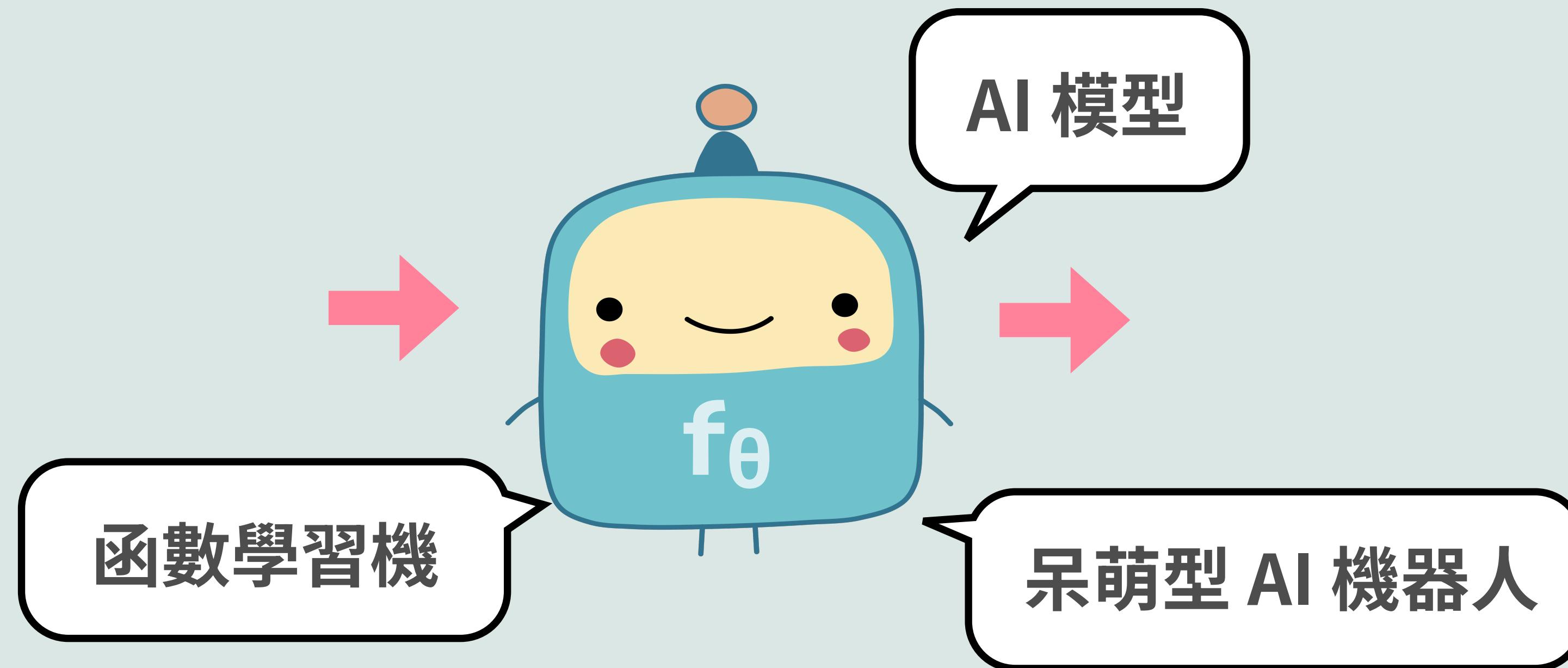
Natural Language Processing

也就是處理正常人
話的 AI。

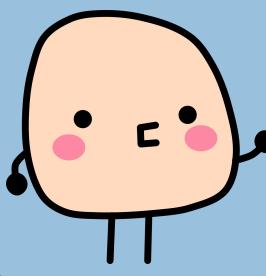




現代的 AI 模型只是一個呆萌 AI 機器人



就是知道輸入是什麼、輸出是長什麼樣子

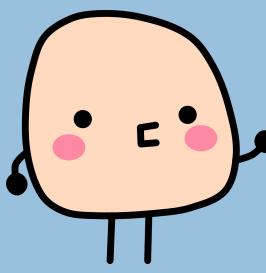


AI 最重要的是要「問個好問題」

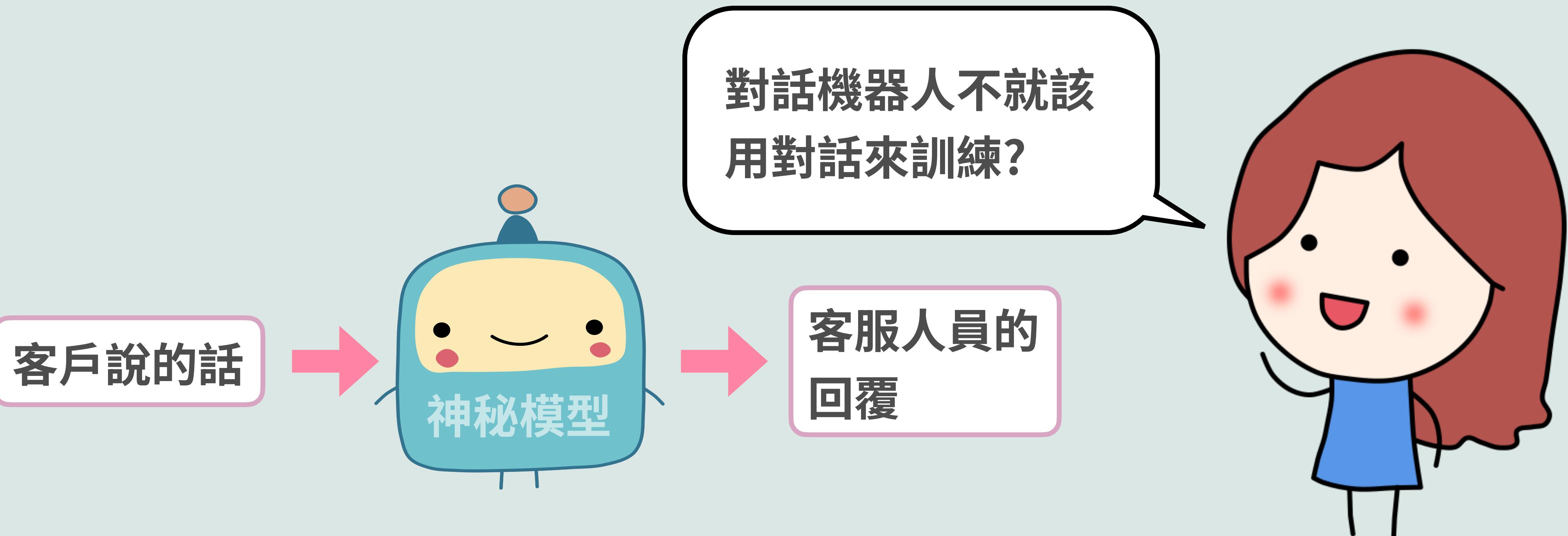


深度學習會專注在
「我們怎麼問這個問題？」

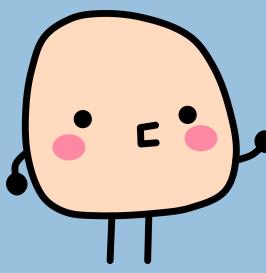
整個判斷由電腦做，所以
一般要**大量資料**！



我們可能會這樣想...



但這樣設計會有問題...



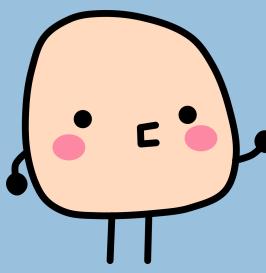
結果不是...

那到底怎麼設
計的呢？

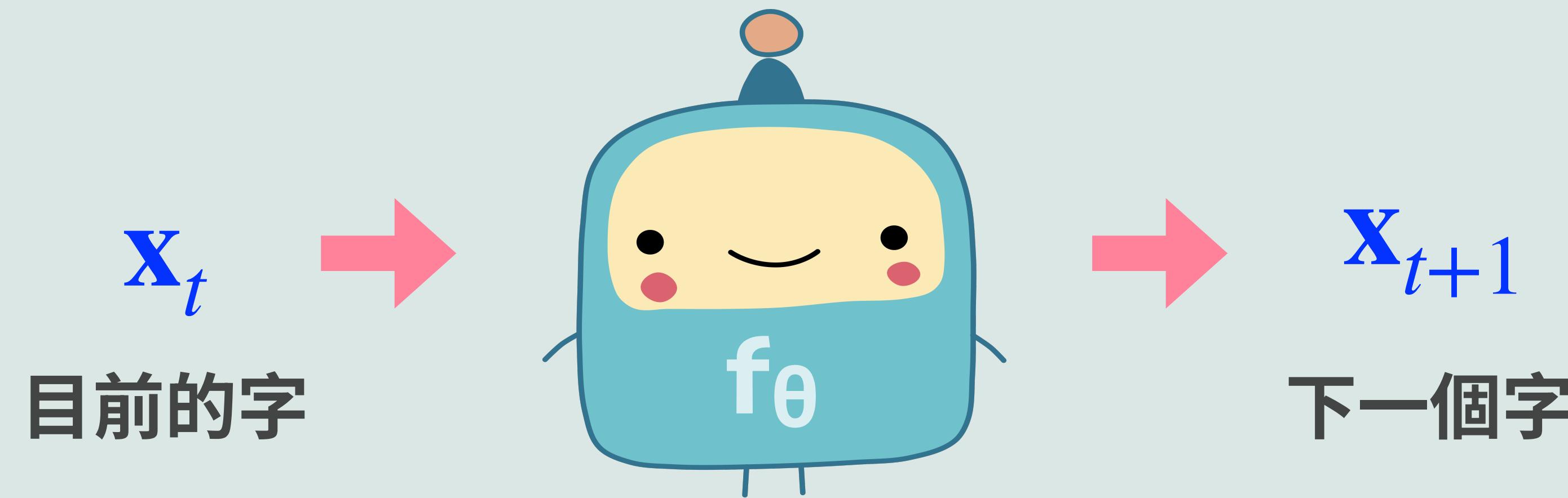


看到答案大家
的驚呆了！

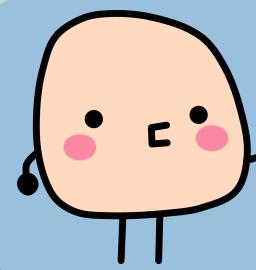




就是前一個字預測下一個字的模型



看到前面一個字、預測下一個字的 AI 機器人

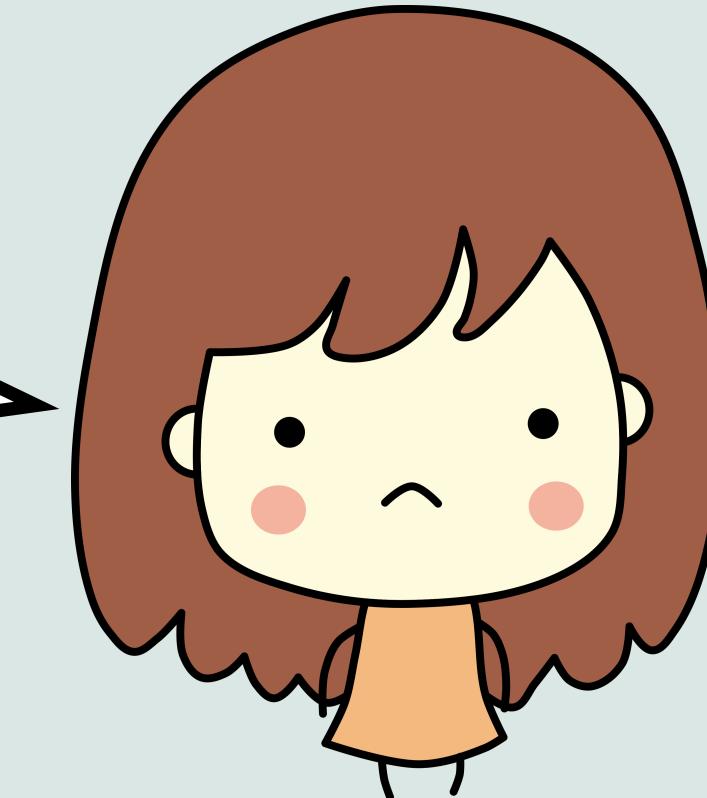


這樣世界上所有句子都可以拿訓練資料

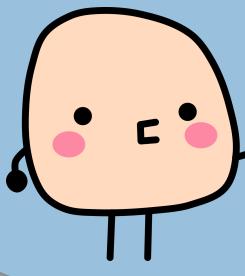
炎龍老師很帥

X1 **X**2 **X**3 **X**4 **X**5 **X**6

**不能舉個實在
一點的例子嗎？**



(炎, 龍), (龍, 老), (老, 師), (師, 很), (很, 帥)
 x_1 y_1 x_2 y_2 x_3 y_3 x_4 y_4 x_5 y_5
(= x₂) (= x₃) (= x₄) (= x₅) (= x₅)



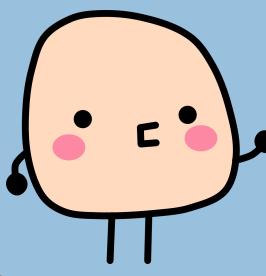
但這不是個函數!

比如說，“今天天氣很好。”這句話拿去訓練，輸入我們的函數就應該是...

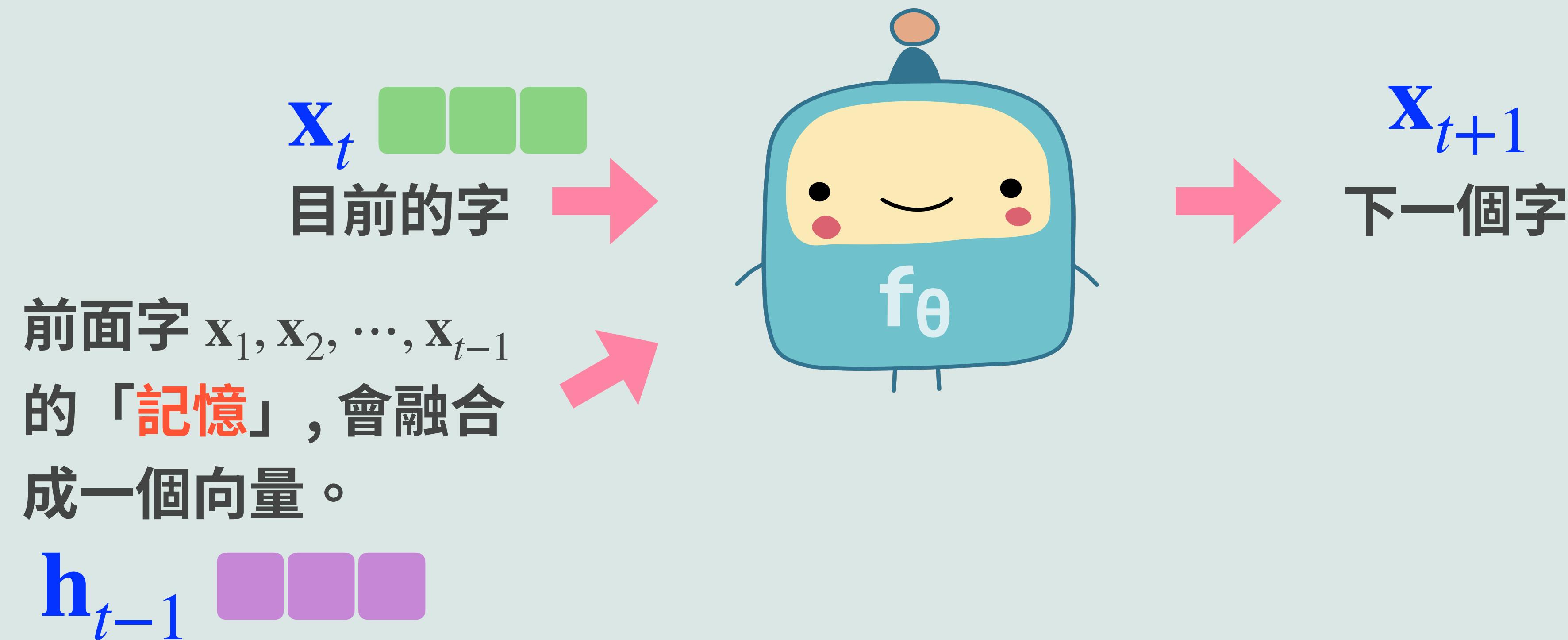
$$\begin{aligned}f_{\theta}(\text{“今”}) &= \text{“天”} \\f_{\theta}(\text{“天”}) &= \text{“天”} \\f_{\theta}(\text{“天”}) &= \text{“氣”} \\f_{\theta}(\text{“氣”}) &= \text{“很”} \\f_{\theta}(\text{“很”}) &= \text{“好”}\end{aligned}$$

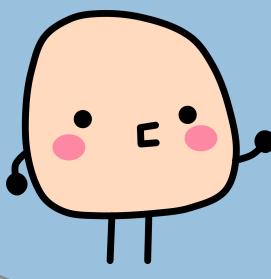
這根本不是函數啊！



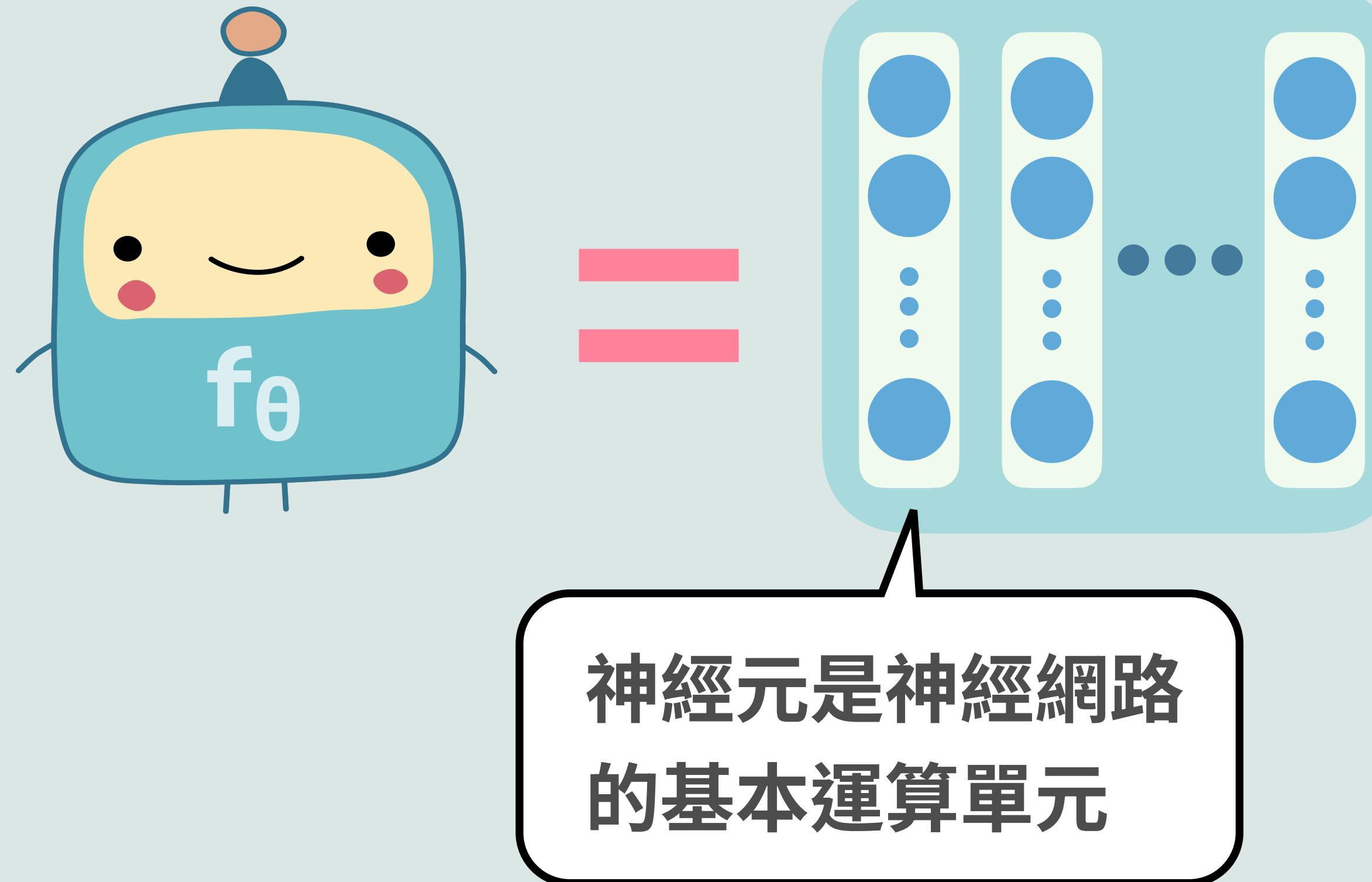


實際上我們用的是「有記憶的」神經網路





用「神經網路」，輕輕鬆鬆可以打造這個機器人！



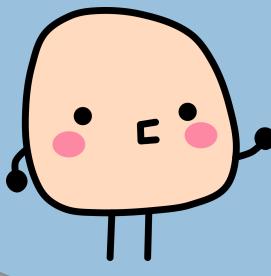
基本上就是建一個個隱藏層，每層放上若干個「神經元」，放法差不多就 3+1 種：

DNN: 標準全連結

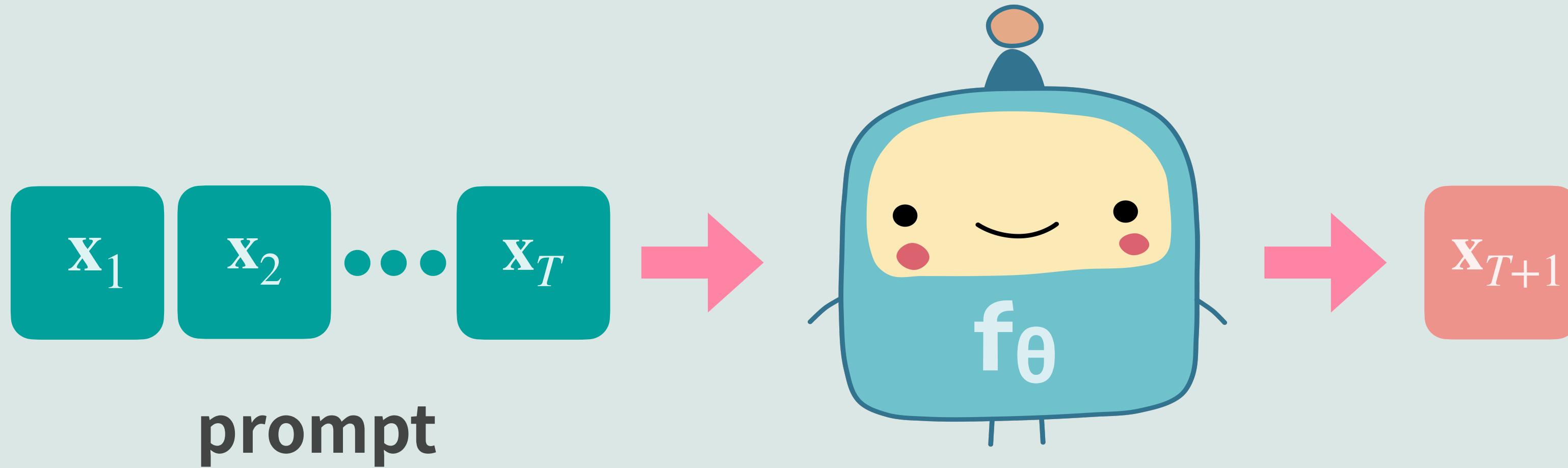
CNN: 圖形辨識很強

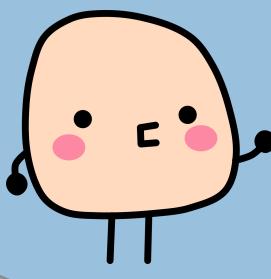
RNN: 有記憶的結構

Transformers

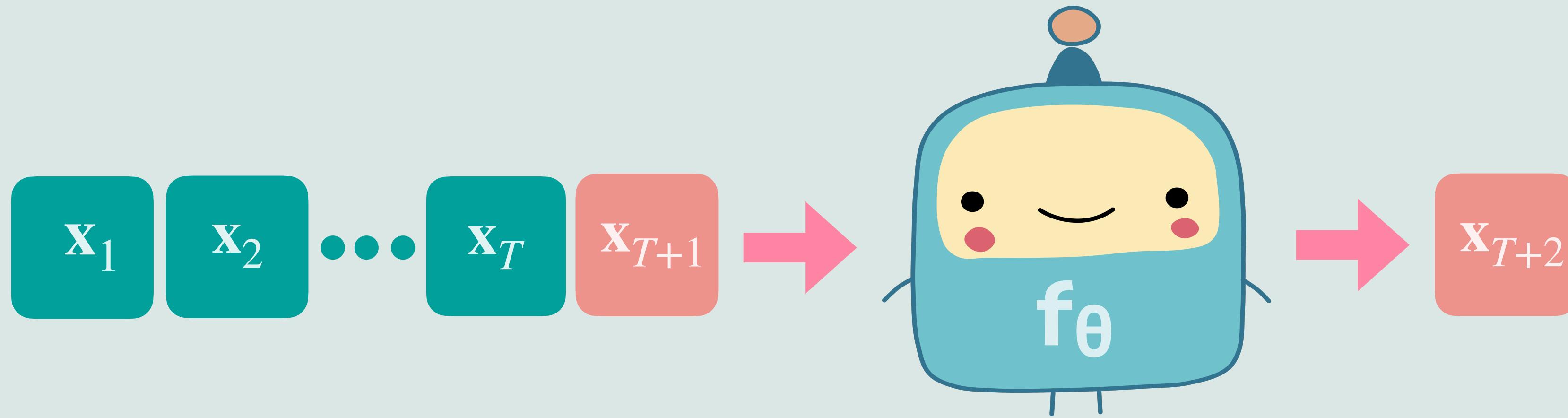


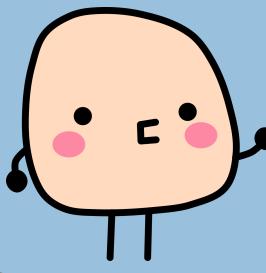
可以想成是一堆字去預測下一個字的模型





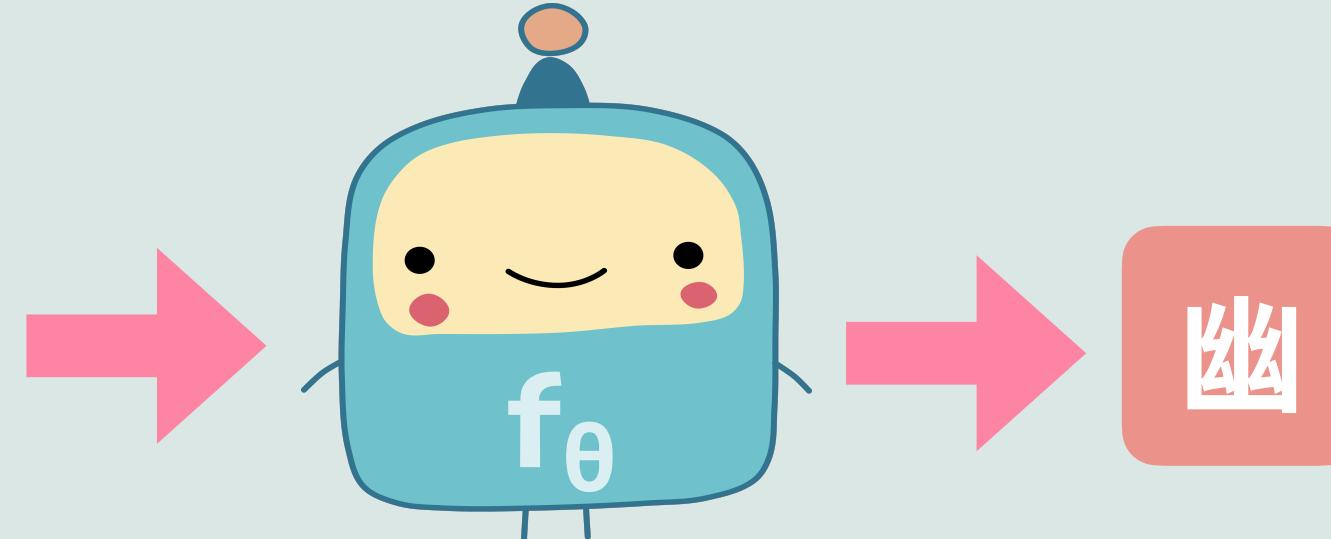
然後就可以一路下去...



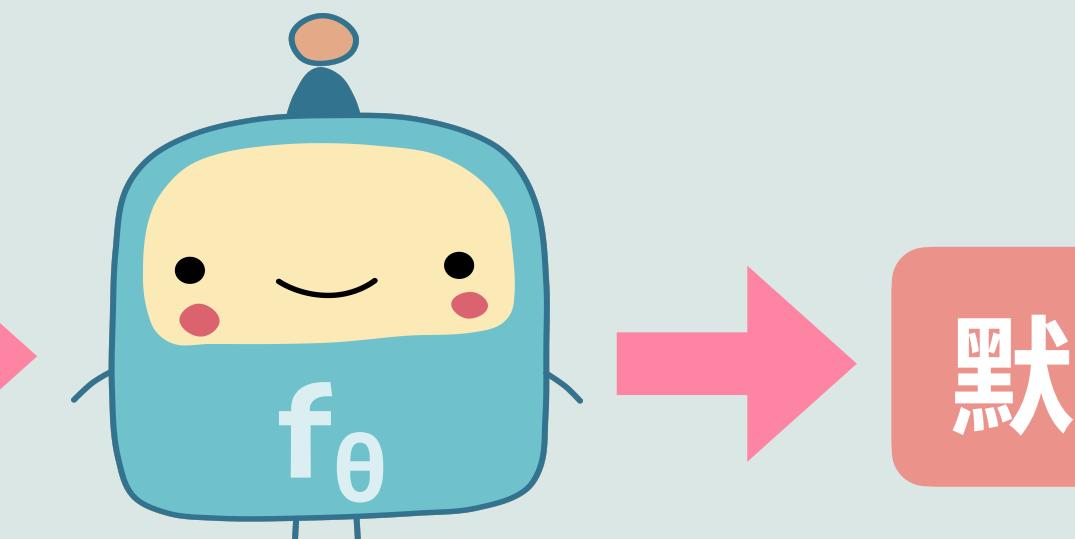


例如 (Tulu 3 70B)

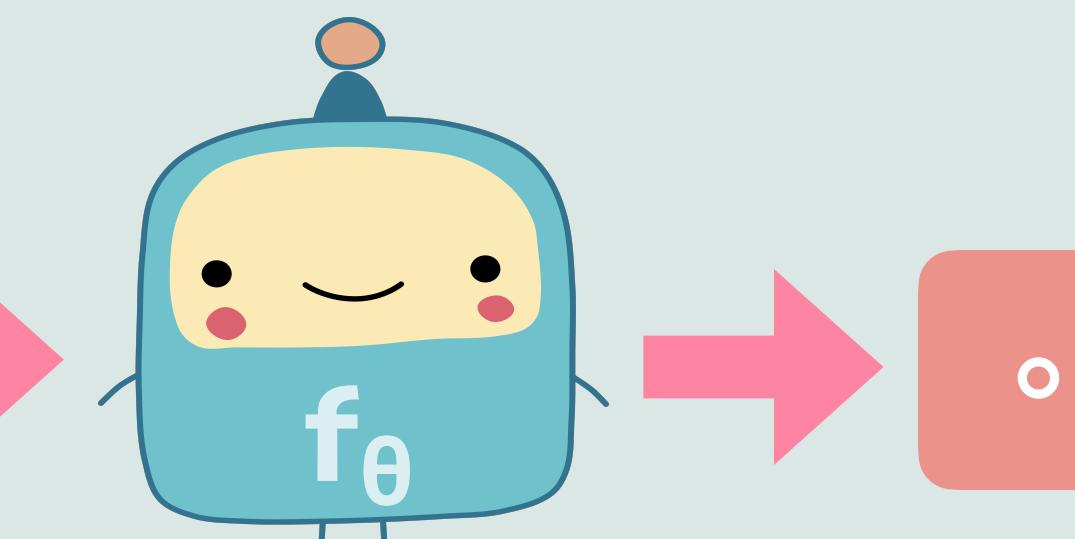
炎 龍 老 師 很

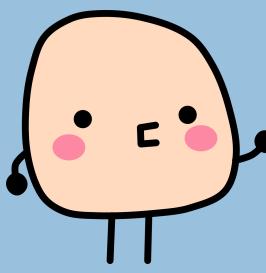


炎 龍 老 師 很 幽



炎 龍 老 師 很 幽 默





電腦只會處理數字

我們希望語言模型
可以生成的字，都給
一個編號。

帥

幽

有

熱

受

87

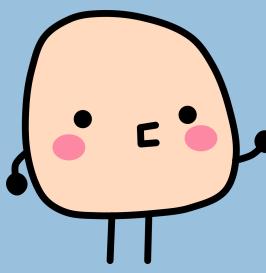
94

112

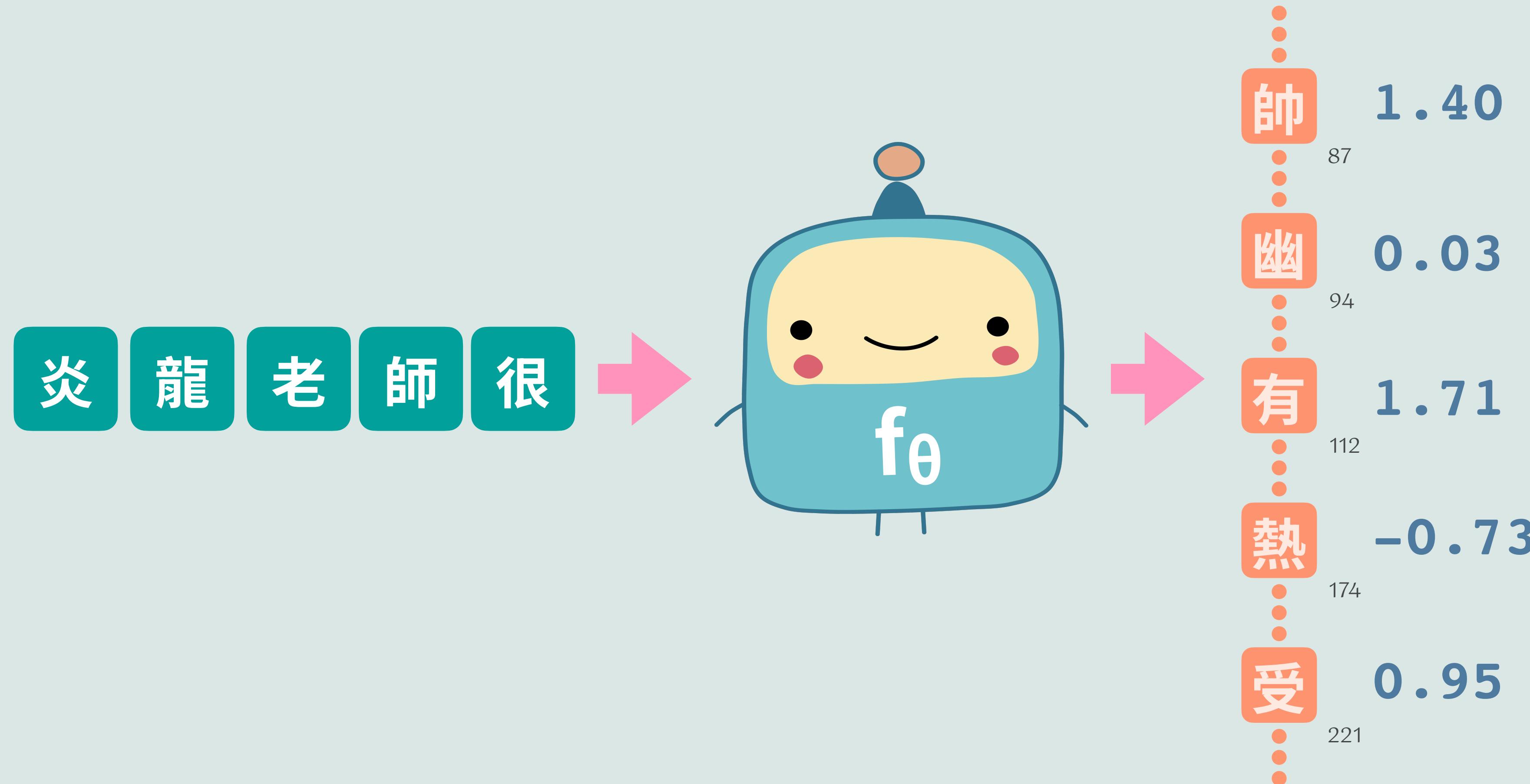
174

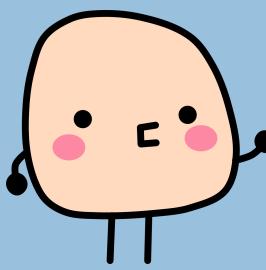
211



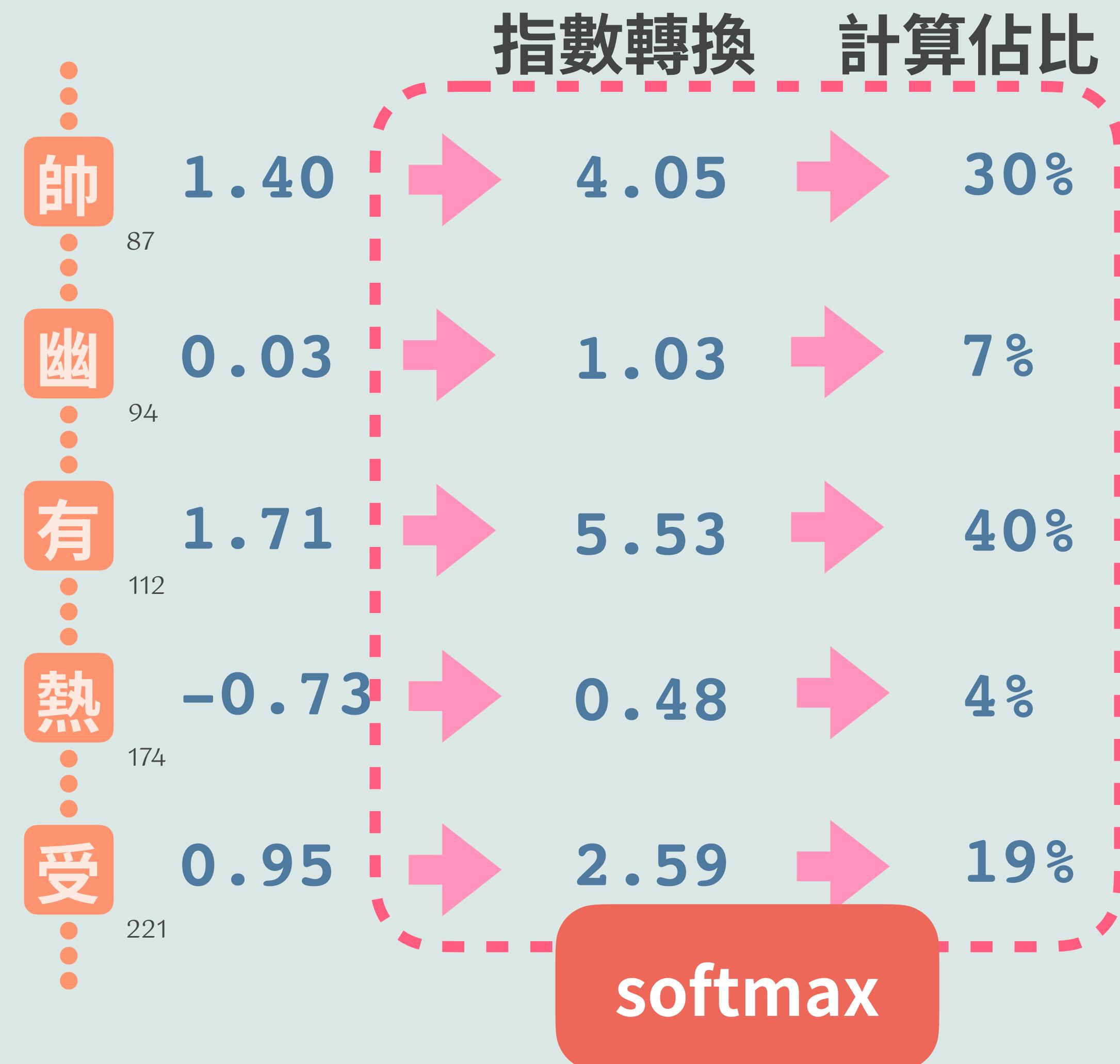


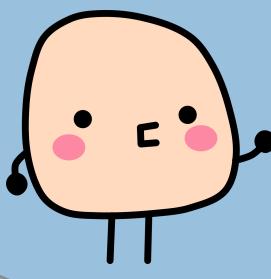
模型其實是把每個可能的字評分



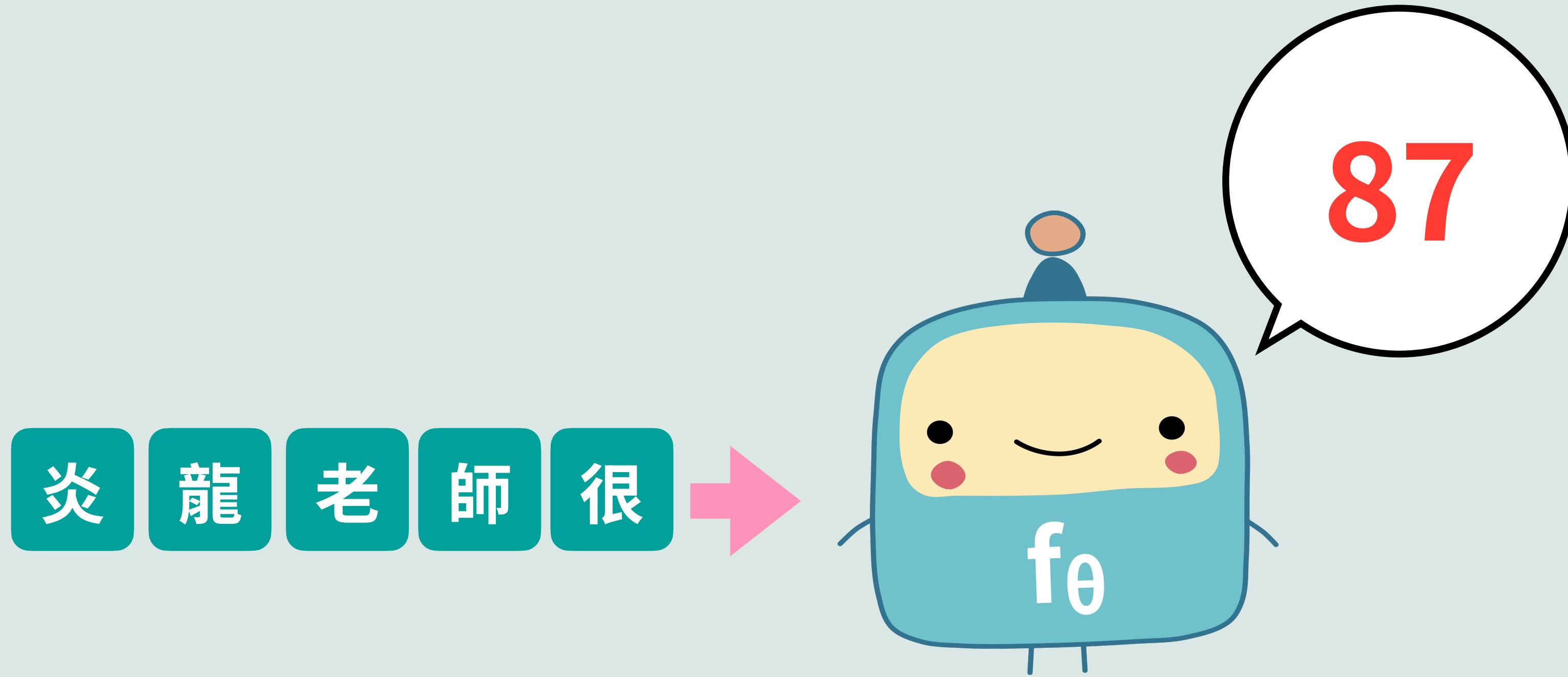


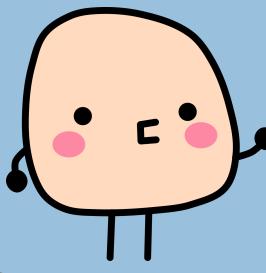
模型其實是把每個可能的字評分





語言模型不知它在做什麼，只覺得下一個字放什麼順就放



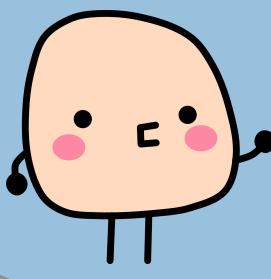


小總結

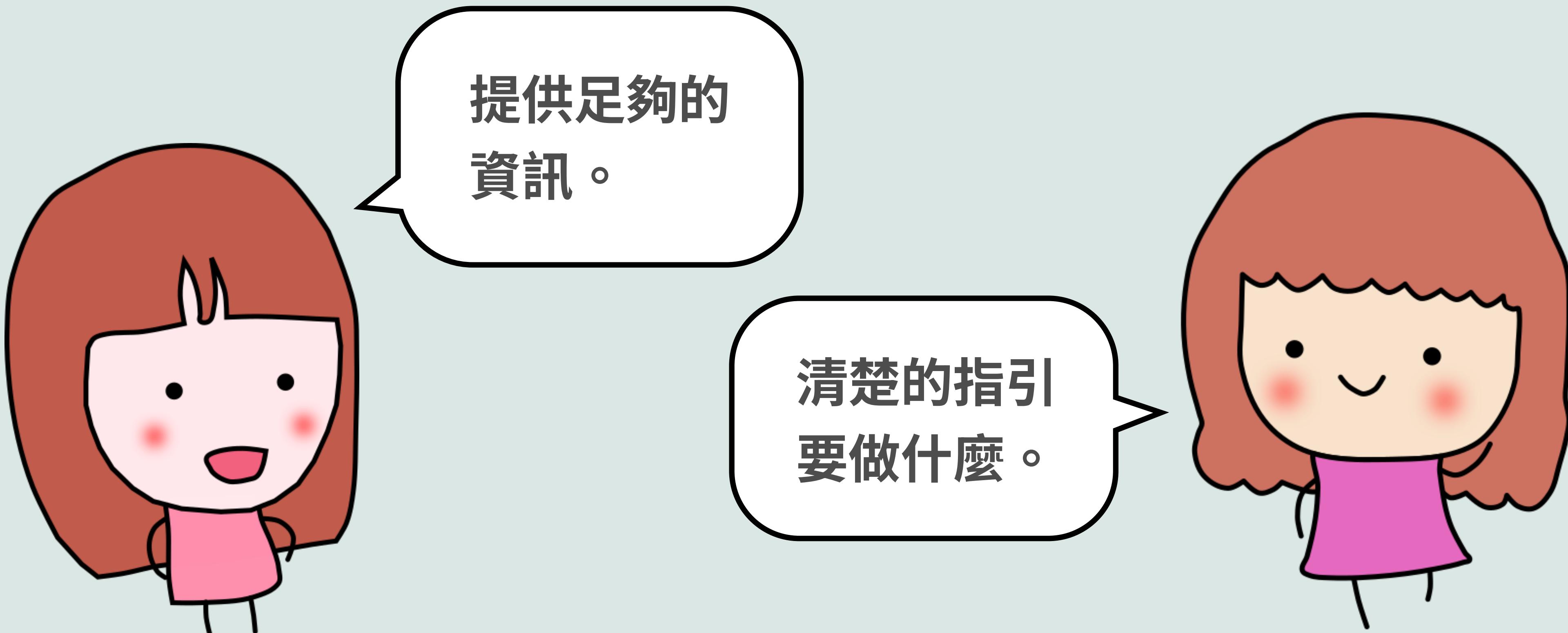
* 文字生成 AI 只是預測下一個字的呆萌型 AI 機器人。

* 訓練目標就是要把它訓練成「很會接話」的唬爛王。





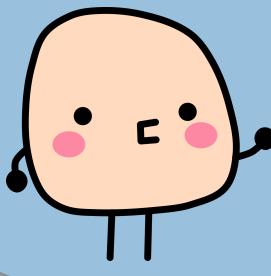
瞭解原理, 就能用好 ChatGPT 等 LLM



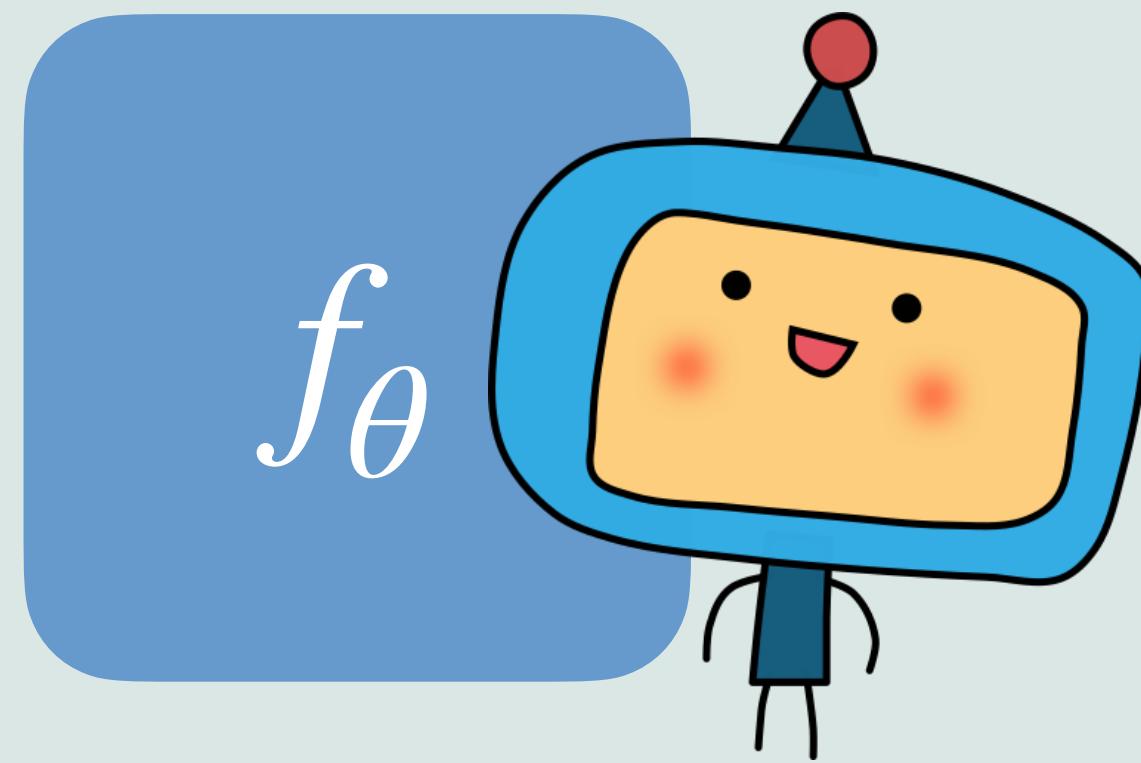


02.

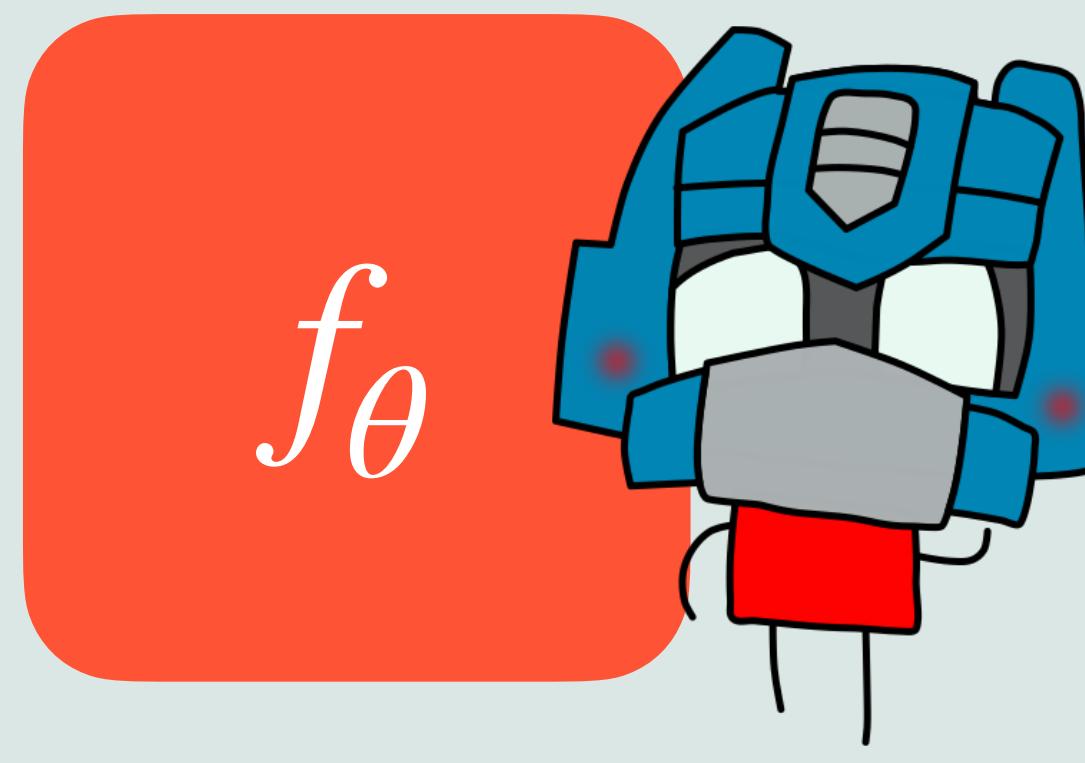
RNN 和變型金剛



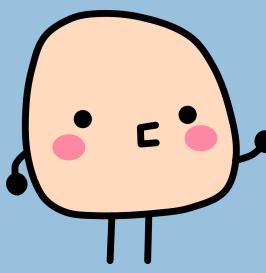
兩種有記憶的神經網路



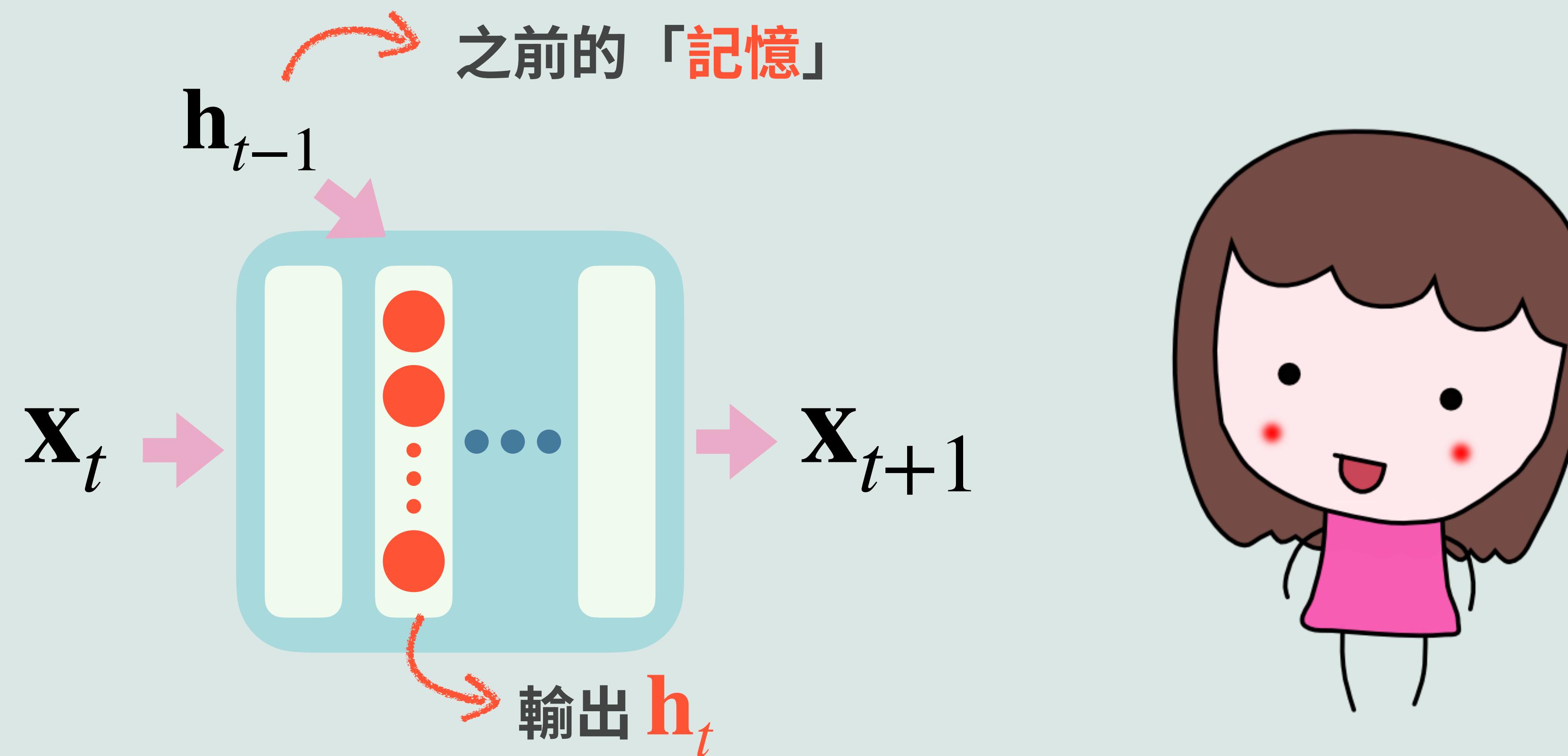
RNN
遞歸神經網路

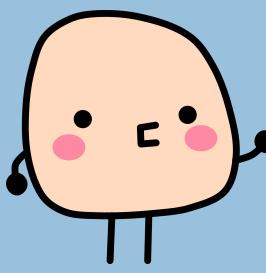


Transformer
變型金剛

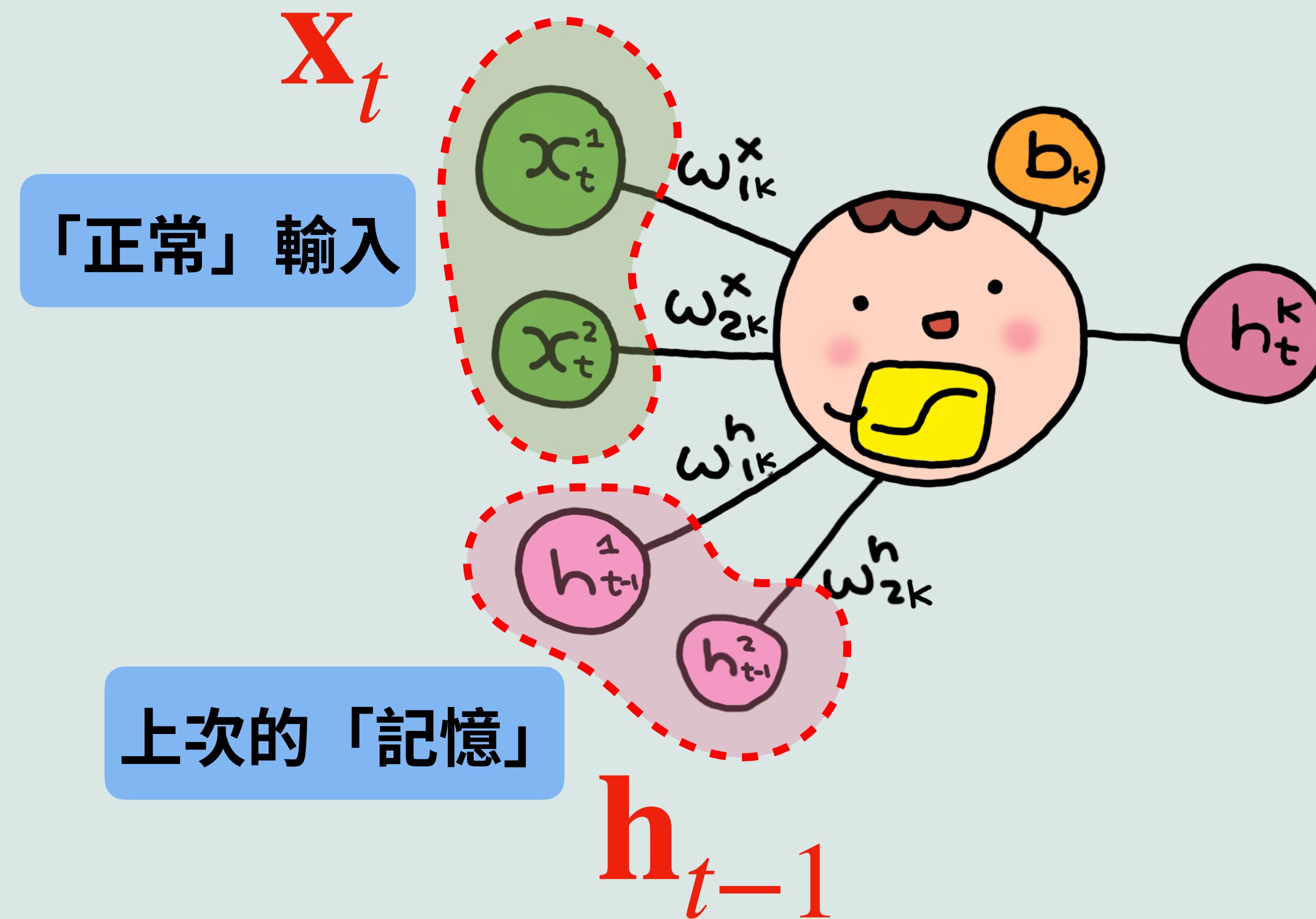


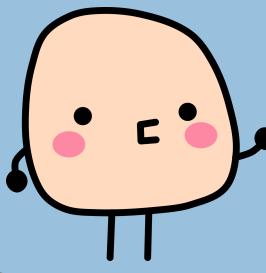
RNN 層會導入上次的記憶



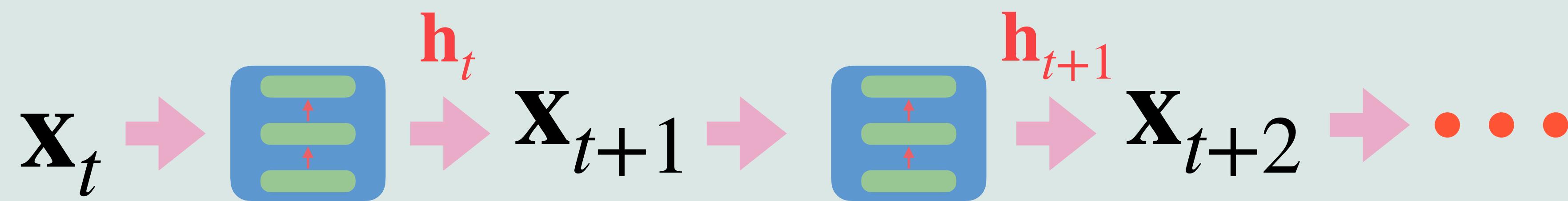


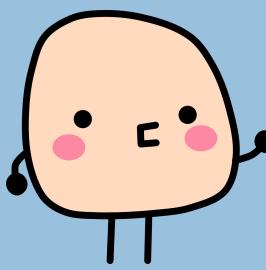
RNN 神經元和正常的其實一樣



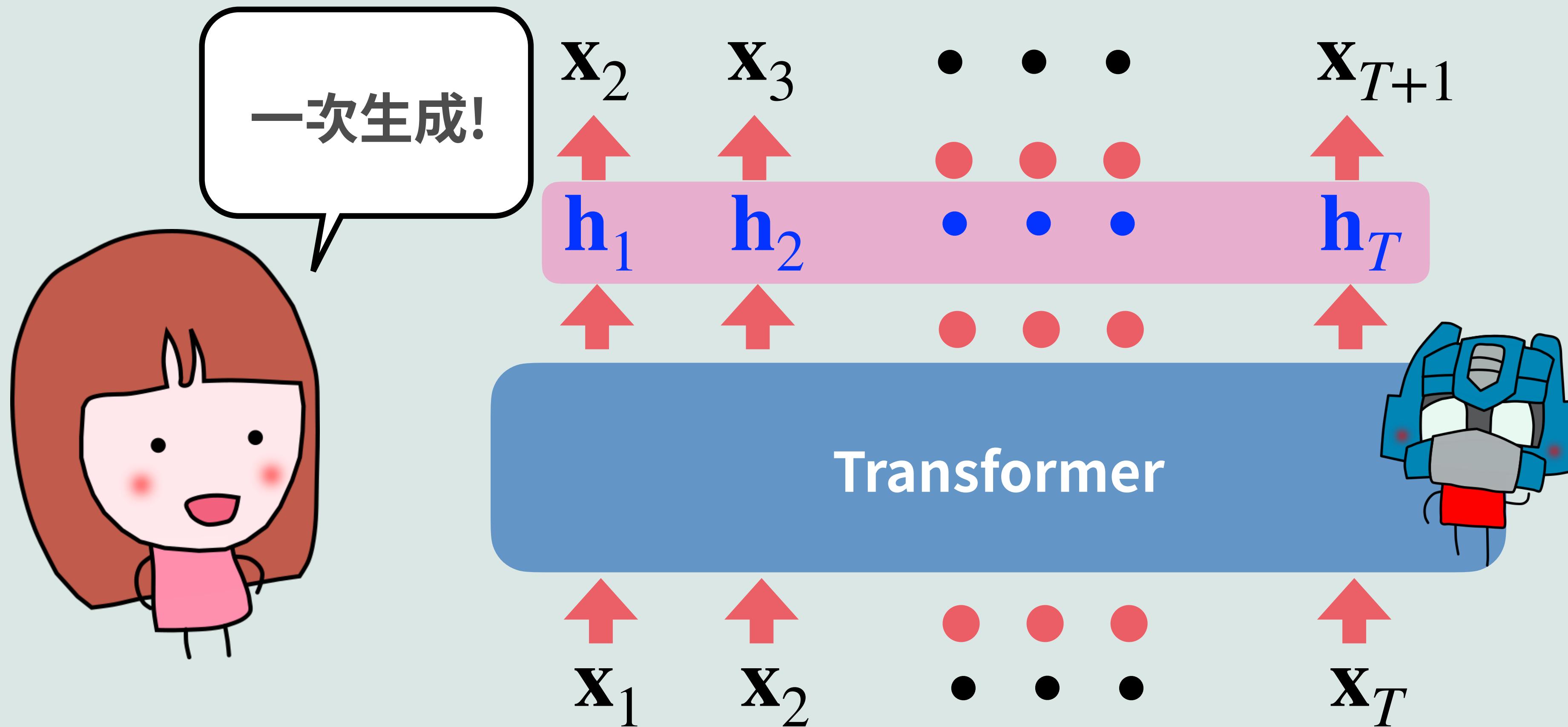


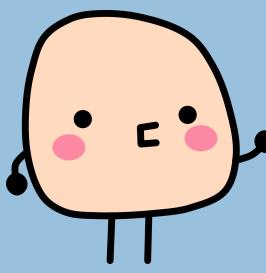
RNN 缺點是遞迴型的運算



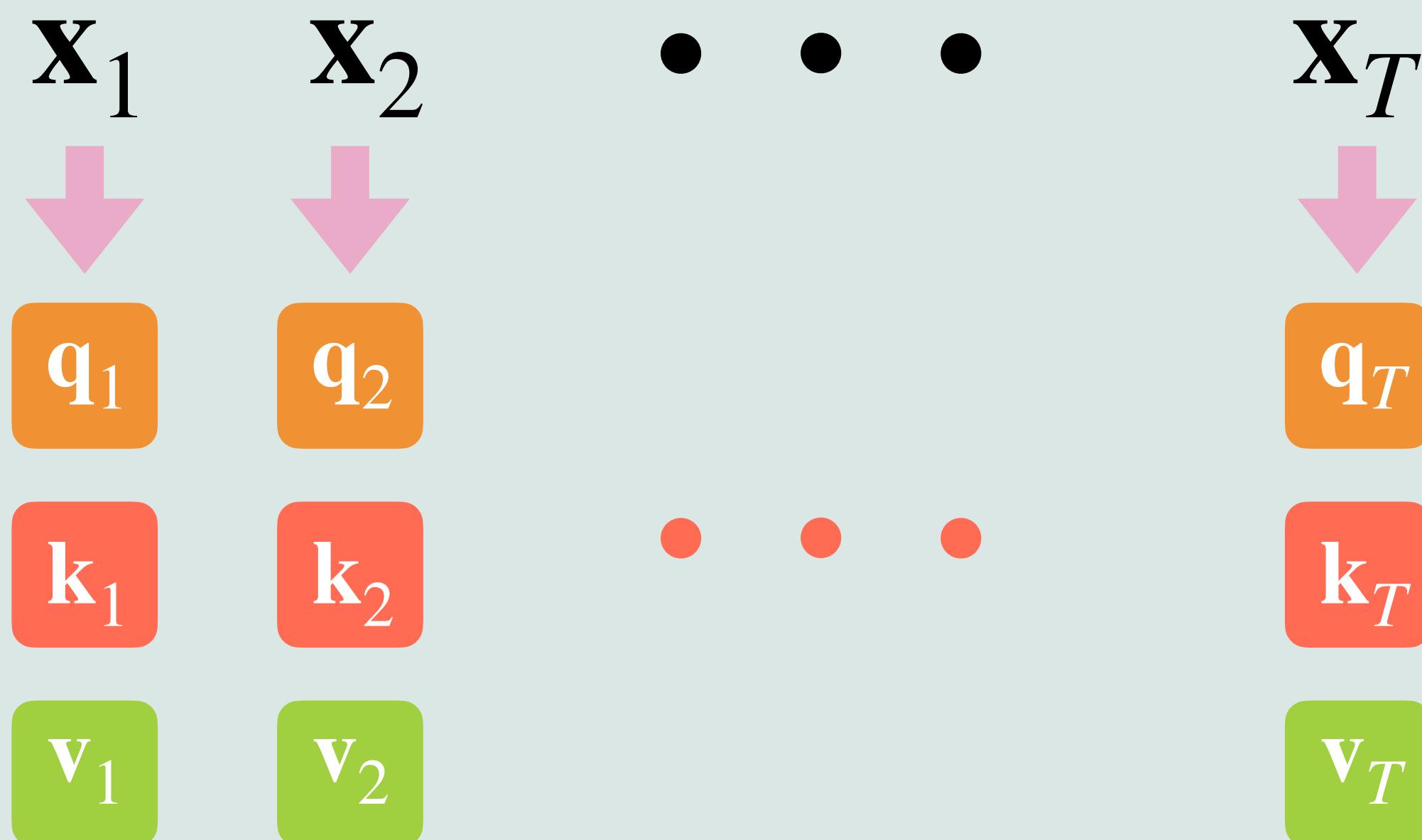


Google 想到一次生成各時間點的「記憶」





每個字生出 Q, K, V 三個特徵代表向量



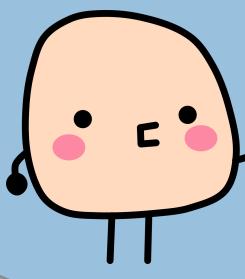
其實只是作線性轉換。

$$q_i = x_i W^Q$$

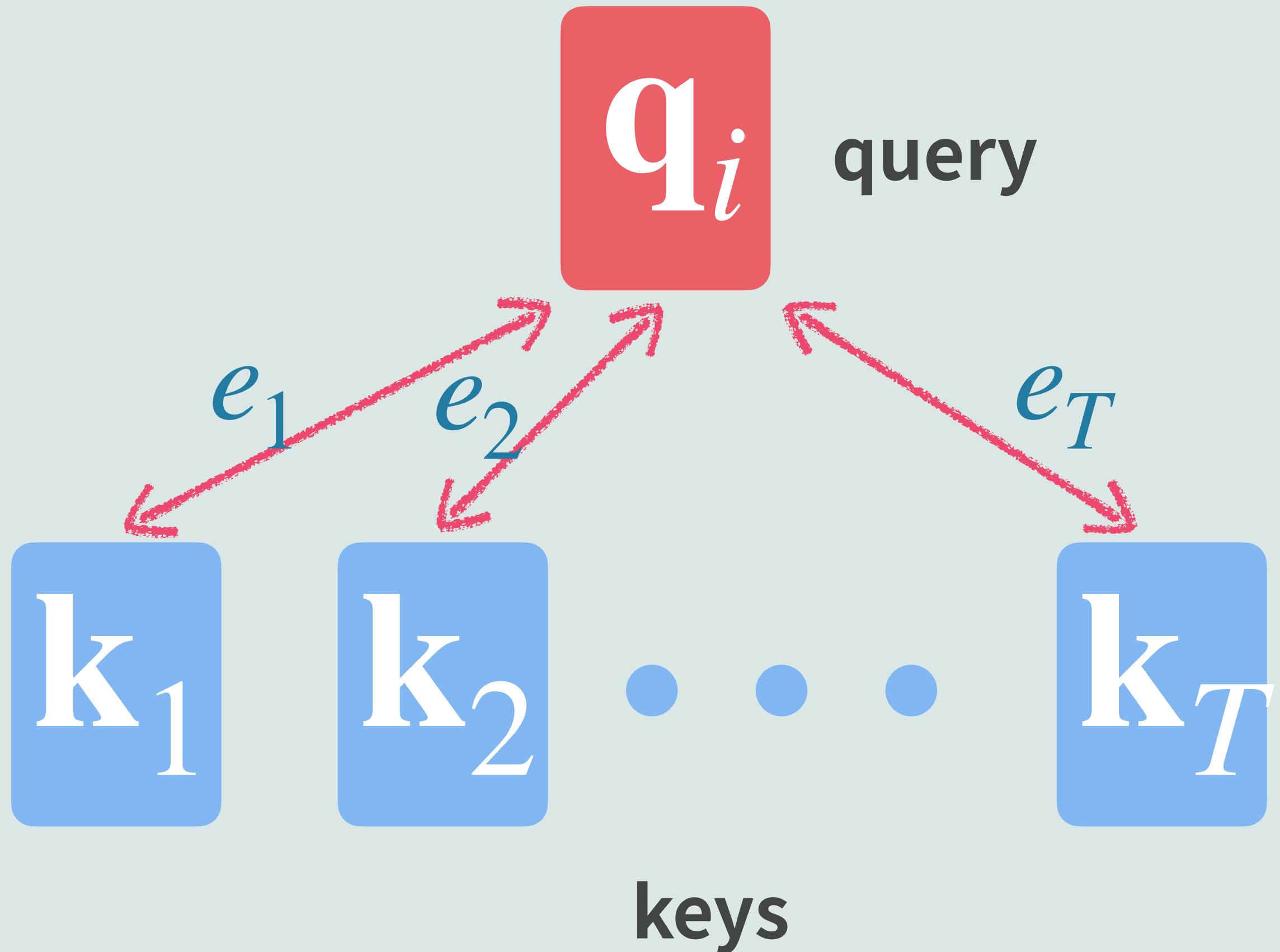
這學習來的



* Google 深愛列向量，所以線性轉換矩陣是乘在後面。



作所謂的 self-attention!

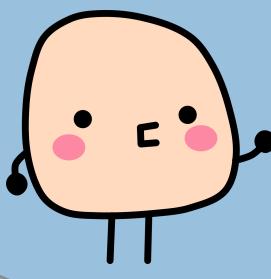


e_j 是 attention 的強度, Google 風
是用內積:

$$e_j = \mathbf{q}_i \mathbf{k}_j^T$$

e_1, e_2, \dots, e_T 經 softmax 得
 $\alpha_1, \alpha_2, \dots, \alpha_T$, 最後令:

$$\mathbf{h}_i = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \dots + \alpha_T \mathbf{v}_T$$

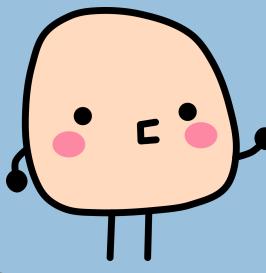


整個過程基本上是矩陣乘法

$$\mathbf{q}_i \cdot \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{k}_3 & \dots & \mathbf{k}_T \end{bmatrix}^T = \begin{bmatrix} e_1 & e_2 & e_3 & \dots & e_T \end{bmatrix}$$

\downarrow softmax

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_T \end{bmatrix}$$



最後就是有名的 transformer 公式

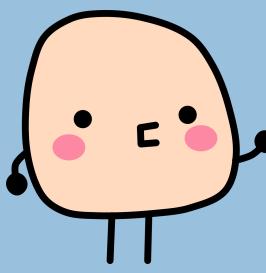
注意基本上就是矩陣乘一乘就好了！

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

d_K 是一個 key 向量的維度，其實除以自己定的一個數就好，這樣寫增加神秘感...



03.
生成模型為什麼
這麼厲害呢？



Andrej Karpathy

Andrej Karpathy blog

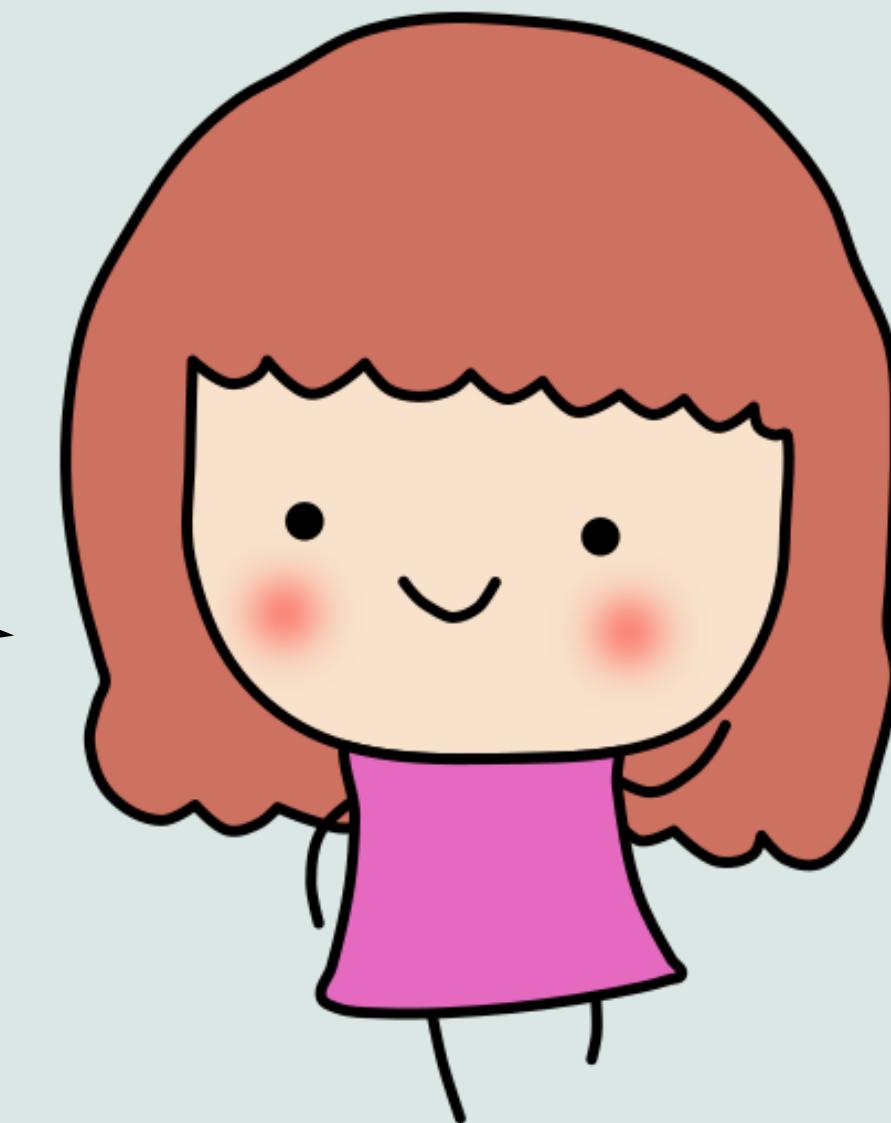
About

The Unreasonable Effectiveness of Recurrent Neural Networks

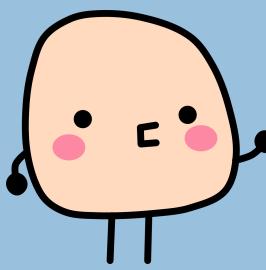
May 21, 2015

There's something magical about Recurrent Neural Networks (RNNs). I still remember when I first trained a recurrent network for [Image Captioning](#). Within a few dozen minutes of training my first basic RNN (with arbitrarily-chosen hyperparameters) started to generate very nice looking descriptions of images that were on the edge of making sense. Sometimes the ratio of how simple your model is to the quality of its outputs is so high that it blows past your expectations, and this was one of those times. What made this result even more surprising at the time was that the common wisdom was that RNNs were supposed to be difficult to train (with the notable exception of [LSTM](#)). I've in fact reached the opposite conclusion). Fast forward about a year: I'm training RNNs all the time and I've witnessed their power and robustness many times, and yet their magical outputs still find ways of amusing me. This post is about sharing some of that magic with you.

引發大家對文字生成 AI 興趣的名作。



<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>



RNN: AI 數學家

To prove study we see that $\mathcal{F}|_U$ is a covering of \mathcal{X}' , and \mathcal{T}_i is an object of $\mathcal{F}_{X/U}$.
 $i > 0$ and \mathcal{F}_p exists and let \mathcal{F}_i be a presheaf of \mathcal{O}_X -modules on \mathcal{C} as a \mathcal{F} -module.
In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1} \mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\text{Sch}/S)_{fppf}^{\text{opp}}, (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \text{Spec}(A))$$

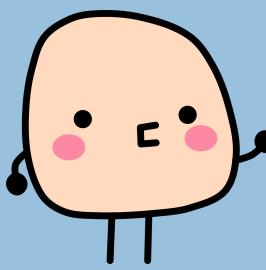
is an open subset of X . Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S .

Proof. See discussion of sheaves of sets. □

電腦覺得自己
是個數學家！



電腦產生代數幾何介紹 "Stacks" 的片段!



RNN: AI 莎士比亞

潘達洛斯：

哎呀，我想他即將來臨的時日，
當微小的負擔終將被承受，卻從未得到餵養，
而他不過是死亡鎖鏈中的一環，
我將無法安眠。

第二位參議員：

這些苦難遠離了他們，卻深深烙印在我的靈魂，
破碎且強烈，當我逝去時，
大地與許多國家的思緒也將隨之埋葬。



電腦覺得自己
是莎士比亞!



看完一本紅樓夢的 AI



紅樓夢生成器

起個頭，幫你完成一段紅樓夢。可以改變 temperature，越小生出的字越固定，越大越隨機。

TEXTS
自孫悟空從石頭中蹦出來之後，

N CHARS
 200

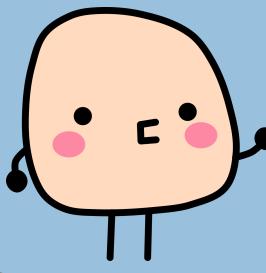
TEMPERATURE
 1

Clear **Submit**

OUTPUT

自孫悟空從石頭中蹦出來之後，說：
「俗，先中用佛嘴罷。」雨村道：「寶玉既回來，又是聞世，也是知道奇異而不必丟。」土隱道：「老先生莫怪拙言，貴族之女，俱屬再聞？據伊禍，我們家兄可傳，舛錯，不如我再有細產，今日，老爺仍是接世。今日弟引近因。」說得賈政嘆了口氣。王夫人這樣一回，說：「老太太給他打聽，打聽發定主意，只要在家裡就完了。」邢夫人說他這樣一個，那裡都願意跟著，毫無蹤跡。裡頭邢夫人更哭著打聽「妞兒不用著了急，她榮兩府的人，也不

Flag



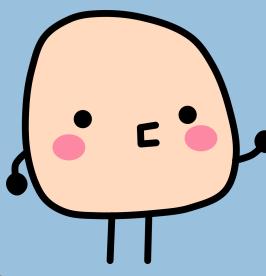
Prompt: 自孫悟空從石頭中蹦出來之後...

自孫悟空從石頭中蹦出來之後，先飛報單的。賈政嘆口氣。手又賞了一本，與他老先邀雨村回話。

那瓦礫明乍寒了一番補入，毫不進，說著：「悟如從此，從天入三洞天。因開燈，古道今日，原來如今的奇處，顏色大大道如此。」士隱自趕的道：「二老爺通報仙草已經失了那仙翁，你把這條猩紅氈的斗篷，向小子同賈珍珠方相認，俟薛姨媽那邊設停寺內。」

次日，鳳姐等分頭取藥那劉姥姥家的，眾人打發人去給邢夫人辭了眾人。王夫人命賈蘭跟了侍兒去，分頭只一顧頭回來看著賈環的相思。王夫人便告訴了平兒，在賈芸家外沒有，不敢答言，只有好心盛妝。

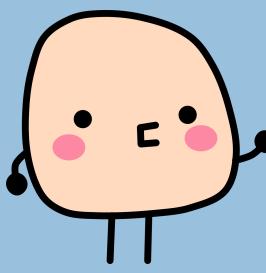
那空空牢牢記著此言，又把五更五更收心起題，寫著書去。回奏：「皇上隆恩，內眷們不及大理的。」賈政帶著賈蘭場中迷失。



Karpathy 認為 AI 生成的都是「幻覺」

在他 2015 年著名的部落格文章
"The Unreasonable Effectiveness of Recurrent Neural Networks" 把 AI 生成的叫 **hallucinated** (幻覺的) 版本。





Karpathy 認為幻覺是 LLM 最偉大的特點



Andrej Karpathy ✅

@karpathy

...

On the "hallucination problem"

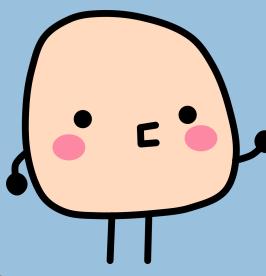
I always struggle a bit with I'm asked about the "hallucination problem" in LLMs. Because, in some sense, hallucination is all LLMs do. They are dream machines.

We direct their dreams with prompts. The prompts start the dream, and based on the LLM's hazy recollection of its training documents, most of the time the result goes someplace useful.

It's only when the dreams go into deemed factually incorrect territory that we label it a "hallucination". It looks like a bug, but it's just the LLM doing what it always does.

At the other end of the extreme consider a search engine. It takes the prompt and just returns one of the most similar "training documents" it has in its database, verbatim. You could say that this search engine has a "creativity problem" - it will never respond with something new. An LLM is 100% dreaming and has the hallucination problem. A search engine is 0% dreaming and has the creativity problem.

- ✓ 從某種意義上來說，幻覺正是大型語言模型 (LLMs) 所做的全部。它們是作夢的機器。
- ✓ 搜索引擎是 0% 的夢境，但不會創造。
- ✓ 人們實際上的意思是，不希望一個 LLM 助理發生幻覺。
- ✓ LLM 沒有「幻覺問題」。幻覺不是一個錯誤，它是 LLM 最偉大的特點。



震驚世界的 GPT 嘘爛王系列

SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

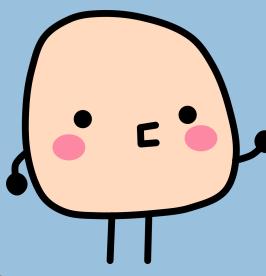
Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

著名的「發現獨角獸」
唬爛文章。

Better Language Models and Their Implications

<https://openai.com/blog>



GPT-2 參數量大到可怕!

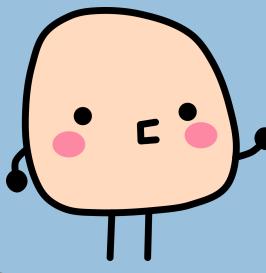
紅樓夢生成器

300萬



GPT-2

15 億!



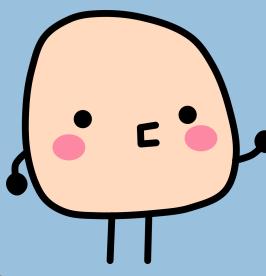
唬爛王第三代 GPT-3

大到無法下載! 只
能申請 API 使用。



1,750億

唬爛功力更是前所未見!



正常人一生讀多少字呢？

ChatGPT 算錯了！總之差不多 2 億字。



假設一個人每天花費30分鐘閱讀，每分鐘可以閱讀250個字，那麼一年下來，這個人可以讀書約27萬字。如果這個人一生閱讀時間持續這樣，那麼他或她在70年的壽命中可以讀書約5300萬字。

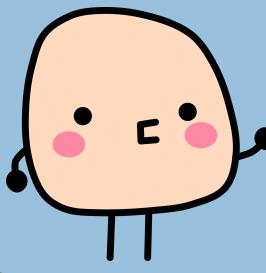
GPT-3 大約精讀了 4990 億的字，一般人要活 9415 次才能達成！



* Gemini 說 10 億字

* Claude 說 3 億字

* Grok 說 2-2.5億，認真閱讀的 5 億



LLM 大型語言模型



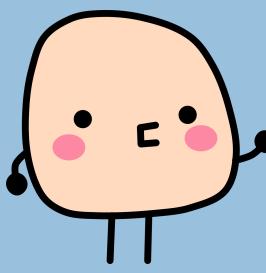
大型語言模型

Large Language Model



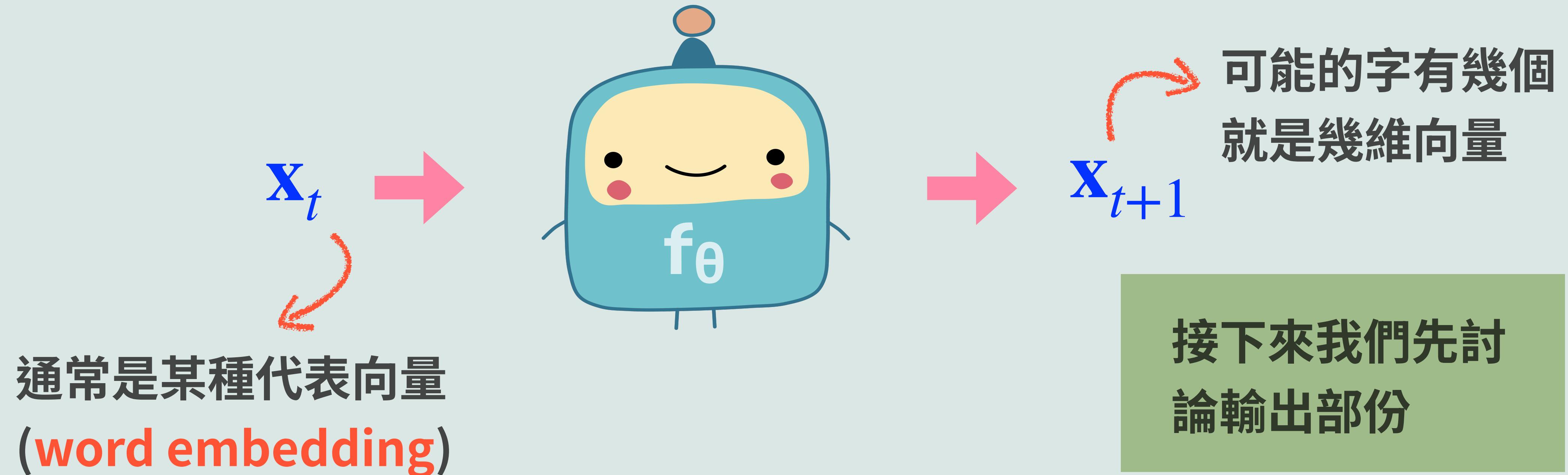
04.

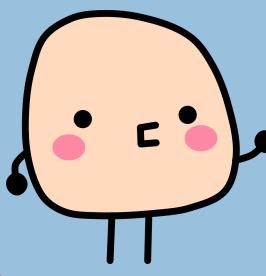
選字的溫度



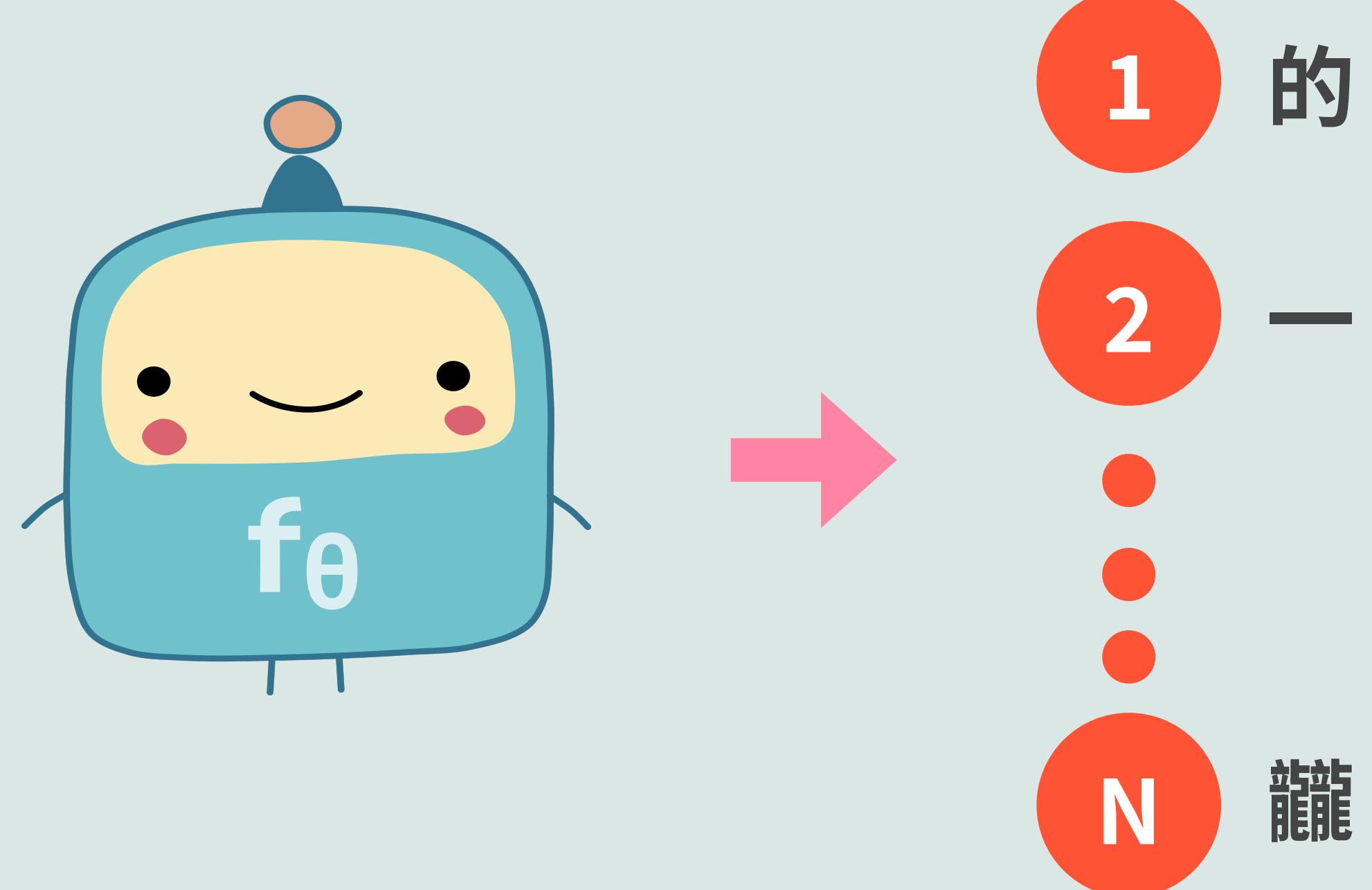
文字生成模型輸入和輸出的樣子其實不同

在 NLP 「前一個字預測下一個字」 的生成模型，輸入和輸出
通當是不同類型...

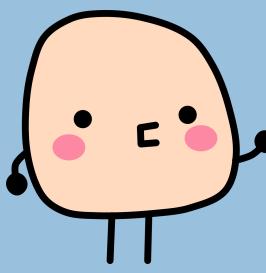




輸出和八哥模型是 87 分像

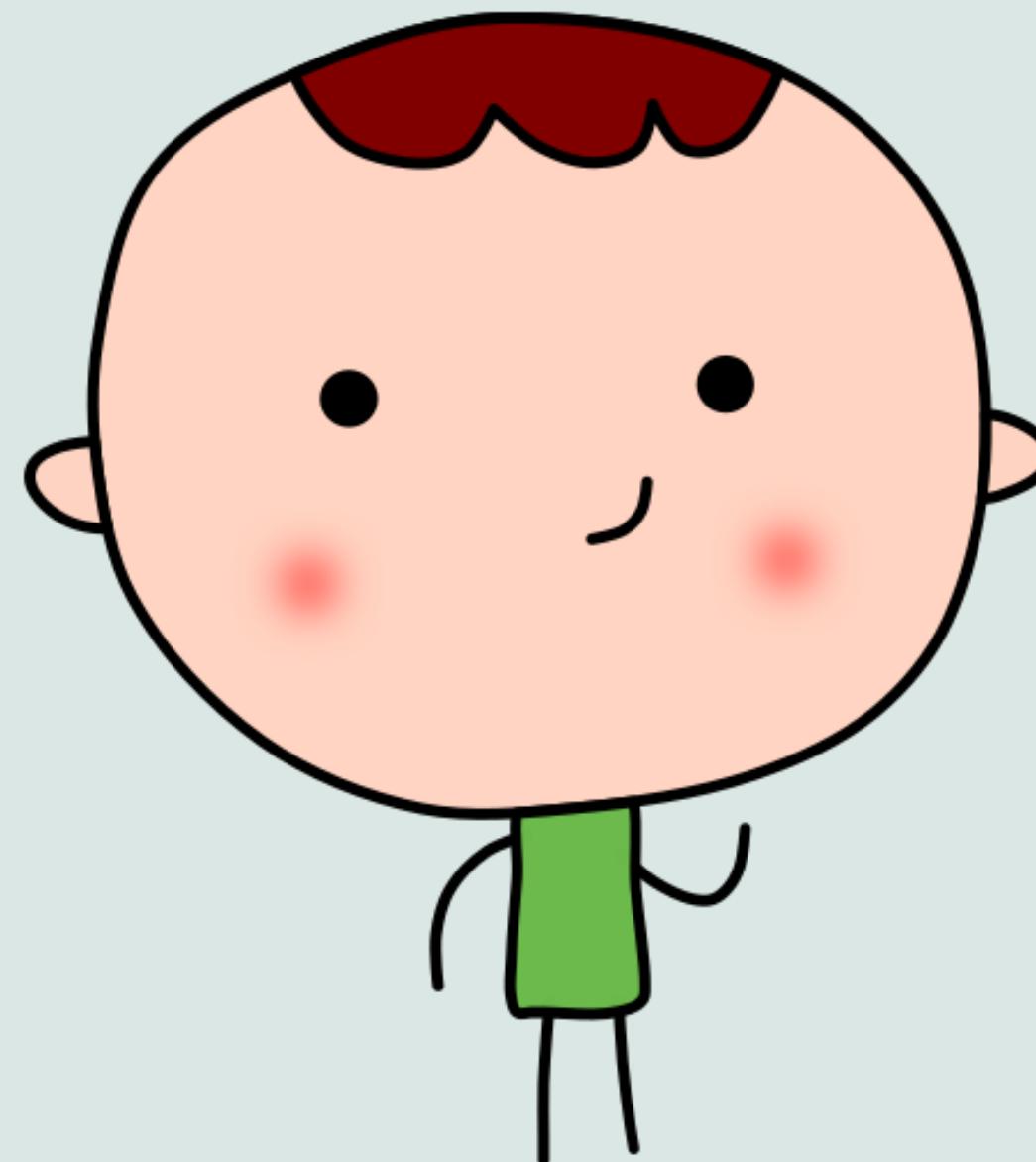
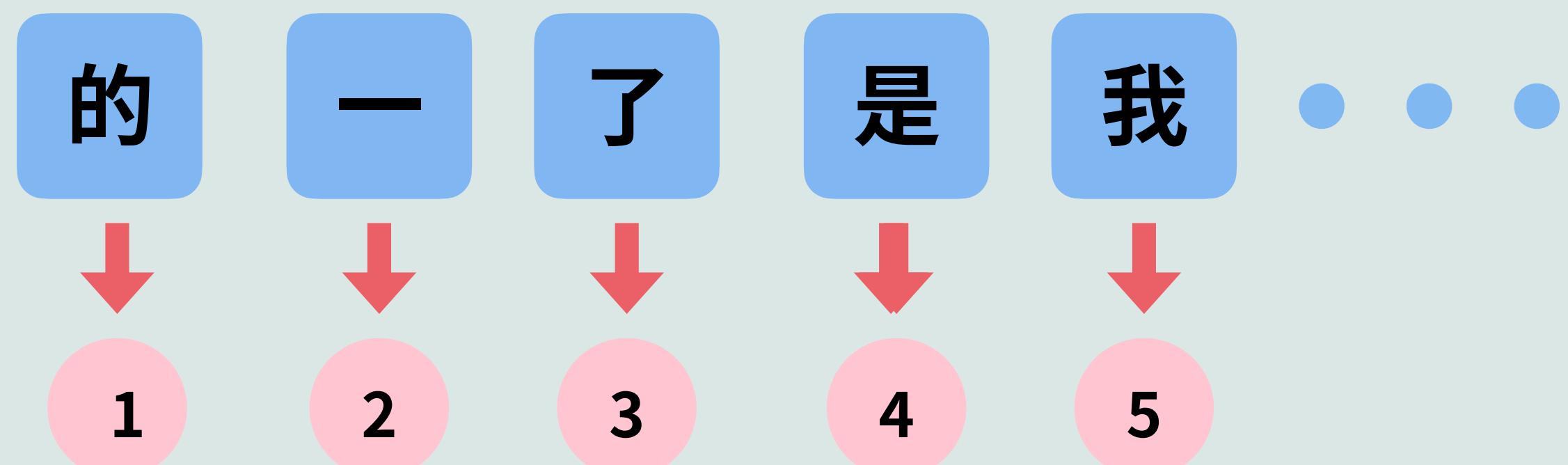


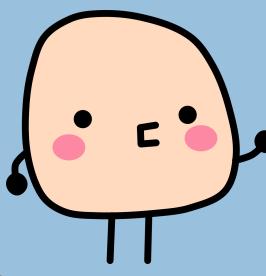
如同一般分類問題，最後會作 softmax，分數（機率）越高
越容易被選中。



依字出現的頻率排

最常見的方式是我們把字依出現的頻率排序，
越常出現給的編號越小。





依頻率排的原因可以問 ChatGPT



為什麼自然語言處理 (NLP) 文字的編號，出現頻率越高，代表的編號越小呢？

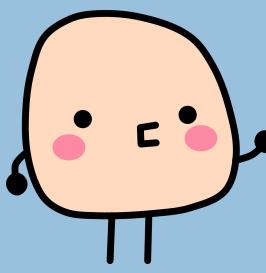


在自然語言處理 (NLP) 中，文字通常會被編號，並且使用頻率越高的文字越小的編號。這種做法可以讓模型訓練和運行更高效，因為使用頻率較低的文字的編號較大，所以在記憶體和計算中會占用更多的空間。



例如，假設你正在處理一個大型英文文本檔案，其中 "the" 和 "and" 是最常用的文字。在這種情況下，你可能會將 "the" 和 "and" 分別指定為編號 0 和 1，而其他較少使用的文字則會有更大的編號。這樣一來，當你建立一個字典或者矩陣來表示文本時，它們會占用較少的空間，並且在訓練和運行模型時更加高效。

另外，使用頻率越高的文字編號越小也可以讓模型的輸出更容易解讀。因為常用的文字通常具有更大的意義，所以使用較小的編號表示它們有助於模型的解讀。



選字原來也是有學問的!

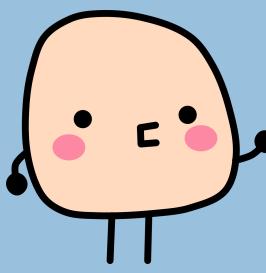
temperature

top-k

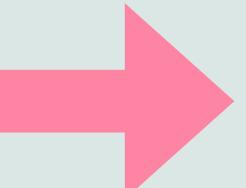
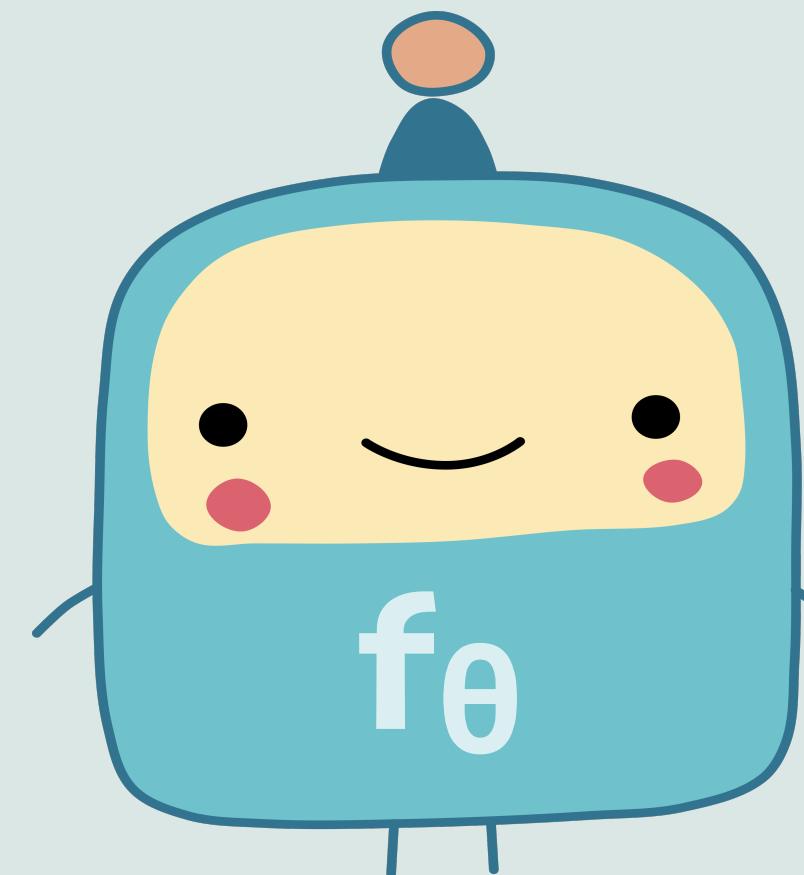
top-p

隨機選字的策略，大概有調整 temperature, top-k, top-p 等等方式。





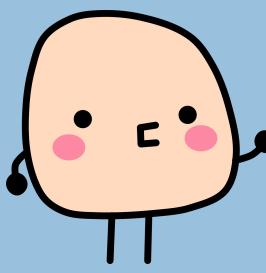
最重要的是依機率抽樣



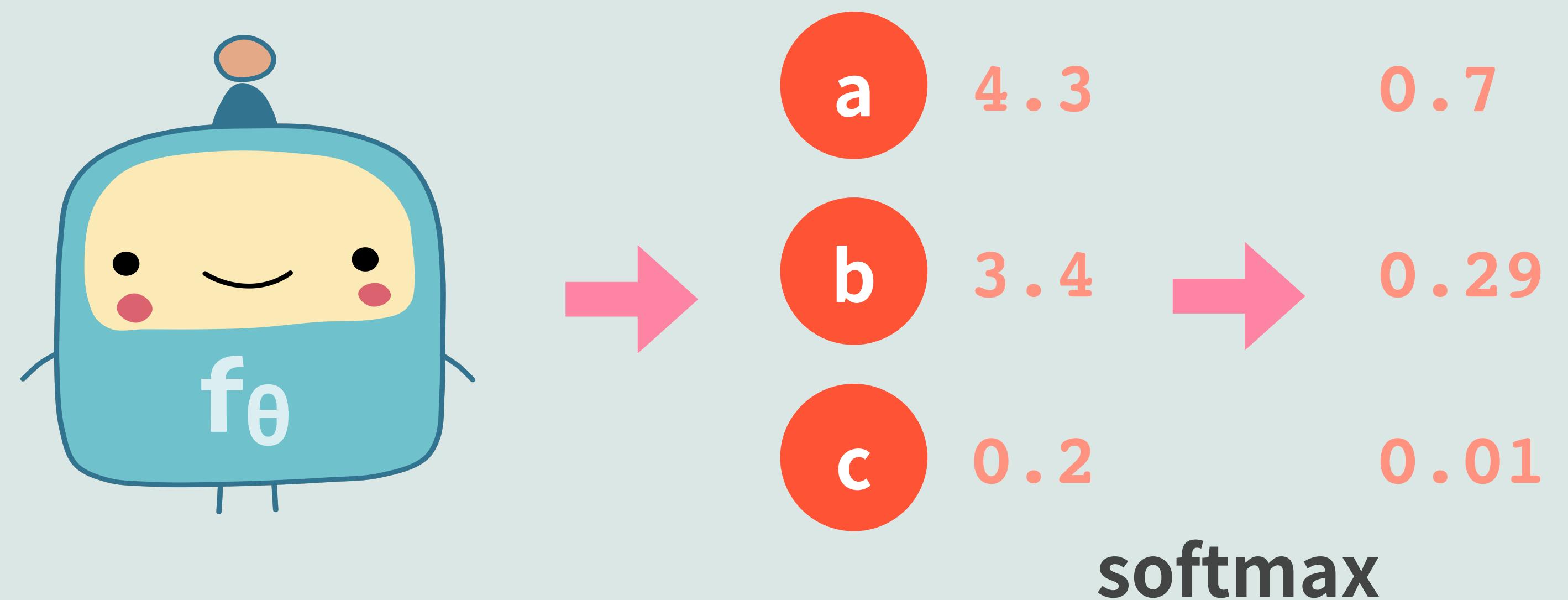
- 1 的 0.7
- 2 一 0.29
- 3 了 0.01

這樣 70% 的機率選到
「的」，29% 會選「一」，
只有 1% 選「了」。

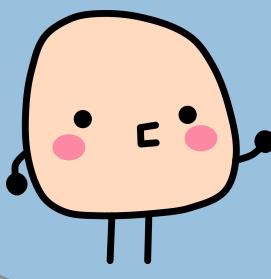




注意 softmax 是「贏者通吃」！



這裡會發現，原來
前兩名感覺沒差
那麼多！



關鍵在做了指數轉換

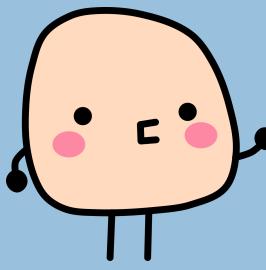
$$a \rightarrow a' = e^a$$

a 經 softmax 轉換為：

$$b \rightarrow b' = e^b$$

$$\frac{a'}{a' + b' + c'}$$

$$c \rightarrow c' = e^c$$



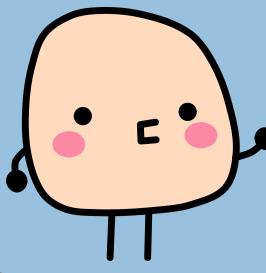
所以常用 temperature 來調整

$$a \rightarrow a' = e^{a/\tau}$$

$$b \rightarrow b' = e^{b/\tau}$$

$$c \rightarrow c' = e^{c/\tau}$$

因此 $\tau > 1$ 時會拉近彼此的機率 (更隨機); 而 $\tau < 1$ 時機率更拉開 (也就是生成文字更固定)。



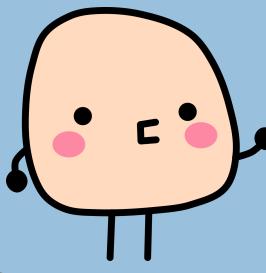
雖然常常是取 log 再除

最後 softmax 出來的機率: p_1, p_2, \dots, p_N

取 log 再除以 τ : $\log(p_1)/\tau, \log(p_2)/\tau, \dots, \log(p_N)/\tau$

這是因為原本我們是用指數轉換, 現在 log
再轉回去 (事實上要取 softmax 才得原機
率, 不過這裡不討論細節)。



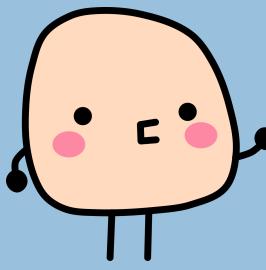


但這樣機率低的還是有可能被選到!!

這樣選字機率低的
字雖然比較難被選
到, 但還是有可能...

於是時常不知在亂說什麼...





Top-p 就是這樣做

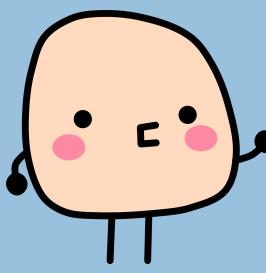
先決定一個機率 (比如 $p = 0.9$)。

從機率最高的字開始加, 看什麼時候大於 p 。

$$\sum_{i=1}^N p_i > p$$

於是只選前 N 個字。





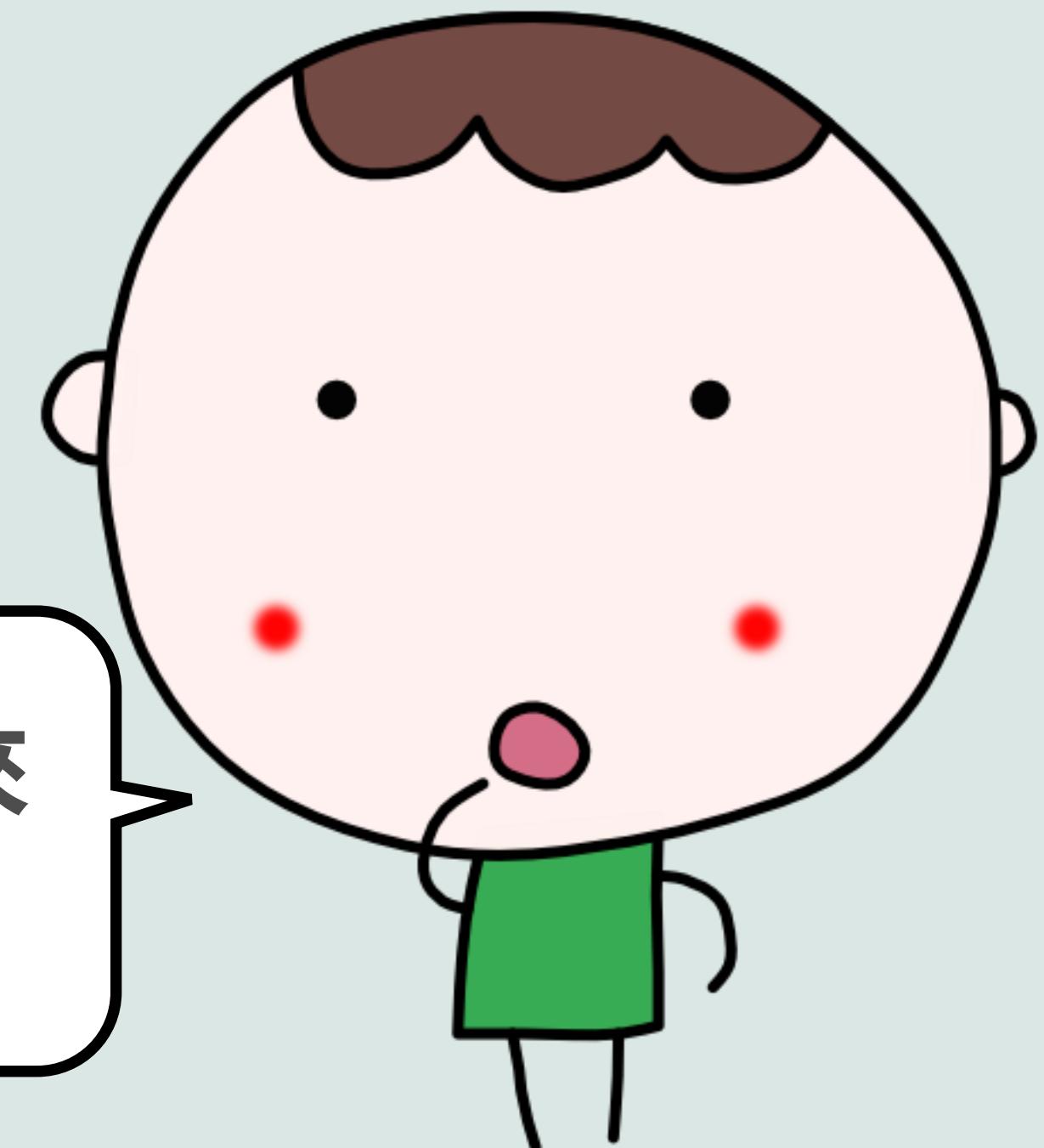
Redistribution

這時前 N 名的字機率加起來通常不是 1。

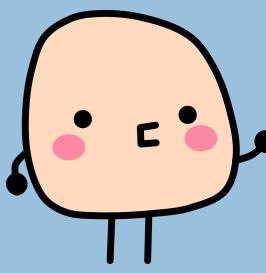
於是我們做「好高級」的 redistribution...

$$p'_k = \frac{p_k}{P}$$

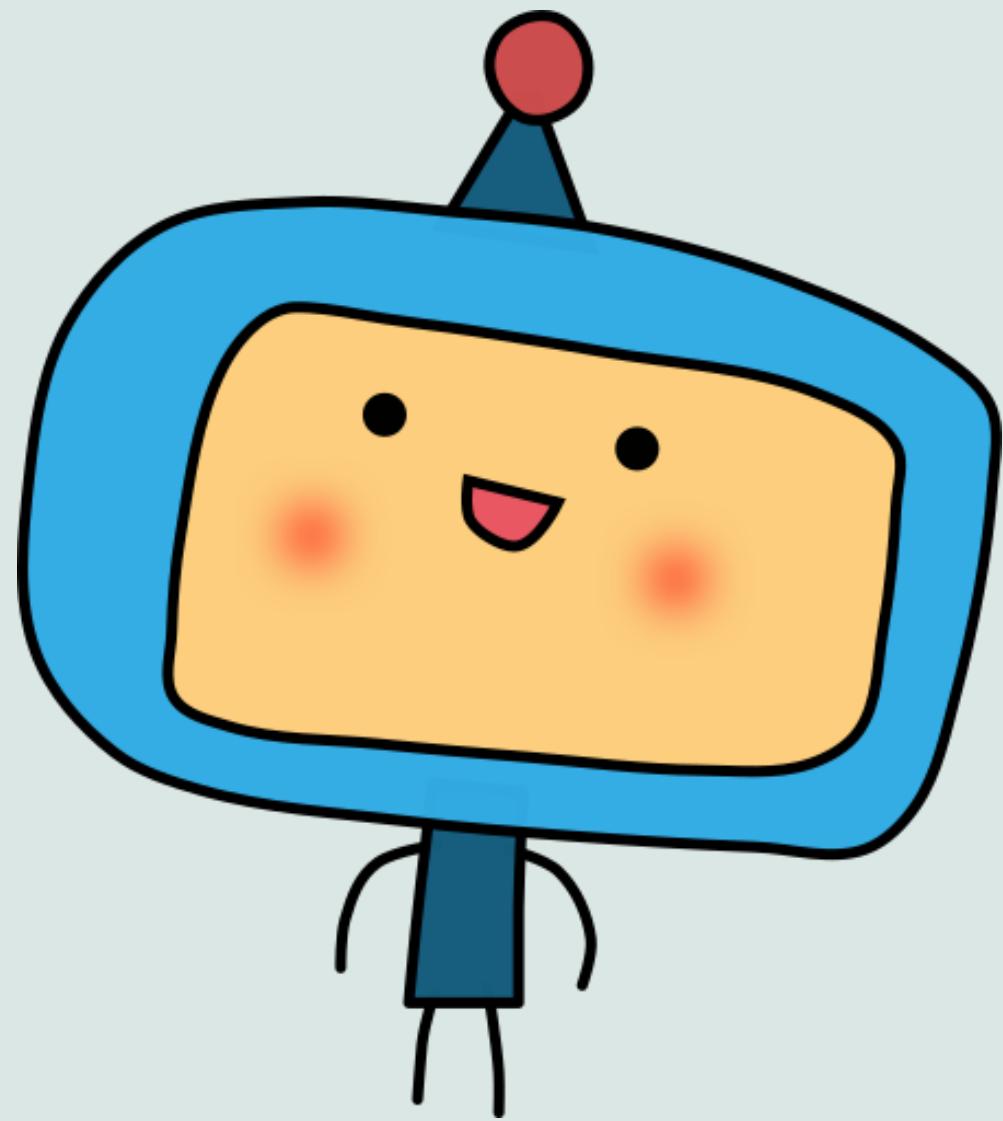
不就最簡單讓加起來
等於 1 的計算...



* 這裡的 $P = \sum_{i=1}^N p_i$

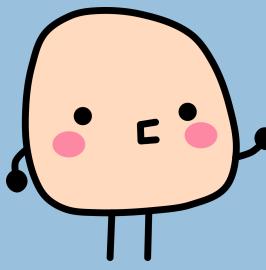


最讓人驚訝的是 top-p 很晚才出現



不要沒禮貌, top-p 正式名稱叫 nucleus sampling, 是 AI 頂會 ICLR 2020 的文章。

- * Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, Yejin Choi, “The Curious Case of Neural Text Degeneration,” ICLR 2020



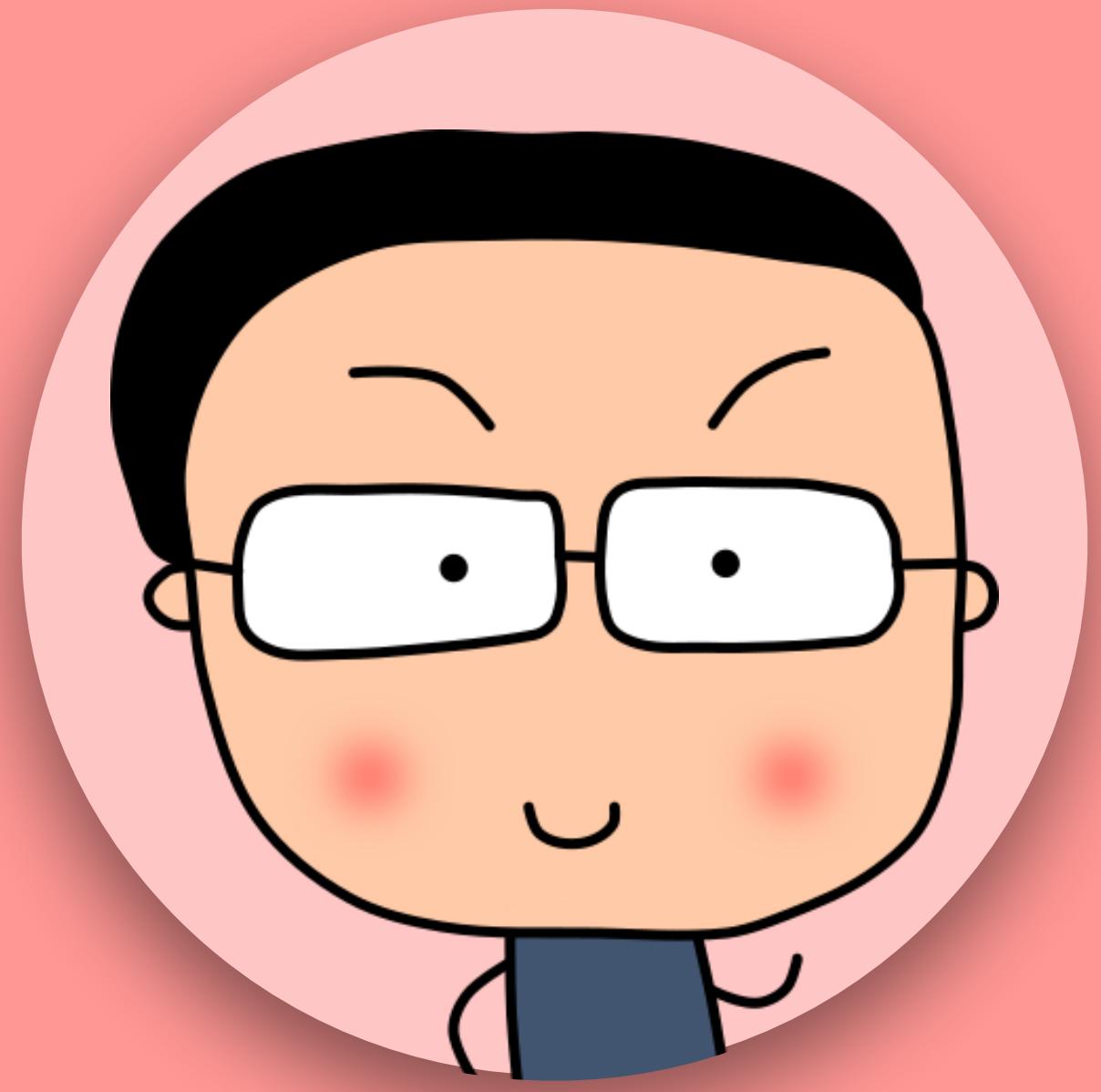
總之文字生成是一堆字生一個字的模型

總之，生成模型就是用前面的字，生出一個機率分佈，再選一個策略，依此 sample 出下一個字。

$$P_{\theta}(w_i \mid x_1, x_2, \dots, x_T)$$

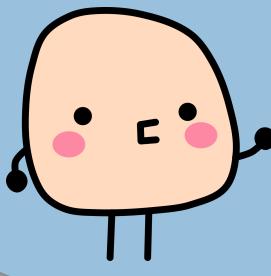
其中 w_i 是我們字庫裡的第 i 個字。





05.

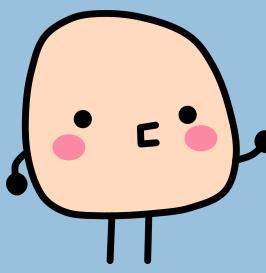
尋找特徵的小任務



輸入比較出複雜一點

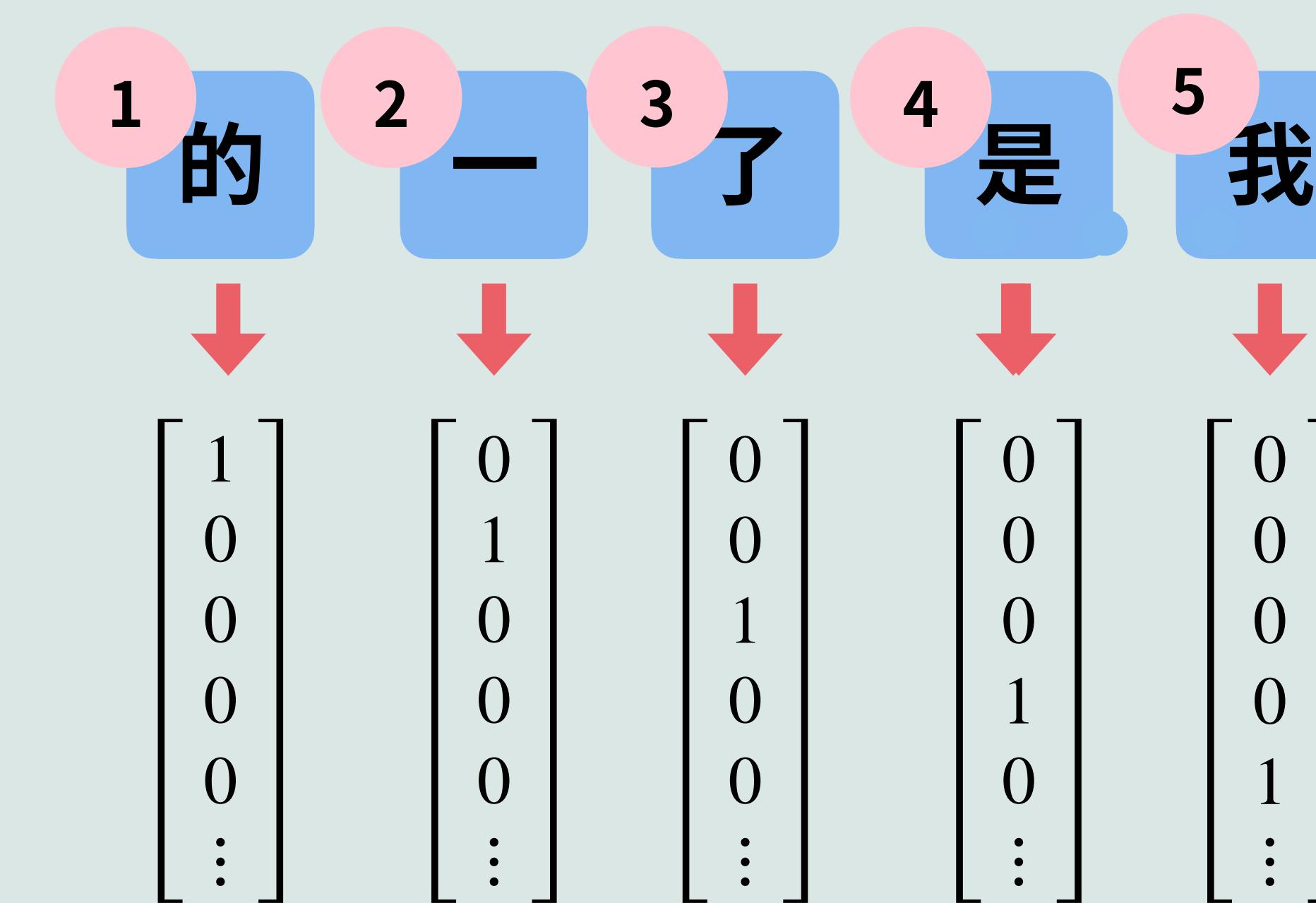
輸入基本上也可以
one-hot encodeing,
但...



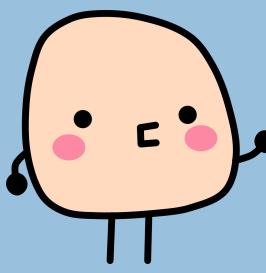


當然也可以用 One-Hot Encoding, 但...

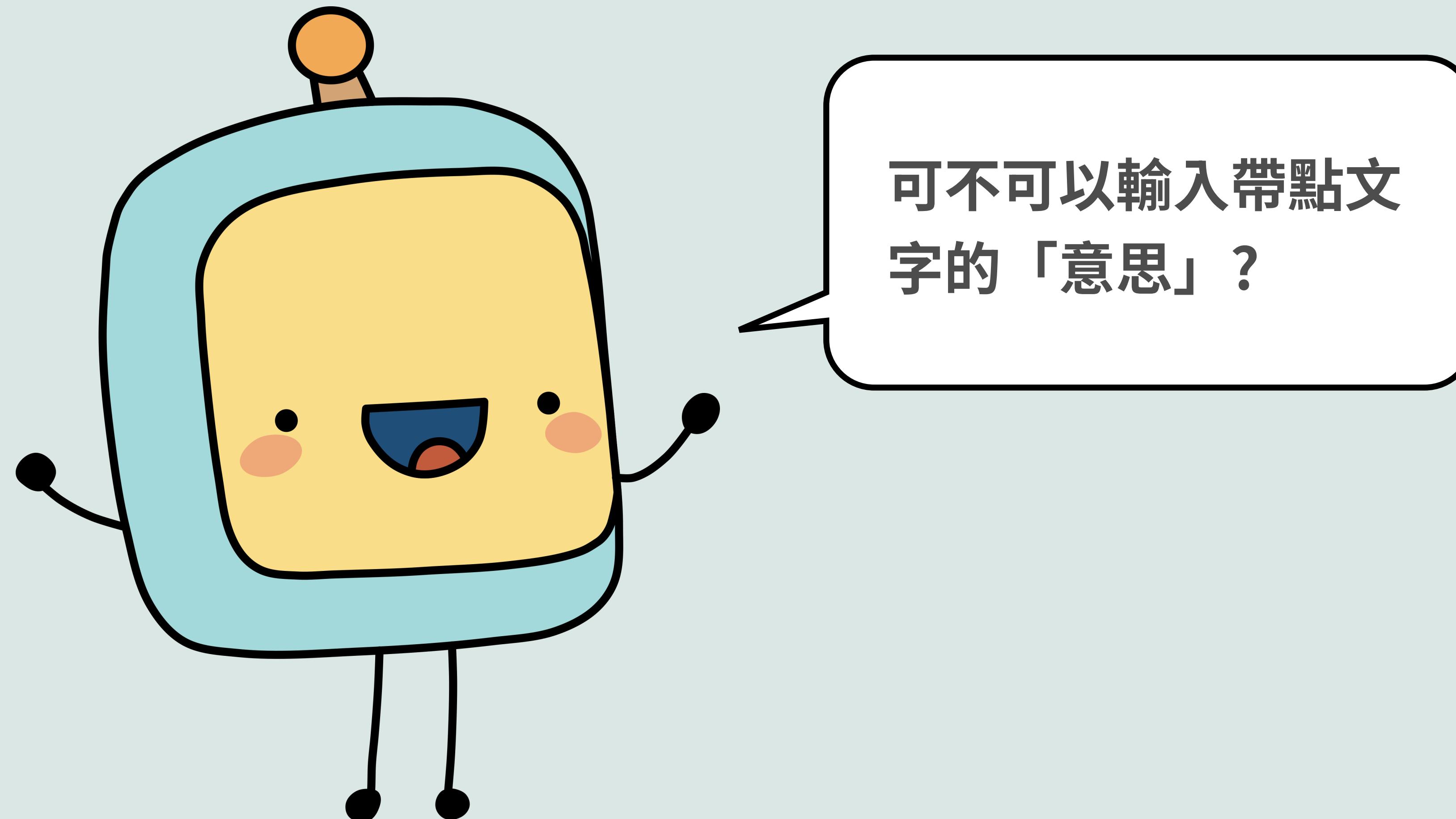
前面說每個字有個編號, 而代表向量就用 one-hot encoding。

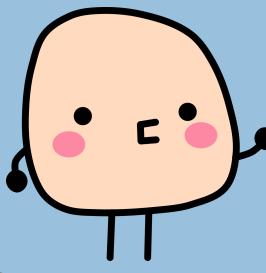


注意 one-hot encoding 後還是只是個編號!



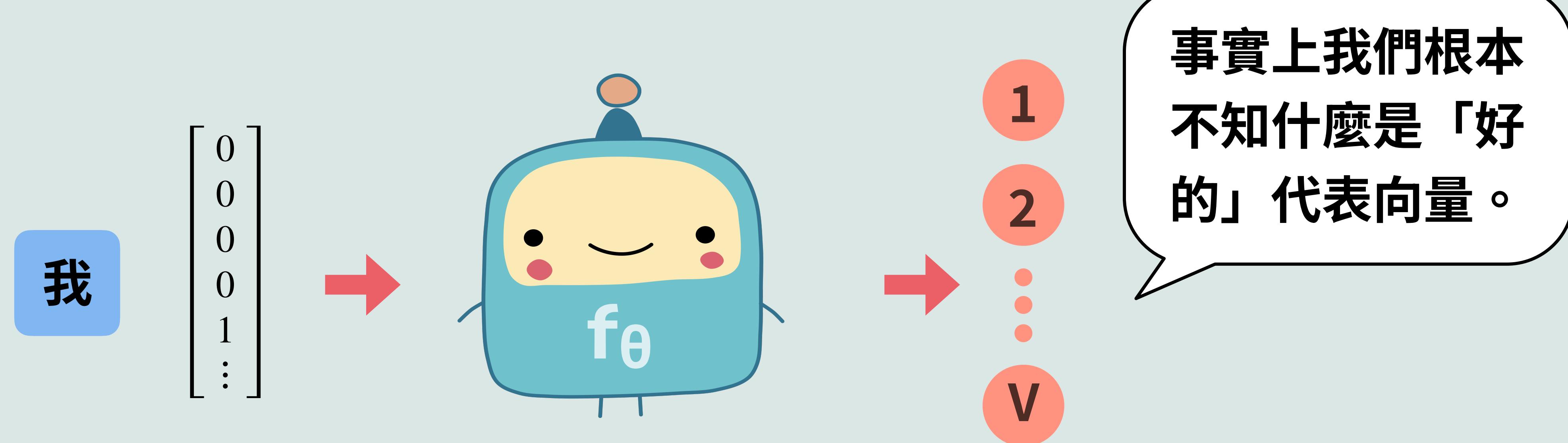
最主要希望有點「意思」



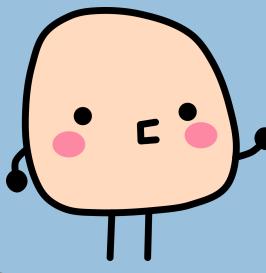


尋找特徵代表向量

一個字 (或一個詞), 我們想給一個比較有意義的「特徵代表向量」。

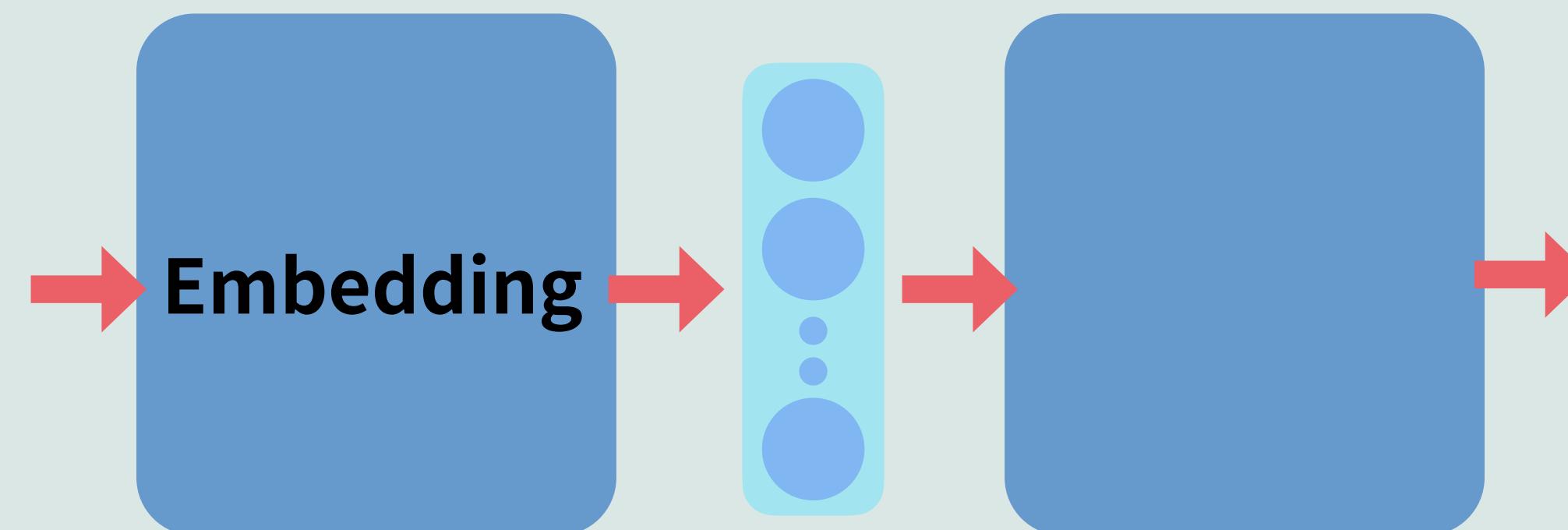


這樣的函數就叫做一個 **word embedding**。

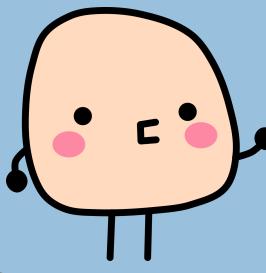


代理任務 (Pretext Task)

我們可以讓電腦去做一些小任務，這個任務是我們覺得「電腦要懂文字的意」才能完成的任務。這種不是我們真正最後的目標，通常是為了訓練好的表示向量的叫 **pretext task**。

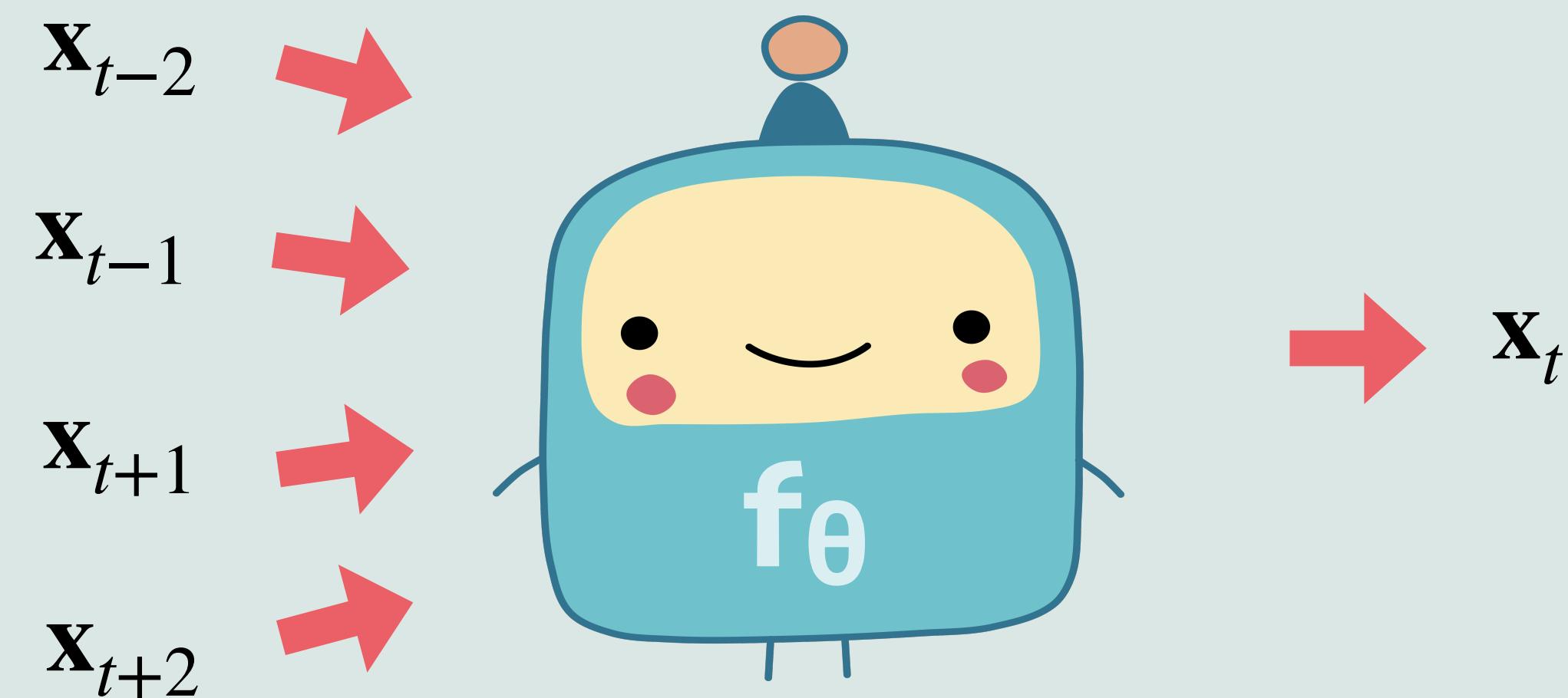


我們看要 embed 到幾維向量，比如說 $V=128$ 維，那就在神經網路中間的隱藏層，放 128 個神經元！



Word2Vec 用了兩種不同的小任務

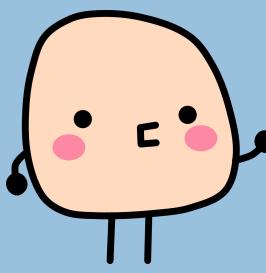
Word2Vec 就設計兩種任務。



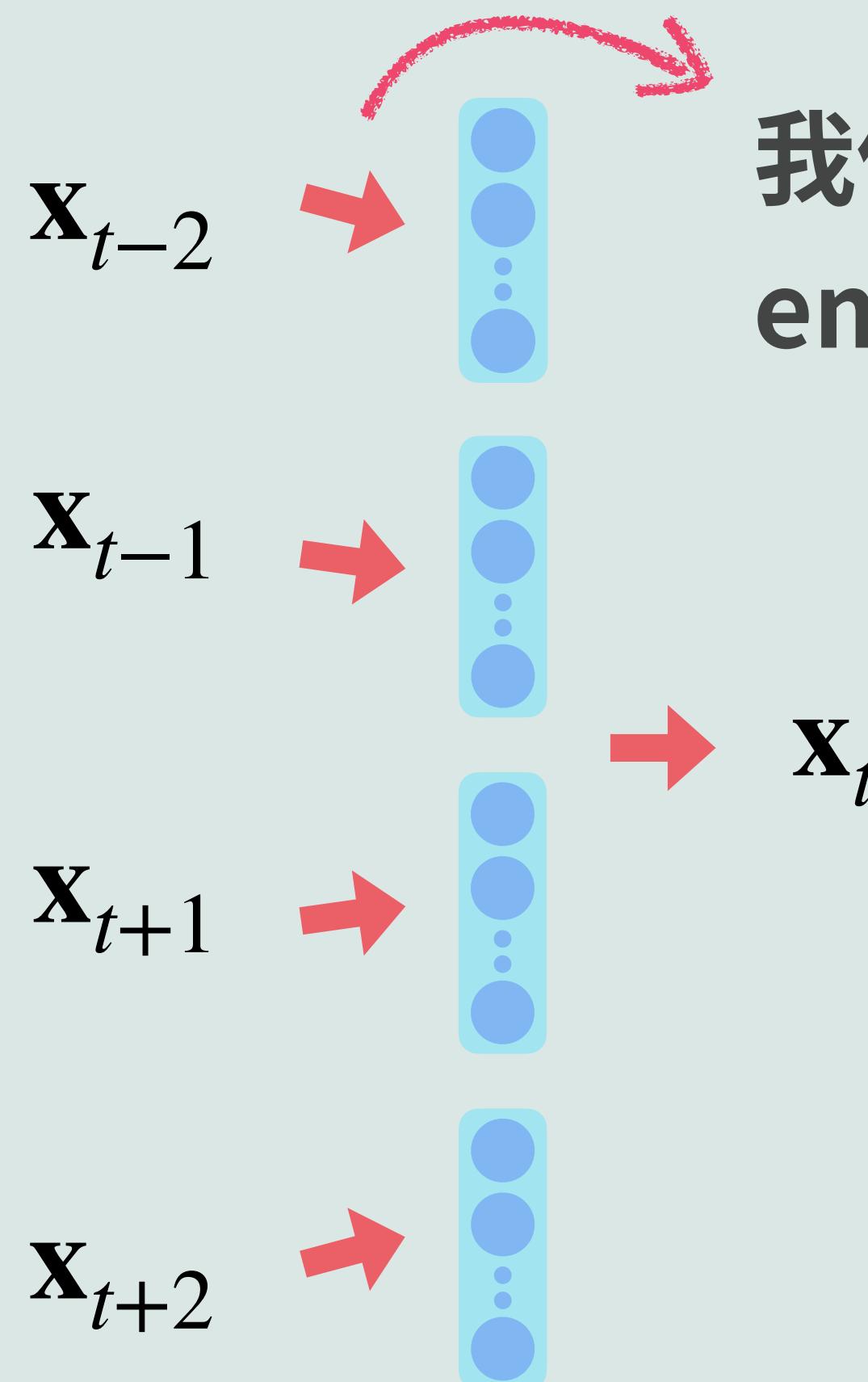
CBOW model

用周圍的字預測中間的字。

x_{t-2} x_{t-1} ? x_{t+1} x_{t+2}

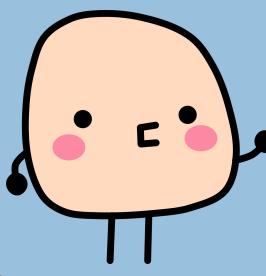


訓練成功, 某層隱藏層輸出當 embedding



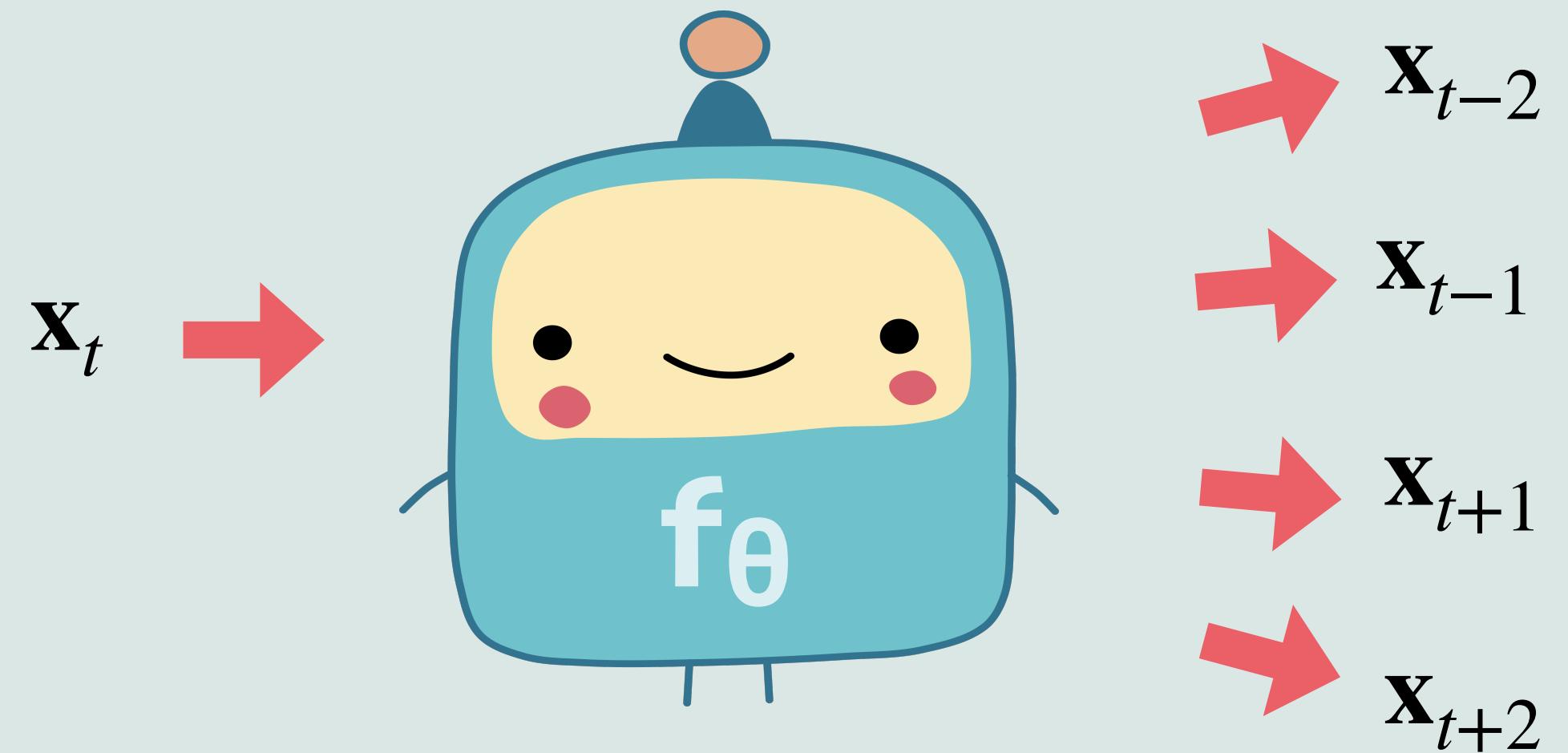
我們可以找到字的
embedding!

CBOW model



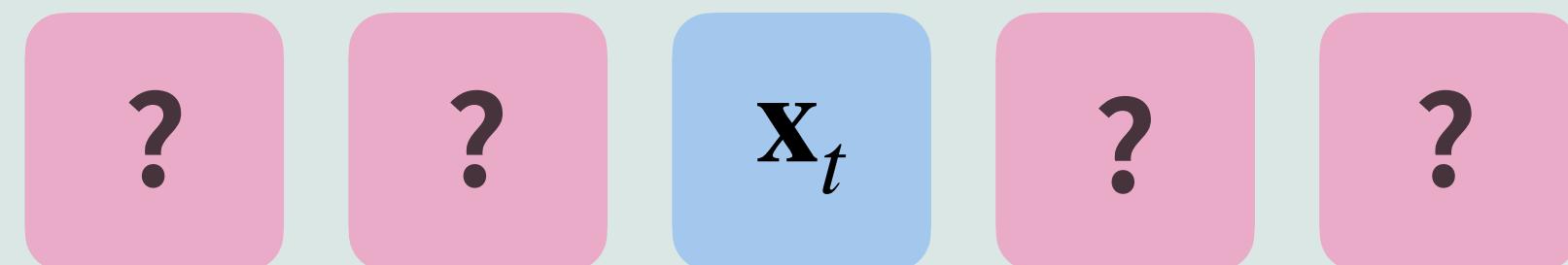
中間的字預測旁邊的會不會更厲害？

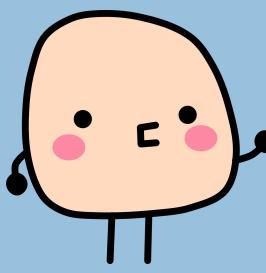
或是更炫的去訓練這樣的函數！



Skip-Gram model

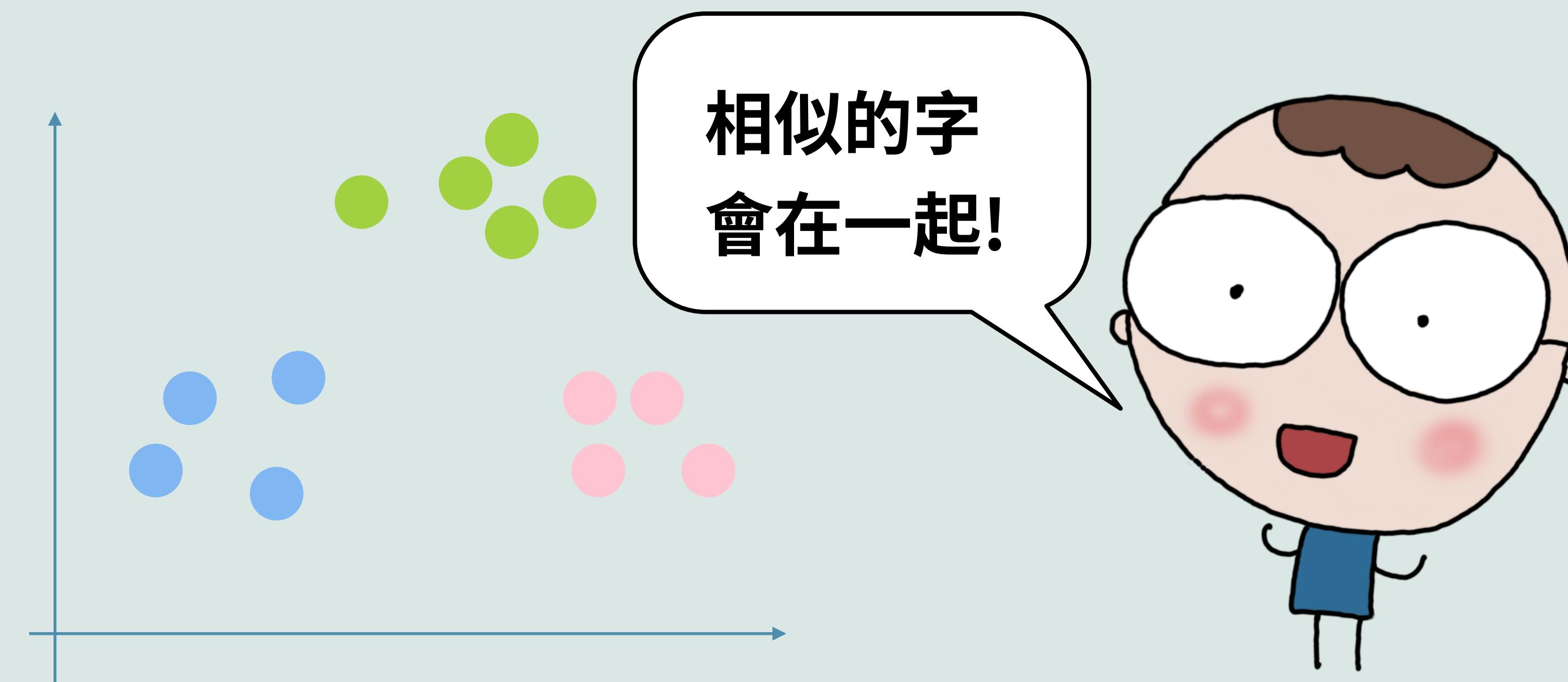
中間的字預測週圍的字



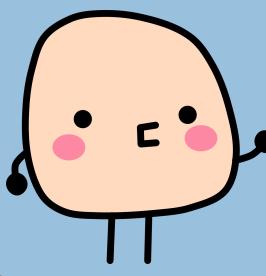


訓練完電腦好像「真的有懂」

結果會發現電腦好像「真的有懂」。



Google 官網: <https://code.google.com/archive/p/word2vec/>



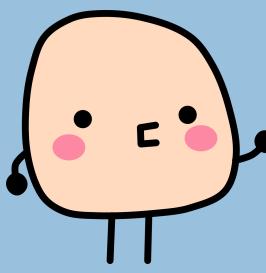
標準的火力展示

訓練好了有很多炫炫的功能。

$$\text{巴黎} + \text{法國} - \text{義大利} = \text{羅馬}$$

$$\text{國王} - \text{男人} + \text{女人} = \text{皇后}$$

T. Mikolov, K. Chen, G. Corrado, J. Dean. Toutanova. Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR, 2013..



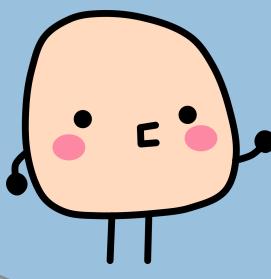
在 Word2Vec 中權重剛好是「記憶」

$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1V} \\ w_{21} & w_{22} & \cdots & w_{2V} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i1} & w_{i2} & \cdots & w_{iV} \\ \vdots & \vdots & & \vdots \\ w_{N1} & w_{V2} & \cdots & w_{NV} \end{bmatrix}$$

word2vec 紿我們一個啟發，就是權重也可以是我們日後要用的部份，可以當成是某種「記憶」。

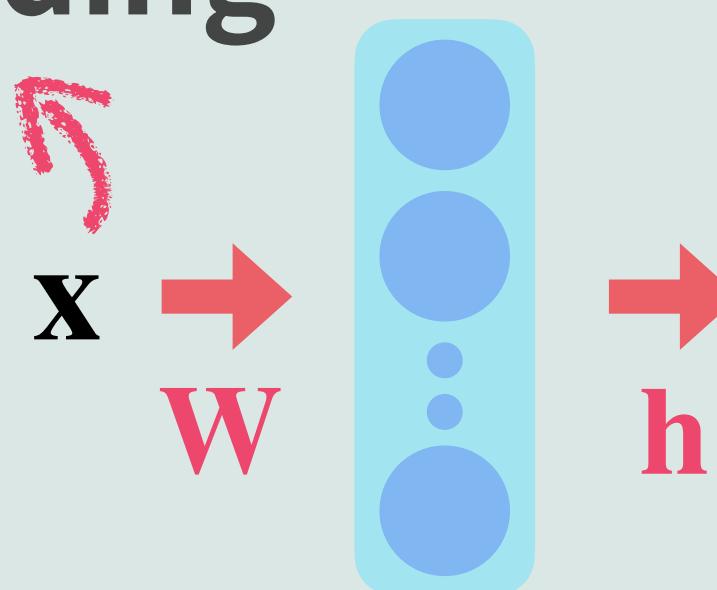
甚至這一小段網路想成是「理解」。





原因其實很簡單

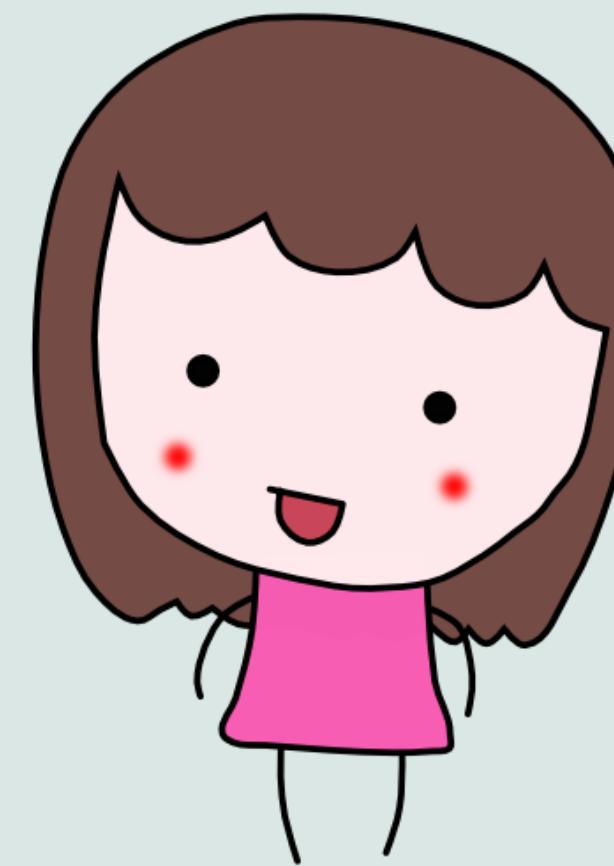
One-hot encoding

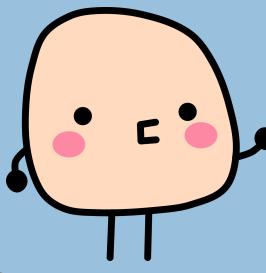


$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & & \vdots \\ w_{i1} & w_{i2} & \cdots & w_{iN} \\ \vdots & \vdots & & \vdots \\ w_{V1} & w_{V2} & \cdots & w_{VN} \end{bmatrix}^T \cdot \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} = h$$

$$W^T \cdot X = h$$

就 word2vec 來說，其實
也可以看成是隱藏層的輸
出！





注意 transformer 其實有自己的 embedding 層

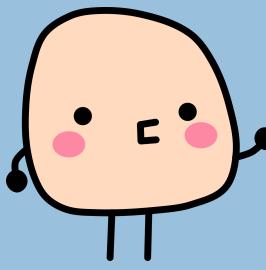
Trasnformers 用 one-hot encoding 輸入後，會連到自己的 embedding 層，也就是訓練時會同時訓練 word embedding。





05.

使用大型語言模型

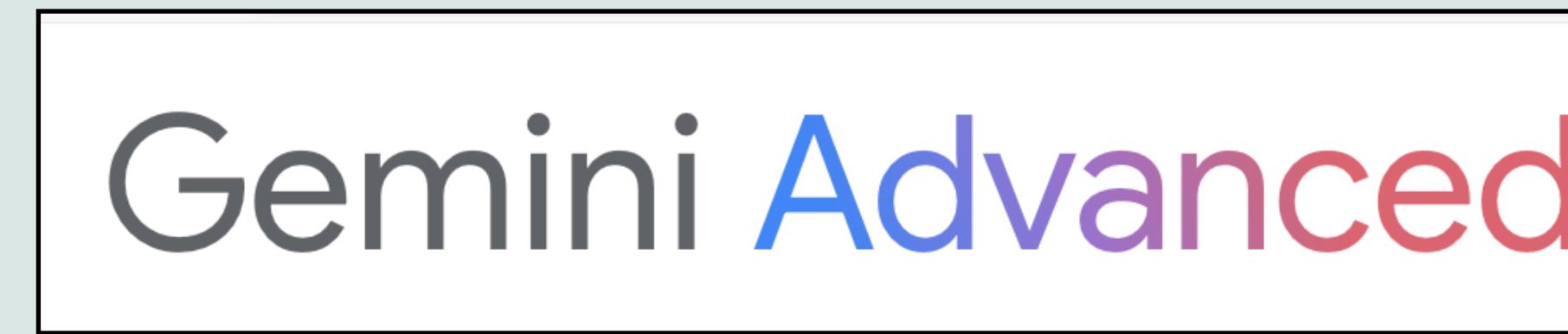


閉源型四大 LLM



ChatGPT

OpenAI



Gemini Advanced

Google



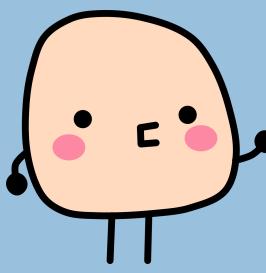
Claude

Anthropic



Grok

xAI



開源型的 LLM

Meta

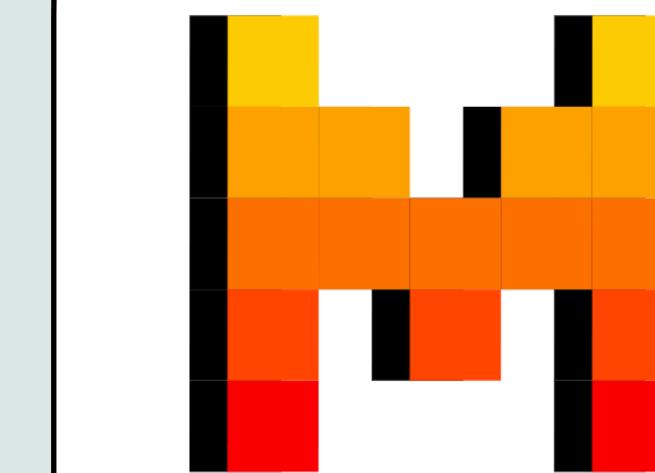
Llama

Google

Gemma



TAIDE/Breeze
加強繁體中文的 AI



MISTRAL
AI_

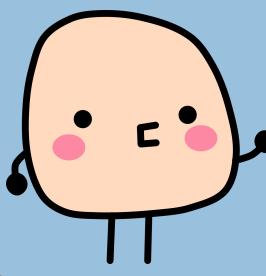
來自法國的 AI 新秀

deepseek

API 开放平台 ↗ English

DeepSeek-R1 已发布并开源，性能对标 OpenAI o1 正式版，在网页端、APP 和 API 全面上线，点击查看详情。

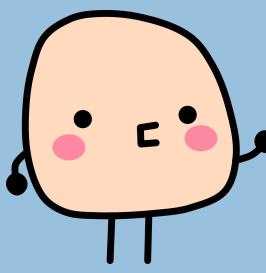
deepseek



還有 OpenAI 終於 Open 了

ChatGPT 出世以來, OpenAI
第一個開源型的語言模型。





怎麼用開源的 LLM 呢？



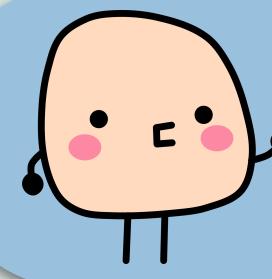
LM Studio

一般人的選擇



Ollama

簡潔、終端機介面



LM Studio

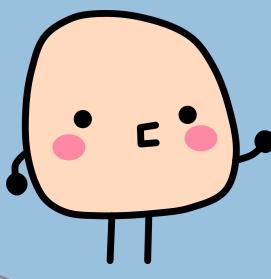


和很多閉源的大型語言模型一樣的界面。

The screenshot shows the LM Studio application window. At the top, there's a toolbar with icons for settings, ejecting, and appearance options like 'Appearance', 'Clear All', and 'Duplicate'. The main interface features a purple header bar with the text 'llama-3.3-70b-instruct'. Below the header, there's a section titled 'Unnamed Chat' with three AI-generated responses:

- "What is the capital of France?"
- "Can you teach me how to solve a Rubik's cube?"
- "Can you show me a cool math proof?"

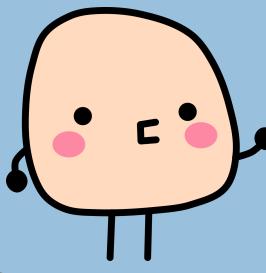
Below these responses is a checkbox labeled 'Show prompt suggestions when creating a new chat' which is checked. A large input field contains the question '生成式 AI 運用於教育有什麼應用呢？'. To the right of the input field are buttons for 'Insert (⌘I)' and 'Send ⌘S'. At the bottom of the input field, it says 'Context is 0.0% full'. The footer of the window displays the text 'LM Studio 0.3.12 (Build 1)', 'User Power User Developer', 'SYSTEM RESOURCES USAGE: RAM: 41.01 GB | CPU: 0.00 %', and a gear icon for settings.



可以考慮的模型

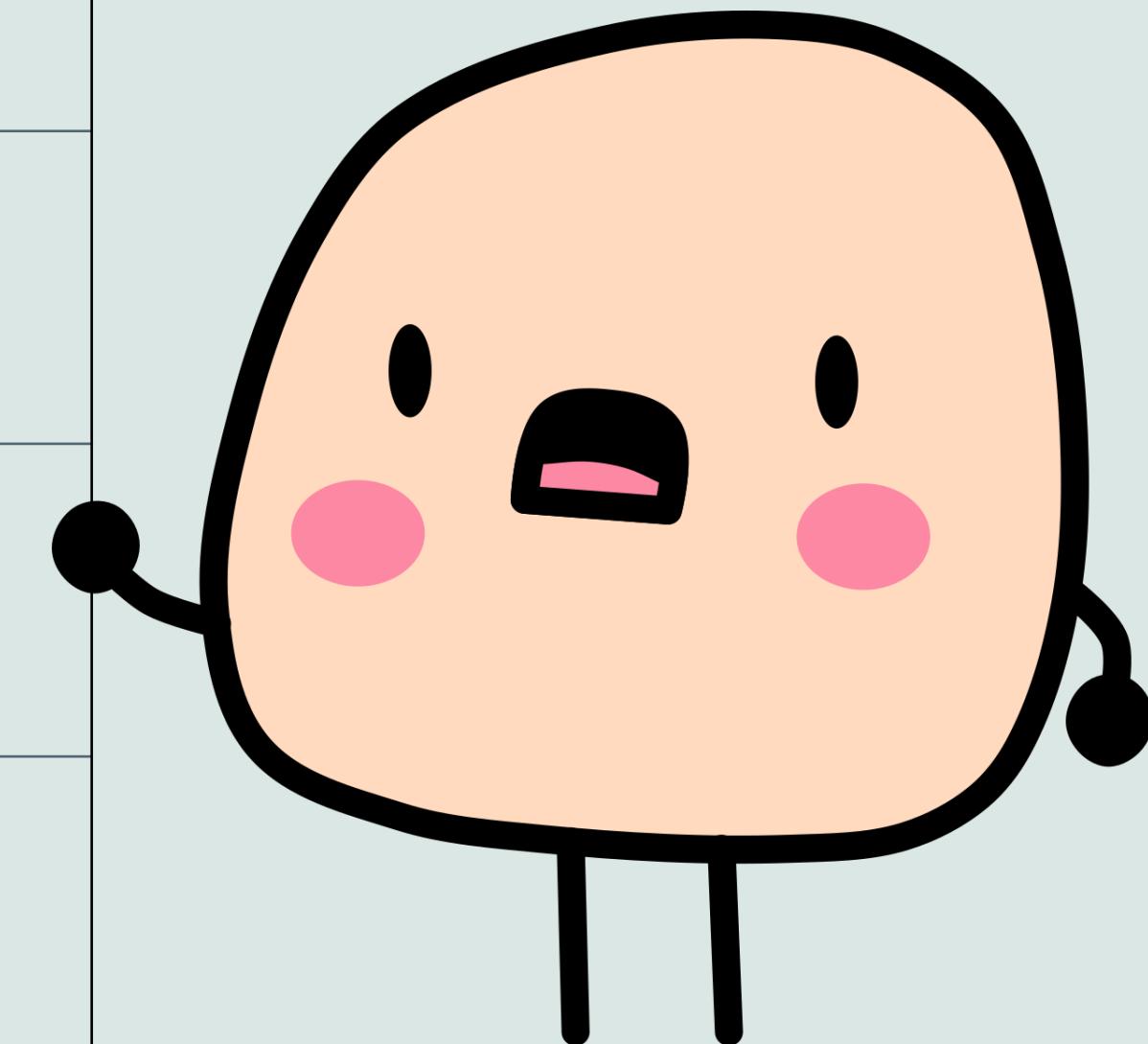


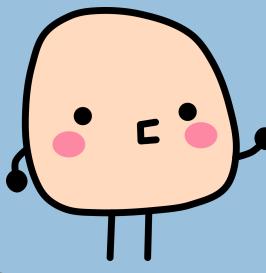
- * **Llama 3.2 3B、Llama 3.3 70B**
- * **TAIDE 7B、Breeze2 8B**
- * **Hermes 3 3B、DeepHermes 3 8B**
- * **Mistral Small 24B**
- * **Microsoft Phi-4 14B、Phi-4 Mini 3.8B**



跑模型的機器考量 (NVIDIA GPU)

GPU	VRAM
RTX 5090	32GB
RTX 4090	24GB
H100	80GB
H800	80GB

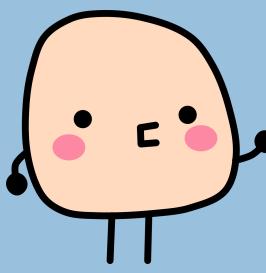




模型大小

gpt-oss 20B

200 億個參數



要多大的 VRAM 才可以跑呢？

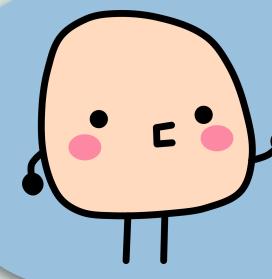


以 4-bit 量化
的版本為例。

$$0.5 \times 20 = 10G$$

模型大小

要超過
10G
實際建議 16G
可跑

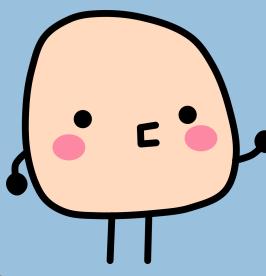


Mac 有多大的記憶體，基本上有多大的 VRAM!

機型	最高 RAM
Mac Mini	64GB
MacBook Pro	128GB
Mac Studio	512GB
Mac Pro	192GB

Mac 運用 unified memory 技術。



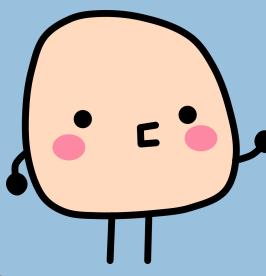


共用記憶體 AMD 篇

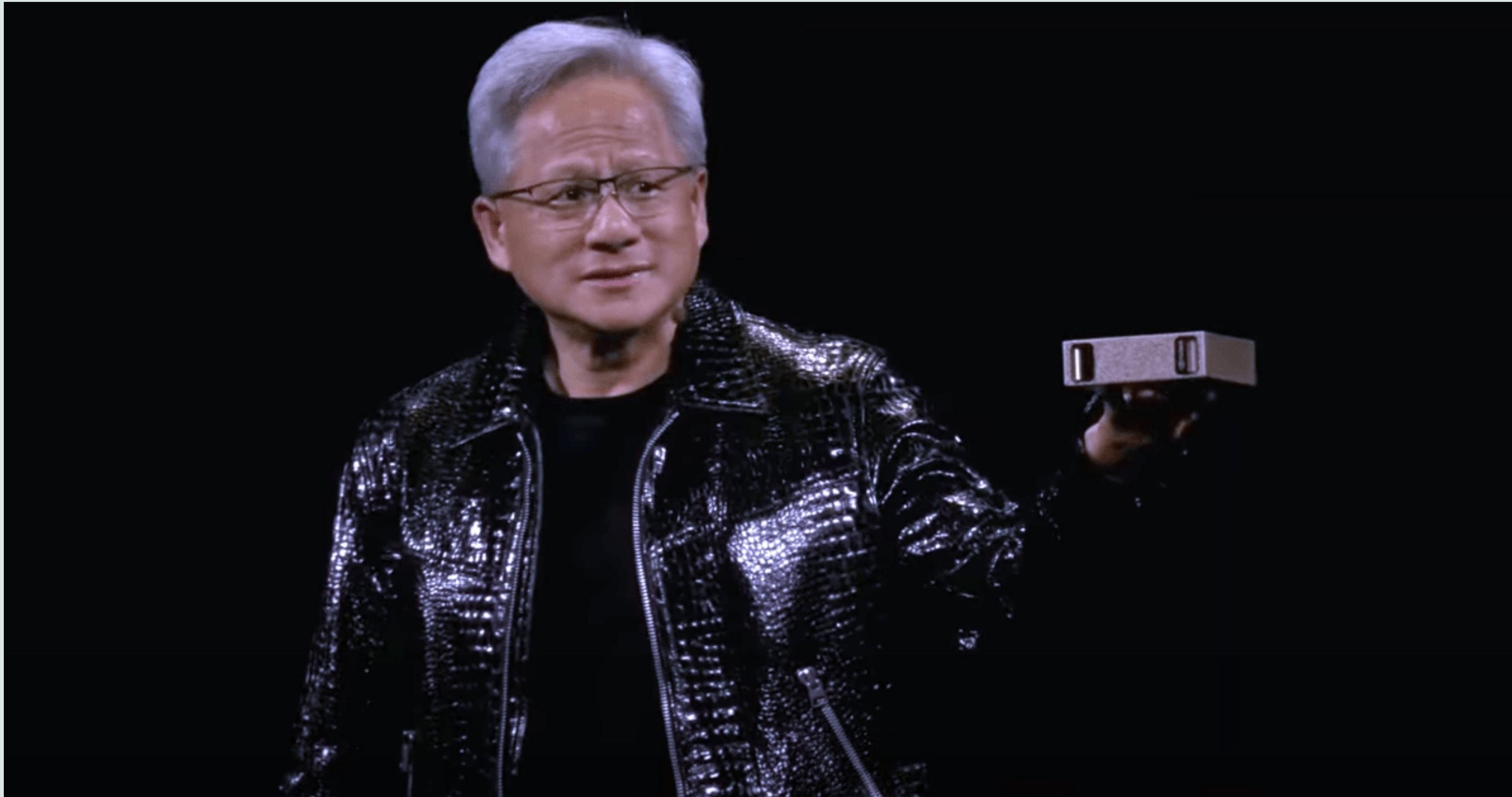


AMD Ryzen AI Max+ 395
也是共用記憶體!

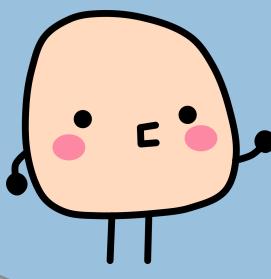
ASUS Flow Z13 (可達 128GB RAM)



共用記憶體 NVIDIA 篇



NVIDIA DGX Spark (Project Digits)



用開源也不一定要裝在自己電腦

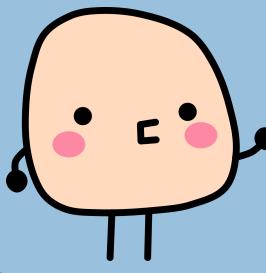
The screenshot shows the Groq website homepage. At the top, there is a navigation bar with a back arrow, a title "Mistral Saba 24B: Specialized language model for the Middle East & South Asia. [Learn more.](#)", and a forward arrow. Below the navigation bar is the Groq logo and a menu with links to "Products", "Developers", "Insights", "About", "DEV CONSOLE" (which is highlighted in red), a search icon, and a menu icon. The main headline "Groq is Fast AI Inference" is displayed in large red text. Below the headline is a search bar with a dropdown menu set to "Llama3-8b-8192" and a "USE FULL CHAT" button. At the bottom of the page, there are three links: "Why fast inference matters", "What is AI inference", and "What is generative AI".

<https://groq.com/>



【作業】

建立自己的 benchmark



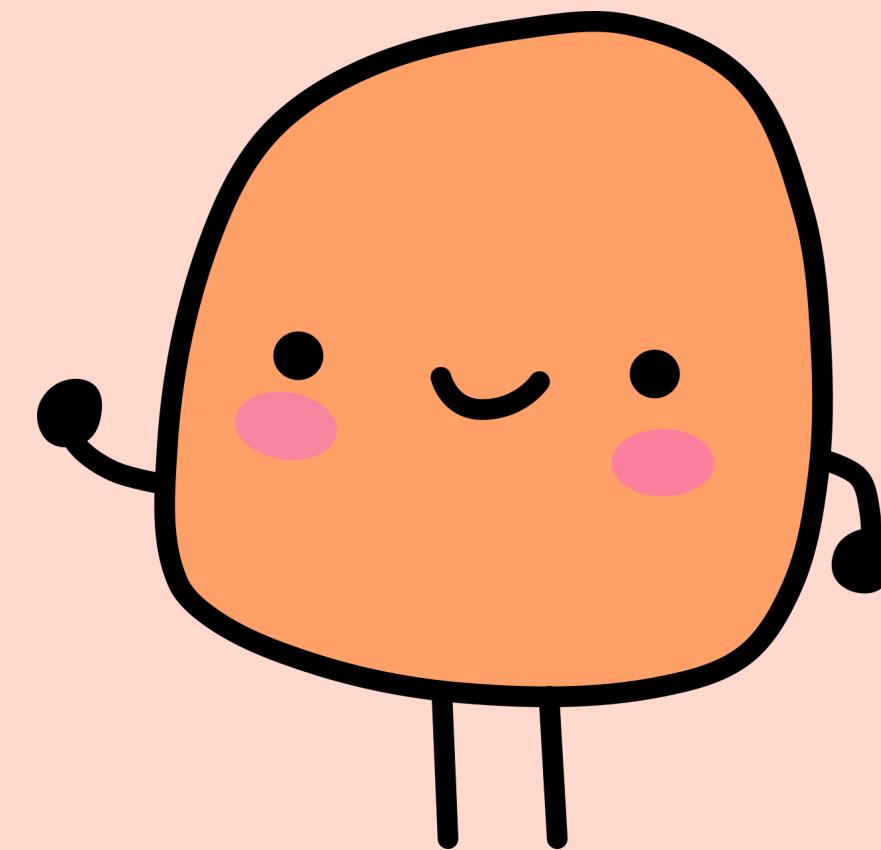
Prompt 其實很簡單

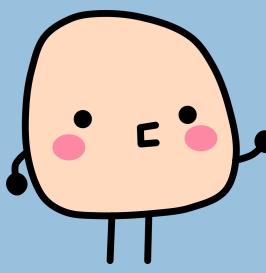
資訊

提供需要的正確資訊。

清楚的指引

例如，以上面的資訊，用什麼樣的格式、風格，來回答使用者的問題。



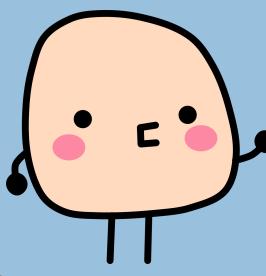


不要問封閉式 (固定答案) 的問題

K-POP 女團 IVE 的成員有誰呢？

(gpt-oss 120B)

成員	韓文名字	中文常用譯名	生日	角色
Yujin (이서연)	李秀妍	Yujin	2003-09-01	Leader、主唱、領舞
Wonyoung (임원영)	林元英	Wonyoung	2004-08-31	主舞、主唱、視覺
Rei (레아)	雷亞	Rei	2005-02-02	主唱、Rapper (偶爾)
Gaeul (가을)	佳乙	Gaeul	2005-09-24	主舞、領唱
Leeseo (이서)	李瑞秀	Leeseo	2007-09-13	主唱、次要領舞



不要問封閉式 (固定答案) 的問題

我想用 LLM 做一個有趣、實用但需要花些時間
調整、完成的專案，可以提供幾個想法嗎？

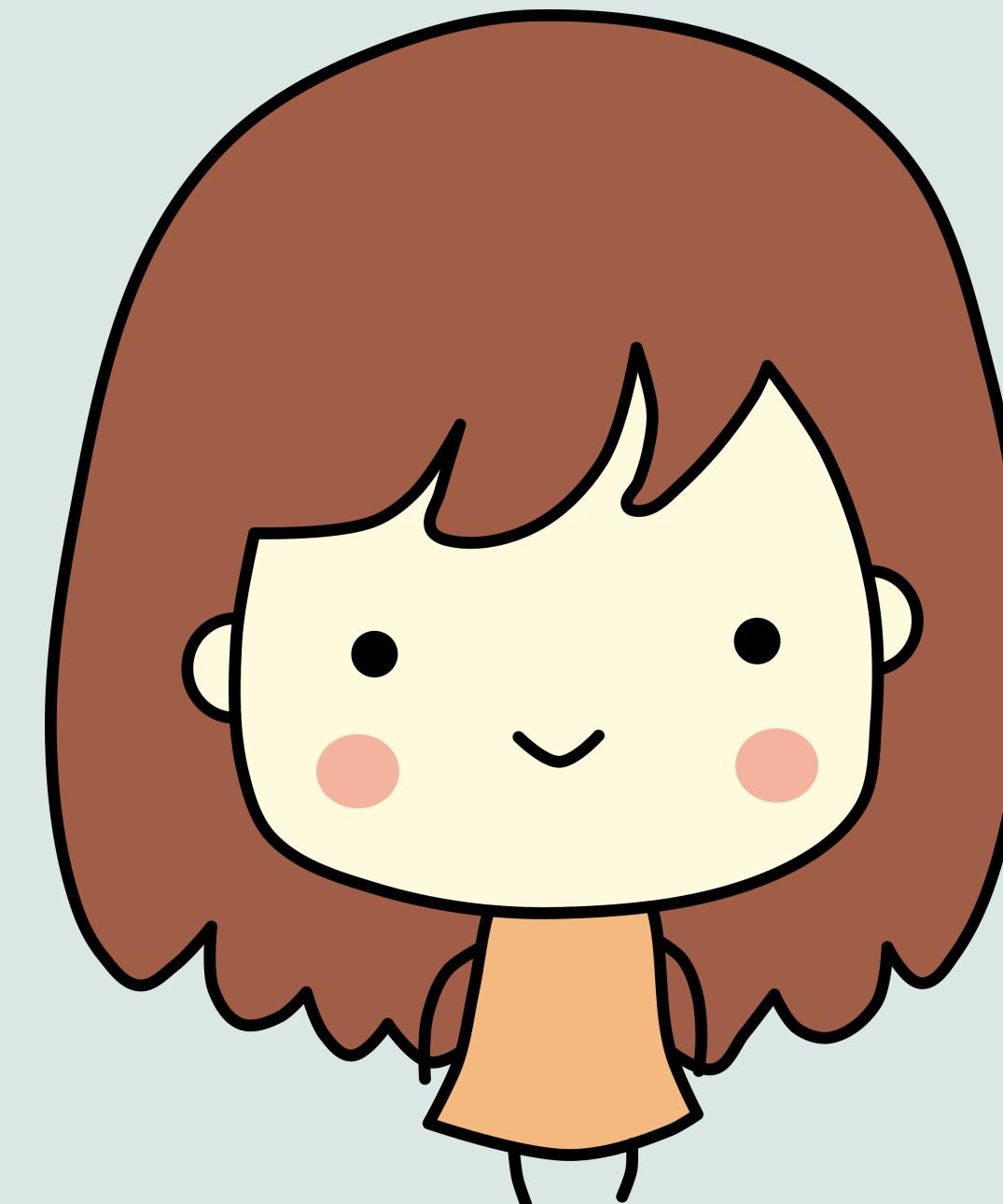
(gpt-oss 120B)

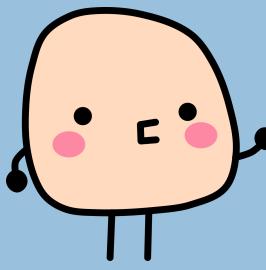
以下提供 5 個結合 LLM、具挑戰性且實用的專案構想，每個都需要一定的資料蒐集、模型微調、系統整合與測試，適合想花時間慢慢打磨的你。

(以下說明皆以 繁體中文 (台灣慣用) 撰寫，若有需要可自行調整技術細節)

1 多語言生活助理 Chatbot (結合行事曆、天氣、交通、食譜等功能)

項目	說明
核心目標	讓使用者只要用自然語言 (中文、英文、日文) 就能查詢/設定行事曆、查天氣、規劃路線、找食譜、提醒事項等。
LLM 任務	1. 意圖辨識與槽位填充 (Intent & Slot) 2. 多輪對話管理 (Context Tracking) 3. 生成自然回覆 (Response Generation)
需要的資料	- 公開的天氣 API、Google Maps / 台灣交通局即時交通資料 - 行事曆 (Google Calendar) OAuth 授權 - 食譜資料庫 (如 openfoodfacts、台灣食譜網站爬蟲)
微調方向	1. 收集日常生活對話 (可自行蒐集或使用公開的 MultiWOZ、DSTC 資料) 2. 針對「行事曆/天氣/交通」等領域的指令做 few-shot / LoRA 微調，提升指令執行正確率。
挑戰點	- 多語言切換與同義詞處理 - 多輪上下文保持 (尤其跨 API 呼叫) - 隱私與授權 (行事曆、位置資訊)
可使用工具	LangChain (對話管理) + Llama-2/Claude-3 (或本地微調的 Mistral) + FastAPI + Docker + Azure/AWS OpenAI 服務 (若不想自行部署)

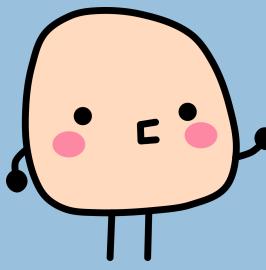




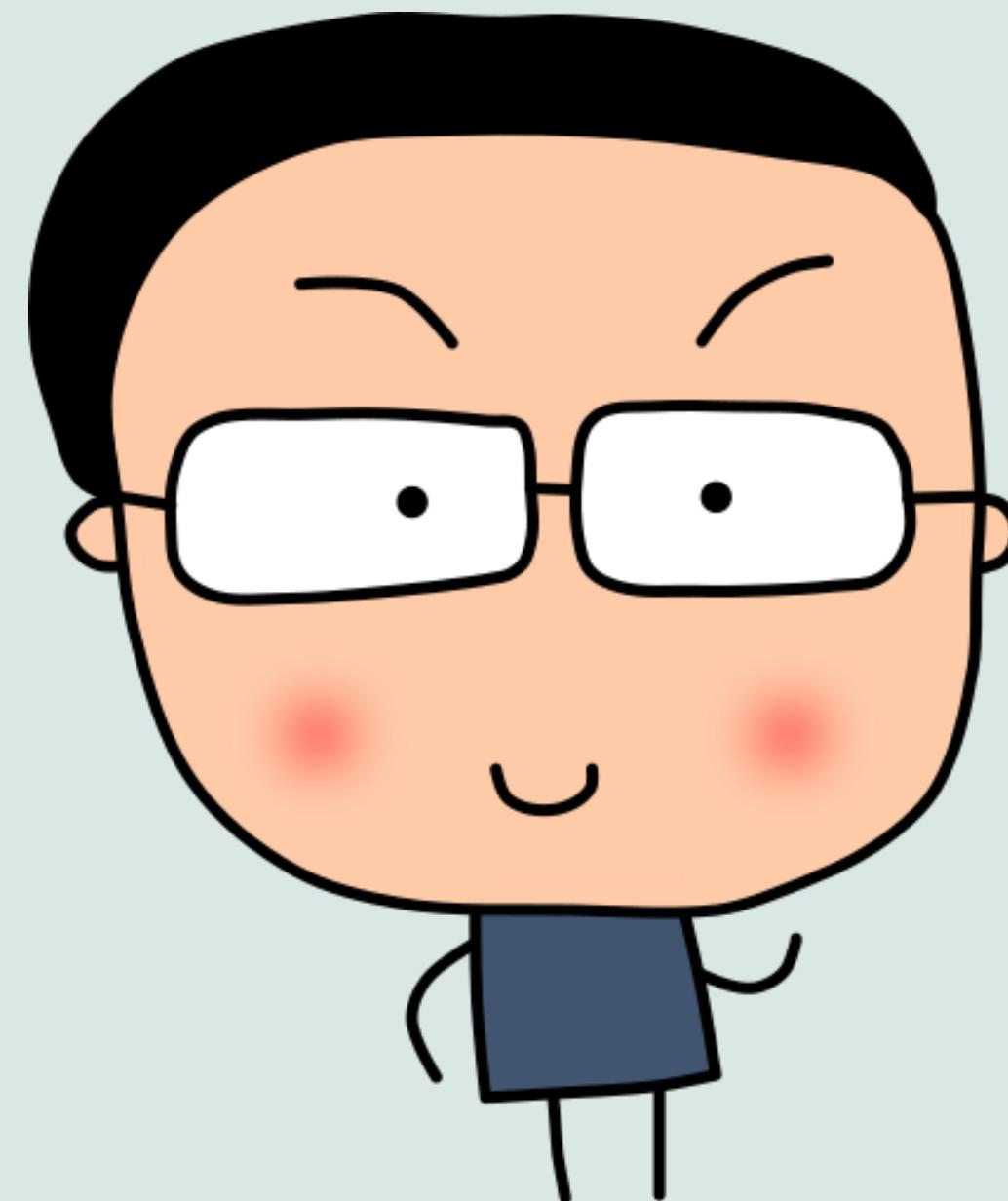
Benchmark 分數高的模型一定比較好？

LLM 一般的 benchmarks 當然有其意義，但相信我們也不一定那麼在意，自己慣用的 LLM 是不是比另外一個會解數學題。

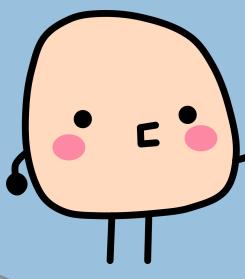




設計幾個標準測試 prompts



這些問題最好你**相當懂**
— 至少不能完全不懂，
你才可以知道他回答得
如何。



例子

1

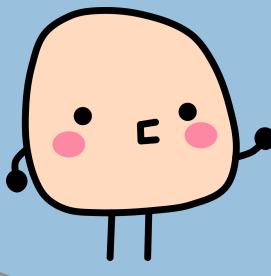
佛法的唯識學感覺名相很多很難學。用很淺顯語言說的話，唯識學到底是什麼，該怎麼學習，和生命有什麼連結呢？

2

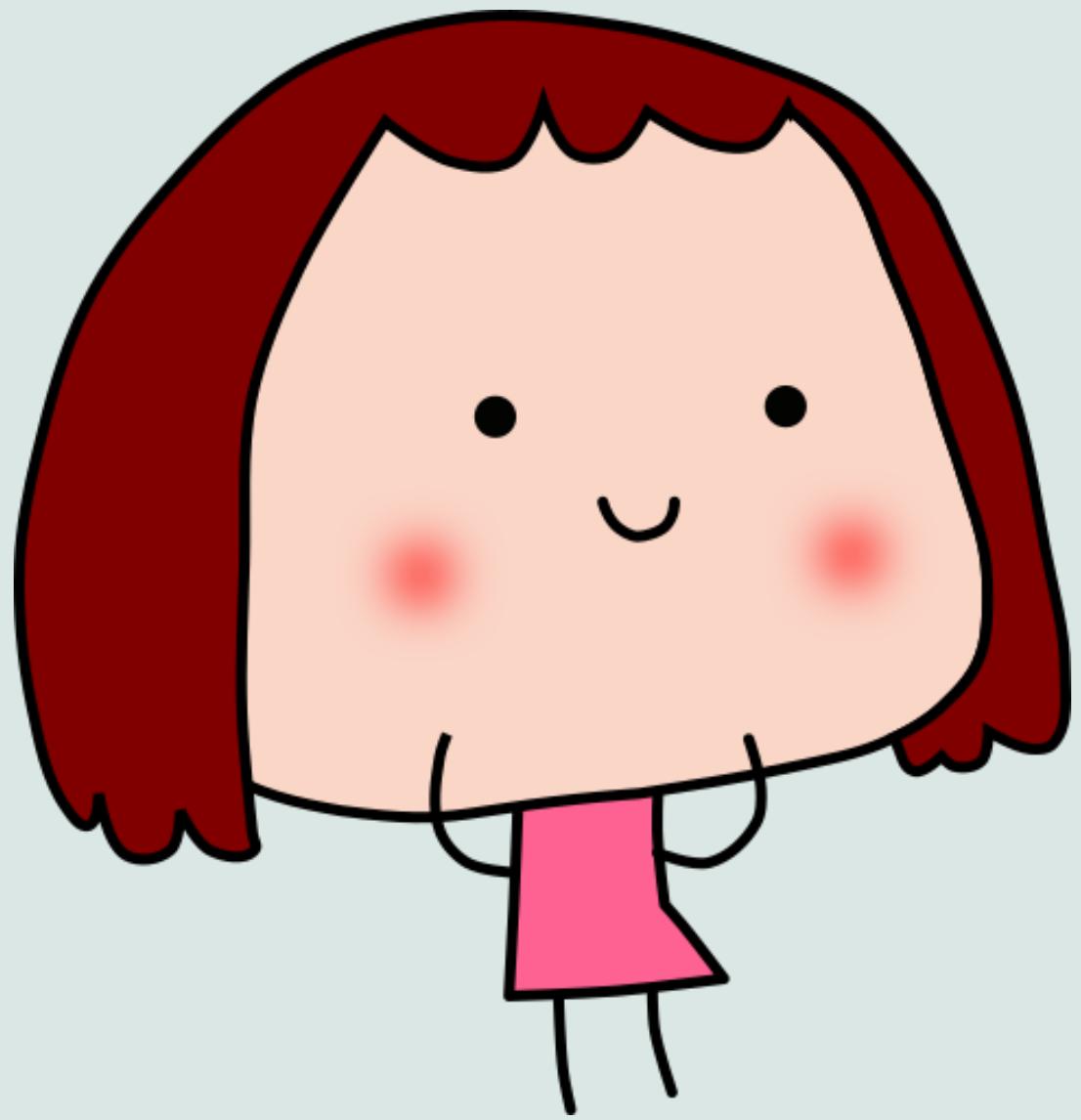
為什麼說「一切唯心造」呢？如果都是心產生的，人不是都只要快樂，怎麼會生出悲傷、痛苦的事呢？車子、房子等等一切硬體的東西，又怎麼能由心生出來呢？

ChatGPT: https://yenlung.me/MindOnly_GPT

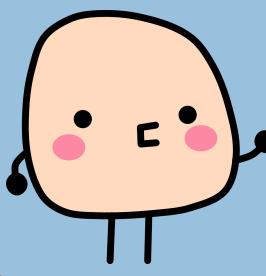
Grok: https://yenlung.me/MindOnly_Grok



【作業】建立自己的 Benchmark



- * 建立一組你自己的基準測試 (prompts)
- * 主題是你有興趣、有點懂的 (才能分辨好壞)
- * 不要考你的 LLM (比如 「IVE 的成員是誰?」)
- * 至少用兩種以上的模型測試
- * 寫下你對這些模型回答的看法, 你比較喜歡哪一個、為什麼



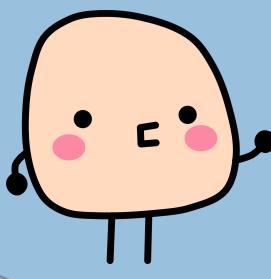
使用者實際測試的 LMArena

Text

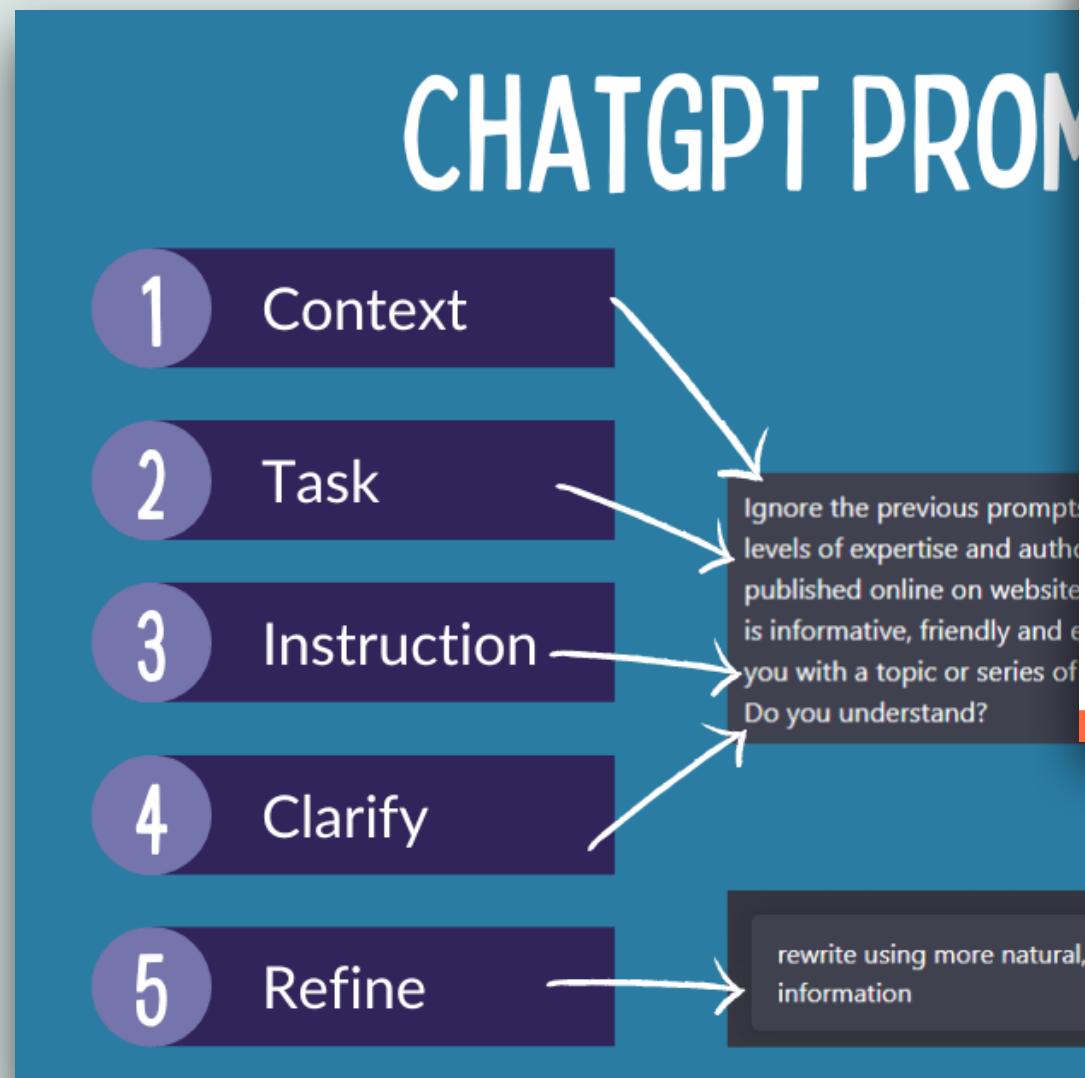
4 days ago

		Rank (UB) ↑ Model ↑	Score ↓	Votes ↓
1	gemini-2.5-pro	1456	46,291	
1	claude-opus-4-1-20250805-thi...	1449	14,843	
2	o3-2025-04-16	1441	46,083	
2	chatgpt-4o-latest-20250326	1441	35,807	
2	gpt-4.5-preview-2025-02-27	1441	14,644	
2	gpt-5-high	1440	17,636	
2	claude-opus-4-1-20250805	1438	23,612	
3	qwen3-max-preview	1430	11,851	
5	gpt-5-chat	1430	14,876	





瞭解原理，會成為一個強大的吸收器！



The ChatGPT Cheat Sheet

Act as a [ROLE]

- Analyst
- Teacher
- Marketer
- Advertiser
- Mindset Coach
- Therapist
- Journalist
- Inventor
- Lawyer
- Ghostwriter
- Website Designer
- Best Selling Author
- Chief Financial Officer
- Expert Copywriter
- Prompt Engineer
- Accountant
- Project Manager
- Sports Coach
- Financial Analyst
- Full Stack Developer
- Linux Terminal
- "Position" Interviewer

Create a [TASK]

- Headline
- Article
- Essay
- Book Outline
- Email Sequence
- Social Media Post
- Product Description
- Cover Letter
- Blog Post
- SEO Keywords
- Summary
- XML
- Markdown
- Recipe
- Sales Copy
- Analysis
- Ad Copy
- Web Page

Show as [FORMAT]

- A Table
- A List
- Summary
- HTML
- Code
- Spreadsheet
- Graphs
- CSV File
- Plain Text file
- JSON
- Rich Text
- PDF
- XML
- Gantt Chart
- Word Cloud
- Infographic
- An Analogy
- Bullet Points

Set restrictions

- Use poetic language
- Adopt a formal tone
- Write short sentences
- Code only in HTML/CSS
- Use Shakespearean style
- Write using basic English
- Use only scientific sources
- Add pop culture references

Prompts For Biz Owners

1. Give me a list of inexpensive ideas on how to promote

ChatGPT Cheat Sheet (Drafting the Perfect Prompt) – Part 1

Master the skill of crafting the perfect prompts to get the most out of LLM tools without upgrading to Pro and saving money.

Prompts for Developers

1. Develop an architecture and code for a <description> website with JavaScript. 2. Help me find mistakes in the following code <paste code below>. 3. I want to implement a sticky header on my website. Can you provide an example using CSS and JavaScript? 4. Please continue writing this code for JavaScript <paste code below>

Prompts for Marketers

1. Can you provide me with some ideas for blog posts about [topic]? 2. Write a product description for my [product or service or company]. 3. Suggest inexpensive ways I can promote my [company] without using social media. 4. How can I obtain high-quality backlinks to improve the SEO of [website name]?

Prompt Structure:

Take on the persona of [optional: *expert persona*], [verb(s)] [optional: *output and length*] [task and objective] The output should include relevant [optional: *data and resources*]. The writing style is [optional: *tone of voice*] targeted towards [optional: *audience*]

Example

Take on the persona of Sr Technical Writer. Please Help me draft 10 lines on a table format for my blog on medium, where I write about AI tools and the latest technologies to improve productivity and save time for a reader from any profession or background. The output should include relevant tool names, pros and cons and pricing. The writing style is *engaging and persuasive*, targeted towards a *working professional, freelancer or business owner and entrepreneur*.

Potential Expert Personas

- Entrepreneur
- Solopreneur
- Digital Content Creator
- Content Creator
- Social media specialist
- Software Engineer
- DevOps Engineer
- Senior Technical Writer
- Accountant
- Financial Consultant
- Stock broker
- Business Coach
- Marketing Consultant
- Professor

C.R.E.A.T.E Formula

1. C: Character - Define the AI's role. E.g., "You are a seasoned 2. R: Request - Be specific with what you need. Instead of "W for the Cougar Hyper Sport, an electric car with top-tier acc 3. E: Examples - Optionally, provide samples for more precise 4. A: Adjustments - Refine the prompt if it's not perfect. Use i 5. T: Type of Output - Describe the desired format. E.g., "Prod 6. E: Extras - Incorporate unique instructions.

Ex: Ignore previous conversations." Ex: "Ask questions before answering." Ex: "Explain your thinking." Ex: "Summarize what you know about me in the first person." Ex: "Use date in 2023." Ex: "Only use reliable sources and cite them." Ex: "Use CAPS LOCK for emphasis."

Using the C.R.E.A.T.E formula ensures better results, although it

10 Prompt Formats and Output options

- Table
- Paragraphs
- Essays
- Reports
- Root Cause analysis
- Bullet points
- Step-by-step instructions
- Code
- Timelines/Schedules
- Summarise
- Eisenhower matrix

Role playing

- Act like Steve Jobs
- Act like Elon Musk
- Act like GaryVee
- Act like Ambani
- Act like a Writer
- Act like an SEO specialist
- Act like a Lawyer
- Act like a Consultant
- Act like a Codier
- Act Like a Doctor
- Act like a Human
- Act Like a Historian
- Act like an artist
- Act Like a Nutritionist
- Act like a Scientist
- Act like a Life Coach
- Act like a Pro Marketer
- Act like a SalesPerson

Tone of Voice

- Persuasive
- Formal/Informal
- Confrontational
- Opinion
- Explanation
- Assertive
- Confident
- Engaging
- Witty
- Direct

Resources

- Top 7 Websites That Pay You to Write Blogs in 2023 – Part 2
- Top 7 Websites that Pay to Write – Part 1
- Top 5 Free AI Courses for Beginners in 2023
- One stop ChatGPT CheatSheet
- Google Bard CheatSheet for Software Engineers
- Best AI tools to generate anything in 2023
- Github: Best AI Tools
- Top 8 AI Tools for Writing
- Best ChatGPT Extensions
- Top 10 AI Design Tools in 2023
- 8 Advanced ChatGPT Prompts
- Top 8 AI tools for Engineers and coders
- Top 8 AI SEO Tools
- Top 8 AI Tools that save 24+ hours every week

Data Insights

Data InkSights Blog

ChatGPT Prompt Frameworks

Unlock the full potential of ChatGPT

R-T-F Prompt Example

Facebook Ad Marketer.
Design a compelling Facebook ad campaign to promote a new line of fitness apparel for a sports brand.

T-A-G Prompt Example

The task is to evaluate the performance of team members
Act as a Direct manager and assess the strengths and weaknesses of team members.

B-A-B Prompt Example

We're nowhere to be seen on SEO rankings
We want to be in top 10 SEO ranking in our niche in 90 days.
Develop a detailed plan for mentioning all the measures we should take also include list of top 20 keywords.

C-A-R-E Prompt Example

We are launching a new line of sustainable clothing
Can you assist us in creating a targeted advertising campaign that emphasizes our environmental commitment?

R-I-S-E Prompt Example

Imagine you are a content strategist.
I've gathered detailed information about our target audience, including their interests & common questions related to our industry.

Created by: Khizer Abbas, Growth Marketing specialist

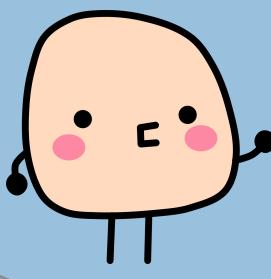
Act as a [ROLE]	Create a [TASK]	Show as [FORMAT]
→ ROLE	→ TASK	→ FORMAT

Define [TASK]	State the [ACTION]	Clarify the [GOAL]
→ TASK	→ ACTION	→ GOAL

Explain Problem [BEFORE]	State Outcome [AFTER]	Ask for the [BRIDGE]
→ BEFORE	→ AFTER	→ BRIDGE

Give the [CONTEXT]	Describe [ACTION]	Clarify the [RESULT]	Give the [EXAMPLE]
→ CONTEXT	→ ACTION	→ RESULT	→ EXAMPLE

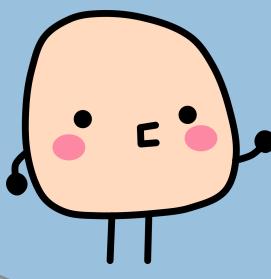
Specify the [ROLE]	Describe [INPUT]	Ask for [STEPS]	Describe the [EXPECTATION]
→ ROLE	→ INPUT	→ STEPS	→ EXPECTATION



自己越強, AI 就越強!



- AI 可以提供客服人員 20% 的生產力
- 對需要產生文案、翻譯、報告、整理簡報及會議記錄的人，增加 70% 的生產力
- 程式設計師更提升 125% 的生產力
- 甚至芥川賞作者九段理江都說她小說 5% 是 ChatGPT 寫的



Q & A

