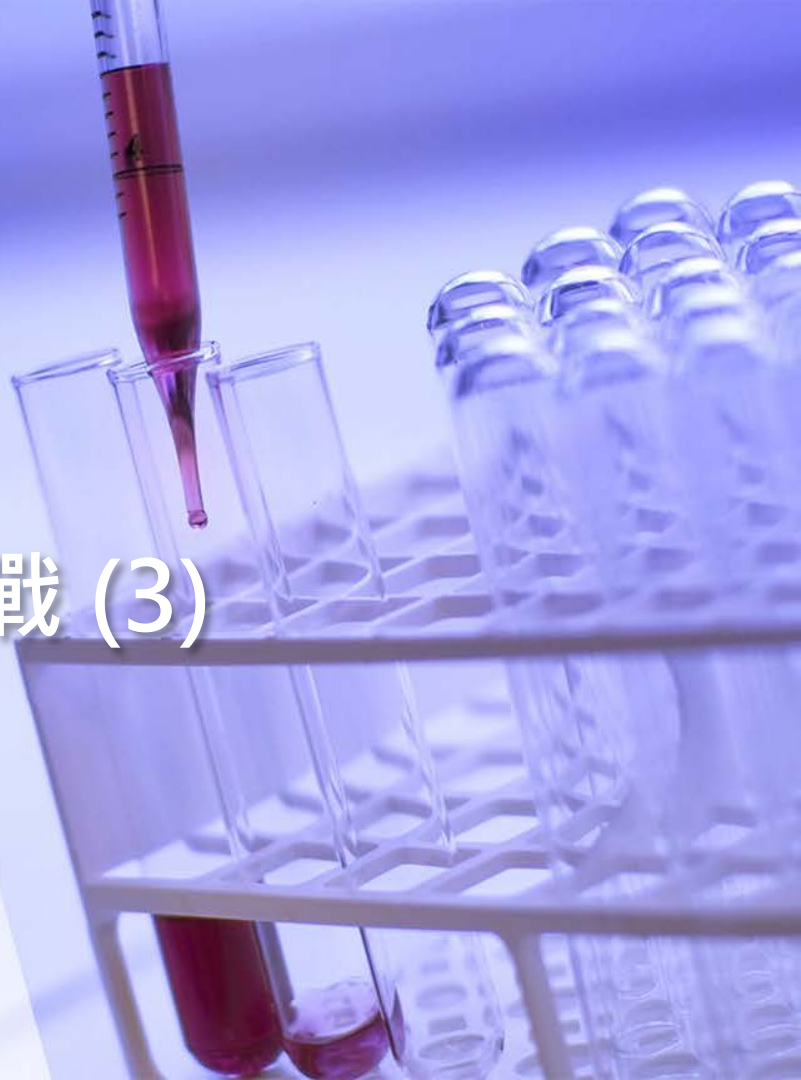


深度學習 最佳入門邁向AI專題實戰 (3)

陳昭明

2021-08-18



北京清華大學同步發行 IT 邦幫忙 AI 組冠軍

深度學習 DEEP LEARNING

最佳入門邁向 AI 專題實戰

陳昭明——著

完整解說必備數學與統計

完整圖片輔助解說

採用最新版 TensorFlow

AI 專題完整程式實戰

神經網路 (NN) 原理與實作

卷積神經網路 (CNN)

物件偵測 (YOLO)

光學文字辨識 (OCR)

車牌辨識 (ANPR)

人臉辨識

生成對抗網路 (GAN)

深度偽造 (DeepFake)

自然語言處理 (NLP)

聊天機器人 (ChatBot)

語言辨識 (ASR)

強化學習 (RL)

本書程式實例可在本書附錄下載



<https://deeppmind.com.tw>

深度學習 DEEP LEARNING

最佳入門邁向 AI 專題實戰

陳昭明——著

遠流出版

大綱

- 第一篇 深度學習導論
- 第二篇 TensorFlow基礎篇
- 第三篇 進階的影像應用
- 第四篇 自然語言處理
- 第五篇 強化學習



大綱



- 第一篇 深度學習導論
- 第二篇 TensorFlow基礎篇
- 第三篇 進階的影像應用
- 第四篇 自然語言處理
- 第五篇 強化學習

涵蓋內容

- 第11章 自然語言處理的介紹
- 第12章 自然語言處理的演算法
- 第13章 聊天機器人(ChatBot)
- 第14章 語音相關應用



涵蓋內容



- 第11章 自然語言處理的介紹
- 第12章 自然語言處理的演算法
- 第13章 聊天機器人(ChatBot)
- 第14章 語音相關應用

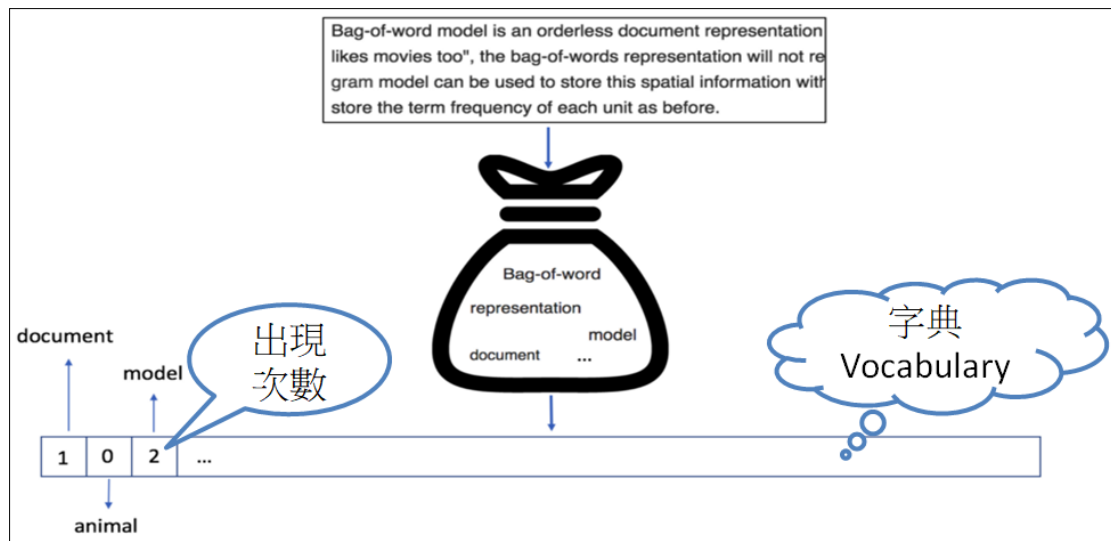
自然語言處理的應用



- 文本分類 (Text classification) 。
- 信息檢索 (Information retrieval) 。
- 文字校對 (Text proofing) 。
- 自然語言生成 (Natural language generation) 。
- 問答系統 (Question answering) 。
- 機器翻譯 (Machine translation) 。
- 自動摘要 (Automatic summarization) 。
- 情緒分析 (Sentiment analysis) 。
- 語音識別 (Speech recognition) 。
- 音樂方面的應用，比如曲風分類、自動編曲、聲音模仿等等。

詞袋(Bag of Words, BOW)

- 把一篇文章進行詞彙的整理，然後統計每個詞彙出現次數，經由前幾名的詞彙猜測全文大意。



實作

- 11_01_BOW.ipynb



TF-IDF (Term Frequency - Inverse Document Frequency)

- 對跨文件常出現的詞彙給予較低的分數，例如 *only* 在每一個文件都出現的話，TF-IDF 對他的評分就相對較低。
- $tf-idf = tf \times idf$

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$TF(\mathbf{term}) = \frac{\text{Number of times } \mathbf{term} \text{ appears in a document}}{\text{Total number of items in the document}}$$

$$idf_{i,j} = \log \frac{|D|}{1 + |D_{t_i}|}$$

$$IDF(\mathbf{term}) = \log \left(\frac{\text{Total number of documents}}{\text{Number of documents with } \mathbf{term} \text{ in it}} \right)$$

實作

- 11_02_TFIDF.ipynb



詞彙前置處理



- 分詞(Tokenizer)
- 去字根(Stemming)
- 詞形還原(Lemmatization)
- 停用詞(Stop Words)
- 詞性(Parts of Speech, POS)
- 命名實體識別 (Named Entity Recognition, NER)
- ngrams
- collocations：片語
- 同義字/相反詞(Synonyms/Antonyms)

詞彙前置處理套件

- NLTK(Natural Language Toolkit)
- spaCy



實作

- 11_03_詞彙前置處理.ipynb

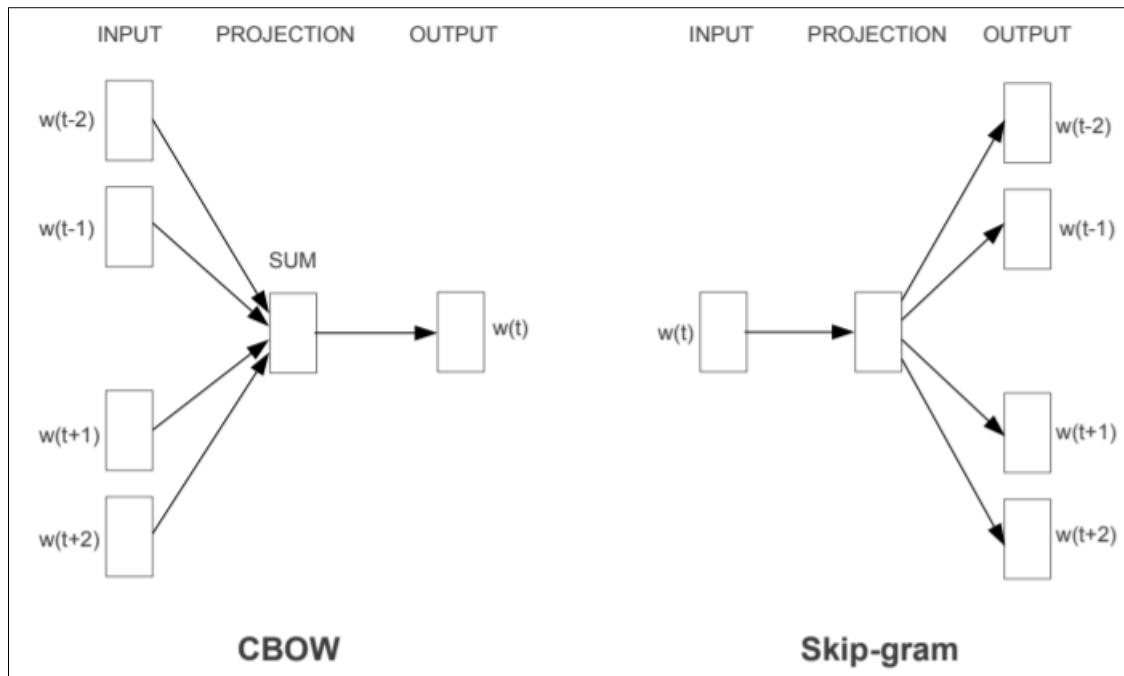


詞向量(Word2Vec)



- BOW和TF-IDF都只著重於詞彙出現在文件中的次數，未考慮語言/文字有上下文的關聯。
- Google研發團隊Tomas Mikolov等人於2013年提出『詞向量』(Word2Vec)，他們蒐集1000億個字(Word)加以訓練，將每個單字改以上下文表達，然後轉換為向量，而這就是『詞嵌入』(Word Embedding)的概念。

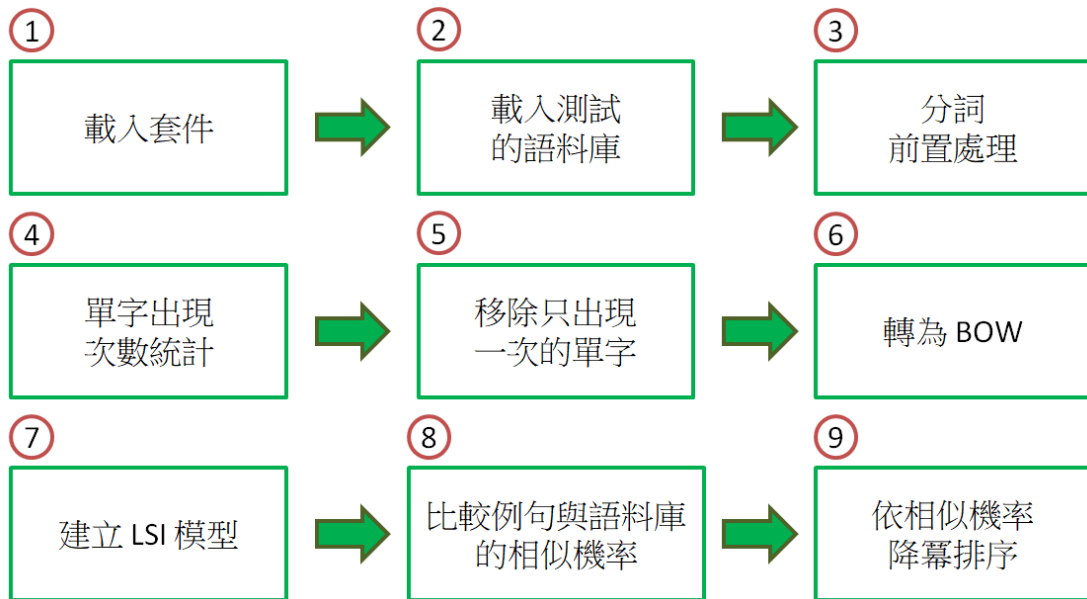
兩種作法



實作 (1)



- 11_04_gensim_相似性比較.ipynb



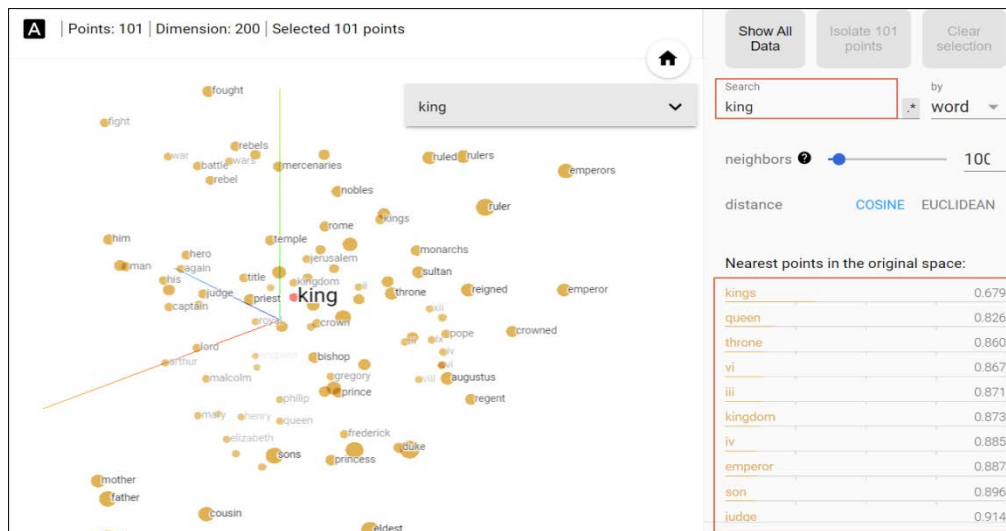
實作 (2)

- 運用Gensim進行Word2Vec訓練與測試
- 11_05_gensim_Word2Vec. ipynb



TensorFlow Embedding Projector

- 詞嵌入的視覺化工具
- <https://projector.tensorflow.org>

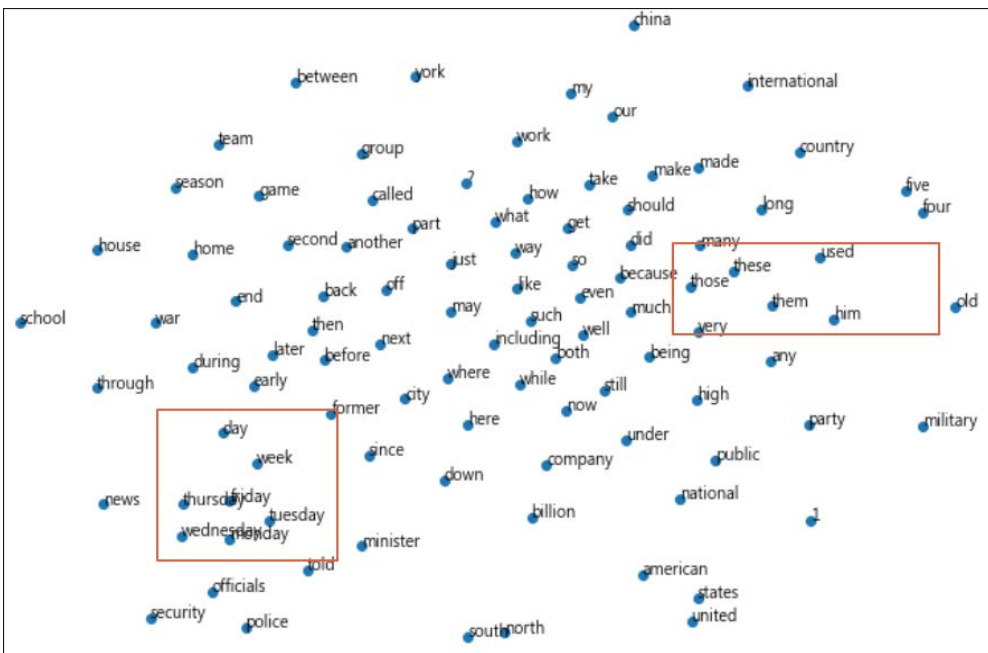


GloVe模型



- 史丹佛大學Jeffrey Pennington等學者於2014所提出的另一套詞嵌入模型，與Word2Vec齊名。
- 認為Word2Vec並未考慮全局的機率分配，只以移動視窗內的詞彙為樣本，沒有掌握全文的資訊，因此，提出了『詞彙共現矩陣』(word-word cooccurrence matrix)，考慮詞彙同時出現的機率。

- 11_06_GloVe.ipynb



中文處理

- Jieba套件
- spaCy套件



實作



- 以Jieba套件進行中文分詞。
- 11_07_中文_NLP.ipynb

全模式：小/ 明/ 硕士/ 毕业/ 于/ 中国/ 中国科学院/ 科学/ 科学院/ 学院/ 计算/ 计算所/ ，/ 后/ 在/ 日本/ 日本京都大学/ 京都/ 京都大学/ 大学/ 深造

精確模式：小明/ 硕士/ 毕业/ 于/ 中国科学院/ 计算所/ ，/ 后/ 在/ 日本京都大学/ 深造

搜索引擎模式： 小明，硕士，毕业，于，中国，科学，学院，科学院，中国科学院，计算，计算所，，后，在，日本，京都，大学，日本京都大学，深造

全模式：新竹/ 的/ 交通/ 交通大/ 大學/ 在/ 新竹/ 的/ 大學/ 大學路/ 學路/ 路上

精確模式：新竹/ 的/ 交通/ 大學/ 在/ 新竹/ 的/ 大學路/ 上

搜索引擎模式： 新竹，的，交通，大學，在，新竹，的，大學，學路，大學路，上

spaCy套件功能

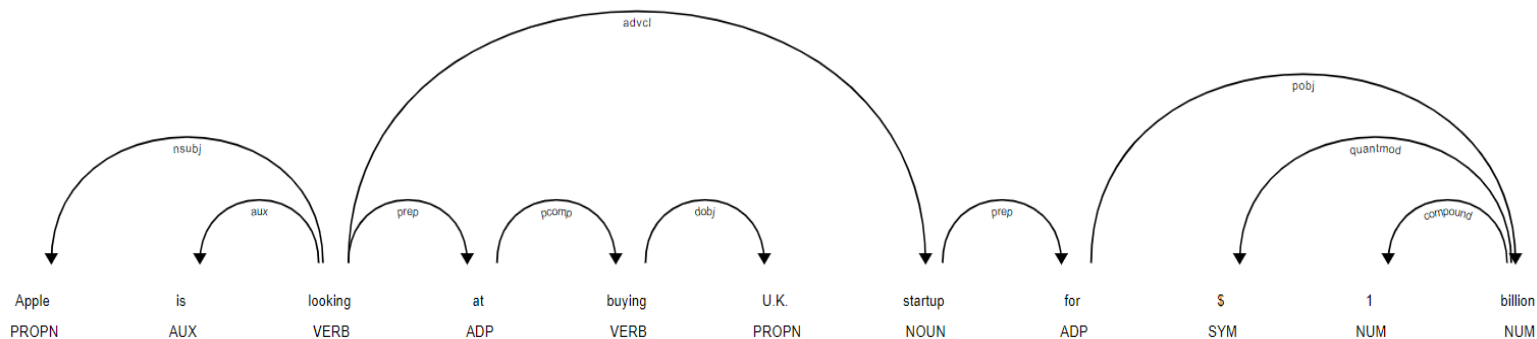


NAME	DESCRIPTION
Tokenization	Segmenting text into words, punctuations marks etc.
Part-of-speech (POS) Tagging	Assigning word types to tokens, like verb or noun.
Dependency Parsing	Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
Lemmatization	Assigning the base forms of words. For example, the lemma of "was" is "be", and the lemma of "rats" is "rat".
Sentence Boundary Detection (SBD)	Finding and segmenting individual sentences.
Named Entity Recognition (NER)	Labelling named "real-world" objects, like persons, companies or locations.
Entity Linking (EL)	Disambiguating textual entities to unique identifiers in a knowledge base.
Similarity	Comparing words, text spans and documents and how similar they are to each other.
Text Classification	Assigning categories or labels to a whole document, or parts of a document.
Rule-based Matching	Finding sequences of tokens based on their texts and linguistic annotations, similar to regular expressions.
Training	Updating and improving a statistical model's predictions.
Serialization	Saving objects to files or byte strings.

<https://spacy.io/usage/spacy-101>

實作

- spaCy 相關功能測試。
- 11_08_spacy_test.ipynb



涵蓋內容

- 第11章 自然語言處理的介紹
- 第12章 自然語言處理的演算法
- 第13章 聊天機器人(ChatBot)
- 第14章 語音相關應用



自然語言處理的演算法

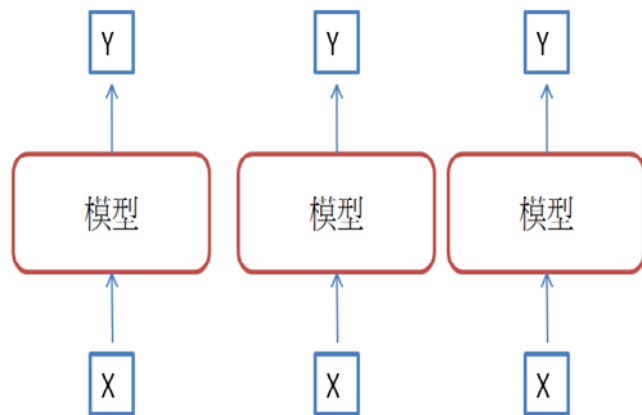


- RNN
- LSTM
- GRU
- Transformer
 - BERT
 - GPT

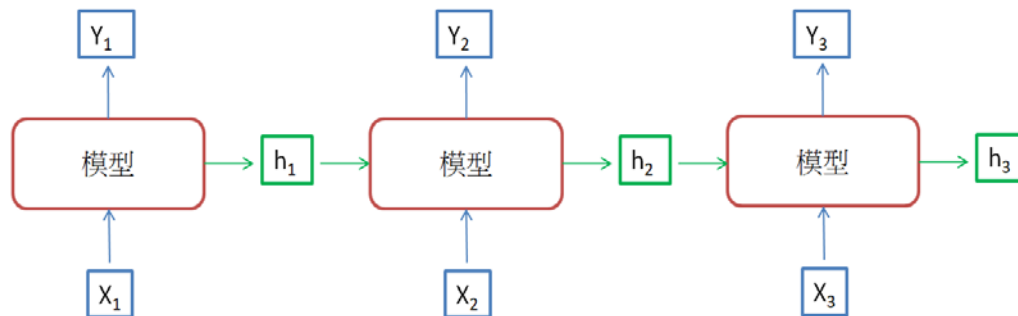
迴歸 vs. RNN



迴歸： $y=Wx+b$



RNN

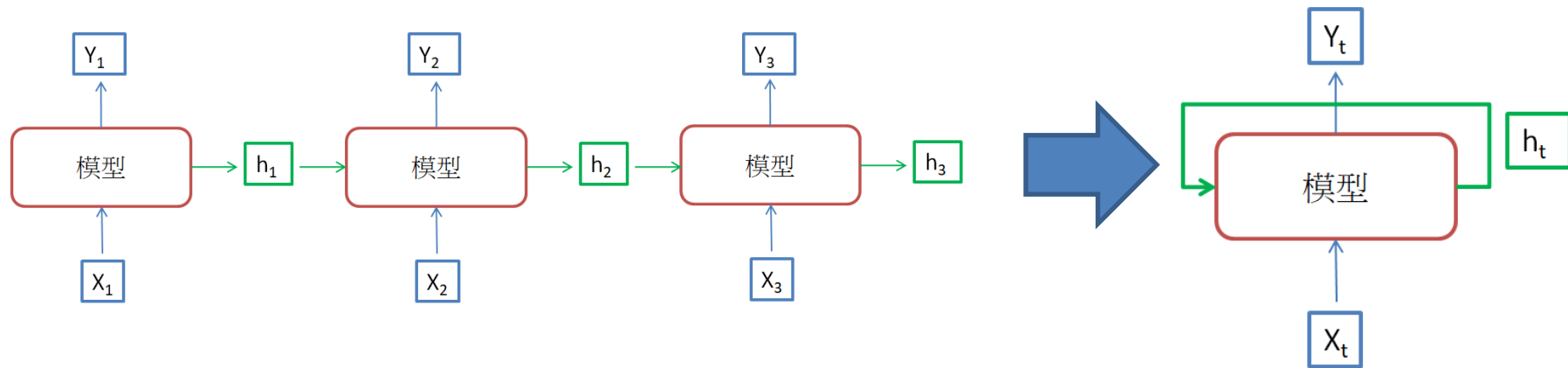


$$h_t = W * h_{t-1} + U * x_t + b$$

$$y = V * h_t$$

其中 W 、 U 、 V 都是權重， h 為隱藏層的輸出。

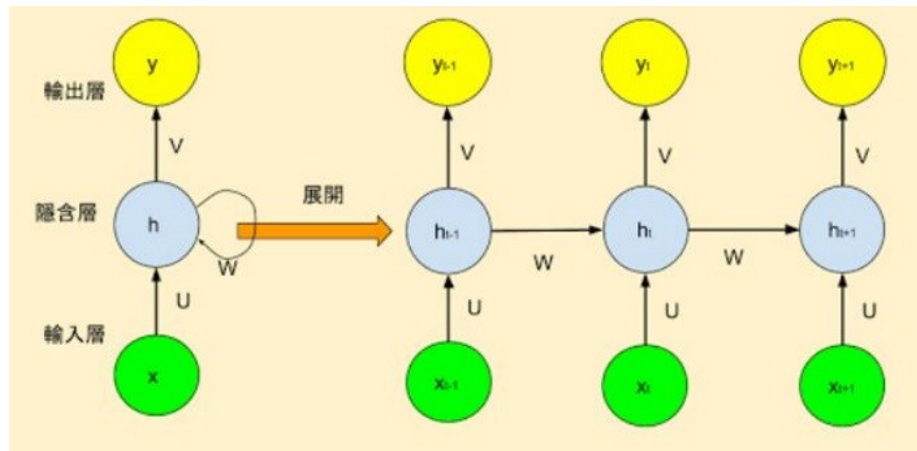
循環神經網路



理解 RNN



- 當前 output 不只受上一層輸入的影響，也受到同一層前一個 output 的影響（即前文）



$$h_t = W * h_{t-1} + Ux_t + b$$

$$y_t = g(V * h_t)$$

上述公式中的 W 、 U 、 V 都是權重，對應不同的輸入來源，與 CNN 一樣，基於『權重共享』（Shared Weights）或稱『參數共享』（Parameter Sharing）的假設， W 、 U 、 V 對每一點推導都共用，即使用相同的值，以簡化計算。

雙向



- 簡單(Vanilla) RNN只考慮上文(上一個神經元)，如果要同時考慮下文，可以直接將simple_rnn包在Bidirectional() 函數內即可。

實作

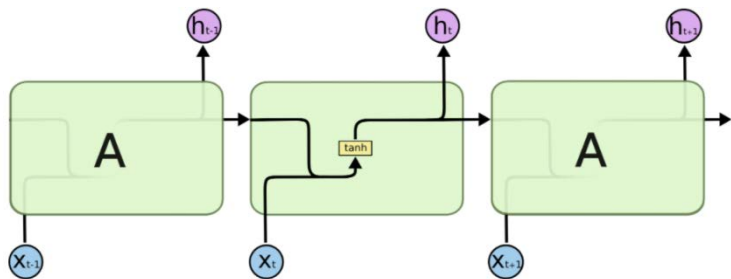
- 簡單的RNN測試。
- 12_01_RNN_test.ipynb



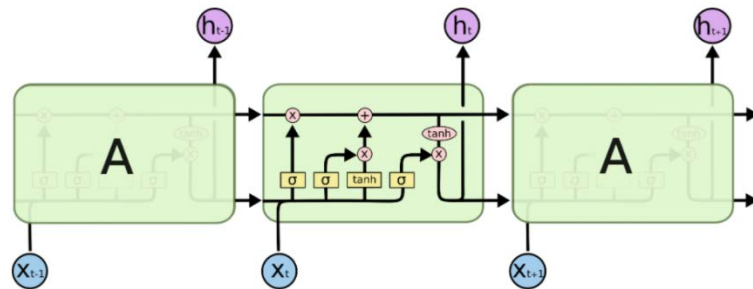
長短期記憶網路(LSTM)



- RNN會造成梯度消失。
- LSTM額外維護一條記憶網路。

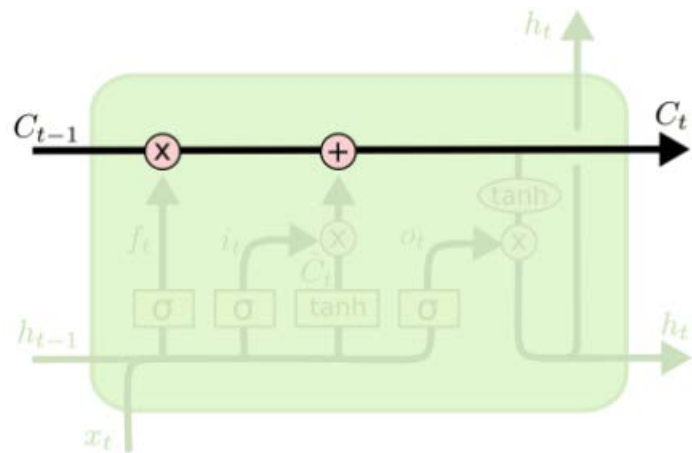


RNN



LSTM

額外維護一條記憶線(Cell state)



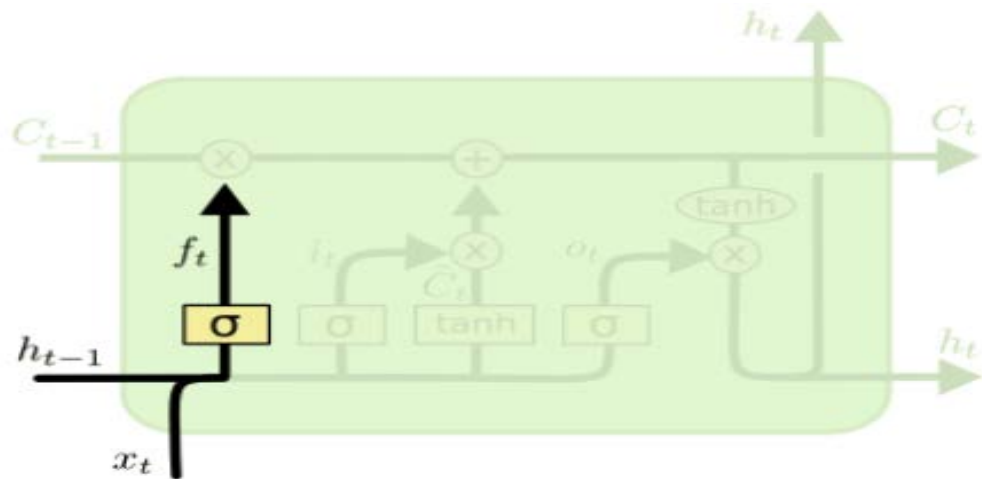
LSTM四個閥(Gate)



- 遺忘閥(Forget Gate)：決定之前記憶是否刪除。
- 輸入閥(Input Gate)：輸入含目前的特徵(x_t)加 $t-1$ 時間點的隱藏層(h_{t-1})。
- 更新閥(Update Gate)：更新記憶(C_t)，為之前的記憶加上目前增加的資訊。
- 輸出閥(Output Gate)：輸出包括目前的正常輸出，乘以更新的記憶。

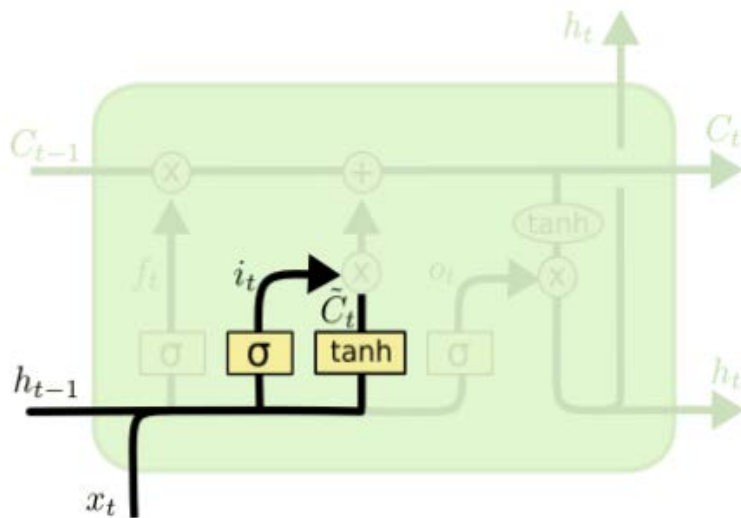
拆解

遺忘閥(Forget Gate)



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

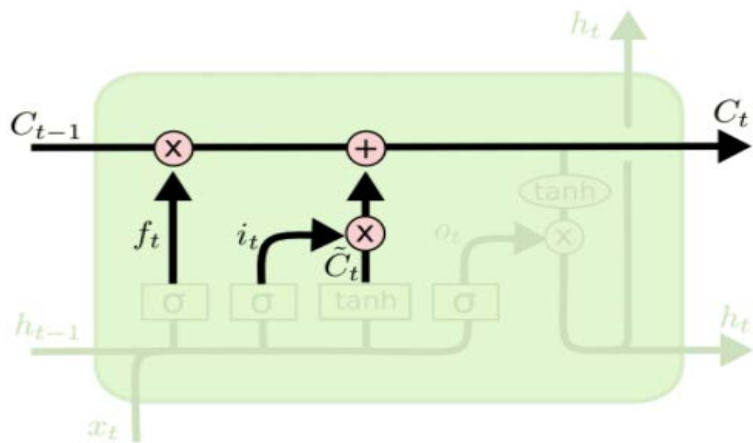
輸入閥(Input Gate)



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

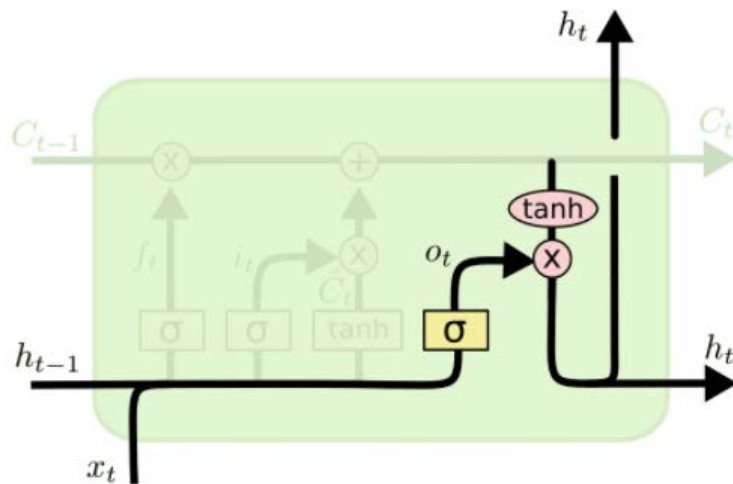
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

更新閥(Update Gate)



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

輸出閥(Output Gate)

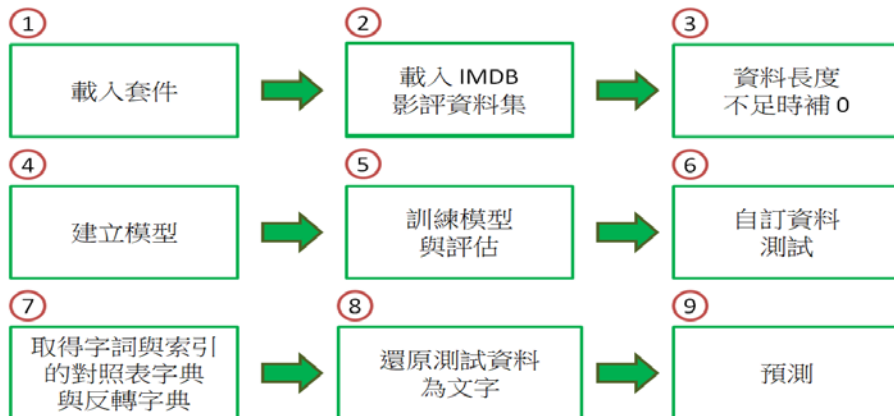


$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

實作

- 以LSTM實作情緒分析(Sentiment Analysis)。
- 資料集：影評資料集(IMDB movie review)。
- 12_02_LSTM_IMDB.ipynb



LSTM重要參數



- `return_sequences`：預設為False，只傳回最後一個神經元的輸出(y)，若為True，則表示每一個神經元的輸出(y)都會傳回。
- `return_state`：預設為False，不會傳回隱藏層狀態(Hidden State)及記憶狀態(Cell State)，反之，則會傳回最後一個神經元的狀態，當作下一週期或批次的輸入。
- `stateful`：預設為False，前一批的隱藏層及記憶狀態不會傳給下一批訓練，延續記憶，反之，則會傳回前一批的隱藏層及記憶狀態，作為下一批訓練的輸入。

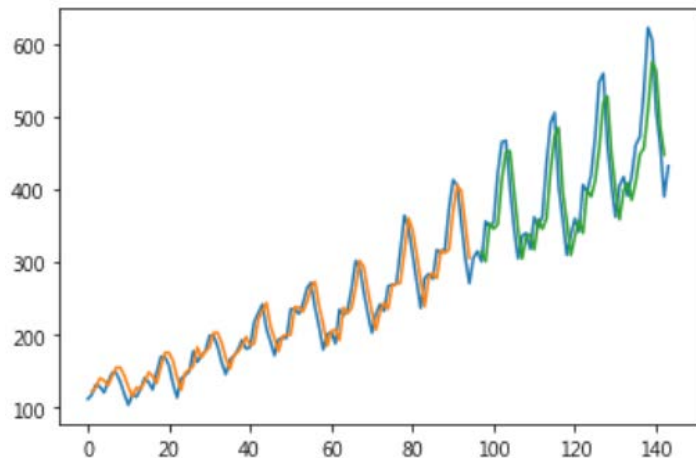
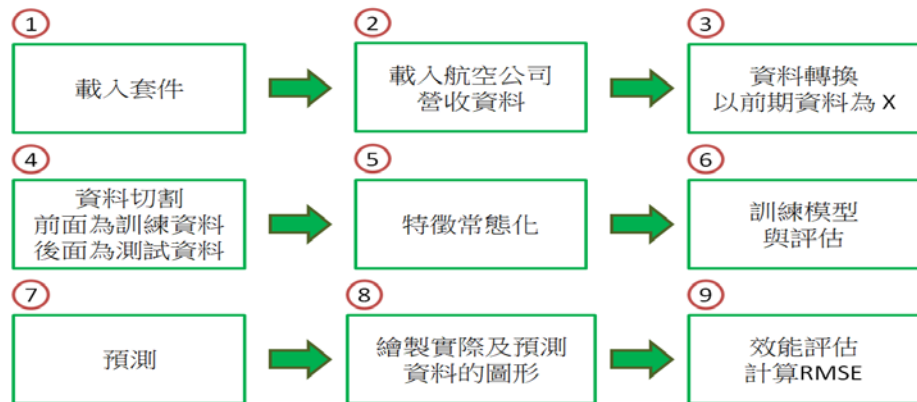
多層LSTM (Stacked LSTM)



- 連續使用多層LSTM。
- 除了最後一層，其他層LSTM均需設定 `return_sequences=True`，確保每一神經元的輸出 (y) 都會傳給下一層。

實作

- LSTM重要參數測試。
- 12_04_LSTM_參數測試.ipynb
- 12_05_Stacked_LSTM.ipynb



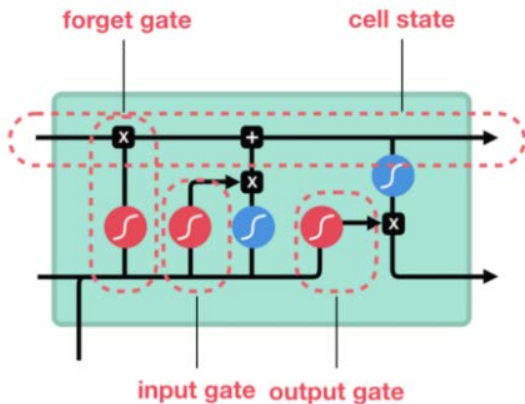
Gate Recurrent Unit (GRU)

- Gate Recurrent Unit (GRU)也是RNN變形的演算法，由Kyunghyun Cho在2014年提出的。
- 改良LSTM缺陷：
 - LSTM計算過慢，GRU可改善訓練速度。
 - 簡化LSTM模型，節省記憶體的空間。
- 將LSTM遺忘閥與輸入閥改由更新閥(Update gate)替代。

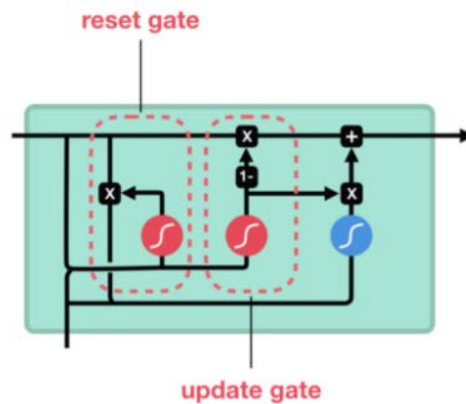
LSTM vs. GRU



LSTM



GRU



sigmoid



tanh



pointwise
multiplication



pointwise
addition

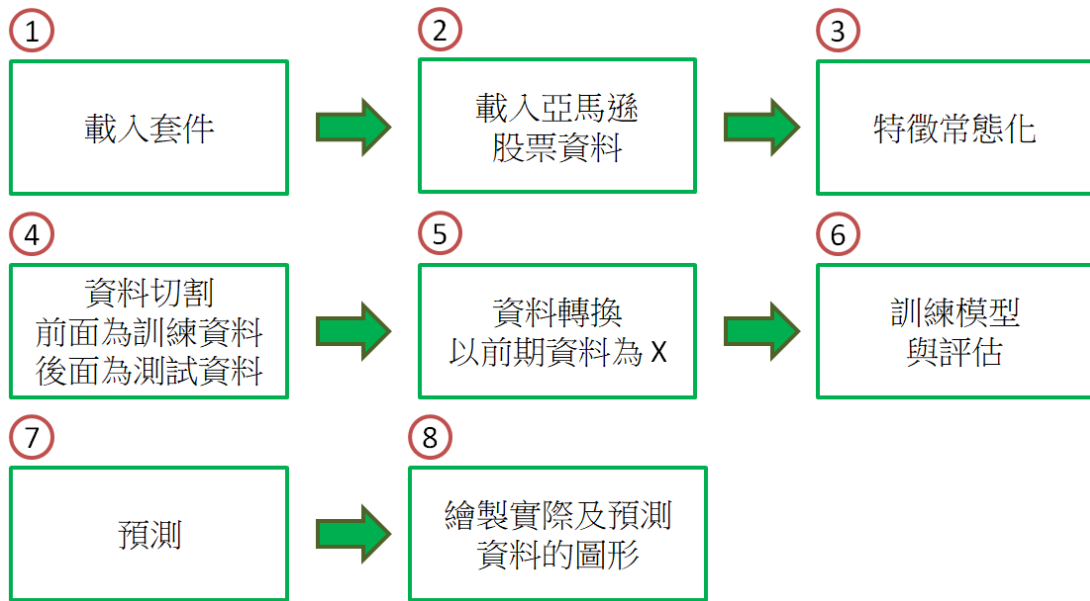


vector
concatenation

實作



- 以LSTM/GRU演算法預測股價。
- 12 06 Stock Forecast. ipynb



注意力機制 (Attention Mechanism)

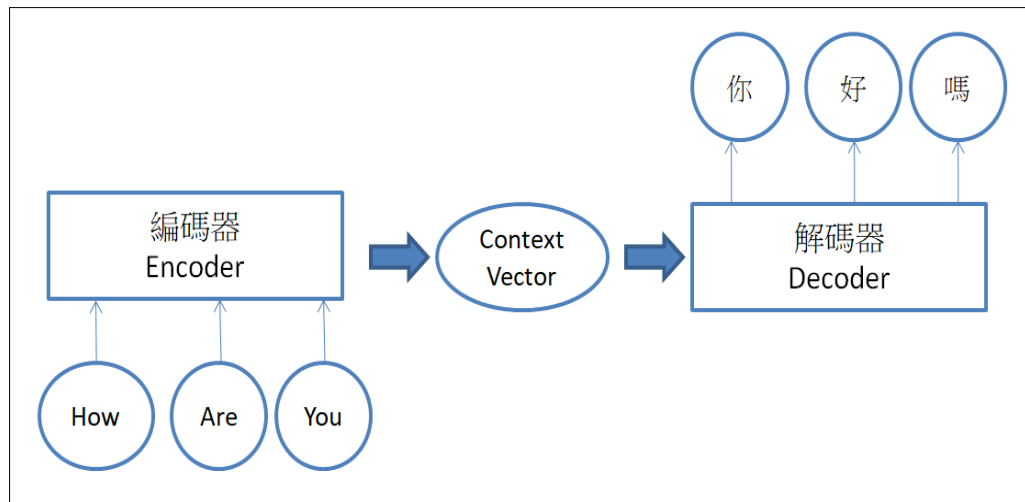
- 在預測時可額外把重點單字或部位納入考量，而不只是上下文。



神經機器翻譯

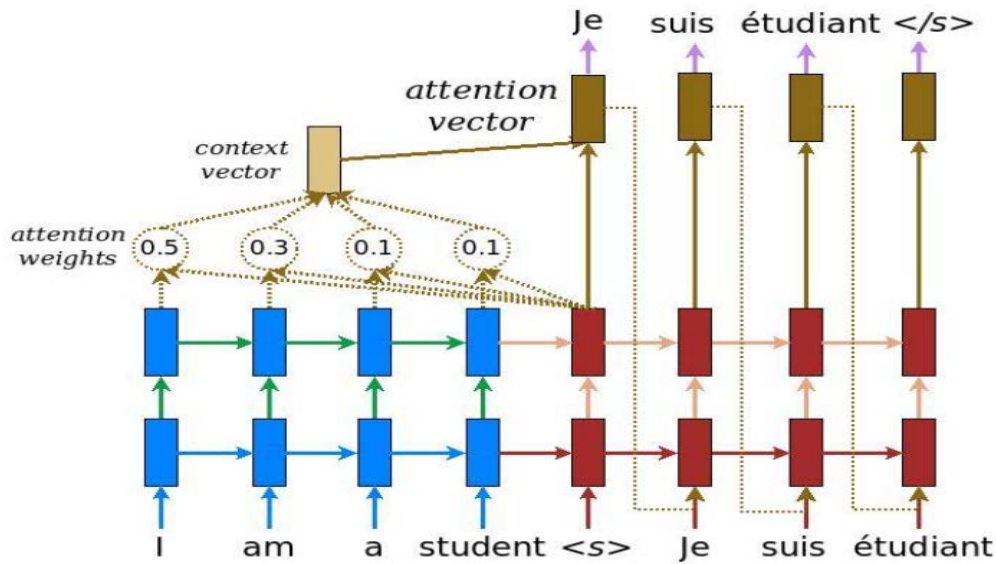
(Neural Machine Translation, NMT)

- 是一種Encoder-Decoder的變形，稱為序列到序列 (Sequence to Sequence, Seq2Seq) 模型



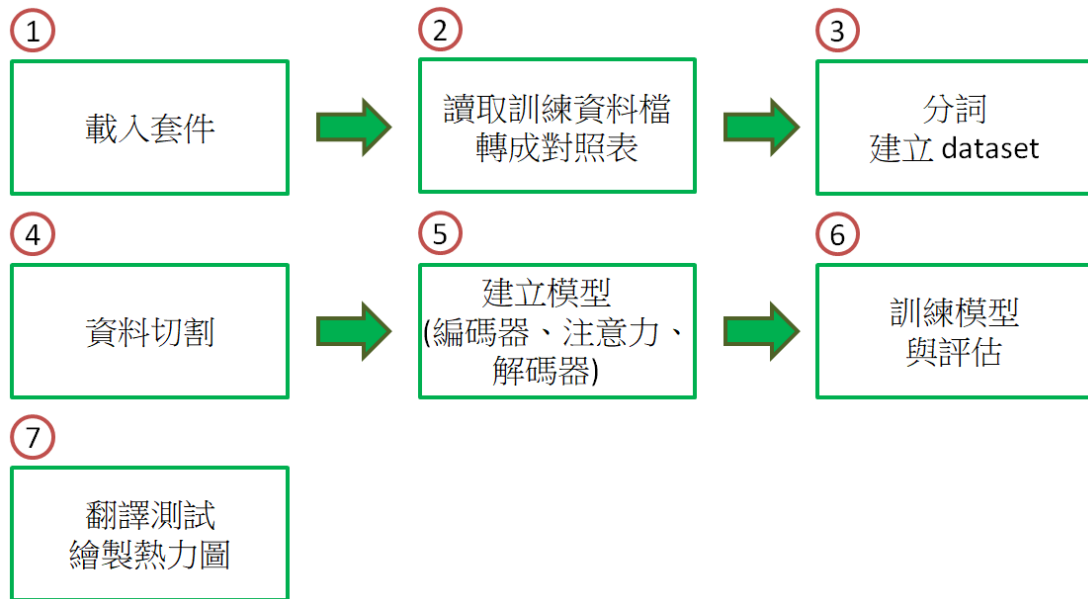
注意力機制在機器翻譯的應用

- 注意力機制就是把要輸入到解碼器(decoder)的詞彙都乘上一個權重(weight)，與Context Vector混合計算成Attention Vector，以預測下一個詞彙。

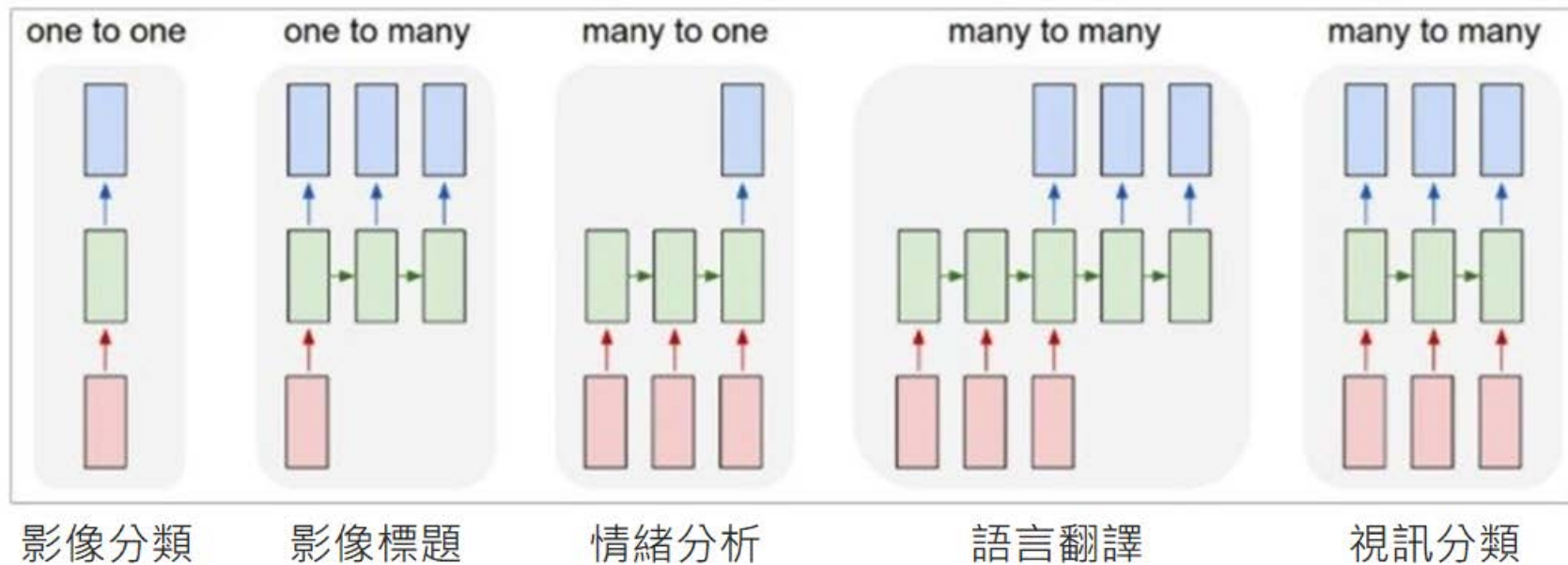


實作

- 12_07_機器翻譯_attention.ipynb



Seq2Seq模型其他型態應用



Transformer架構

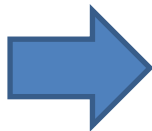


- Google的學者Ashish Vaswani等人於2017年依照Seq2Seq模型加上注意力機制，提出了Transformer架構，如下圖所示。架構一推出後，馬上躍身為NLP近年來最夯的演算法。
- 必讀論文：[Attention Is All You Need](#)。
- 改善：
 - Word2Vec不支援一詞多義
 - RNN序列太長，順序執行太慢。

Word2Vec 缺點



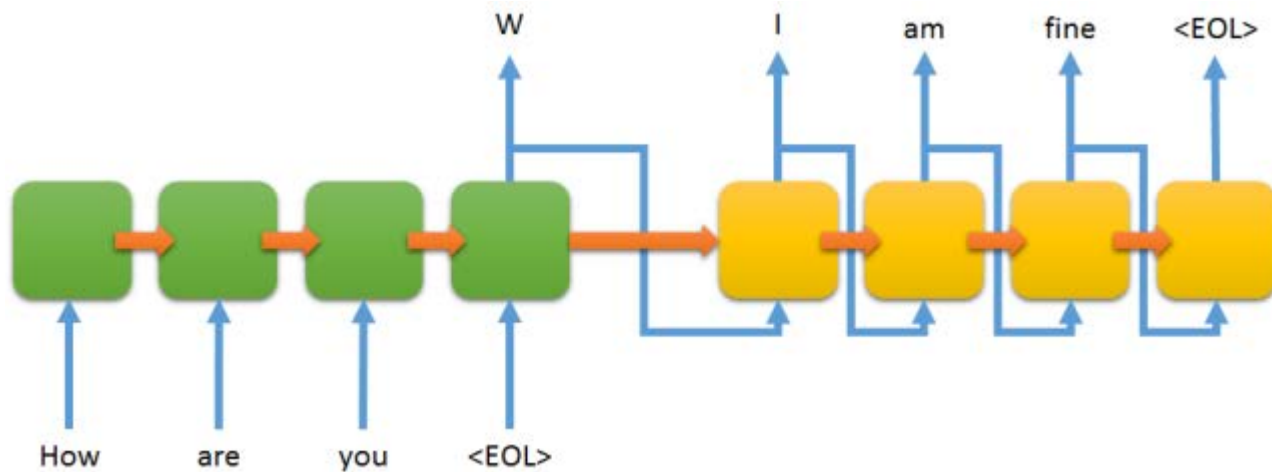
- 一詞多義
 - Apple
 - 蘋果電腦公司
 - 蘋果
 - Saw
 - 看
 - 縫紉
 - Bank
 - Bank Account
 - The bank of the river (河岸)



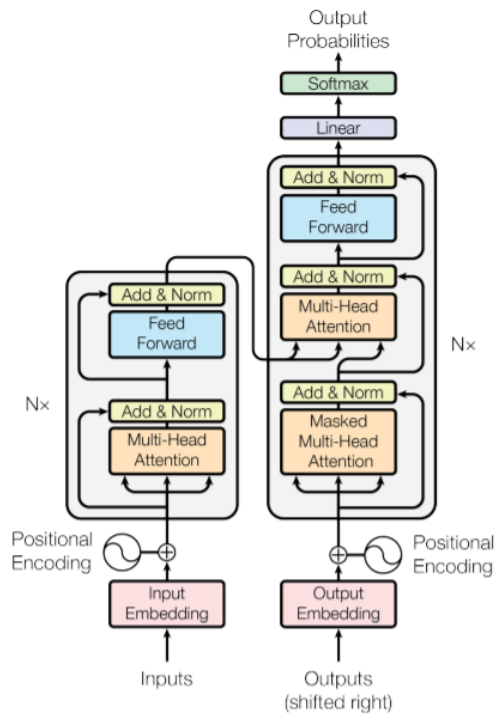
上下文相關
Context Dependent

RNN 缺點

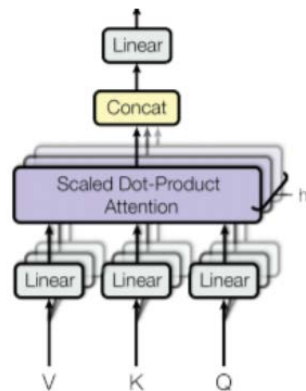
- 序列太長，需順序執行。



模型架構



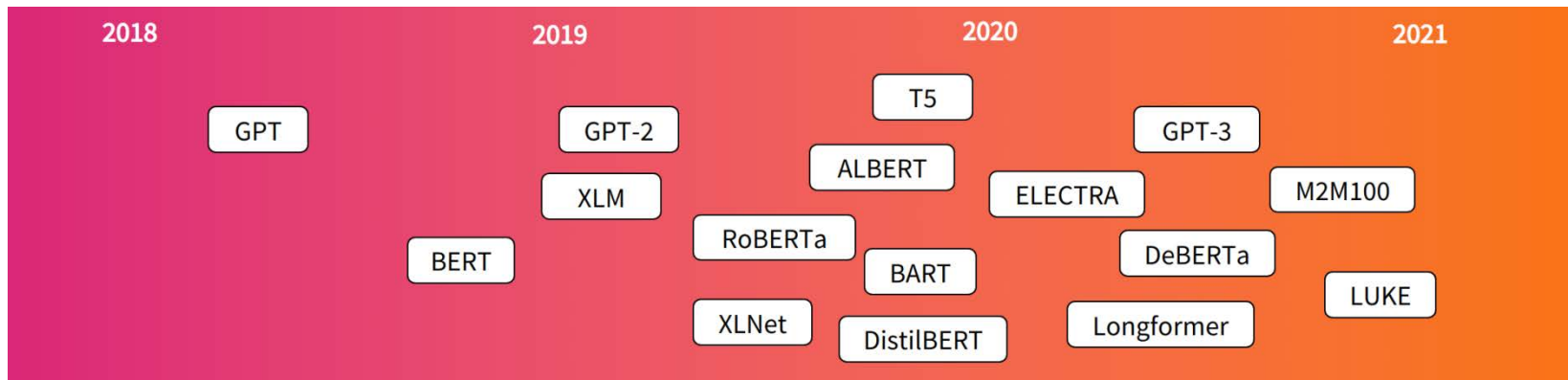
$N \times$ Nx : 重複6次



自注意力機制是多頭(Multi-Head)

實踐Transformer架構的模型

- Transformer : Seq2Seq 模型 + 自注意力機制 (Self-Attention)
- 百家爭鳴 : <https://zhuanlan.zhihu.com/p/120315111>
- [A bit of Transformer history](#)



BERT



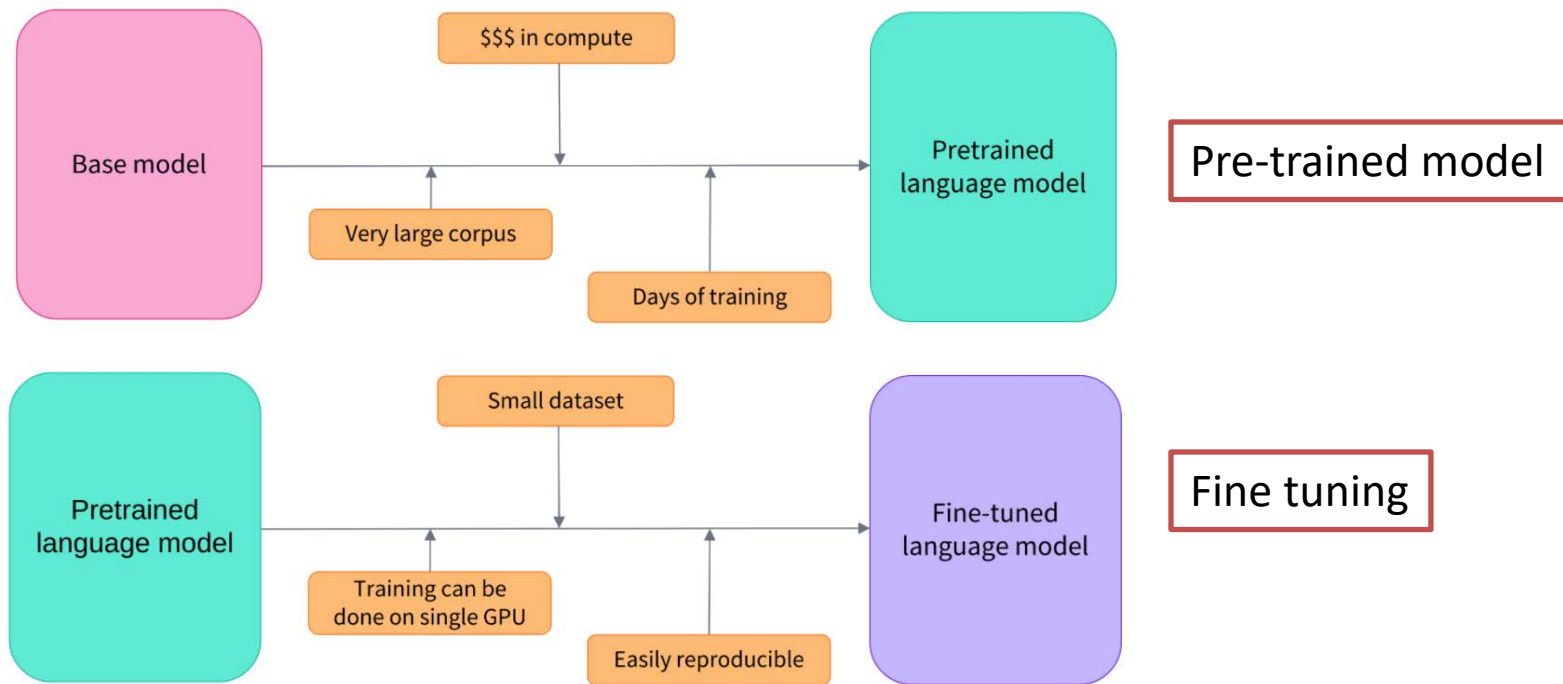
- BERT (Bidirectional Encoder Representations from Transformers) 顧名思義，就是雙向的 Transformer。
- Google Jacob Devlin 等學者於 2018 年發表，參閱『[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#)』。

兩階段遷移學習 (Transfer Learning)



- Pre-trained model
 - 拿很多的語料庫餵入訓練，得到通用模型
- Fine tuning
 - 將該模型拿來做特徵擷取或 fine tune 下游的（監督式）任務
 - 自訂資料集

Pre-trained model vs. Fine tuning



Pretraining 時間



- 1.1 億參數的 12 層 BERT-BASE 得用 16 個 [TPU chips](#) 跑上整整 4 天，花費 500 鎊
- 24 層的 BERT-LARGE 則有 3.4 億個參數，得用 64 個 TPU chips (約 7000 鎊) 訓練
- 使用 [TensorFlow](#) 或是 [PyTorch](#) 將已訓練好的 BERT 載入，就能省去預訓練步驟的所有昂貴成本。

Fine-tuning 配備

- using a GPU like a Titan X or GTX 1080.
<https://github.com/google-research/bert>
- 訓練也需要很多個小時



Transformers套件



- 情緒分析(Sentiment analysis)。
- 文字生成(Text generation)：限英文。
- 命名實體識別(Named Entity Recognition, NER)。
- 問題回答(Question Answering)。
- 克漏字填空(Filling masked text)。
- 文字摘要(Text Summarization)：將文章節錄出大意。
- 翻譯(Translation)。
- 特徵萃取(Feature extraction)：類似詞向量，將文字轉換為向量。

實作



- 12_08_BERT_情緒分析.ipynb
- 12_09_BERT_問題回答.ipynb
- 12_10_BERT_填漏字.ipynb
- 12_11_GPT2_文字生成.ipynb
- 12_12_BERT_NER.ipynb
- 12_13_文字摘要.ipynb
- 12_14_T5_翻譯.ipynb
- 12_15_BERT_Text_classification_on_GLUE.ipynb

涵蓋內容

- 第11章 自然語言處理的介紹
- 第12章 自然語言處理的演算法
- 第13章 聊天機器人(ChatBot)
- 第14章 語音相關應用



第13章 聊天機器人(ChatBot)

- ChatBot類別
- ChatBot設計
- ChatBot實作
- ChatBot工具套件
- Dialogflow實作

ChatBot類別



- 不限話題的機器人：可以與人天南地北的閒聊。
- 任務型機器人：例如專家系統，具備特定領域的專業知識。
- 常見問答集(Frequently Asked Questions, FAQ)。
- 資訊檢索：利用全文檢索的功能，搜尋關鍵字的相關資訊。
- 資料庫應用：藉由SQL指令來查詢、篩選或統計資料。

ChatBot設計



- 訂定目標：根據規劃的目標，選擇適合的ChatBot類別，可以是多種類別的混合體。
- 收集應用案例(Use Case)：收集應用的各種狀況和場景，整理成案例。
- 提供的內容：內容行銷(Content Marketing)。
- 挑選開發平台。
- 佈署平台：可選擇雲端或本地端。
- 用戶偏好(Preference)與面貌(Profile)：考量要儲存哪些與業務相關的用戶資訊。

開發平台

- 套裝軟體
- ChatBot平台
- 開發工具
- 自行構建



ChatBot相關術語



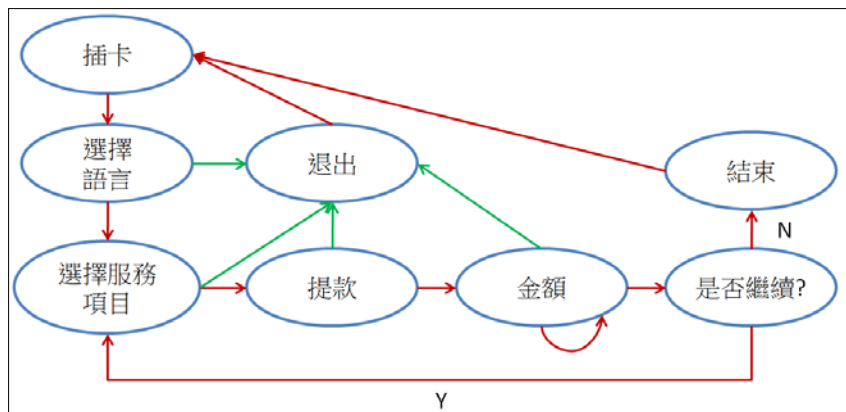
- 技能(Skill)：例如銀行的技能包含存提款、定存、換匯、基金購買、房貸。
- 意圖(Intent)：技能中每一種對談的用意，例如，技能是旅館訂房，意圖則是有查詢某日是否有空的雙人房、訂房、換日期、退房、付款、...等。
- 實體(Entity)：關鍵的人事時地物。
- 例句(Utterance)：因為不同的人表達同一意圖會有各種不同表達方式，所以需要收集大量的例句。
- 行動(Action)：所需資訊均已收集完整後，即可作出回應(Response)與相關的動作。
- 開場白(Opening Message)：例如歡迎詞(Welcome)、問候語(Greeting)等。

對話設計



- 對話管理
- 整合社群媒體
- 人機整合
 - 必須設定跳脫條件，一旦察覺對話不合理，就應停止或轉由客服人員處理，避免引起使用者不快，造成反效果。

- 槽位填充(Slot Filling)
- 我要訂3/21雙人房
- 槽位
 - 日期：3/21
 - 房型：雙人房



ChatBot實作

- NLP加上相似度比較，製作簡單ChatBot。
- 13_01_simple_chatbot.ipynb

```
請輸入:  
hello  
你問的是: greeting, Hello  
回答: Hello, thanks for asking  
請輸入:  
How you could help me  
你問的是: options, what help you provide?  
回答: I can guide you through Adverse drug reaction list, Blood pressure tracking, Hospitals and Pharmacies  
請輸入:  
Adverse drug reaction  
你問的是: adverse_drug, How to check Adverse drug reaction?  
回答: Navigating to Adverse drug reaction module  
請輸入:  
blood pressure result  
你問的是: blood_pressure_search, Show blood pressure results for patient  
回答: Patient ID?  
請輸入:  
123  
我不懂你的意思, 請再輸入一次.  
請輸入:  
pharmacy  
你問的是: pharmacy_search, Find me a pharmacy  
回答: Please provide pharmacy name  
請輸入:  
hospital  
你問的是: hospital_search, Hospital lookup for patient  
回答: Please provide hospital name or location  
請輸入:  
回答: Bye! Come back again soon.
```

ChatBot工具套件



- [ChatterBot](#)：採配接器模式(Adapter Pattern)，是一個可擴充式的架構，支援多語系。
- [ChatBotAI](#)：以樣板(Template)語法訂定各式的樣板。
- [Rasa](#)：以Markdown格式訂定意圖(Intent)、故事(Story)、回應(Response)、實體(Entity)與對話管理等功能。

ChatterBot實作

- 13_02_ChatterBot_test.ipynb



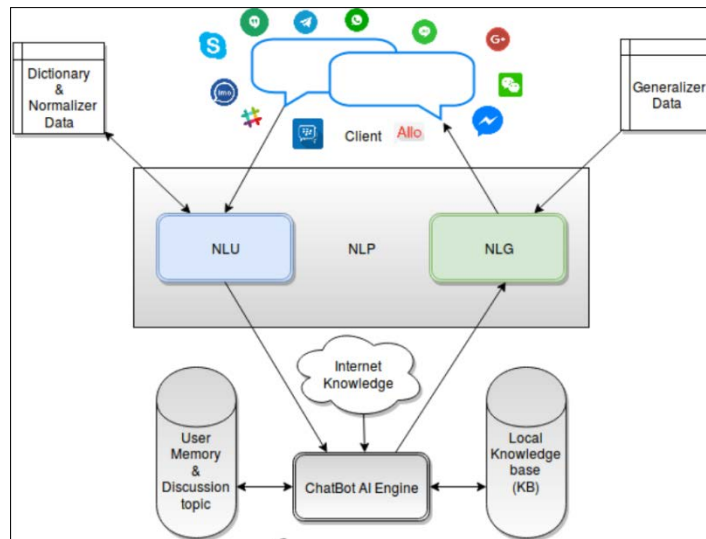
ChatterBot 實作

- 13_02_ChatterBot_test.ipynb



chatbotAI 實作

- 13_03_ chatbotAI_test.ipynb



Rasa 實作



- 依書中13-4-3說明安裝與測試。

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hello
Hey! How are you?
Your input -> I am fine
Great, carry on!
Your input -> what is you name
I am a bot, powered by Rasa.
Your input -> I am disappointed
Here is something to cheer you up:
Image: https://i.imgur.com/nGF1K8f.jpg
Did that help you?
Your input -> yes
Great, carry on!
Your input -> great
Great, carry on!
Your input -> bye
Bye
Your input -> /stop
2021-05-04 22:15:07 INFO      root - Killing Sanic server now.
```

Dialogflow 版本



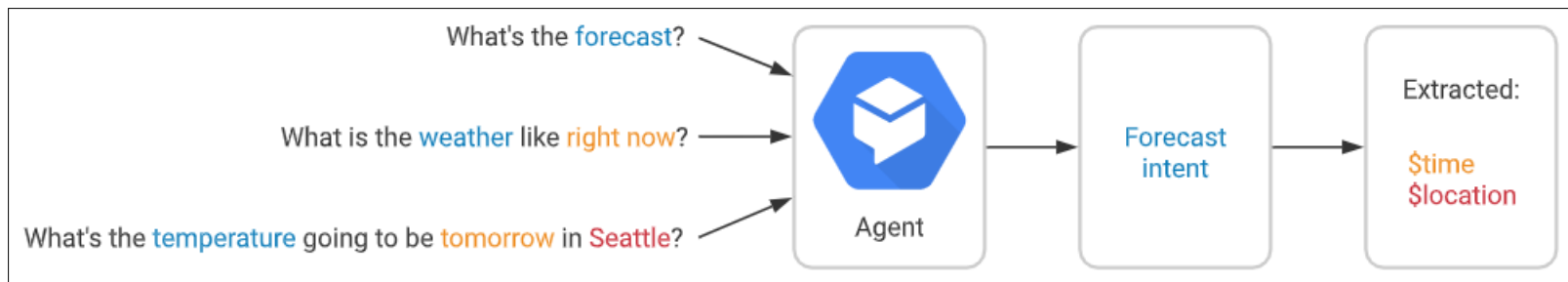
- Dialogflow CX：為進階版本。
- Dialogflow ES：為標準版，可免費試用。
- 比較表可參閱：
<https://cloud.google.com/dialogflow/docs/editions>。

Dialogflow 術語定義



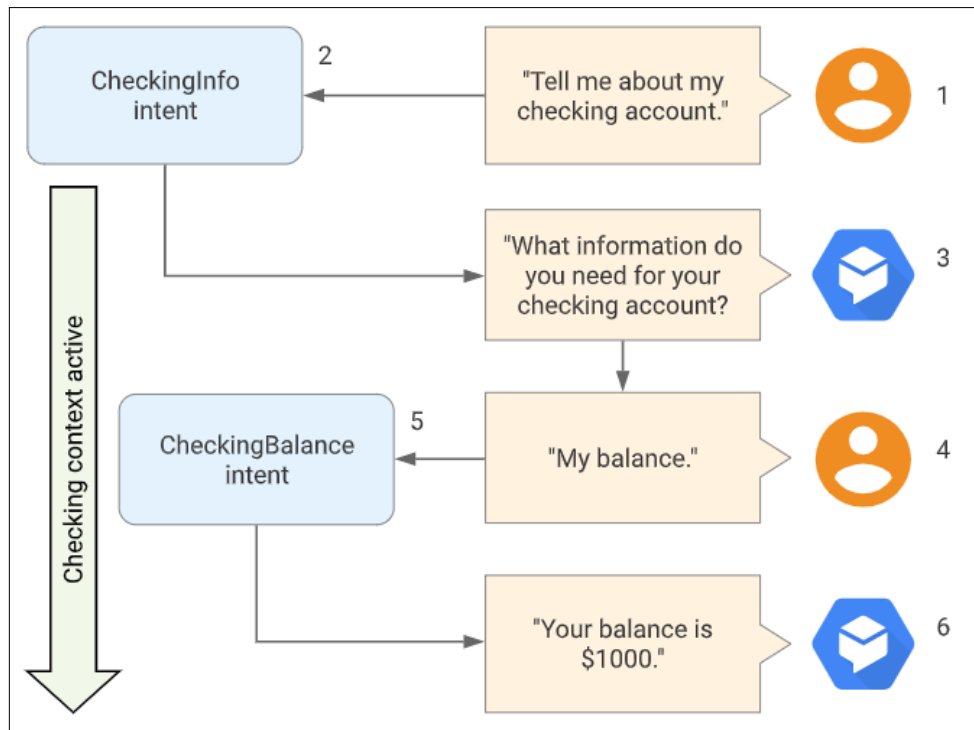
- Agent：即ChatBot本身。
- 意圖(Intent)：訓練的片語(Training phrases)。
- 行動(Action)：ChatBot接收到意圖後採取的行動。
- 參數(Parameter)：定義槽位填充所需的資訊，包括必填的獲選填的參數，Dialogflow可以從使用者的表達中找出對應的實體(Entity)。
- 回應(Response)：行動完畢後，回應使用者的文字或語音。

從意圖中找出時間和地點



Dialogflow可從意圖中找出時間和地點

槽位填充(Slot Filling) 機制



Dialogflow 實作



ChatBot

你要訂房嗎?

Yes

回答

No

要訂幾人房?

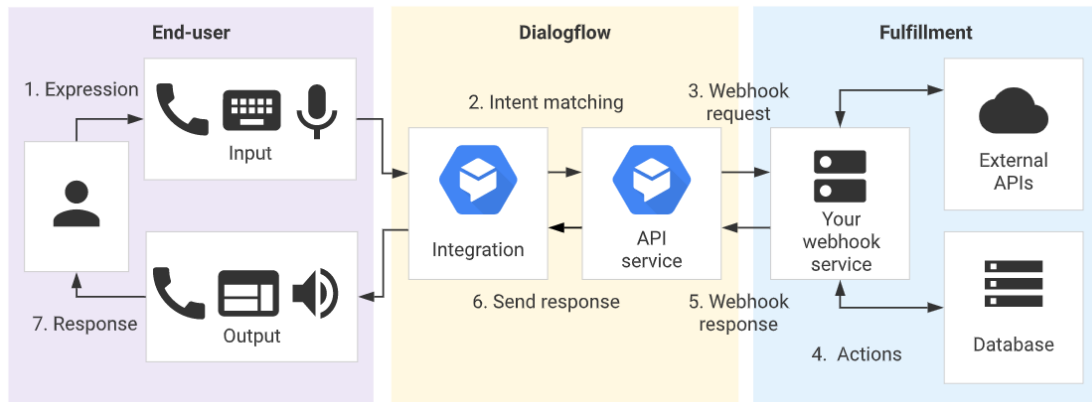
再見

追問意圖(Follow-up intent):

- 可依據使用者的回答定義不同的回答方式，以追問意圖，透過此功能可以建立有限狀態機。

履行(Fulfillment)

- ChatBot除了回應文字之外，也能夠與資料庫或社群軟體整合，開發者可以撰寫一個服務，整合各種軟硬體。



Dialogflow 開通



- 請參閱
 - <https://cloud.google.com/dialogflow/es/docs/quick/setup>
- 開通
 - <https://cloud.google.com/dialogflow/es/docs/console>
 - 請按書中13-5-1說明。

建立 Agent



<https://dialogflow.cloud.google.com/#/newAgent>

HotelOrder

CREATE

DEFAULT LANGUAGE ?

English — en

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+8:00) Asia/Hong_Kong

Date and time requests are resolved using this timezone if not provided in the API requests.

GOOGLE PROJECT

quickstart-1590025178539

Enables Cloud functions, Actions on Google and permissions management.

AGENT TYPE

☒ Set as Mega Agent

Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. [mega agent](#)).

建立意圖



- Dialogflow會預先建立2個意圖
 - Default Fallback Intent
 - Default Welcome Intent
- 建立意圖：點擊『Create Intent』按鈕，輸入意圖名稱，並點擊『Add Training Phrases』超連結，就可輸入多組問句與回應。

建立意圖截圖



• get-agent-name

SAVE

...

Contexts ?

Events ?

Training phrases ?

Search training phras

“ Add user expression

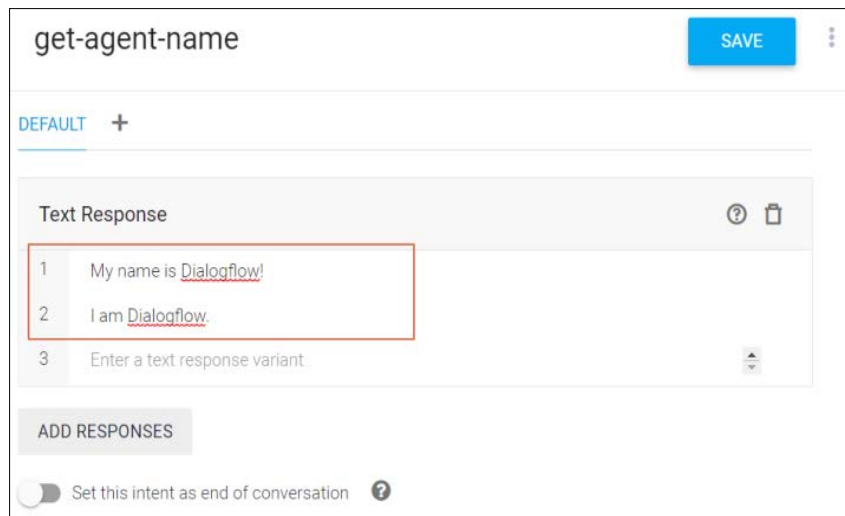
“ Do you have a name?

“ Tell me your name

“ What is your name?

建立回應

- 回應：點擊『Add Response』超連結。



get-agent-name SAVE

DEFAULT +

Text Response ?

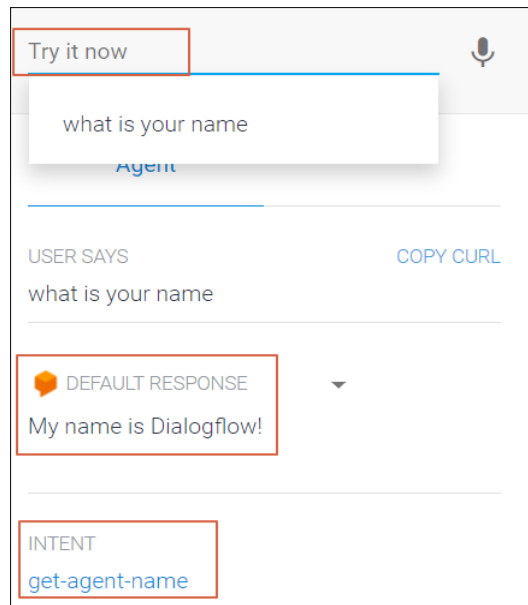
- 1 My name is Dialogflow!
- 2 I am Dialogflow.
- 3 Enter a text response variant ↑

ADD RESPONSES

☐ Set this intent as end of conversation ?

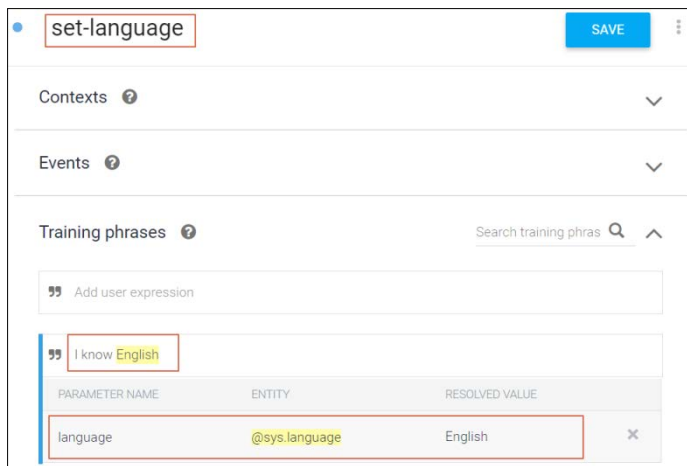
測試

- 測試：存檔，並確定訓練完成的訊息出現之後，即可在畫面右側測試。
- 可鍵盤或語音輸入。



建立參數

- 輸入的例句如果包含內建的實體(Entity)，則會被解析出來，當作參數，可進一步設定參數屬性。點選畫面左側Intent旁的『+』。



set-language SAVE

Contexts ?

Events ?

Training phrases ? Search training phrases

” Add user expression

” I know English

PARAMETER NAME	ENTITY	RESOLVED VALUE
language	@sys.language	English

設定參數屬性



- Required: 是否必要輸入。
- Parameter Name: 參數名稱。
- Entity: 選擇Entity類別，可修改為其他類別。
- Value: 參數的名稱，回應(Response)可以此名稱取得參數值。
- Is List: 參數值是否為List，即一參數含多個值。
- Prompts: 若輸入的問句或回答欠缺此參數，Agent會顯示此提示，詢問使用者。

夾帶參數的回應



- 輸入回應(Response)：『Wow! I didn't know you knew \$language.』，其中\$language會自使用者的問句取得變數值。
- 測試：輸入『I speak english』，回應的
\$language = english

進階


- 建立自訂的實體(Entity)。
- 追問意圖(Follow-up intent)




建立履行(Fulfillment)



- Webhook：撰寫一個網頁服務(Web service)，Dialog透過POST請求送給Webhook，並接收回應。
- 設定：啟用『Enable webhook call for this intent』。

 Fulfillment

Webhook

ENABLED 

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

撰寫程式



- 可使用多種語言撰寫，這裡我們使用Python加上Flask套件，撰寫Web程式，完整程式請參考 `dialogflow\webhook\app.py`。
- 佈署
 - 先在本機測試：使用 `ngrok.exe` 將內部網址對應到外部網址。
 - 測試成功後，再將程式佈署到Heroku或其他網站測試。

整合



- Dialogflow還可以整合語音交換機、社群媒體、Spark等，詳情可參閱『[Dialogflow Integrations說明](#)』。
- Dialogflow也內建許多應用程式，可參閱『[Dialogflow Prebuilt Agents說明](#)』。