遞歸神經網絡
Recurrent Neural Network

aGENDA

- Sequence Data的分析(1)Time Series
- Sequence Data的分析(2)NLP

- 遞歸神經網絡
  Recurrent Neural Network

- NLP in security

Sequence Data
的分析(1)
Time Series

Time Series

https://en.wikipedia.org/wiki/Time_series

一組按照時間發生先後順序進行排列的數據點序列。

通常一組時間序列的時間間隔為一恆定值（如1秒，5分鐘，12小時，7天，1年），因此時間序列可以作為離散時間數據進行分析處理。

時間序列廣泛應用:
數理統計、
信號處理、
模式識別、
計量經濟學、
數學金融、
天氣預報、
地震預測、
腦電圖、
控制工程、航空學、通信工程以及絕大多數涉及到時間數據測量的應用科學與工程學。

Time Series Analysis

時間序列模型基本概念：AR, MA, ARMA, ARIMA 模型

https://mropengate.blogspot.com/2015/11/time-series-analysis-ar-ma-arma-arima.html

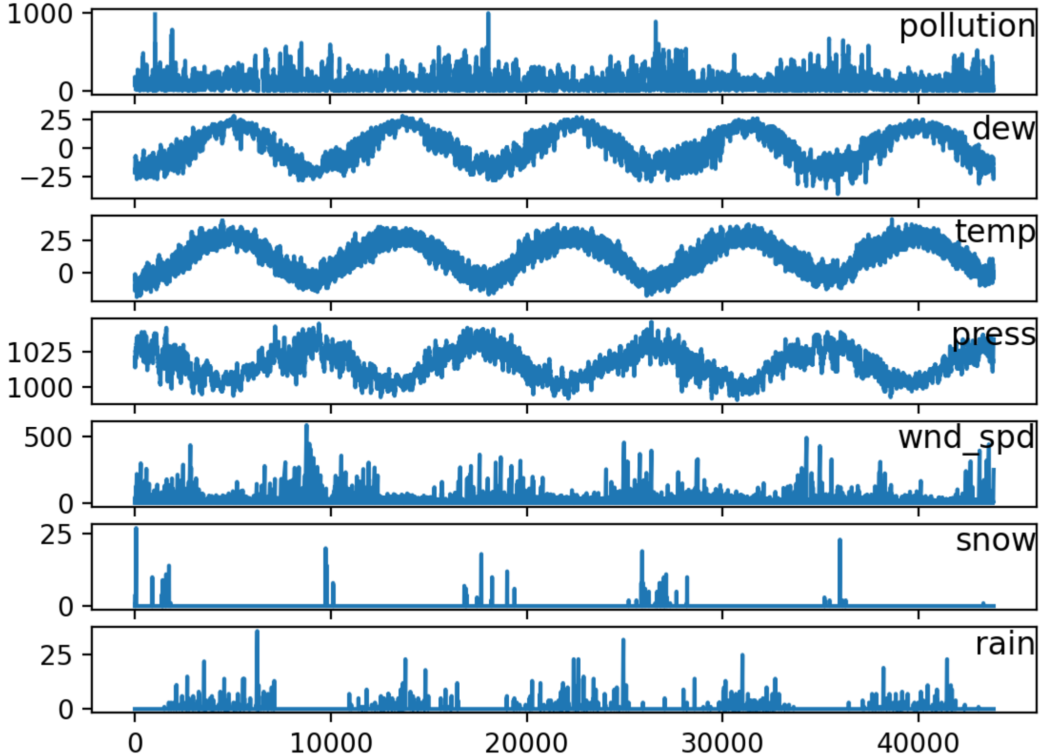
http://yongfeng.me/attach/time-series-analysis-zhang.pdf https://ocw.mit.edu/courses/mathematics/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/video-lectures/lecture-8-time-series-analysis-i/

Deep Time Series Classification

- 預測股票每日最高價
- 空氣污染預測
- 地震預測
- PM2.5預測
- GDP預測
- 人口數預測

空氣污染預測

https://raw.githubusercontent.com/jbrownlee/Datasets/master/pollution.csv



Multivariate LSTM Forecast Model
https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

https://www.itread01.com/content/1544492764.html

https://www.kaggle.com/

The 3rd YouTube-8M Video Understanding Challenge
Temporal localization of topics within video
Research · a month to go · video data, object detection
$25,000
222 teams

Open Images 2019 - Object Detection
Detect objects in varied and complex images
Research · a month to go · image processing, image data
$25,000
453 teams

Open Images 2019 - Visual Relationship
Detect pairs of objects in particular relationships
Research · a month to go · image processing, image data
$25,000
145 teams

https://faridrashidi.github.io/kaggle-solutions/

Kaggle Solutions

Fork me on GitHub

The Most Comprehensive List of Kaggle Solutions and Ideas

This is a list of almost all available solutions and ideas shared by top performers in the past Kaggle competitions. This list will get updated as soon as a new competition finished. If you find other solutions beside the ones listed here I would suggest you to contribute to this repo by making a pull request. The symbols were used in this list is described here.

If you found it interesting you can give a star or make a fork

Star 20 Fork 6

Check out the following markdown pages about Top Kagglers Tips/Tricks and all Kernels of The week.

- Top Kagglers Interviews and Lectures
- Kernels of The Week

https://www.kaggle.com/c/career-con-2019

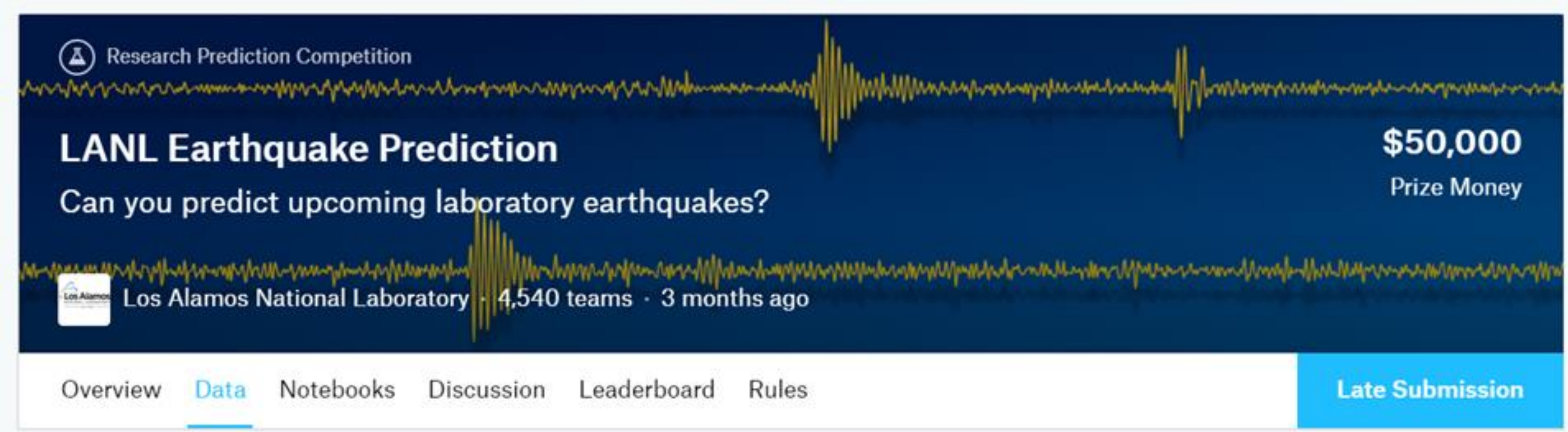
https://www.kaggle.com/purplejester/pytorch-deep-time-series-classification

https://github.com/ammar1y/My-Solution-to-Help-Navigate-Robots-Competition

https://www.youtube.com/watch?v=ageh45rxyXU

Kaggle Earthquake Prediction Challenge

https://www.kaggle.com/c/LANL-Earthquake-Prediction/data



https://github.com/Kaggle/kaggle-api

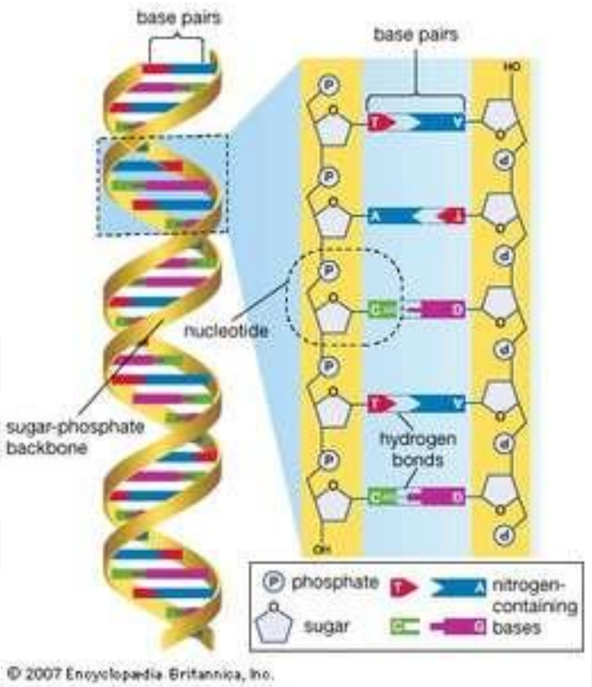
https://www.youtube.com/watch?v=TffGdSsWKlA

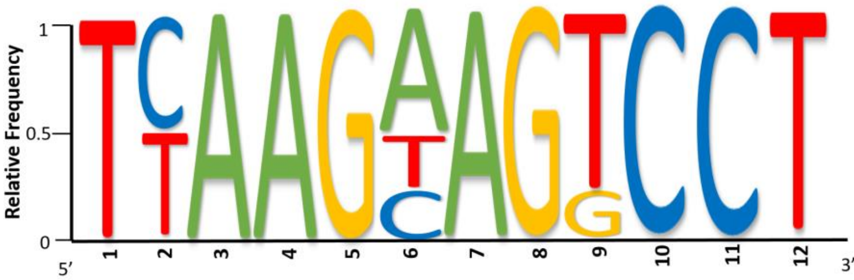
https://github.com/llSourcell/Kaggle_Earthquake_challenge

DEMO Kaggle_challenges/LANL_Earthquake_Challenge_DNN_1D_CNN_LSTM_with_TPU.ipynb

DNA Sequence Data Analysis

Working with DNA sequence data for ML
https://www.kaggle.com/thomasnelson/working-with-dna-sequence-data-for-ml





Time series analysis in cybersecurity

International Journal of Research in Engineering, Science and Management
Volume-3, Issue-7, July-2020
journals.resaim.com/ijresm | ISSN (Online): 2581-5792

Analyzing and Predicting Cyber Hacking with Time Series Models

C. Soundarya[1*], S. Usha[2]
[1]*PG Scholar, Department of Computer Science and Engineering, Anna University, BIT-Campus, Tiruchirappalli, India*
[2]*Assistant Professor, Department of Computer Science and Engineering, Anna University, BIT-Campus, Tiruchirappalli, India*
*Corresponding author: csoundarya1995@gmail.com

A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems

January 2018 · International Journal of Engineering & Technology 7(3):479-482

Authors:

Ranjit Panigrahi
Sikkim Manipal Institute of Technology

Samarjeet Borah
Sikkim Manipal Institute of Technology

https://www.researchgate.net/publication/329045441_A_detailed_analysis_of_CICIDS2017_dataset_for_designing_Intrusion_Detection_Systems

class imbalance problem

Table 2: Overall characteristics of CICIDS2017 dataset

Dataset Name	CICIDS2018
Dataset Type	Multi class
Year of release	2017
Total number of distinct instances	2830540
Number of features	83
Number of distinct classes	15

Table 3: Class wise instance occurrence of CICIDs2017 dataset

Class Labels	Number of instances
BENIGN	2359087
DoS Hulk	231072
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652

Table 4: Class prevalence ratio of CICIDS2017 dataset

Sl No	Normal / Attack Labels	Number of in-stances	% of preva-lence w.r.t. the majority class	% of preva-lence w.r.t. the total instances
1	BENIGN	2359087	1	83.34406
2	Bot	1966	0.000833	0.06946
3	DDoS	41835	0.017734	1.47799
4	DoS GoldenEye	10293	0.004363	0.36364
5	DoS Hulk	231072	0.09795	8.16353
6	DoS Slow-httptest	5499	0.002331	0.19427
7	DoS slowloris	5796	0.002457	0.20477
8	FTP-Patator	7938	0.003365	0.28044
9	Heartbleed	11	0.000005	0.00039
10	Infiltration	36	0.000015	0.00127
11	PortScan	158930	0.067369	5.61483
12	SSH-Patator	5897	0.0025	0.20833
13	Web Attack – Brute Force	1507	0.000639	0.05324
14	Web Attack – Sql Injection	21	0.000009	0.00074
15	Web Attack – XSS	652	0.000276	0.02303

Table 5: Characteristics of new attack labels with their prevalence rate in CICIDS2017 dataset

Sl No	*New Labels*	*Old Labels*	*Number of instances*	*% of prevalence w.r.t. the majority class*	*% of prevalence w.r.t. the total instances*
1	Normal	Benign	2359087	100	83.34
2	Botnet ARES	Bot	1966	0.083	0.06
3	Brute Force	FTP-Patator, SSH-Patator	13835	0.59	0.48
4	Dos/DDos	DDoS, DoS GoldenEye, DoS Hulk, DoS Slow-httptest, DoS slowloris, Heartbleed	294506	12.49	10.4
5	Infiltration	Infiltration	36	0.001	0.001
6	PortScan	PortScan	158930	6.74	5.61
7	Web Attack	Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS	2180	0.092	0.07

Sequence Data
的分析(2)
NLP

NLP Natural Language Processing 自然語言處理

https://zh.wikipedia.org/wiki/自然語言處理

- 自然語言處理（Natural Language Processing，NLP）是人工智慧和語言學領域的分支學科。
- 此領域探討如何處理及運用自然語言；自然語言處理包括多方面和步驟，基本有認知、理解、生成等部分。
- 自然語言認知和理解是讓電腦把輸入的語言變成有意思的符號和關係，然後根據目的再處理。**自然語言生成系統**則是把電腦資料轉化為自然語言。

http://research.sinica.edu.tw/nlp-natural-language-processing-chinese-knowledge-information/

Browse SoTA > Natural Language Processing

Natural Language Processing

1700 benchmarks • 517 tasks • 1498 datasets • 15621 papers with code

https://paperswithcode.com/area/natural-language-processing

Language Modelling

Task	Benchmarks	Papers with code
Language Modelling	364 benchmarks	1950 papers with code
Speech Recognition	302 benchmarks	698 papers with code
Long-range modeling	2 benchmarks	14 papers with code
Sentence Pair Modeling	7 benchmarks	5 papers with code
Cross-Document Language Modeling	2 benchmarks	1 papers with code

2D Classification

Task	Benchmarks	Papers with code
Language Modelling	364 benchmarks	1950 papers with code
Neural Network Compression	2 benchmarks	58 papers with code
Music Source Separation	3 benchmarks	38 papers with code

Question Answering

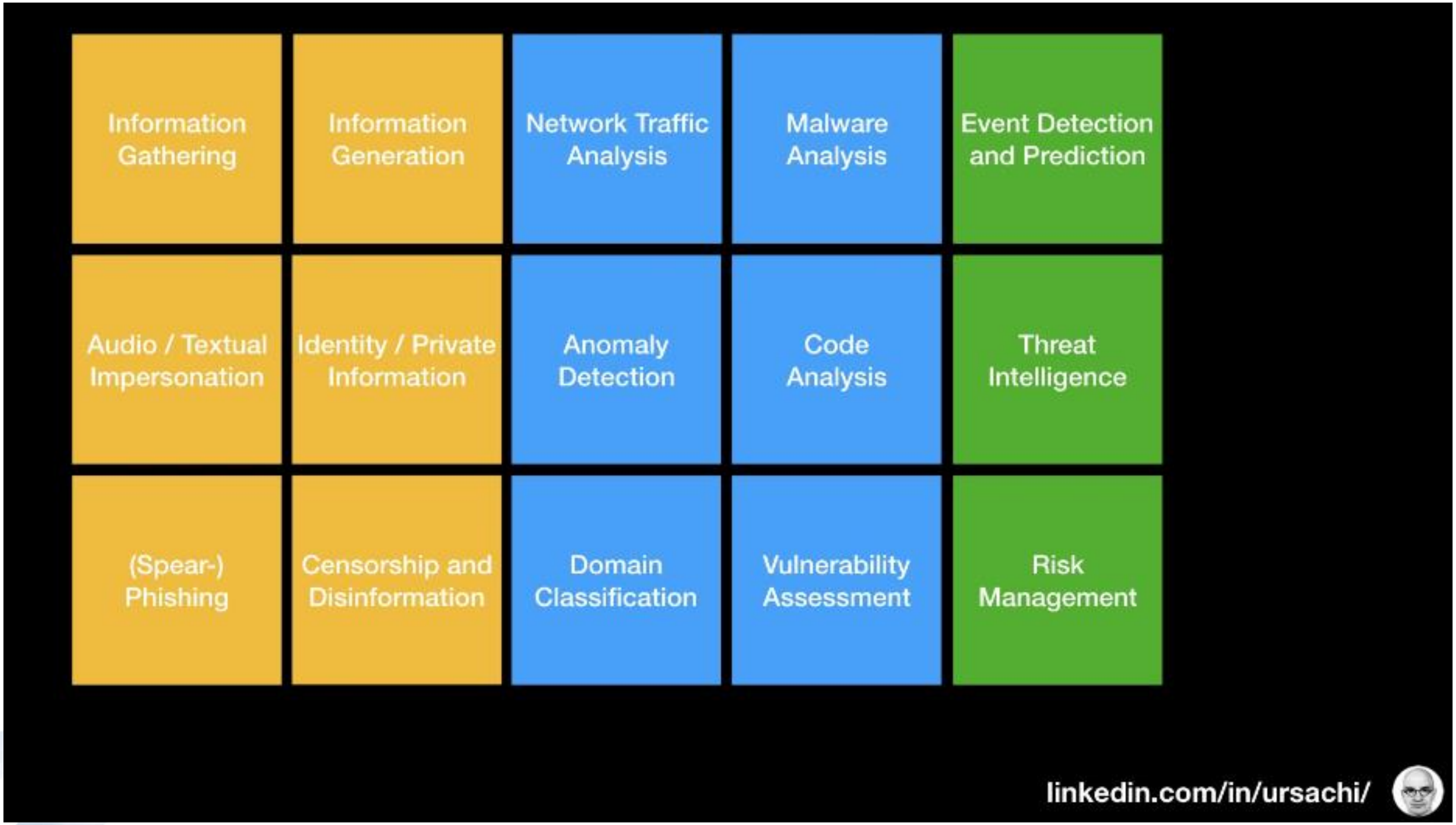
Task	Benchmarks	Papers with code
Question Answering	147 benchmarks	1608 papers with code
Open-Domain Question Answering	14 benchmarks	116 papers with code
Community Question Answering	2 benchmarks	39 papers with code
Answer Selection	3 benchmarks	38 papers with code
Conversational Question Answering		36 papers with code

See all 14 tasks

Machine Translation

Task	Benchmarks	Papers with code
Machine Translation	74 benchmarks	1581 papers with code
Transliteration		32 papers with code
Unsupervised Machine Translation	9 benchmarks	27 papers with code
Bilingual Lexicon Induction		26 papers with code
Automatic Post-Editing		24 papers with code

See all 9 tasks



https://www.masernet.com/project/role-and-applications-of-nlp-in-cybersecurity

RNN for.....

列到序列翻譯（sequence to sequence translation），包括將語音轉為文字或翻譯不同語言

Sequential patterns

Text

Speech

Audio

Video

Physical processes

Anything embedded in time (almost everything)

https://brohrer.mcknote.com/zh-Hant/how_machine_learning_works/how_rnns_lstm_work.html

NLP應用

文字朗讀（Text to speech）/語音合成（Speech synthesis）
語音識別（Speech recognition）
中文自動分詞（Chinese word segmentation）
詞性標註（Part-of-speech tagging）
句法分析（Parsing）
自然語言生成（Natural language generation）
文字分類（Text categorization）
資訊檢索（Information retrieval）
資訊抽取（Information extraction）
文字校對（Text-proofing）
問答系統（Question answering）
機器翻譯（Machine translation）
自動摘要（Automatic summarization）
文字蘊涵（Textual entailment）
命名實體辨識（Named entity recognition）


SPAM
偵測詐騙郵件
這部電影太糟了
Negative(負雷)
情緒分析
Google
台灣
搜尋建議更正
John saw the saw.
PN V D N
詞類標示
“Machine learing...”
機器學習
機器翻譯
大家好……
語音辨識
這位是新垣結衣
人名辨識擷取
W
summry
摘要文本大綱
S
NP VP
NP CC NP VV NP
NN NN 及 NP NP 與 NN
機器 學習 PN NN 結構化
其 深層
句法分析

Chinese Word Segmentation中文分詞

Chinese word segmentation is the task of splitting Chinese text (a sequence of Chinese characters) into words.

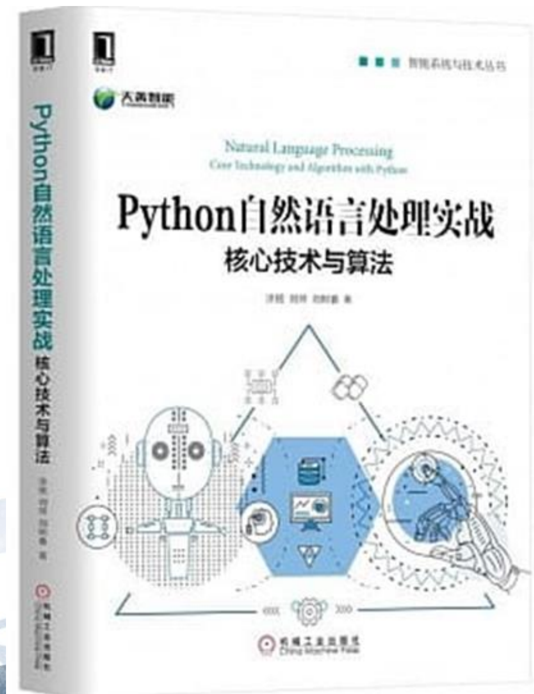
'上海浦東開發與建設同步' → ['上海', '浦東', '開發', ‘與', ’建設', '同步']

♠ marks the system that uses character unigram as input. ♣ marks the system that uses character bigram as input.

- Huang et al. (2019): BERT + model compression + multi-criterial learing ♠
- Yang et al. (2018): Lattice LSTM-CRF + BPE subword embeddings ♠♣
- Ma et al. (2018): BiLSTM-CRF + hyper-params search♠♣
- Yang et al. (2017): Transition-based + Beam-search + Rich pretrain♠♣
- Zhou et al. (2017): Greedy Search + word context♠
- Chen et al. (2017): BiLSTM-CRF + adv. loss♠♣
- Cai et al. (2017): Greedy Search+Span representation♠
- Kurita et al. (2017): Transition-based + Joint model♠
- Liu et al. (2016): neural semi-CRF♠
- Cai and Zhao (2016): Greedy Search♠
- Chen et al. (2015a): Gated Recursive NN♠♣
- Chen et al. (2015b): BiLSTM-CRF♠♣

https://github.com/sebastianruder/NLP-progress/blob/master/chinese/chinese_word_segmentation.md

Chinese Word Segmentation中文分詞



chapter-3 中文分詞技術
chapter-4 詞性標注與命名實體識別
chapter-5 關鍵字提取
chapter-6 句法分析

chapter-7 文本向量化
chapter-8 情感分析
chapter-9 NLP中用到的機器學習演算法
chapter-10 基於深度學習的NLP演算法

https://github.com/nlpinaction/learning-nlp

Chinese Word Segmentation中文分詞

https://github.com/fxsjy/jieba

支持繁體分詞
支持自訂字典
MIT 授權協議

支持三種分詞模式：

全模式，把句子中所有的可以成詞的詞語都掃描出來, 速度非常快，但是不能解決歧義；

精確模式，試圖將句子最精確地切開，適合文本分析；

搜索引擎模式，在精確模式的基礎上，對長詞再次切分，提高召回率，適合用於搜尋引擎分詞。

jieba
“結巴”中文分詞：做最好的 Python 中文分片語件
"Jieba" (Chinese for "to stutter") Chinese text segmentation: built to be the best Python Chinese word segmentation module.

```
import jieba

sent = '中文分詞是文本處理不可或缺的一步！'

seg_list = jieba.cut(sent, cut_all=True)
print('全模式：', '/ ' .join(seg_list))

seg_list = jieba.cut(sent, cut_all=False)
print('精確模式：', '/ '.join(seg_list))

seg_list = jieba.cut(sent)
print('預設精確模式：', '/ '.join(seg_list))

seg_list = jieba.cut_for_search(sent)
print('搜尋引擎模式', '/ '.join(seg_list))
```

```python
import jieba

sent = '中文分詞是文本處理不可或缺的一步！'

seg_list = jieba.cut(sent, cut_all=True)

print('全模式：', '/ ' .join(seg_list))

seg_list = jieba.cut(sent, cut_all=False)
print('精確模式：', '/ '.join(seg_list))

seg_list = jieba.cut(sent)
print('預設精確模式：', '/ '.join(seg_list))

seg_list = jieba.cut_for_search(sent)
print('搜尋引擎模式', '/ '.join(seg_list))

```

```
全模式： 中文/ 分/ 詞/ 是/ 文本/ 處/ 理/ 不可/ 不可或缺/ 或缺/ 的/ 一步/ /
精確模式： 中文/ 分詞/ 是/ 文本/ 處理/ 不可或缺/ 的/ 一步/ ！
預設精確模式： 中文/ 分詞/ 是/ 文本/ 處理/ 不可或缺/ 的/ 一步/ ！
搜尋引擎模式 中文/ 分詞/ 是/ 文本/ 處理/ 不可/ 或缺/ 不可或缺/ 的/ 一步/ ！
```

搜索引擎模式，在精確模式的基礎上，對長詞再次切分，提高召回率，適合用於搜尋引擎分詞。

詞性標註Part-of-speech tagging

http://terms.naer.edu.tw/detail/1678982/

字詞（**word**）是語言系統中具有獨立語意或扮演特定語法功能，且可以自由使用的最小語言單位。

依據字詞在句法結構或語言形態上扮演的角色，
經由詞性分類賦予語句中每個字詞適當之詞性符號或標記的過程，
則稱為詞性標記（**part-of-speech tagging**，或稱**POS tagging**）。

隨著語言的不同，詞性分類的方式也有所差異。
基本上可分為動詞（**verb**）、名詞（**noun**）、形容詞（**adjective**）、副詞（**adverb**），以及其他類別，例如：代名詞（**pronoun**）、介係詞（**preposition**）、連接詞（**conjunction**）或感嘆詞（**interjection**）

詞性標註Part-of-speech tagging

http://terms.naer.edu.tw/detail/1678982/

詞性標記的範例

Tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part of speech.

Tagging_VBG is_BEZ the_**AT** task_NN of_IN labeling_VBG (_(or_CC tagging_VBG)_) each_DT word_NN in_IN a_AT sentence_NN with_IN its_PP$ appropriate_JJ part_NN of_IN speech_NN ._.

VBG為動名詞或現在分詞，BEZ代表is，AT為冠詞，NN為名詞，IN為介係詞，CC為連接詞，DT為限定詞，PP$為所有格，JJ為形容詞。

詞性標註Part-of-speech tagging

!pip list | grep nltk

```
import nltk
from nltk import word_tokenize, pos_tag

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

print(pos_tag(word_tokenize("I'm learning NLP")))
# [('I', 'PRP'), ("'m", 'VBP'), ('learning', 'VBG'), ('NLP', 'NNP')]
```

The complete list of POS tags in nltk with examples
https://medium.com/@faisal-fida/the-complete-list-of-pos-tags-in-nltk-with-examples-eb0485f04321

pos_tag load the Standard treebank POS tagger

1. CC Coordinating conjunction
2. CD Cardinal number
3. DT Determiner
4. EX Existential there
5. FW Foreign word
6. IN Preposition or subordinating conjunction
7. JJ Adjective
8. JJR Adjective, comparative
9. JJS Adjective, superlative

……………………………

請參閱https://blog.csdn.net/fxjtoday/article/details/5841453

詞性標註Part-of-speech tagging

```
nltk.download()
```

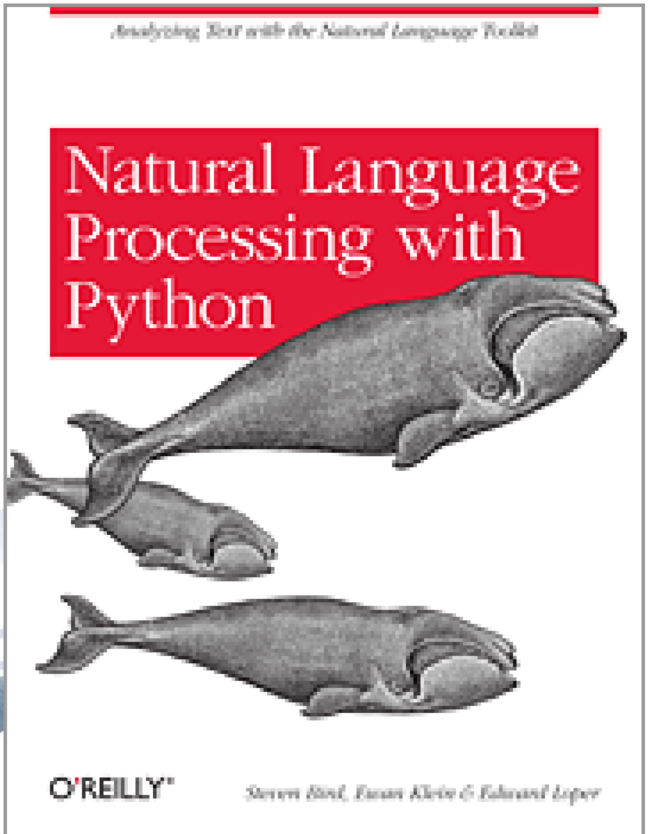
```
NLTK Downloader
---------------------------------------------------------------------------
    d) Download   l) List    u) Update   c) Config   h) Help   q) Quit
---------------------------------------------------------------------------
Downloader> l

Packages:
  [ ] abc................. Australian Broadcasting Commission 2006
  [ ] alpino.............. Alpino Dutch Treebank
  [*] averaged_perceptron_tagger Averaged Perceptron Tagger
  [ ] averaged_perceptron_tagger_ru Averaged Perceptron Tagger (Russian)
  [ ] basque_grammars..... Grammars for Basque
  [ ] biocreative_ppi..... BioCreAtIvE (Critical Assessment of Information
                           Extraction Systems in Biology)
  [ ] bllip_wsj_no_aux.... BLLIP Parser: WSJ Model
  [ ] book_grammars....... Grammars from NLTK Book
  [ ] brown............... Brown Corpus
  [ ] brown_tei........... Brown Corpus (TEI XML Version)
  [ ] cess_cat............ CESS-CAT Treebank
  [ ] cess_esp............ CESS-ESP Treebank
```

詞性標註Part-of-speech tagging

Complete guide for training your own Part-Of-Speech Tagger

https://nlpforhackers.io/training-pos-tagger/

詞性標註Part-of-speech tagging



http://www.nltk.org/book/

Natural Language Processing with Python

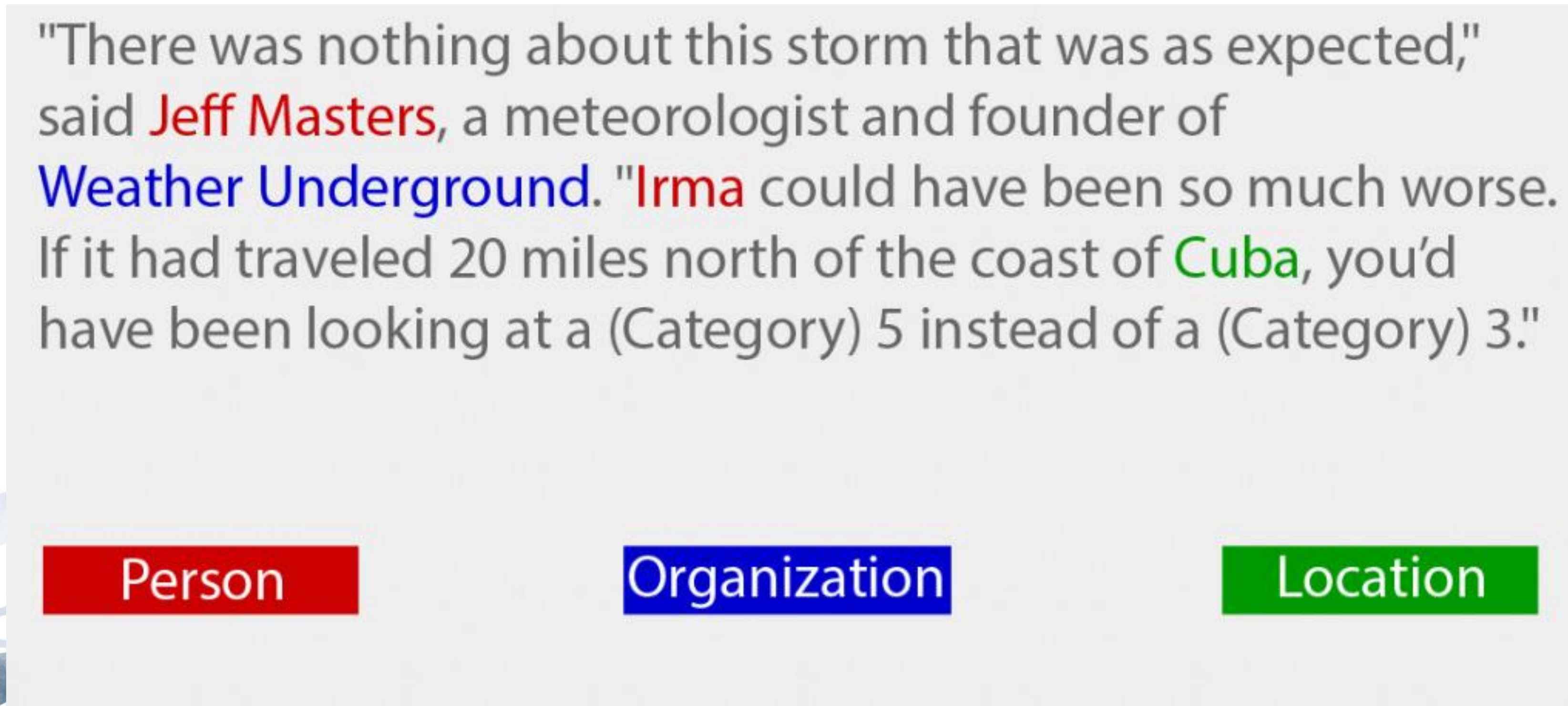
– Analyzing Text with the Natural Language Toolkit

Steven Bird, Ewan Klein, and Edward Loper

This version of the NLTK book is updated for Python 3 and NLTK 3. The first edition of the book, published by O'Reilly, is available at http://nltk.org/book_1ed/. (There are currently no plans for a second edition of the book.)

命名實體識別(Named Entity Recognition, NER)

在句子的序列中，定位並識別人名、地名、機構名等任務。



圖片來源：https://blog.paralleldots.com/data-science/named-entity-recognition-milestone-models-papers-and-technologies/

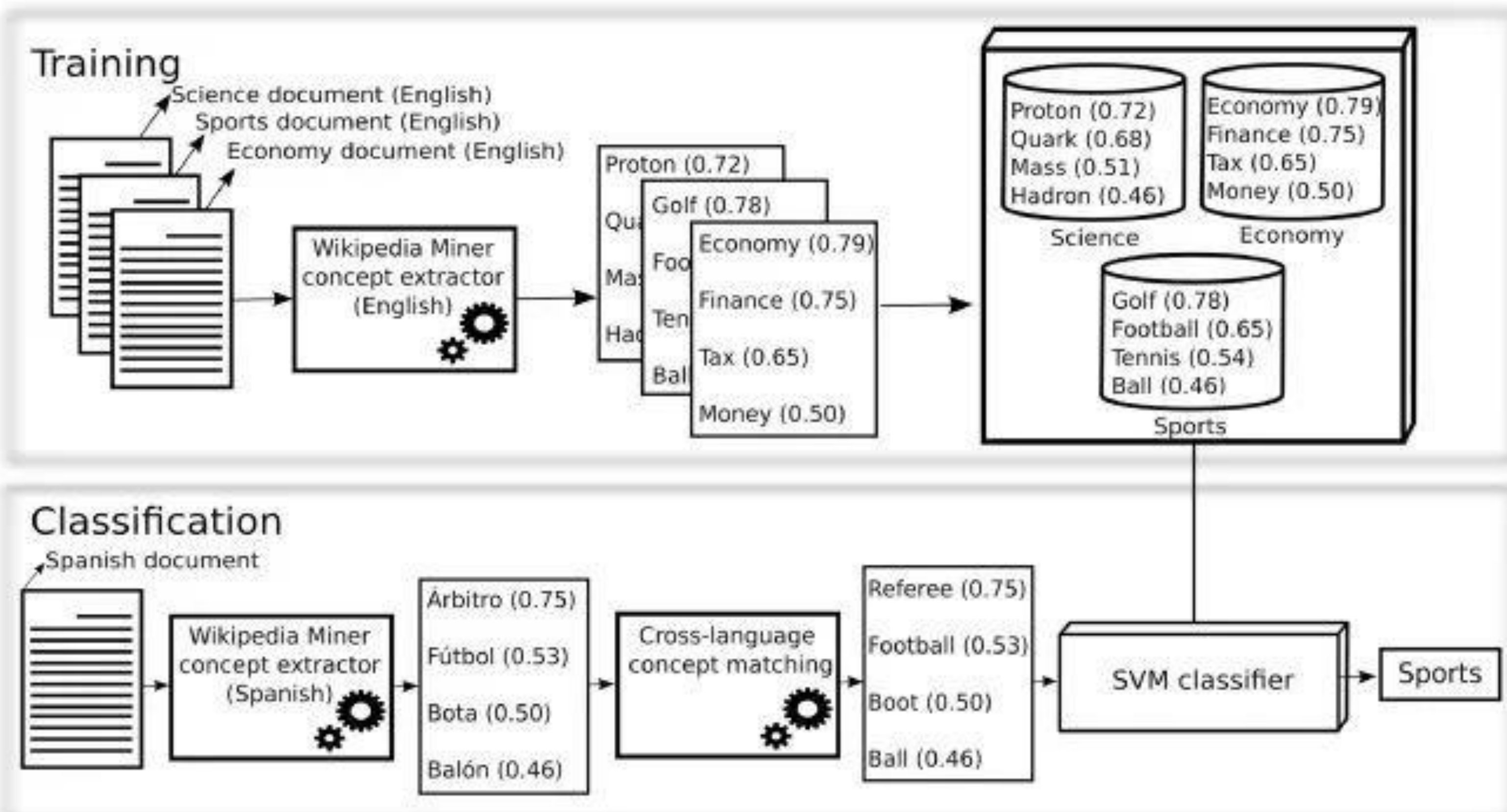
https://ithelp.ithome.com.tw/articles/10209418

Text classification 文本分類

1. 詞嵌入向量化：word2vec, FastText等等
2. 卷積神經網路特徵提取：Text-CNN, Char-CNN等等
3. 上下文機制：Text-RNN， BiRNN， RCNN等等
4. 記憶存儲機制：EntNet， DMN等等
5. 注意力機制：HAN等等

text classification 文本分類

傳統機器學習時代的主流分類器:Naive Bayes，Maximum Entropy， K-NN，和SVM。
經典特徵模型：經典距離定義模型Vector Space Model（Rocchio）， N-grams等等。



Text classification 文本分類

1. 詞嵌入向量化：word2vec, FastText等等
2. 卷積神經網路特徵提取：Text-CNN, Char-CNN等等
3. 上下文機制：Text-RNN， BiRNN， RCNN等等
4. 記憶存儲機制：EntNet， DMN等等
5. 注意力機制：HAN等等

Question-Answering system 智能問答

Question-Answering Systems

- Systems that answer questions asked in natural language
- Goal-oriented (to complete one or more tasks):
  - Obtain information/facts
  - Seek recommendations
  - Make an appointment
  - Ask for help
  - ...

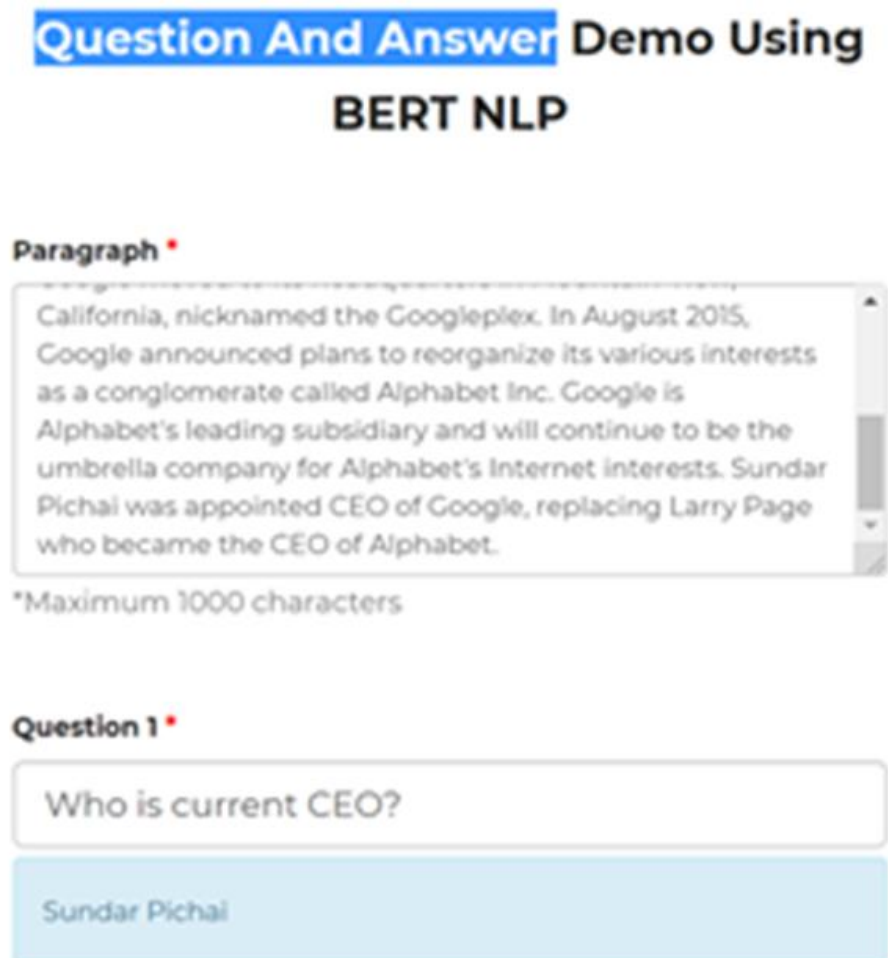
[AAAI 2019 tutorial] End-to-end goal-oriented question answering systems
https://www.slideshare.net/QiHe2/aaai-2019-tutorial-endtoend-goaloriented-question-answering-systems

Question-Answering system 智能問答

https://github.com/l11x0m7/book-of-qna-code

智能問答與深度學習
王海良, 等 電子工業出版社
2018-11-01

https://www.tenlong.com.tw/products/9787121349218



https://www.pragnakalp.com/demos/BERT-NLP-QnA-Demo/

NLP Transfer Learning 遷移學習

面向文本分類的通用語言模型微調
Universal Language Model Fine-tuning for Text Classification
Jeremy Howard & Sebastian Ruder
https://arxiv.org/abs/1801.06146

https://kknews.cc/code/6qqggj3.html
https://medium.com/mlreview/understanding-building-blocks-of-ulmfit-818d3775325b

Inductive transfer learning has greatly impacted computer vision, but existing approaches in NLP still require task-specific modifications and training from scratch.

We propose **Universal Language Model Fine-tuning (ULMFiT),** an effective transfer learning method that can be applied to any task in NLP, and introduce techniques that are key for fine-tuning a language model.

Our method significantly outperforms the state-of-the-art on six text classification tasks, reducing the error by 18-24% on the majority of datasets. Furthermore, with only 100 labeled examples, it matches the performance of training from scratch on 100x more data. **We open-source our pretrained models and code.**

機器翻譯 Machine Translation

Transformer(2017)

BERT(2018)

XLNet(2019)

ELMO, BERT, GPT
https://www.youtube.com/watch?v=UYPa347-DdE

Seq2Seq(2014)

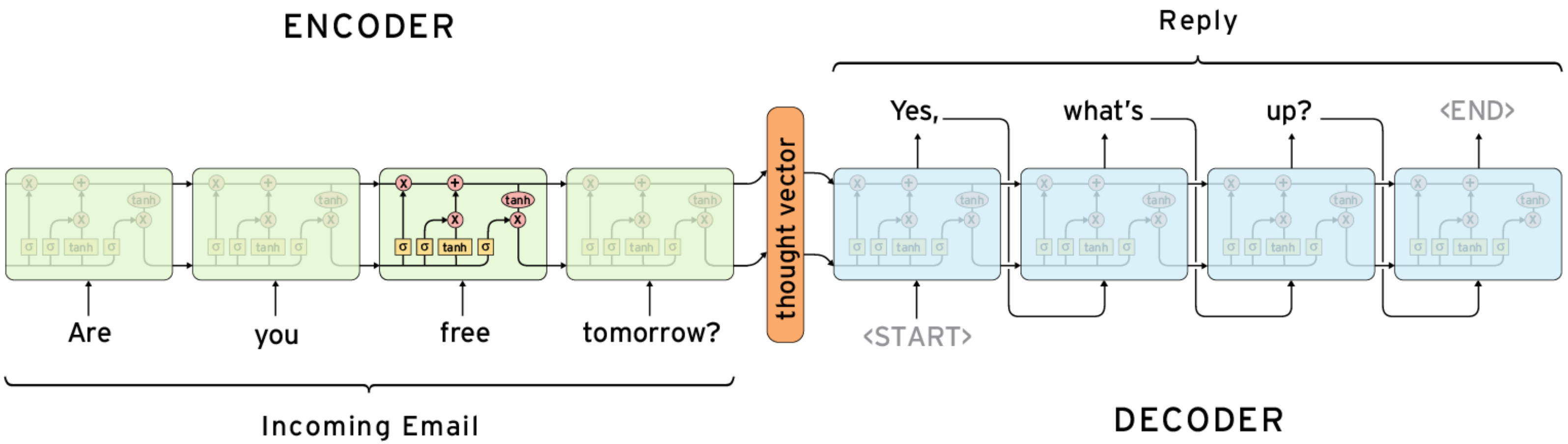
Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Sequence to Sequence
是由 Encoder 與 Decoder 兩個 RNN 構成

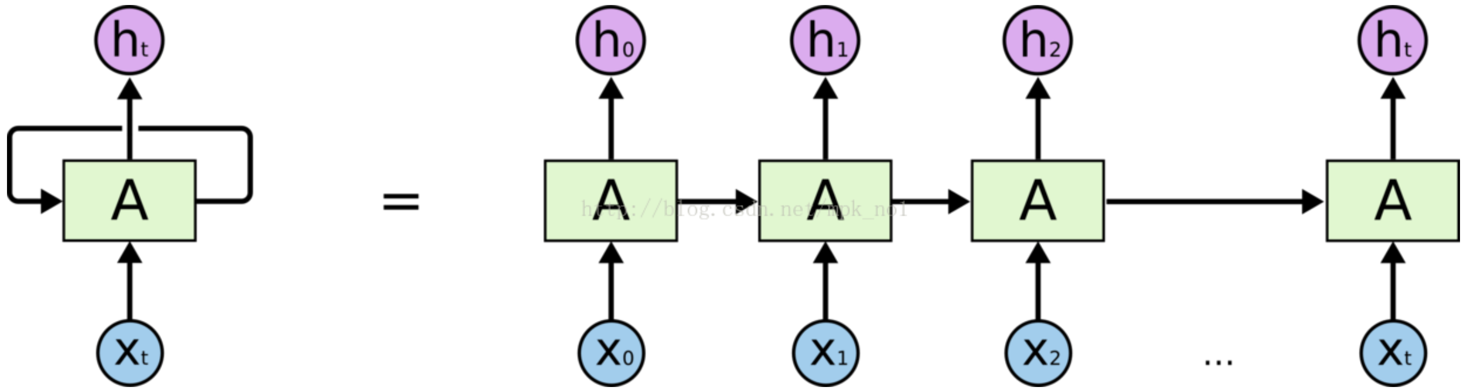


運作原理:當看到一段話時，會先將這句話理解吸收，再根據我們理解的內容說出回覆，Sequence to Sequence 就是在模擬這個過程。

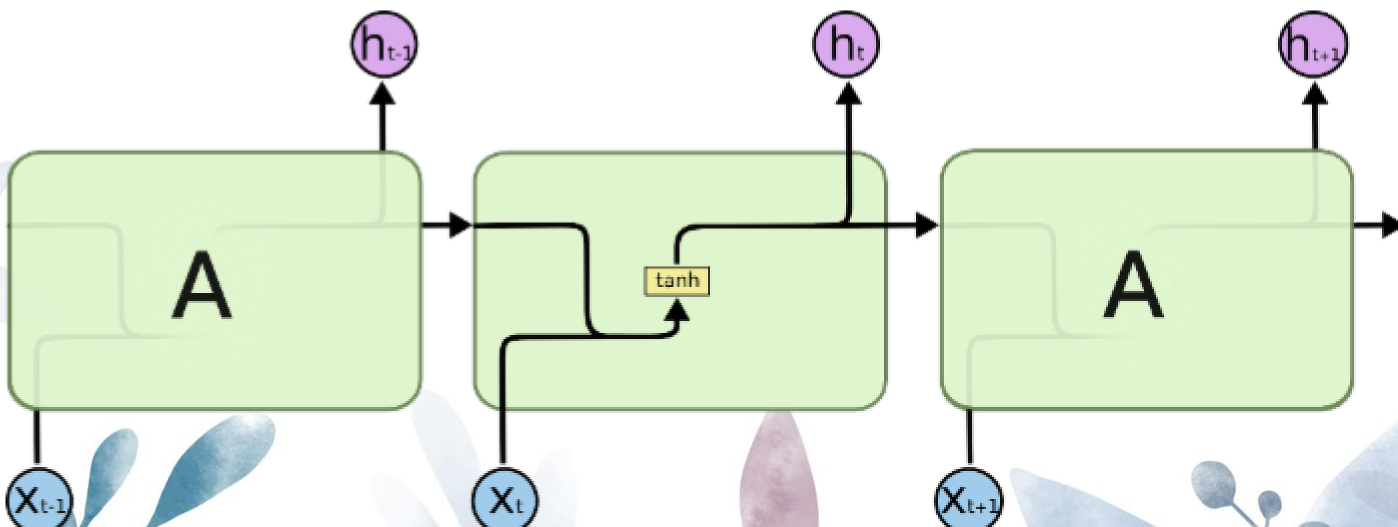
https://zake7749.github.io/2017/09/28/Sequence-to-Sequence-tutorial/#Sequence_to_Sequence

遞歸神經網絡
Recurrent
Neural Network
模型

SimpleRNN

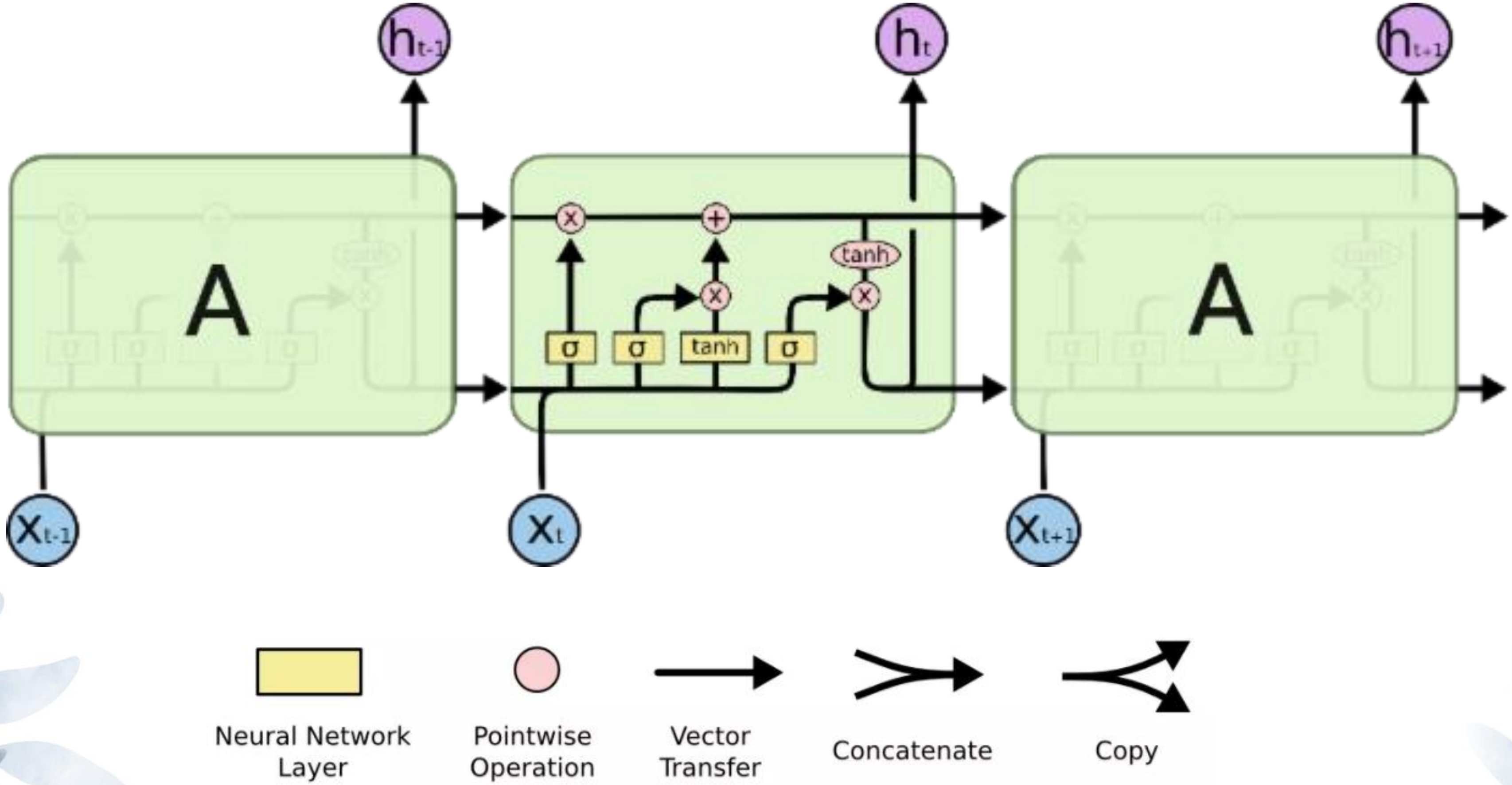


所有迴圈神經網路結構都是由完全相同結構的（神經網路）模組進行複製而成的。
在普通的RNNs 中，這個模組結構非常簡單，比如僅是一個單一的 tanh 層



LSTM

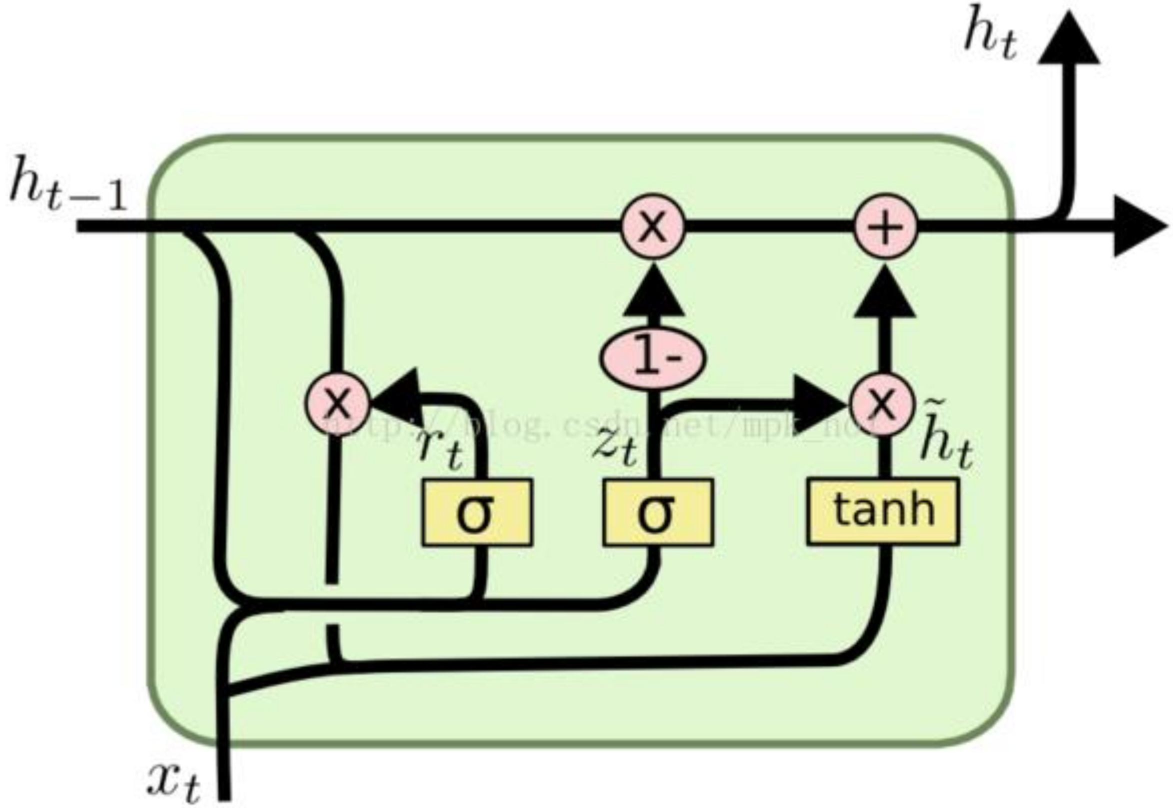
LSTMs 也有類似的結構,但是它們不再只是用一個單一的 tanh 層，而是用了四個相互作用的層



https://brohrer.mcknote.com/zh-Hant/how_machine_learning_works/how_rnns_lstm_work.html

https://blog.csdn.net/fendouaini/article/details/80198994

GRU



https://blog.csdn.net/mpk_no1/article/details/72875185 colah's blog   Blog   About   Contact

Understanding LSTM Networks

Posted on August 27, 2015

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://blog.csdn.net/menc15/article/details/71271566

https://blog.csdn.net/Jerr__y/article/details/58598296

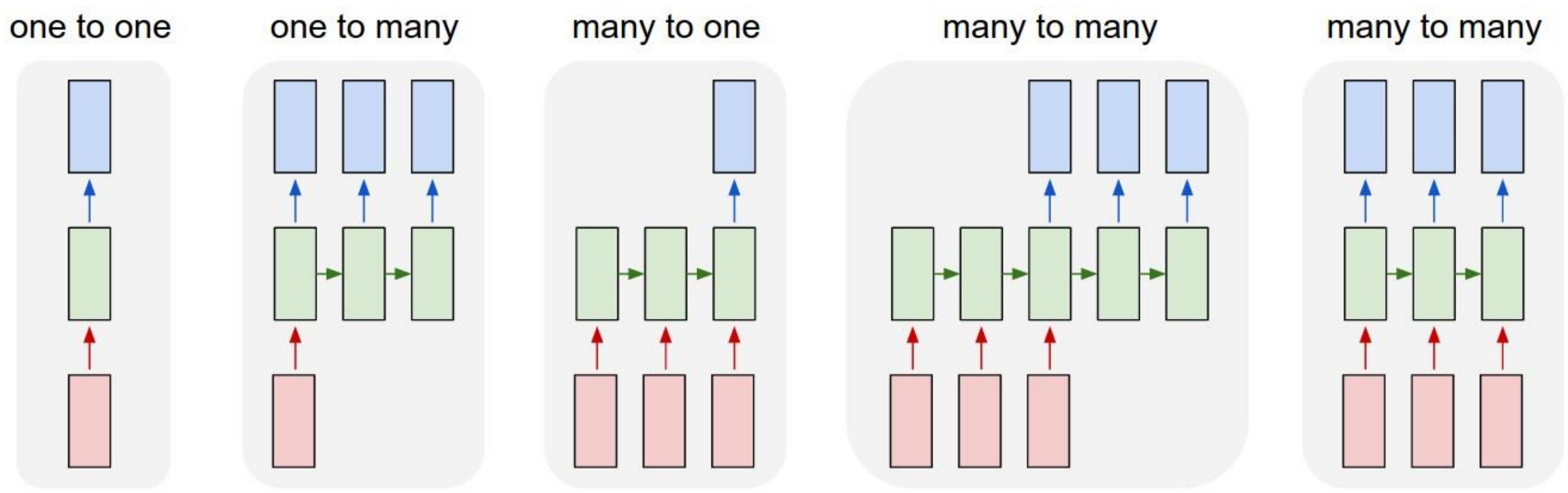
Andrej Karpathy blog   About   Hacker's guide to Neural Networks

The Unreasonable Effectiveness of Recurrent Neural Networks

May 21, 2015

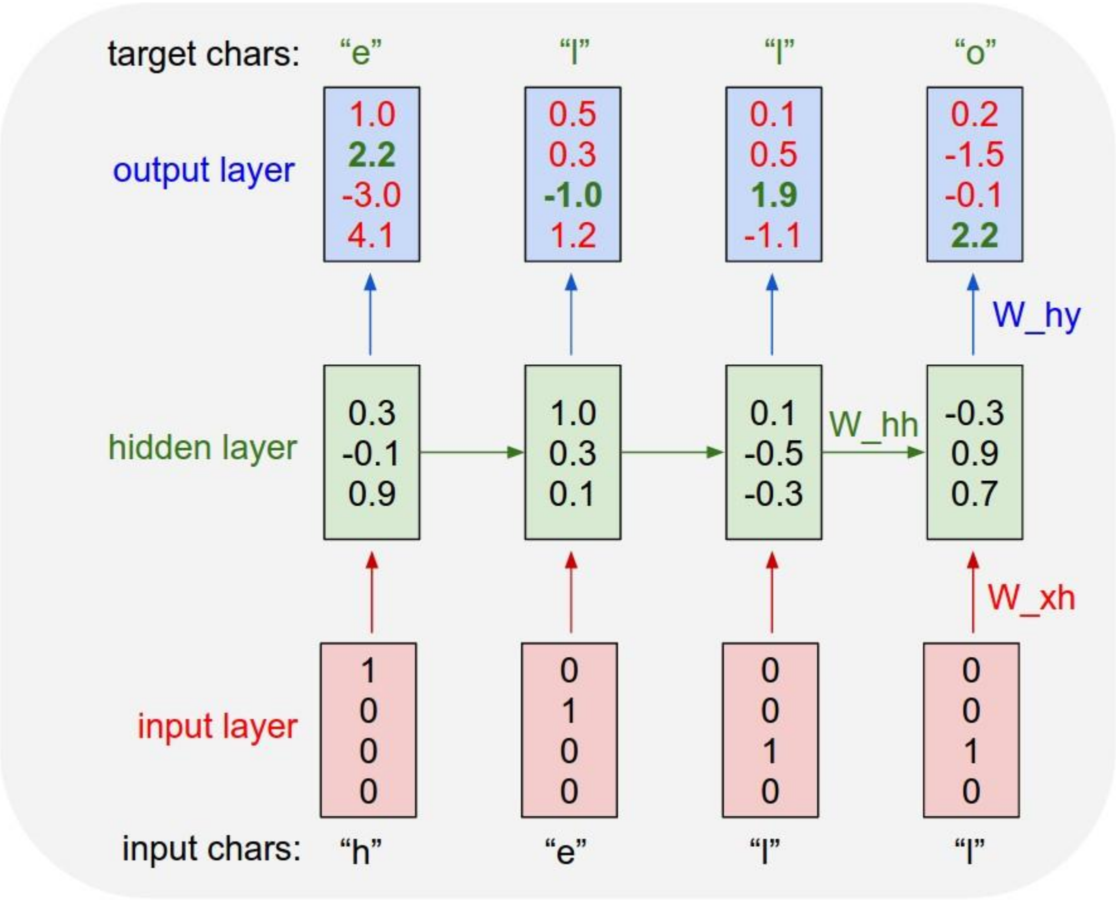
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

https://github.com/karpathy/char-rnn

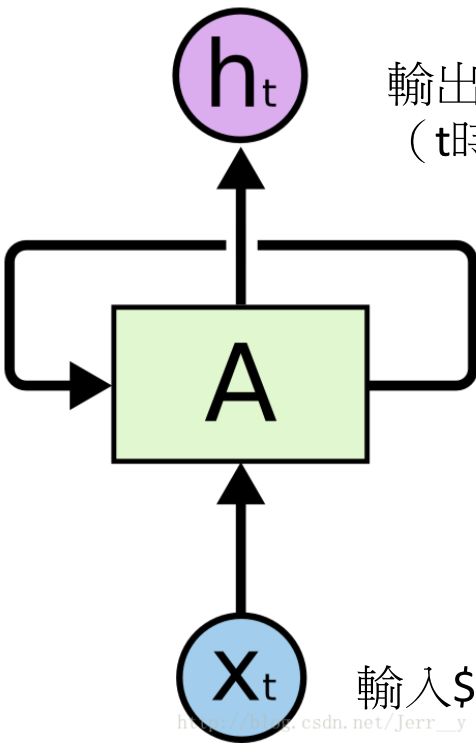


http://karpathy.github.io/2015/05/21/rnn-effectiveness/

An example RNN with 4-dimensional input and output layers, and a hidden layer of 3 units (neurons)



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



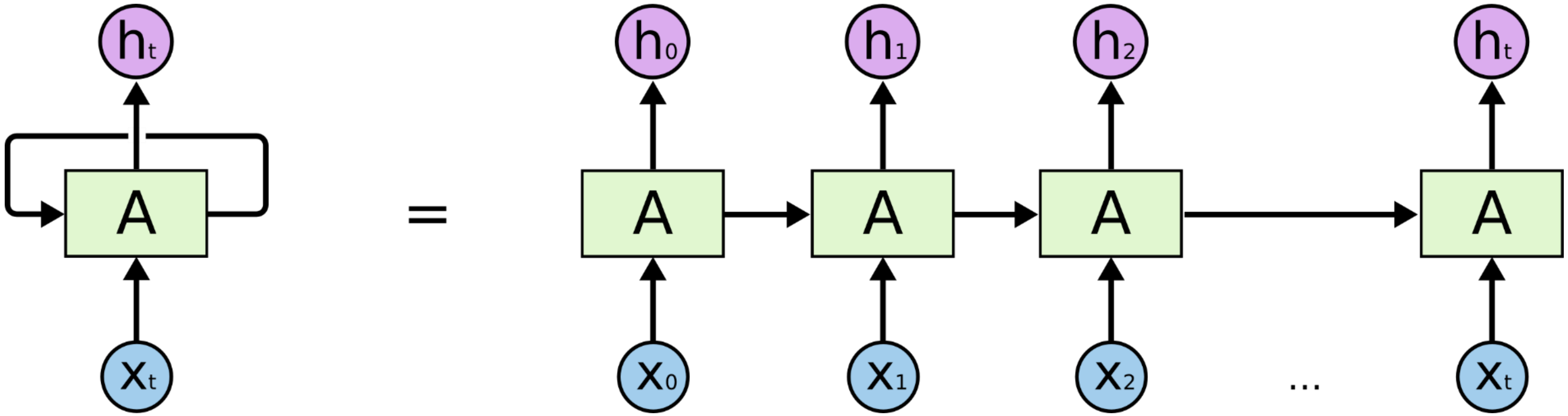
輸出一個結果 h_t $
（t時刻的狀態或者輸出）

網路中的迴圈結構使得某個時刻的狀態能夠傳到下一個時刻

A loop allows information to be passed from one step of the network to the next

輸入$ x_t （t時刻的特徵向量）

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.



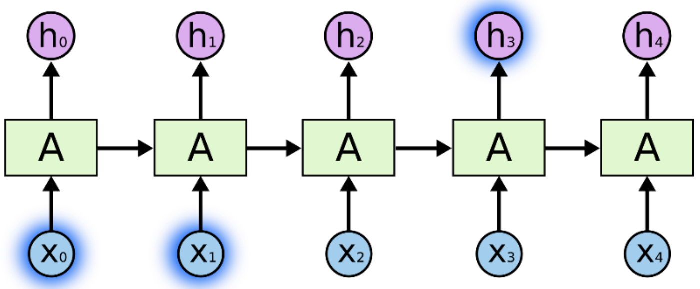
This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists.

They’re the natural architecture of neural network to use for such data.

The Problem of Long-Term Dependencies

長時期依賴的問題

consider a language model trying to predict the next word based on the previous ones.



the clouds are in the ?

the clouds are in the sky

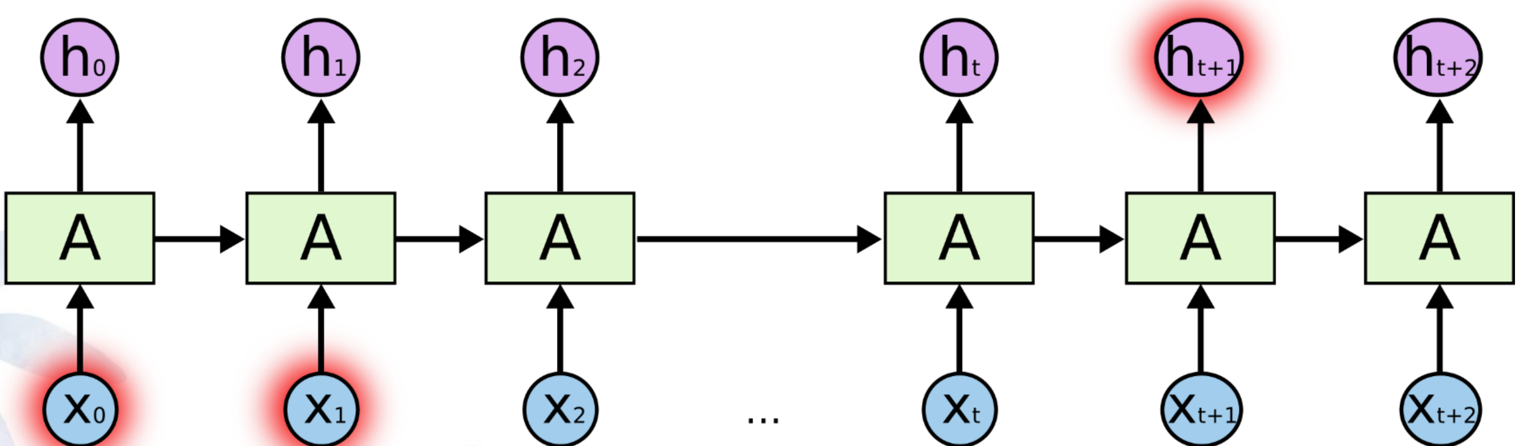
不需要更多的資訊，我們就能夠自然而然的想到下一個詞應該是 “sky”。

所要預測的內容和相關資訊之間的間隔很小，這種情況下 RNNs 就能夠利用過去的資訊， 很容易的實現

長時期依賴的問題

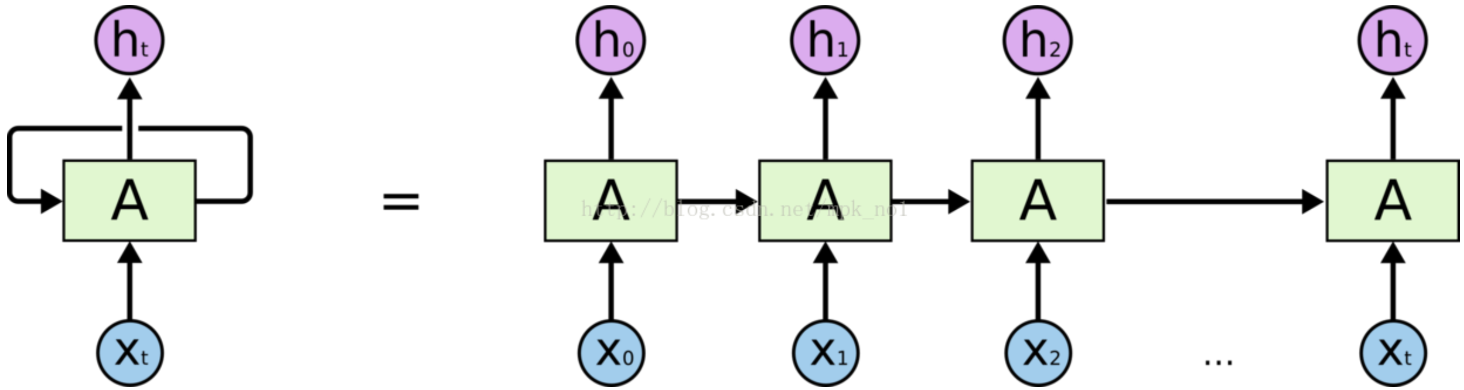
預測 “I grew up in France … (此處省略1萬字)… I speak ?”

“I grew up in France… I speak fluent **French**.”

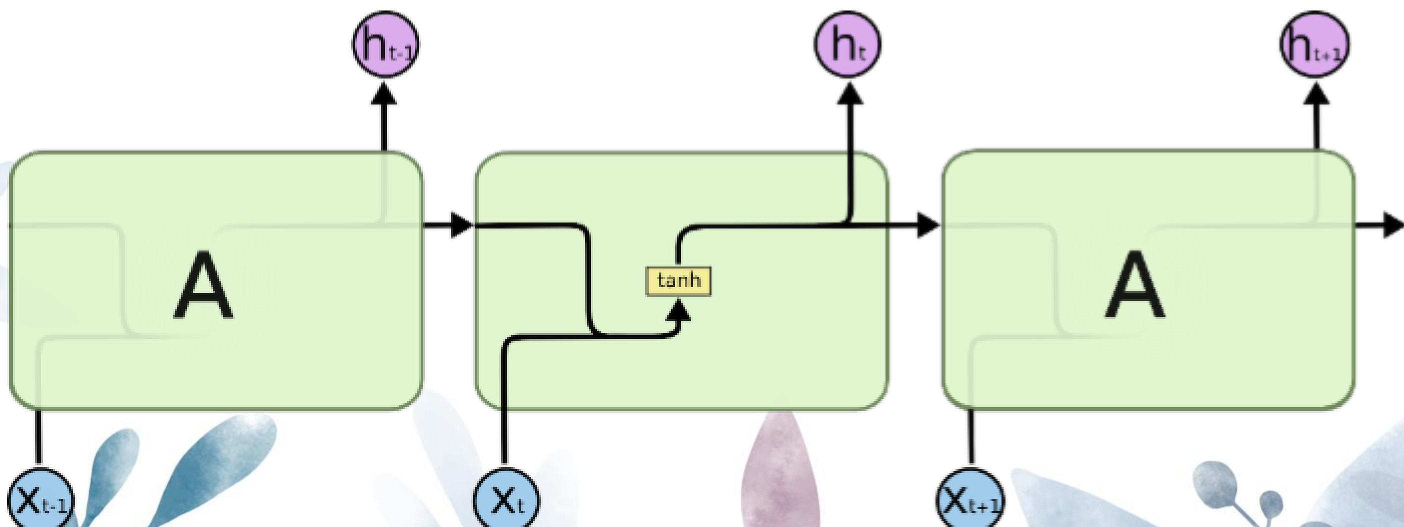


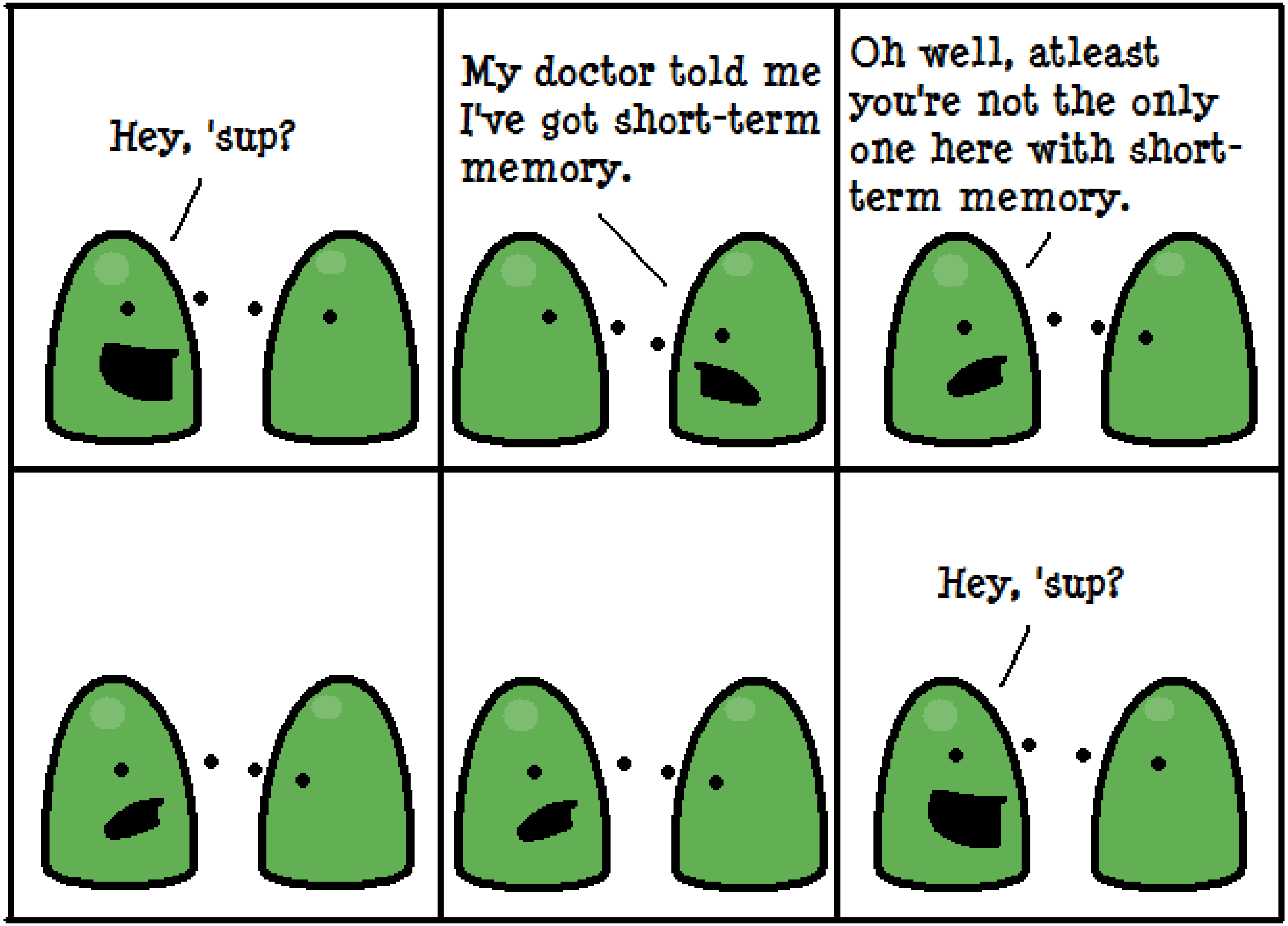
預測資訊和相關資訊間的間隔增大， RNNs 很難去把它們關聯起來

SimpleRNN



所有迴圈神經網路結構都是由完全相同結構的（神經網路）模組進行複製而成的。在普通的RNNs 中，這個模組結構非常簡單，比如僅是一個單一的 tanh 層





https://www.deviantart.com/toxicpaprika/art/Short-term-memory-88267010

LONG SHORT-TERM MEMORY

NEURAL COMPUTATION 9(8):1735–1780, 1997

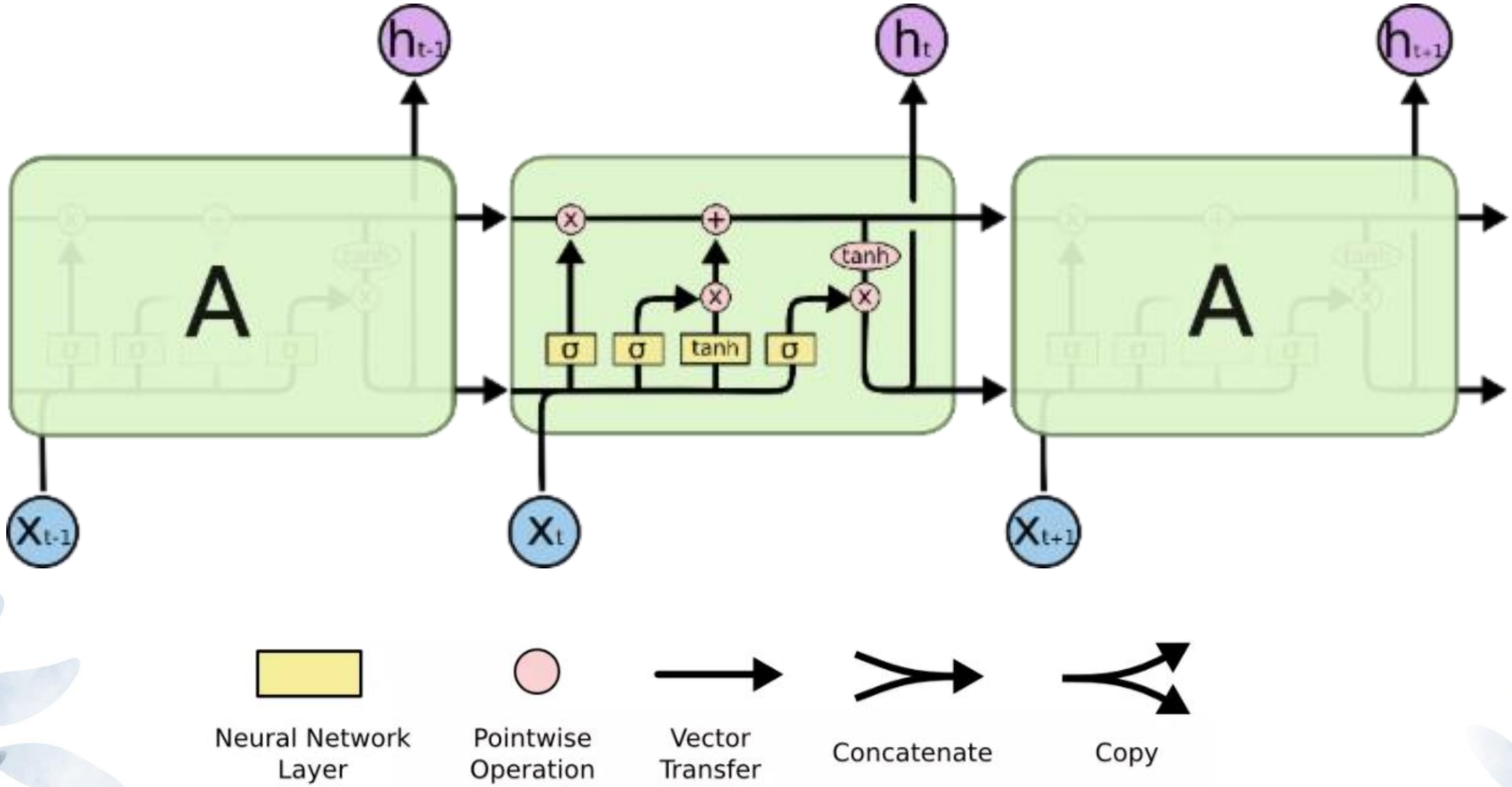
Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen

設計初衷是希望能夠解決神經網路中的長期依賴問題，讓記住長期資訊成為神經網路的預設行為，而不是需要很大力氣才能學會。

LSTM

LSTMs 也有類似的結構,但是它們不再只是用一個單一的 tanh 層，而是用了四個相互作用的層



https://brohrer.mcknote.com/zh-Hant/how_machine_learning_works/how_rnns_lstm_work.html

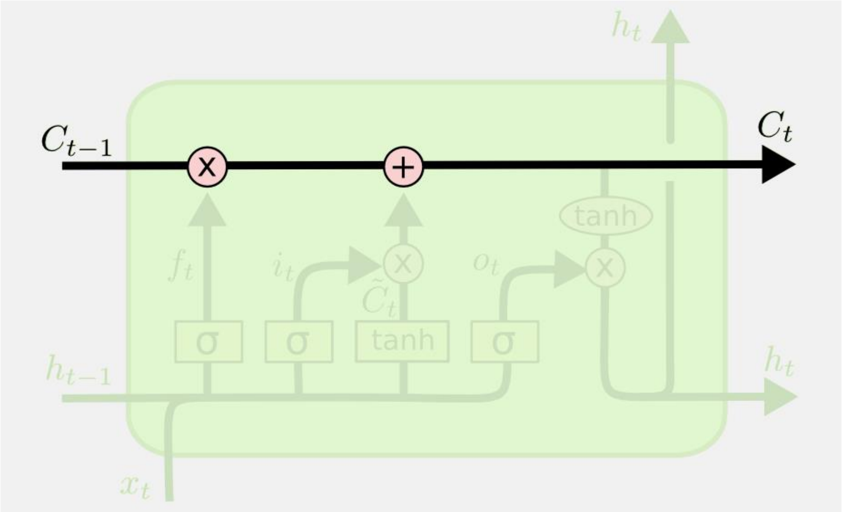
https://blog.csdn.net/fendouaini/article/details/80198994

cell state **長時期的記憶保留**

LSTMs 最關鍵的地方在於 cell 狀態 和 結構圖上面的那條橫穿的水平線。

cell 狀態的傳輸就像一條傳送帶， 向量從整個 cell 中穿過，只是做了少量的線性操作。這種結構能夠很輕鬆地實現資訊從整個 cell 中穿過而不做改變。（這樣就可以實現長時期的記憶保留）

傳送帶結構



使用Gate來添加或者刪除資訊

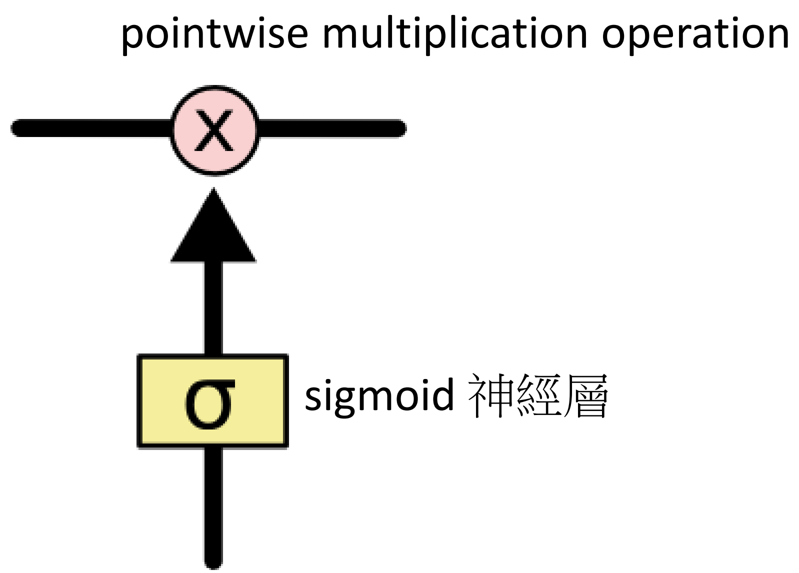
每個 LSTM 有三個這樣的門結構，來實現保護和控制資訊。
（分別是 "forget gate layer", 遺忘門； "input gate layer"，傳入門；
"output gate layer", 輸出門）

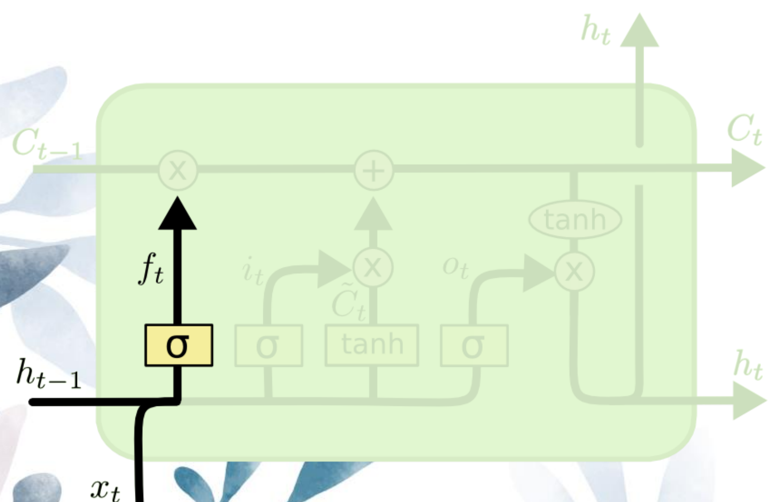
遺忘門 (forget gates)

決定要讓那些資訊繼續通過這個 cell?

sigmoid 神經層
輸出是一個數值都在 0，1 之間的向量（向量長度和 cell 的狀態 $ C_{t-1} $ 一樣），表示讓 $C_{t-1} $ 的各部分資訊通過的比重。

0 表示“不讓任何資訊通過”，
1 表示“讓所有資訊通過”。



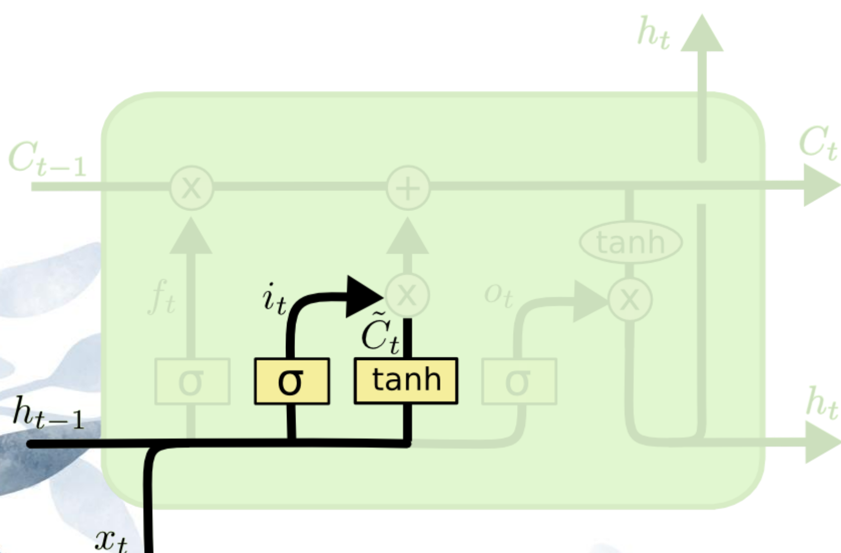


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

傳入門 (input gates)

決定讓多少新的資訊加入到 cell 狀態

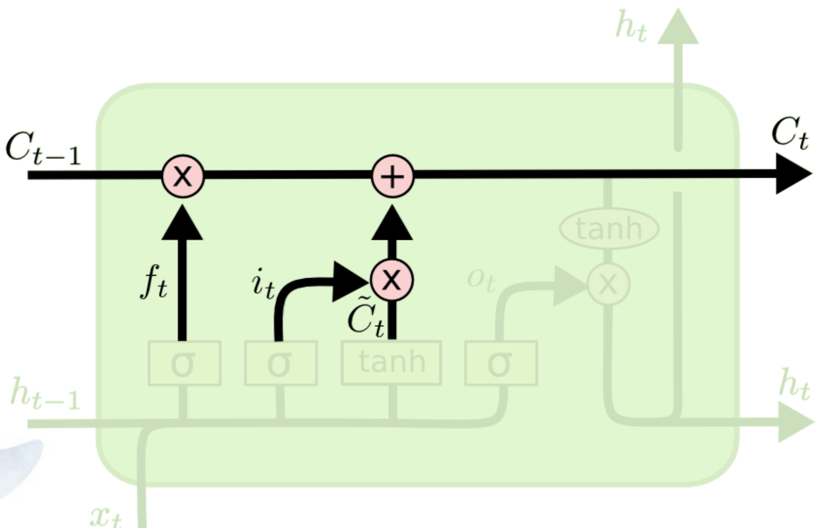
[1]a sigmoid layer called the “input gate layer” decides which values we’ll update.
[2]a tanh layer creates a vector of new candidate values, C~t, that could be added to the state



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

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

combine these two to create an update to the state



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

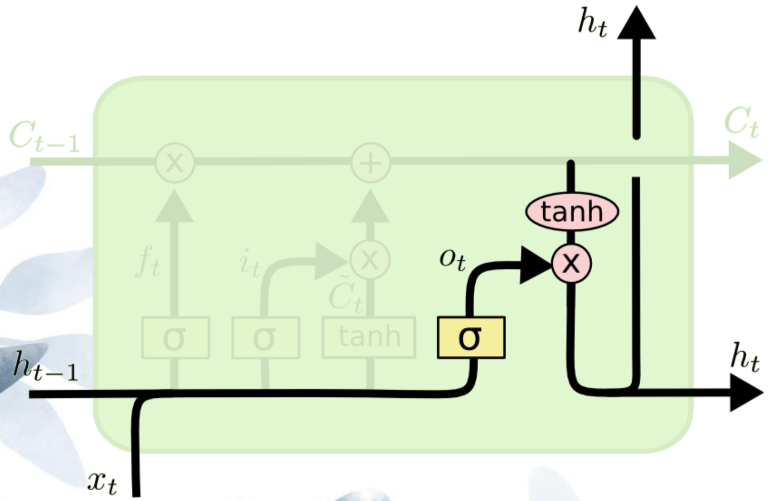
輸出門

決定輸出什麼值

This output will be based on our cell state, but will be a filtered version

First, we run a sigmoid layer which decides what parts of the cell state we're going to output.

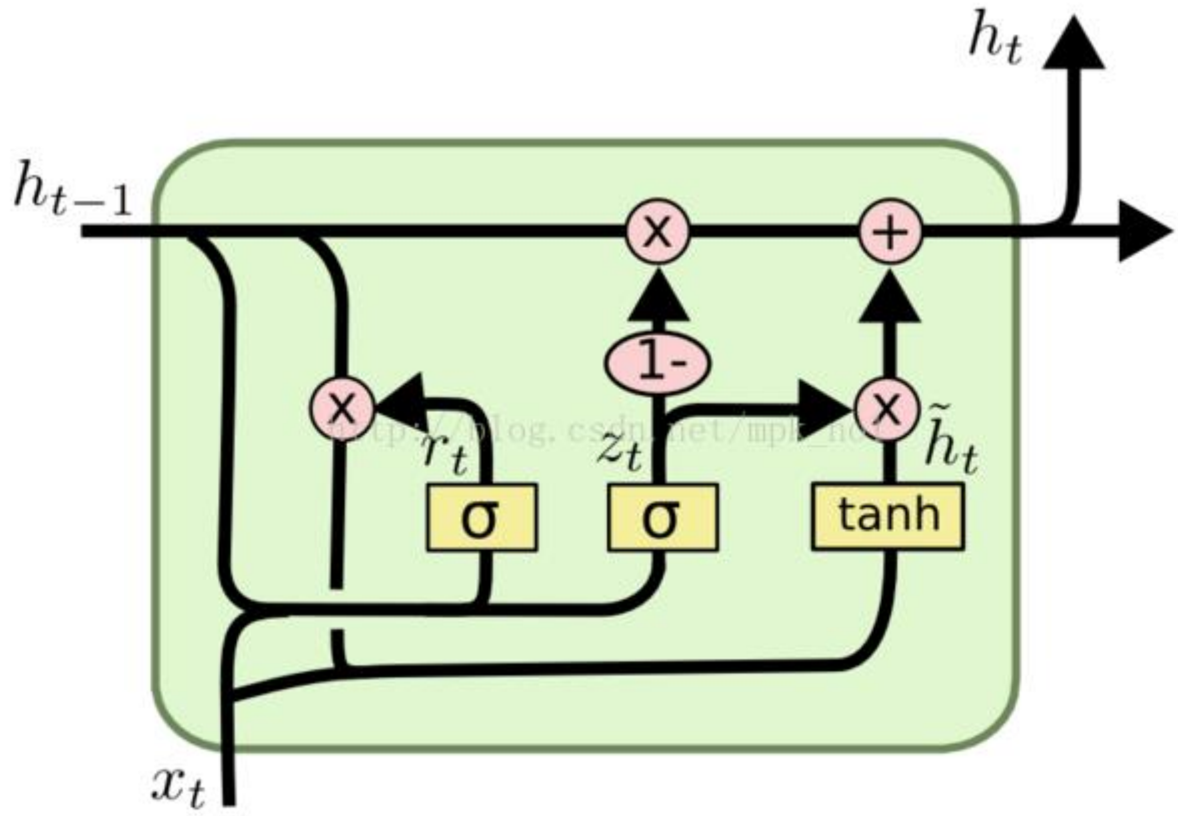
Then, we put the cell state through tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.



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

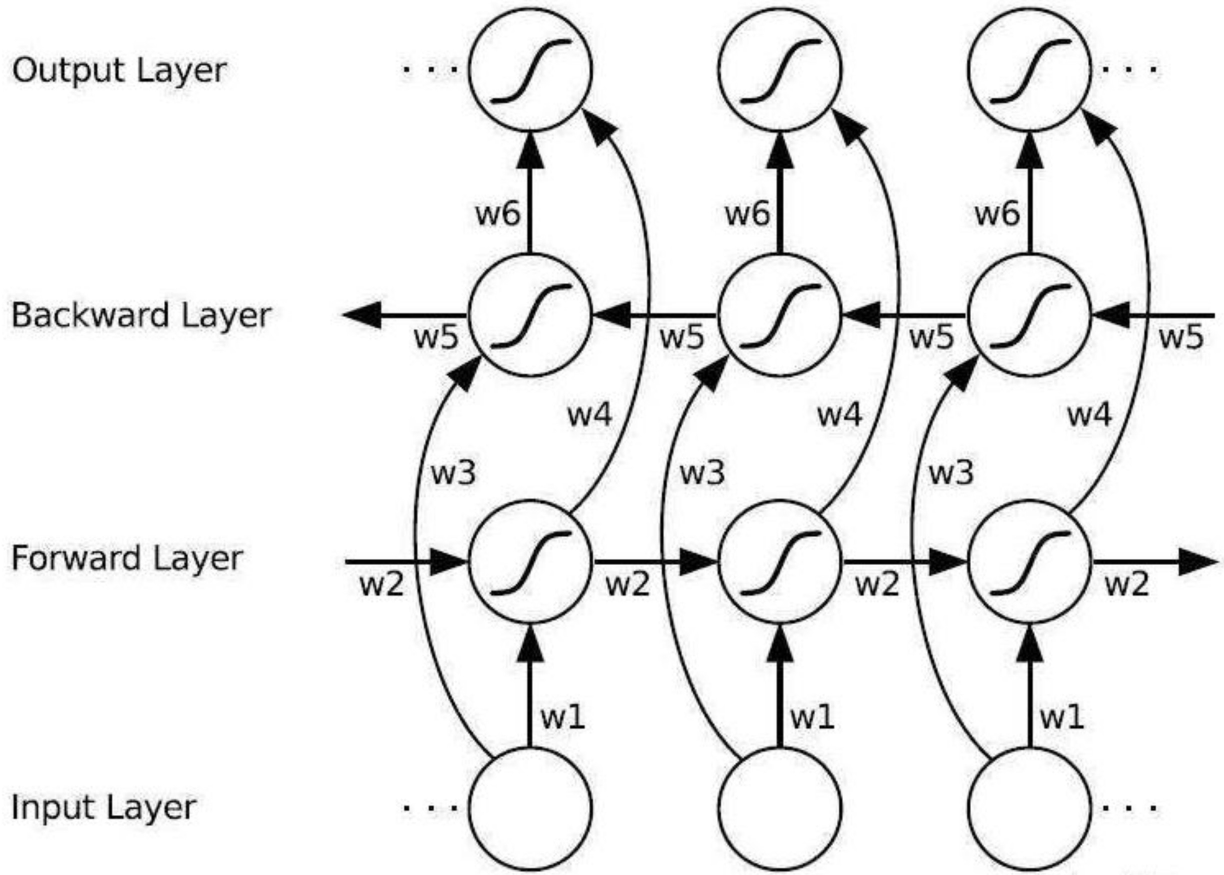
$$h_t = o_t * \tanh\left(C_t\right)$$

GRU



https://blog.csdn.net/mpk_no1/article/details/72875185

雙向LSTM(Bi-directional LSTM)



https://blog.csdn.net/fendouaini/article/details/80198994