



# 語言模型及循環神經網路

報告人: 蘇佳益

指導老師: 陳聰毅

國立高雄科技大學建功校區電子工程系

<https://github.com/chaiyisu/Natural-Language-Processing>

# Agenda

- 語言模型
- 計算圖
- 循環神經網路 (RNN)
- 反向傳播及損失函數
- 演算法複雜度及句子的長度
- Text Generation Example
- 語言模型的評估
- RNN Example



# 語言模型

# 語言模型介紹

- 根據前一個語詞預測下一個語詞

- 他有很多 \_\_\_\_\_

錢

書籍

- And many words

房子

# 語言模型：公式定義

- 假設我們有一個序列  $x^1 \dots x^t$ ， $x^{t+1}$  的計算方法如下：
- $p(x^{t+1} | x^t, \dots, x^1)$
- $x^{t+1}$  為所有單詞中其中一個語詞
- 下一個語詞所會出現的機率的運算稱語言模型。

# 語言模型

- 語言模型也可以用來產生一段文字的機率
- 假設我們要產生一段文字  $x^1 \dots x^T$ ，其計算方式如下
- $p(x^1, \dots, x^T) = p(x^1) * p(x^2|x^1) * \dots * p(x^T|x^{T-1}, \dots, x^1)$   
 $= \prod_{t=1}^T p(x^t|x^{t-1}, \dots, x^1)$

# Language model is everywhere and every day.

face|

- face**book**
- face**book** login
- face**book** sign up
- face**book** messenger
- face**book** app
- face **shield**
- face**book** stock
- face**book** video downloader
- face**time**
- face**book** logo

goog|

- google
- google **translate**
- google **maps**
- google **classroom**
- google **news**
- google **scholar**
- google **docs**
- google **images**
- google **mail**
- google **play**

amaz|

- amazon **prime video**
- amazon **prime**
- amazon
- amazon **talker**
- amazon **taiwan**
- amazon **stock**
- amazon **us**
- amazon **jobs**
- amazon **india**
- amazon **kindle**

# How to learn a language model?

- N-gram (Pre-Deep Learning)
- RNN
- LSTM
- Seq2Seq
- GPT
- BERT
- ...



# N-gram 語言模型的問題

- 以投資方面，他有很多 \_\_\_\_\_
- Sparsity (4-grams)
  - $\frac{C(\text{有很多}\_\_\_)}{c(\text{有很多})}$ 
    - Numerator : add smoothing term
    - Denominator : Back off
    - This problem gets worse when n is increased.
- Storage
  - Needs to store all the counts

# N-gram 語言模型的實例：路透社語料庫

- Today the \_\_\_\_
- Probability Distribution
  - company 0.153
  - bank 0.153
  - price 0.077
  - italian 0.039
  - emirate 0.039
- Have a sparsity problem.
- <https://nlpforhackers.io/language-models>

# Tri-Gram 語言模型：文本生成

- Today the \_\_\_\_
- Probability Distribution
  - company 0.153
  - bank 0.153
  - price 0.077
  - italian 0.039
  - emirate 0.039

# Tri-Gram 語言模型：文本生成

- Today the price —
- Probability Distribution
  - of 0.308
  - for 0.050
  - it 0.046
  - to 0.046
  - is 0.031
- ...

# Tri-Gram 語言模型：文本生成

- today the price of gold per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .
- 文法上有些地方合理有些不合理，但是句子間的關聯性不大
- $n$ 越大，語言模型表現越好
- 然而， $n$ 越大，越難找到相對應的語詞，每個語詞的機率也越小 (Sparsity Problem)

# Neural Language Model : Fixed-window

• ~~as the proctor started the clock the students opened their~~ \_\_\_\_

output distribution

$$\hat{y} = \text{softmax}(Uh + b_2) \in \mathbb{R}^{|V|}$$

hidden layer

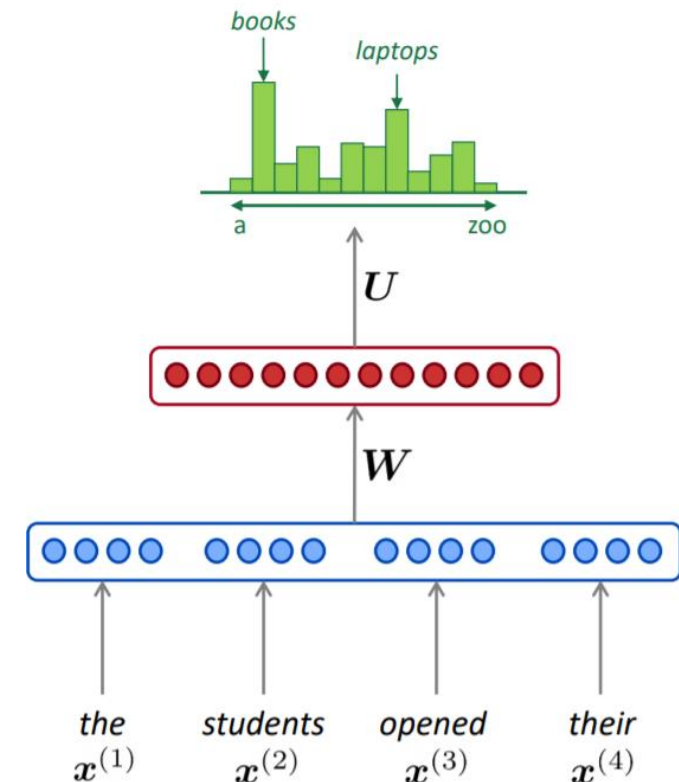
$$h = f(We + b_1)$$

concatenated word embeddings

$$e = [e^{(1)}; e^{(2)}; e^{(3)}; e^{(4)}]$$

words / one-hot vectors

$$x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$$



# 固定視窗大小語言模型的好處及壞處

- 好處
  - 解決了 N-Gram 模型的問題
- 壞處
  - 視窗太小資訊量可能不足
  - 當視窗大小增加權種大小也會增加
  - 視窗大小永遠不夠大
  - 每個輸入都有不同的權重



$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

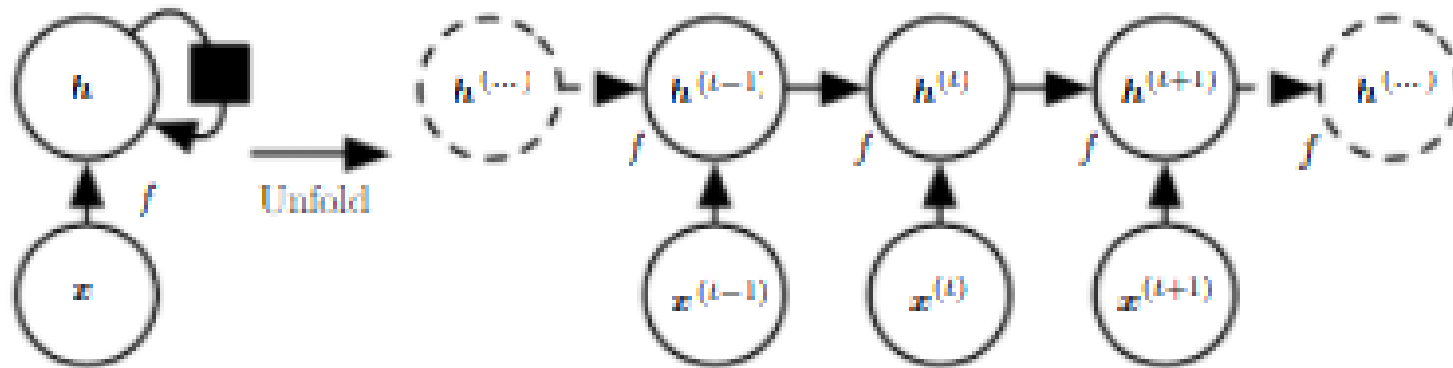
- 因此：我們需要一個網路來處理任意長度的句子



# 計算圖



# 循環圖及展開圖



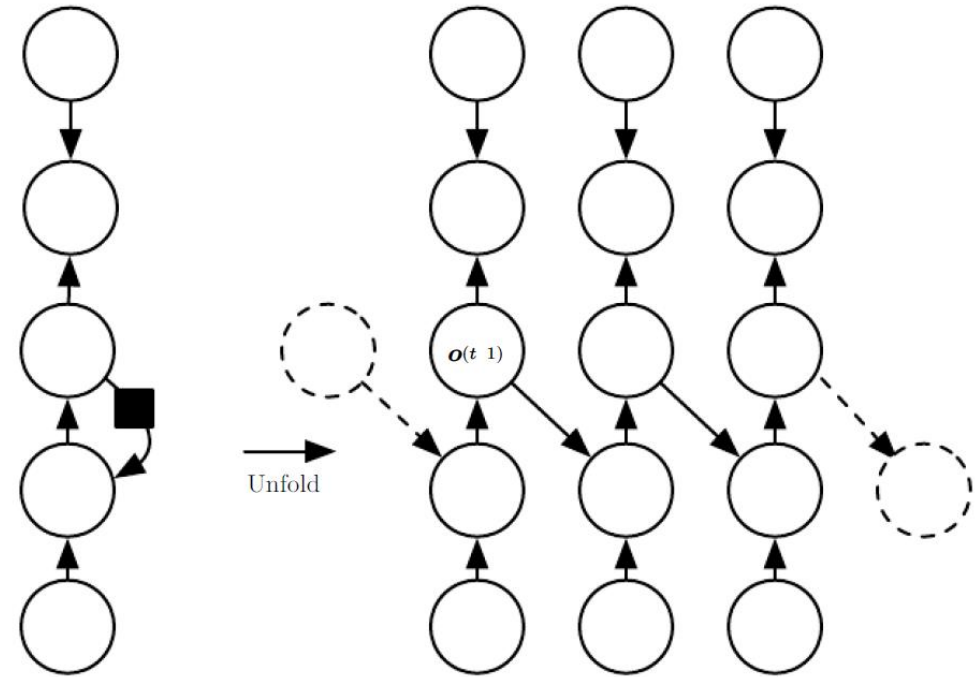
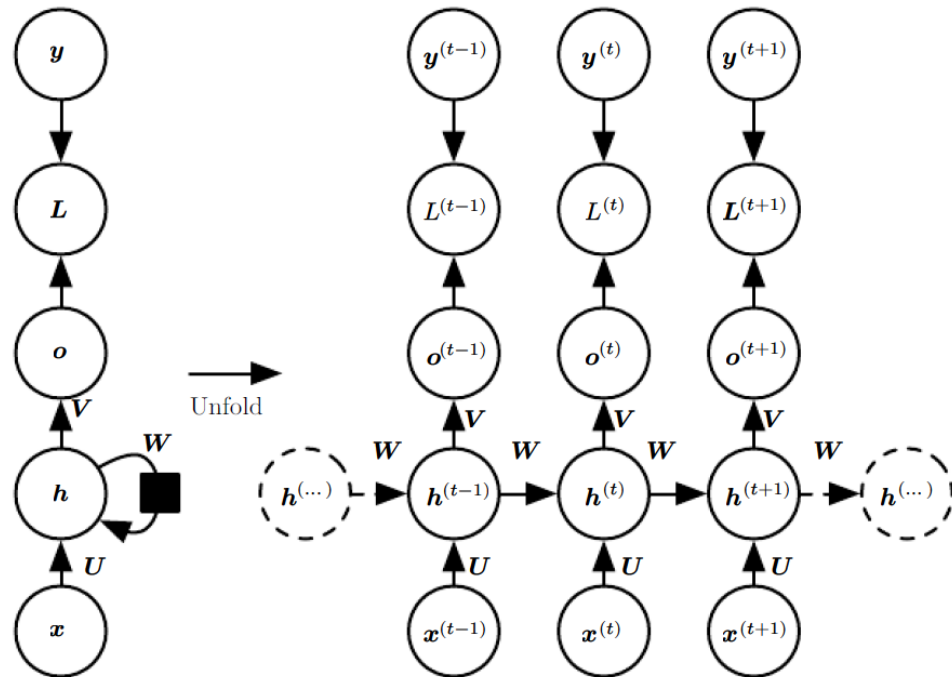
# 循環圖與展開圖的比較

- 循環圖
  - 簡潔
- 展開圖
  - 較明確
  - 能夠清楚的展示前向及反向傳播的資訊

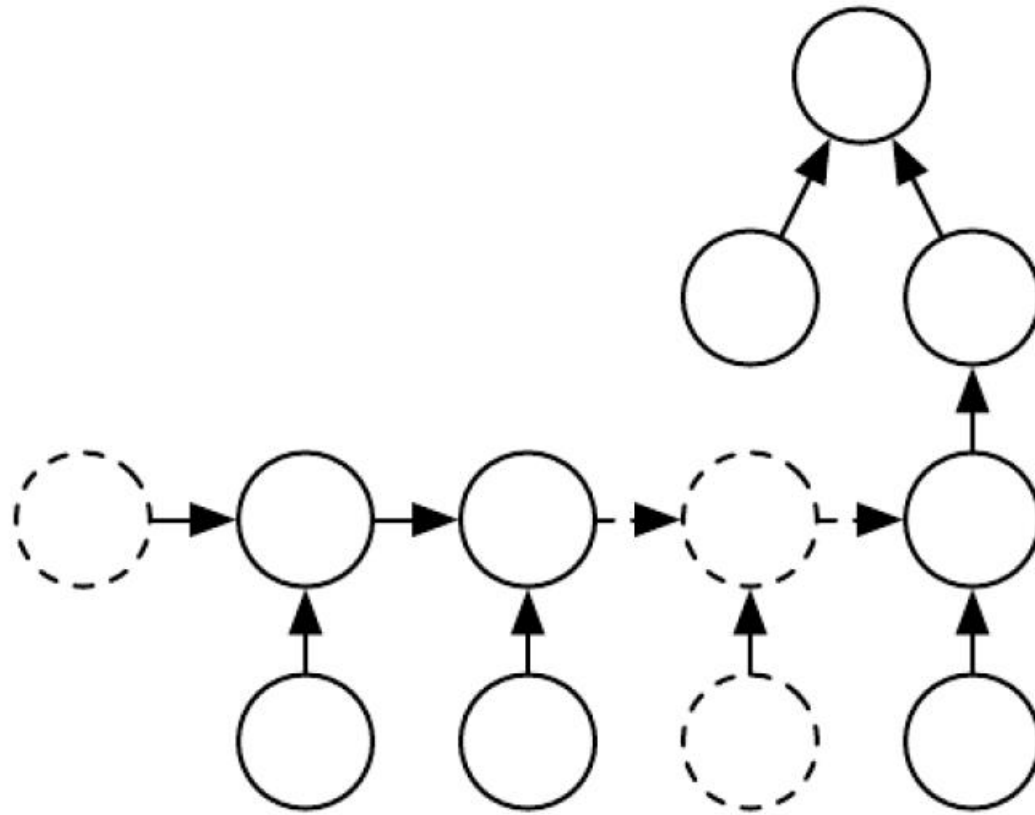


# 循環神經網路 (RNN)

# RNN Design Patterns

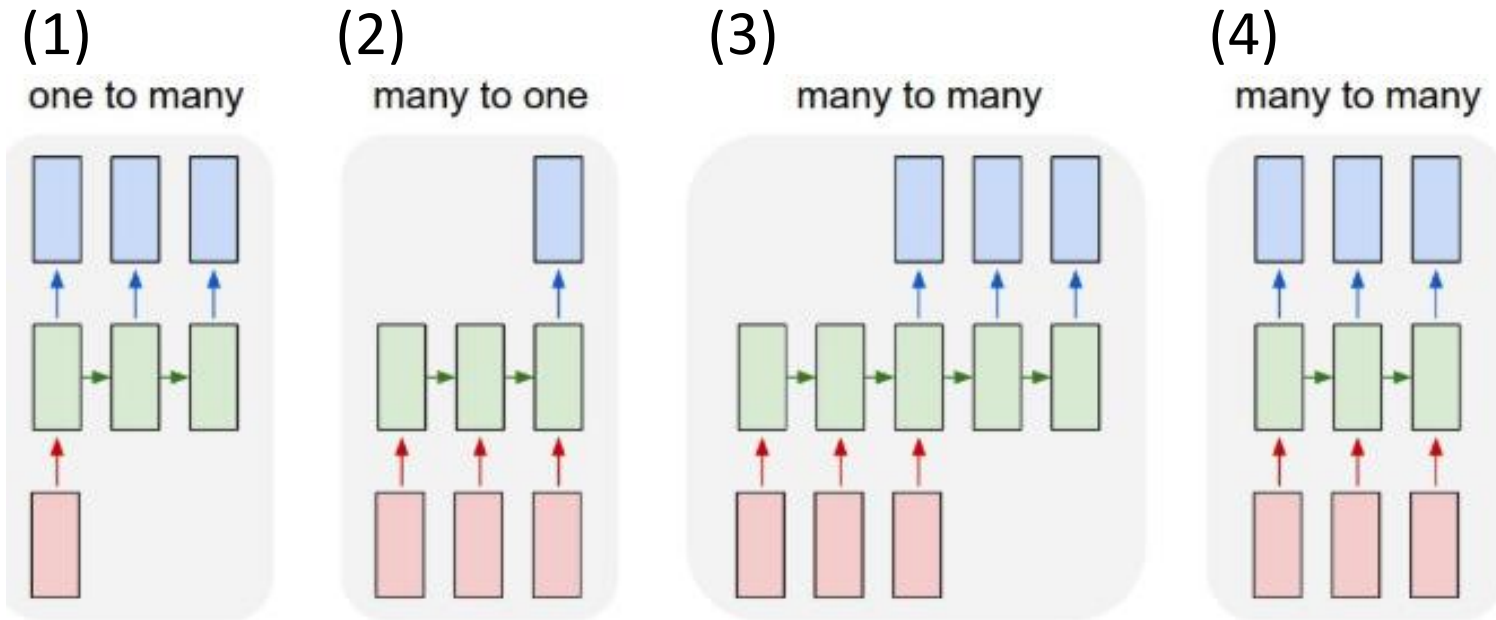


# RNN Design Patterns



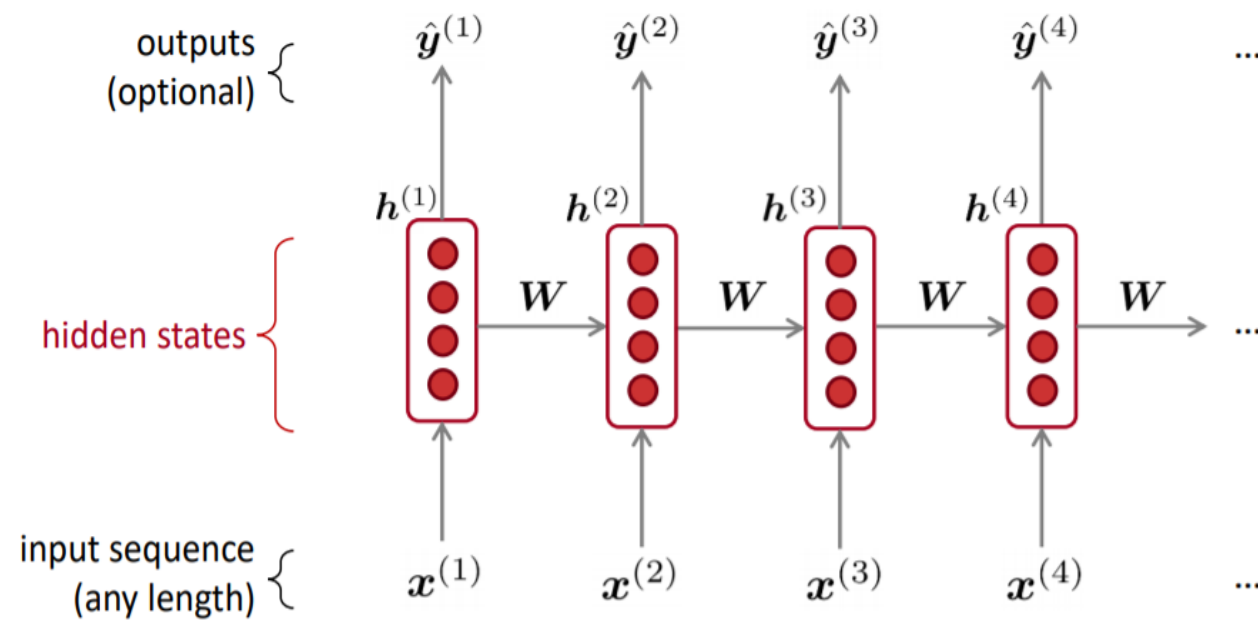
# RNN Overview

- (1) Image Captioning (2) Sentiment Analysis (3) Machine Translation (4) Video Classification



# 循環神經網路(RNN)

- $W$ 通常會重複使用
- 隱含層的計算會採用之前的資訊以及目前所輸入的資訊
- 輸入的句子可以是任意長度
- RNN不一定每個輸入都要有輸出



# Detail of RNNLM

## output distribution

$$\hat{y}^{(t)} = \text{softmax} \left( U h^{(t)} + b_2 \right) \in \mathbb{R}^{|V|}$$

## hidden states

$$h^{(t)} = \sigma \left( W_h h^{(t-1)} + W_e e^{(t)} + b_1 \right)$$

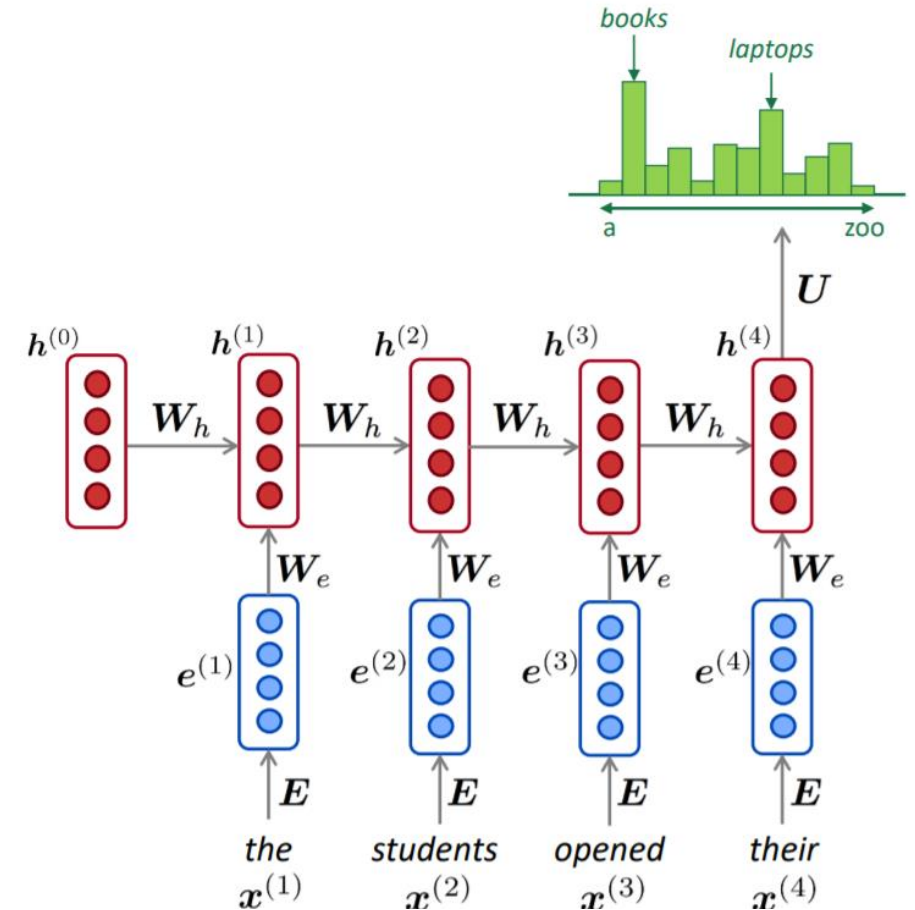
$h^{(0)}$  is the initial hidden state

## word embeddings

$$e^{(t)} = E x^{(t)}$$

words / one-hot vectors

$$x^{(t)} \in \mathbb{R}^{|V|}$$





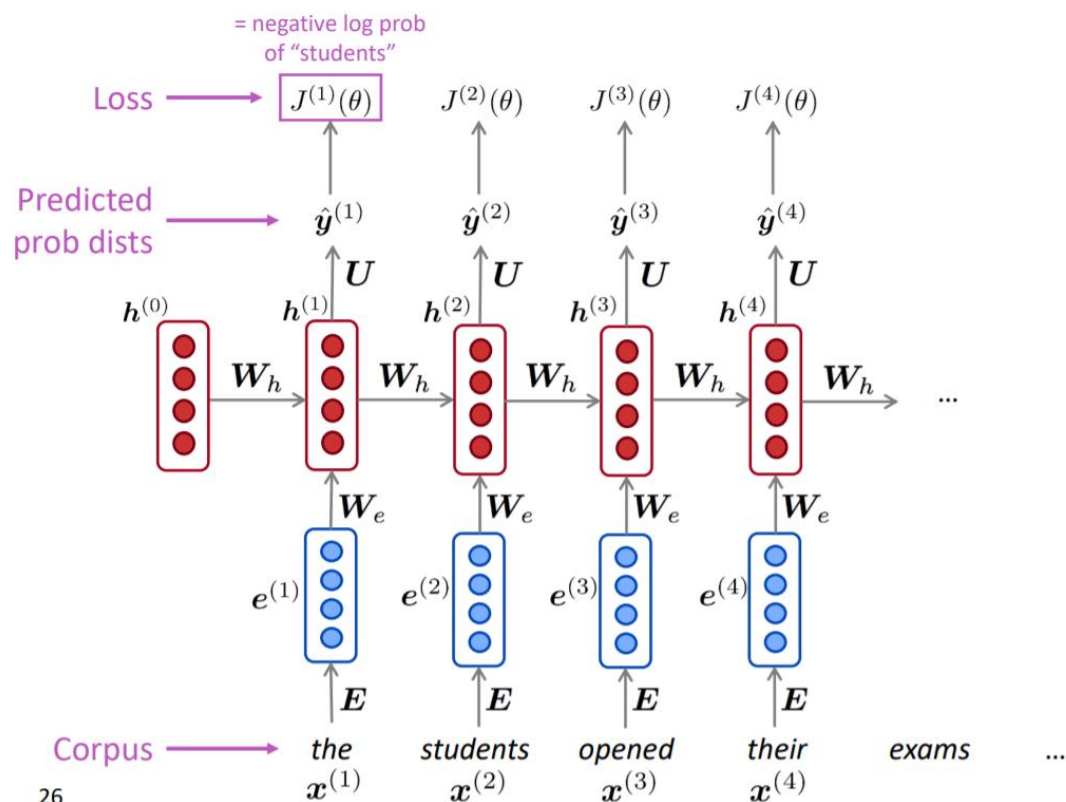
# RNNLM: 好處及壞處

- 好處:
  - 可以處理任何長度的輸入
  - 理論上可以運用到歷史資訊
  - 當輸入增加時，模型的大小不會跟著增加
  - 因為每個權重的輸入都相同，所以每個輸入的處理過程都相同
- 壞處
  - 運算速度較慢
  - 實際上無法有效運用歷史資訊 (梯度消失、爆炸)

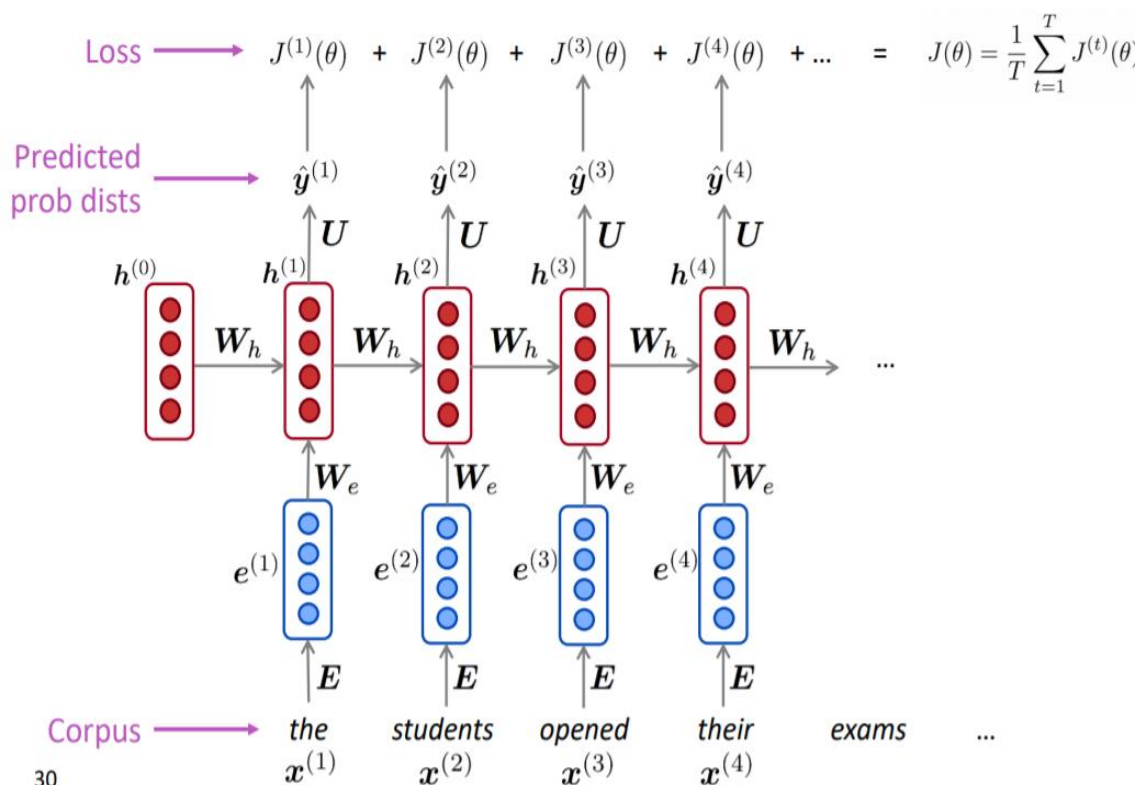
# RNNLM : Training

- 將語詞序列輸入到RNN；計算出每一個時序的機率分布
- 計算損失值
- 將損失值取平均

# RNNLM : Training



26



30

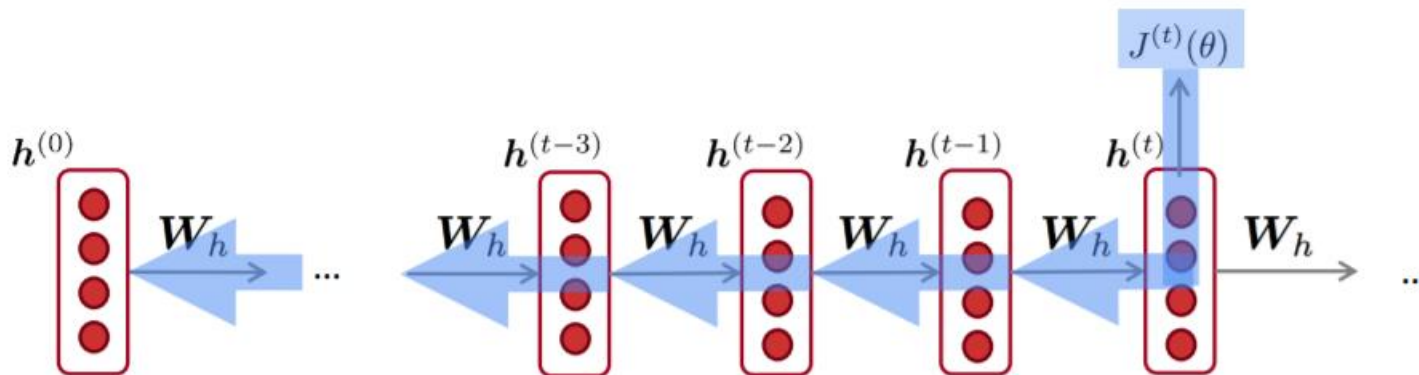
# RNNLM : Training

- 利用整個文本來計算梯度非常消耗資源
- 因此，通常我們採用小Batch的句子或文本來計算梯度



# 反向傳播及損失函數

# RNN : 反向傳播



- $\frac{\partial J^{(t)}}{\partial W_h} = \sum_{i=1}^t \frac{\partial J^{(t)}}{\partial W_h|_i}$

How to Calculate?

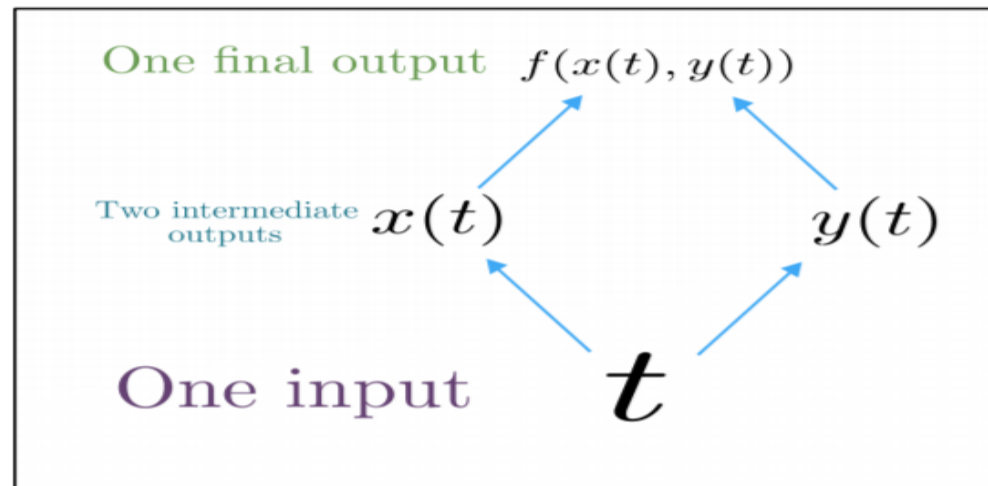
- This is called “backpropagation through time”

# Backpropagation : Multivariable Chain Rule

- Given a multivariable function  $f(x, y)$ , and two single variable functions  $x(t)$  and  $y(t)$ , here's what the multivariable chain rule says:

$$\underbrace{\frac{d}{dt} f(x(t), y(t))}_{\text{Derivative of composition function}} = \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt}$$

Derivative of composition function

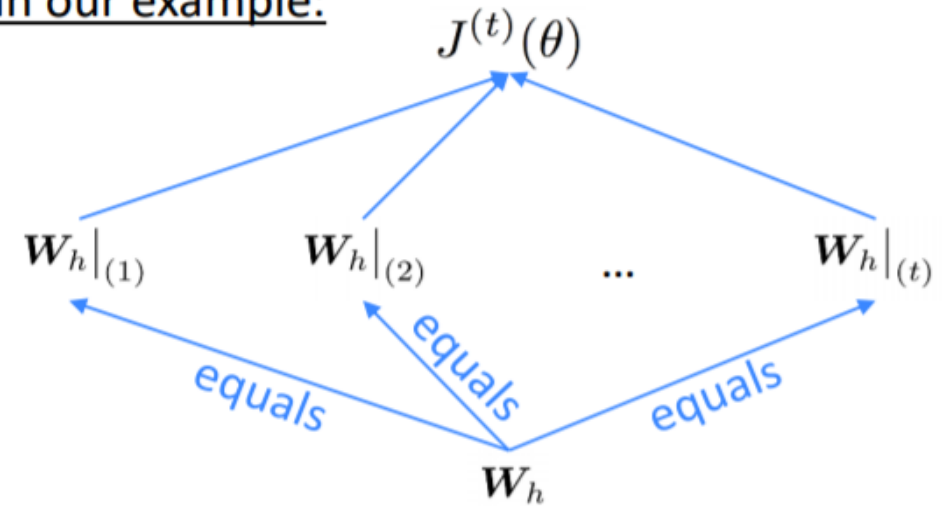


- <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/slides/cs224n-2019-lecture06-rnnlm.pdf>

# Backpropagation : RNN

- $\frac{\partial J^{(t)}}{\partial W_h} = \sum_{i=1}^t \frac{\partial J^{(t)}}{\partial W_h|_i} * \frac{\partial W_h|_i}{\partial W_h}$
- $\frac{dx}{dx} = ?$
- Timestep is from  $t$  to 1.

In our example:



- <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/slides/cs224n-2019-lecture06-rnnlm.pdf>



# RNN：損失函數

- 損失函數的選擇會根據不同的任務而不同
- 如果我們任務是輸出一個機率分布
  - Cross-Entropy Loss
- 如果我們的任務為預測下一個語詞
  - 目標將會是最大化Log-Likelihood 或是最小化負的Log-Likelihood



# 演算法複雜度及句子的長度

# 如何決定序列的長度?

- 在文本中加一個特殊符號，當輸出此特殊符號就停止輸出 (Schmidhuer, 2012)
- 加一個Bernoulli的模型判斷是否繼續輸出 (better and more general)
- 加一個而外的輸出來預測輸出序列長度 (Goodfellow, 2014d)

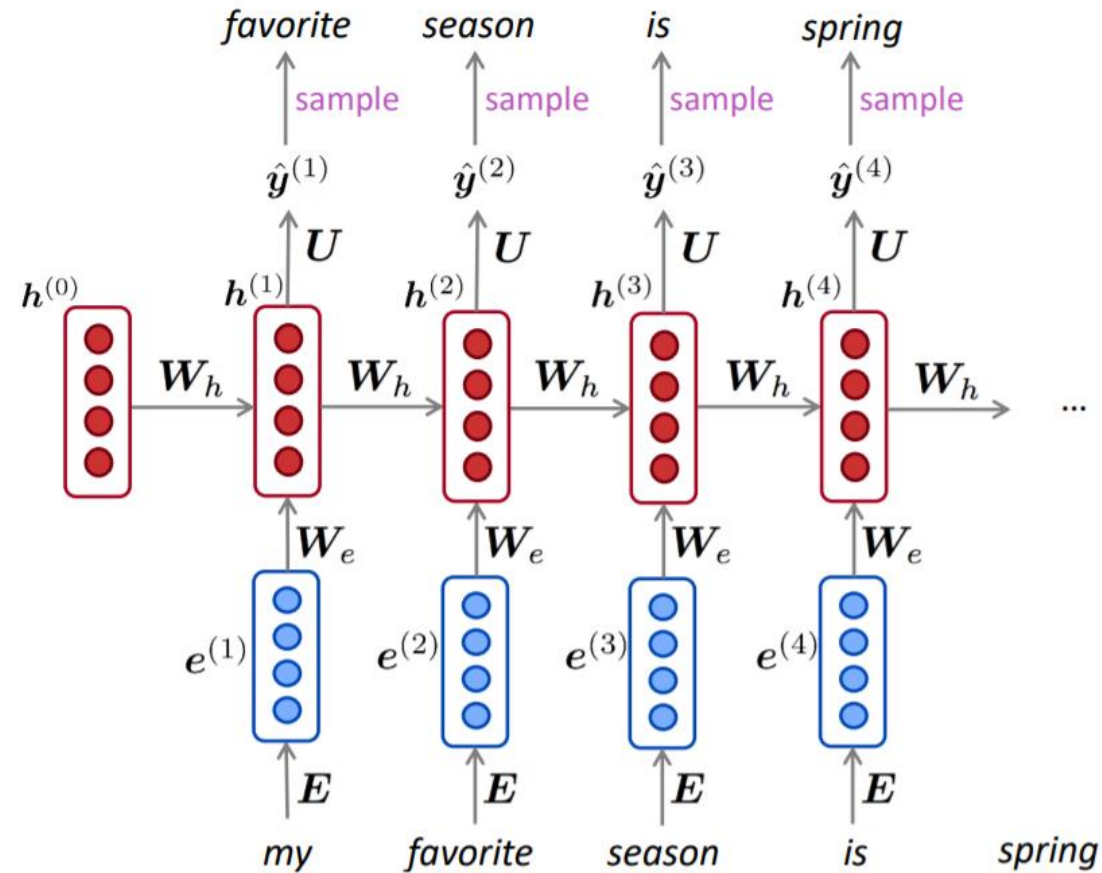
# RNN：演算法複雜度

- Forward Propagation Runtime (cannot be parallelized)
  - $O(\tau)$
- Forward Propagation Memory Cost
  - $O(\tau)$
- Backward Propagation Runtime
  - $O(\tau)$
- Parameter Complexity
  - $O(1)$



# Text Generation Example

# RNNLM : Text Generation



# RNNLM trained on Harry Potter

- Have the characteristic's name but still difficult to read. It has run-on sentences.

“Sorry,” Harry shouted, panicking—“I’ll leave those brooms in London, are they?”

“No idea,” said Nearly Headless Nick, casting low close by Cedric, carrying the last bit of treacle Charms, from Harry’s shoulder, and to answer him the common room perched upon it, four arms held a shining knob from when the spider hadn’t felt it seemed. He reached the teams too.

- <https://medium.com/deep-writing/harry-potter-written-by-artificial-intelligence-8a9431803da6>

# RNNLM trained on Recipes

- Challenging task.
- Still fluent but nonsensical

Title: CHOCOLATE RANCH BARBECUE

Categories: Game, Casseroles, Cookies, Cookies

Yield: 6 Servings

2 tb Parmesan cheese -- chopped

1 c Coconut milk

3 Eggs, beaten

Place each pasta over layers of lumps. Shape mixture into the moderate oven and simmer until firm. Serve hot in bodied fresh, mustard, orange and cheese.


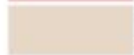

Combine the cheese and salt together the dough in a large skillet; add the ingredients and stir in the chocolate and pepper.

- <https://gist.github.com/nylki/1efbaa36635956d35bcc>



# RNNLM trained on Paint Color Name

- character lever RNNLM
- Trained by using RGB

|   |                            |   |                            |
|---|----------------------------|---|----------------------------|
|    | Ghasty Pink 231 137 165    |    | Sand Dan 201 172 143       |
|    | Power Gray 151 124 112     |    | Grade Bat 48 94 83         |
|    | Navel Tan 199 173 140      |    | Light Of Blast 175 150 147 |
|    | Bock Coe White 221 215 236 |    | Grass Bat 176 99 108       |
|    | Horble Gray 178 181 196    |    | Sindis Poop 204 205 194    |
|    | Homestar Brown 133 104 85  |    | Dope 219 209 179           |
|    | Snader Brown 144 106 74    |    | Testing 156 101 106        |
|   | Golder Craam 237 217 177   |   | Stoner Blue 152 165 159    |
|  | Hurky White 232 223 215    |  | Burple Simp 226 181 132    |
|  | Burf Pink 223 173 179      |  | Stanky Bean 197 162 171    |
|  | Rose Hork 230 215 198      |  | Turdly 190 164 116         |

- <https://aiweirdness.com/post/160776374467/new-paint-colors-invented-by-neural-network>



# 語言模型的評估

# 困惑度 (Perplexity)

- 用來評估語言模型
- 困惑度越低越佳
- 困惑度公式如下

$$\begin{aligned} perplexity &= \prod_{t=1}^T \left( \frac{1}{P(x^{(t+1)} | x^t \dots x^1)} \right)^{1/T} \\ &= \exp \left( \frac{1}{T} \sum_{t=1}^T -\log \hat{y}_{x_{t+1}}^{(t)} \right) = \exp(J(\theta)) \end{aligned}$$

# RNNs have greatly improved perplexity

| Model   | Perplexity |
|---|------------|
| Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)            | 67.6       |
| RNN-1024 + MaxEnt 9-gram (Chelba et al., 2013)                  | 51.3       |
| RNN-2048 + BlackOut sampling (Ji et al., 2015)                  | 68.3       |
| Sparse Non-negative Matrix factorization (Shazeer et al., 2015) | 52.9       |
| LSTM-2048 (Jozefowicz et al., 2016)                             | 43.7       |
| 2-layer LSTM-8192 (Jozefowicz et al., 2016)                     | 30         |
| Ours small (LSTM-2048)  | 43.9       |
| Ours large (2-layer LSTM-2048)                                  | 39.8       |

- <https://research.fb.com/building-an-efficient-neural-language-model-over-a-billion-words/>



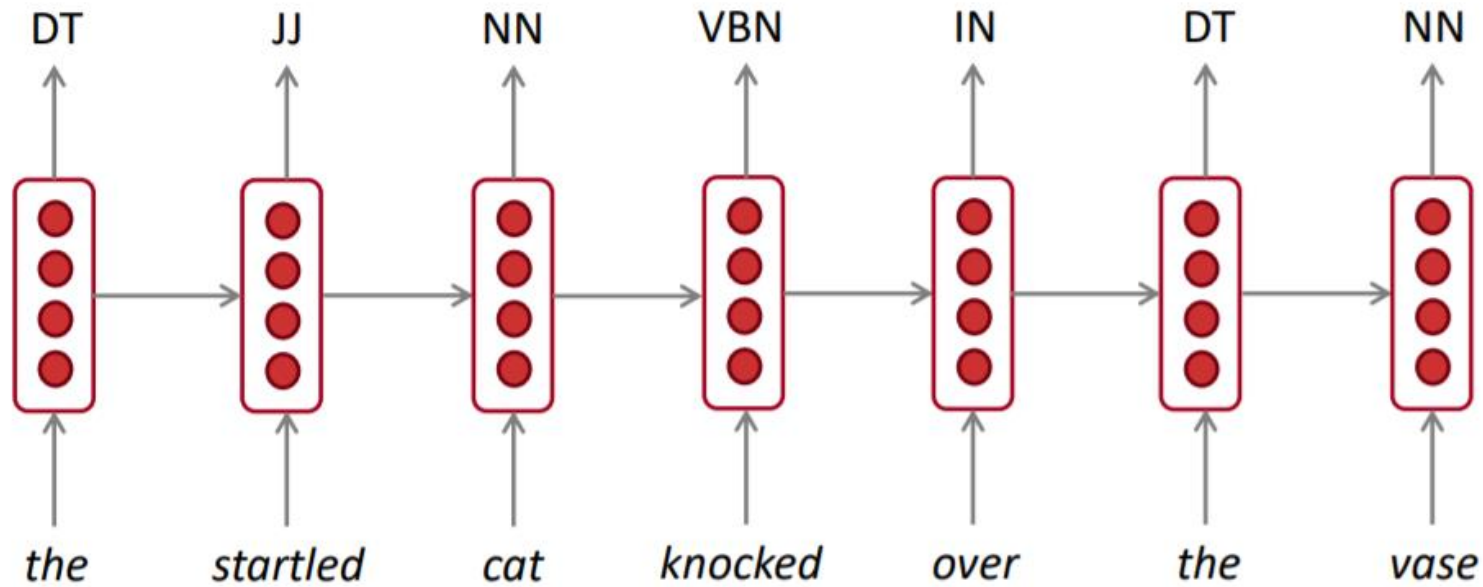
# RNN Example

# Why is Language Model important?

- Language Model is a benchmark task so it can help us understand how well the computer understands human language.
- Language Model is involved in many NLP tasks. E.g.
  - Predictive Typing
  - Speech Recognition
  - Handwriting Recognition
  - Spelling/Grammar Correction
  - Authorship Identification
  - Machine Translation
  - Summarization
  - Dialogue

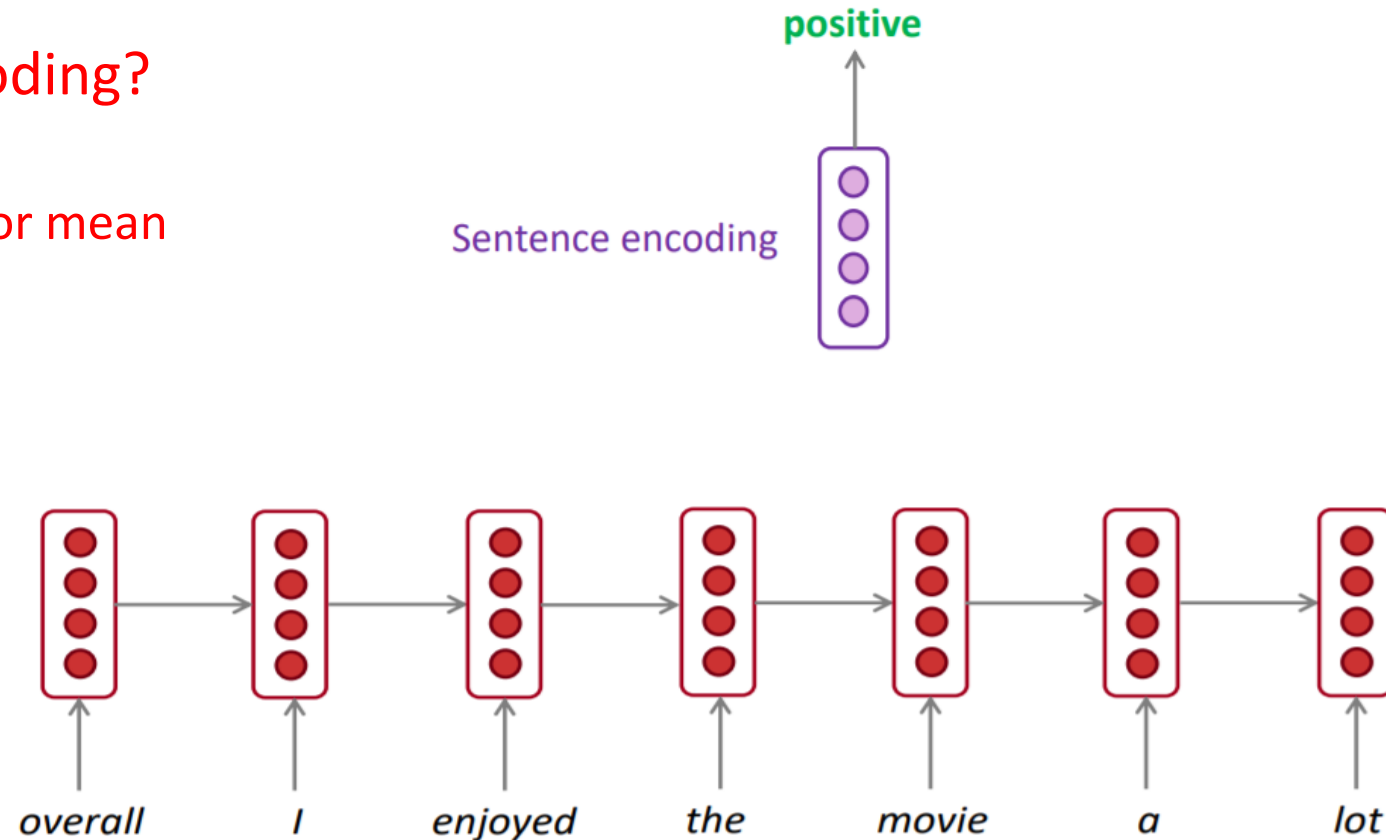
# RNN for Tagging

- E.g. Part-of-Speech, Name Entity Recognition



# RNN for Sentence Classification

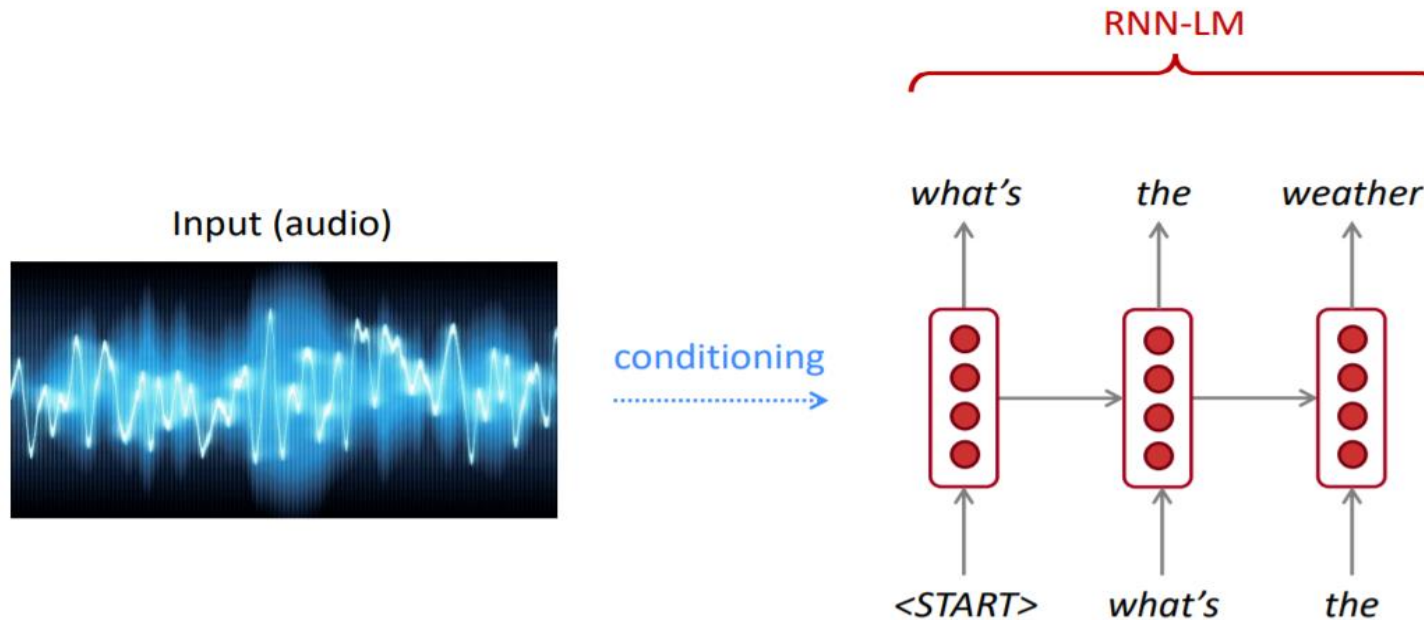
- e.g. Sentiment Classification
- How to choose encoding?
  - Final Hidden State
  - Element-wise max or mean





# RNN for Text Generation

- Speech Recognition, Machine Translation etc.



# Image Captioning

- The task of describing image by using natural language.



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."

# Fun Examples

```
{ { cite journal | id=Cerling Nonforest Department|format=NewLynxlated|none } }
'www.e-complete'.
```

'''See also''': [[List of ethical consent processing]]

== See also ==

- \*[[Iender dome of the ED]]
- \*[[Anti-autism]]

=== [[Religion|Religion]] ===

- \*[[French Writings]]
- \*[[Maria]]
- \*[[Revelation]]
- \*[[Mount Agamul]]

== External links ==

- \* <http://www.biblegateway.nih.gov/entrepre/> Website of the World Festival. The labour of India

==External links==

- \* <http://www.romanoology.com/> Constitution of the Netherlands and Hispanic Competition for Bila

```
<page>
<title>Antichrist</title>
<id>865</id>
<revision>
<id>15900676</id>
<timestamp>2002-08-03T18:14:12Z</timestamp>
<contributor>
<username>Paris</username>
<id>23</id>
</contributor>
<minor />
<comment>Automated conversion</comment>
<text xml:space="preserve">#REDIRECT [[Christianity]]</text>
</revision>
</page>
```

For  $\bigoplus_{n=1,\dots,m} \mathcal{L}_{m,n}$  where  $\mathcal{L}_{m,n} = 0$ , hence we can find a closed subset  $H$  in  $H$  and any sets  $\mathcal{F}$  on  $X$ ,  $U$  is a closed immersion of  $S$ , then  $U \rightarrow T$  is a separated algebraic space.

*Proof.* Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by  $\coprod Z \times_U U \rightarrow V$ . Consider the maps  $M$  along the set of points  $Sch_{fppf}$  and  $U \rightarrow U$  is the fibre category of  $S$  in  $U$  in Section, ?? and the fact that any  $U$  affine, see Morphisms, Lemma ?? . Hence we obtain a scheme  $S$  and any open subset  $W \subset U$  in  $Sh(G)$  such that  $\mathrm{Spec}(R') \rightarrow S$  is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that  $f_i$  is of finite presentation over  $S$ . We claim that  $\mathcal{O}_{X,x}$  is a scheme where  $x, x', s' \in S'$  such that  $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}_{X',x'}$  is separated. By Algebra, Lemma ?? we can define a map of complexes  $\mathrm{GL}_S(x'/S'')$  and we win.  $\square$

To prove study we see that  $\mathcal{F}|_U$  is a covering of  $\mathcal{N}'$ , and  $\mathcal{T}_i$  is an object of  $\mathcal{F}_{X/S}$  for  $i > 0$  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

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

is a unique morphism of algebraic stacks. Note that

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

and

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

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.  $\square$

The result for prove any open covering follows from the less of Example ?? . It may replace  $S$  by  $X_{spaces, \acute{e}tale}$  which gives an open subspace of  $X$  and  $T$  equal to  $S_{Zar}$ , see Descent, Lemma ?? . Namely, by Lemma ?? we see that  $R$  is geometrically regular over  $S$ .

# References

- 齋藤康毅, Deep Learning 2: 用Python進行自然語言處理的基礎理論實作.
- Manning et al., CS224n Natural Language Processing with Deep Learning, Stanford University.