

吳恩達老師_深度學習_卷積神經網路_第三週_序列模型與注意力機制

tags: deep learning Andrew Ng

3-1.Basic models



deeplearning.ai

Sequence to
sequence models

Basic models

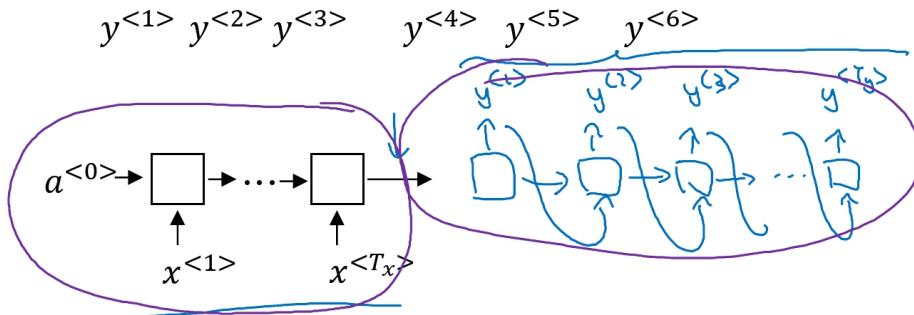
課程說明：基礎模型

Sequence to sequence model

Sequence to sequence model

$x^{<1>} \quad x^{<2>} \quad x^{<3>} \quad x^{<4>} \quad x^{<5>}$
 Jane visite l'Afrique en septembre

→ Jane is visiting Africa in September.



[Sutskever et al., 2014. Sequence to sequence learning with neural networks] ↩

[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation] ↩

Andrew Ng

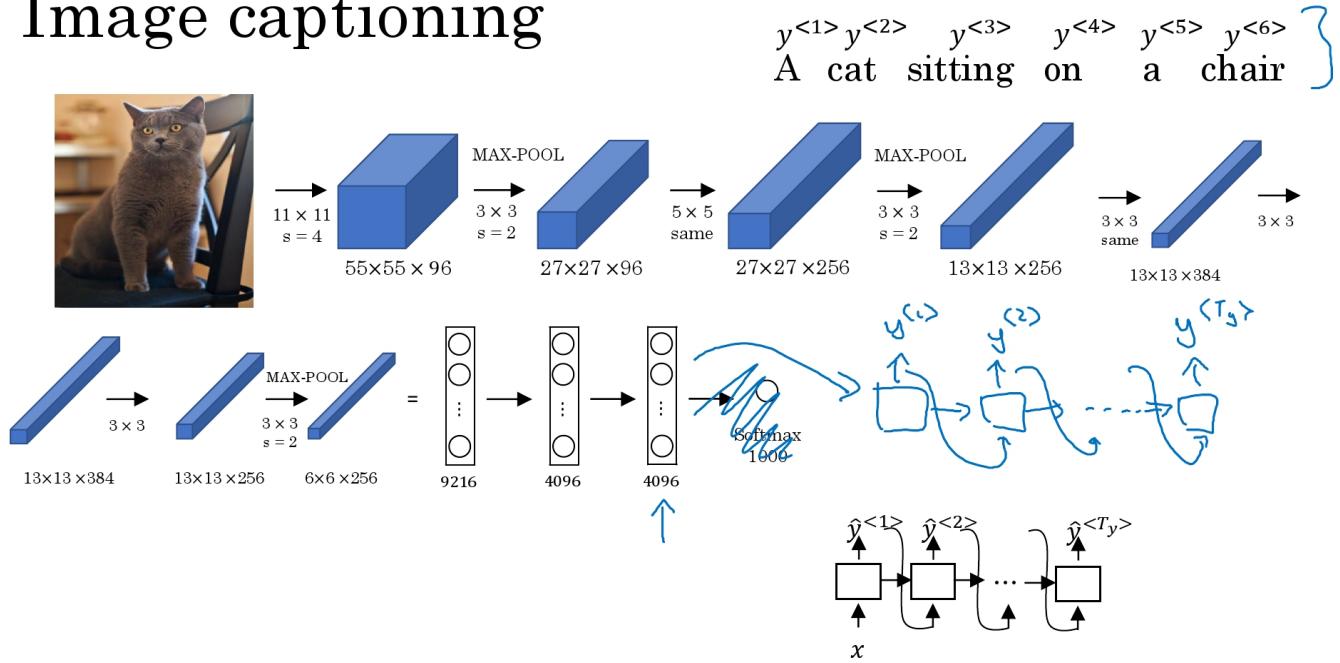
假設我們輸入一句法文，希望將它翻譯成英文，如之前課程說明，以 $x^{<1>} \dots x^{<5>}$ 來表示輸入的句子，而以 $y^{<1>} \dots y^{<6>}$ 來表示翻譯後的英文句子。

- 建立一個網路，編碼網路(encoder network)
 - RNN，可以是GRU也可以是LSTM
 - 輸出一個向量來代表輸入序列
- 建立一個網路，解碼網路(decoder network)
 - 以encoder network的輸出為輸入
 - 輸出翻譯後的單字

在給定足夠的英法相關文本資料情況下，這個模型非常有效。

Image captioning

Image captioning



[Mao et. al., 2014. Deep captioning with multimodal recurrent neural networks]

[Vinyals et. al., 2014. Show and tell: Neural image caption generator]

[Karpathy and Li, 2015. Deep visual-semantic alignments for generating image descriptions]

Andrew Ng

另外有一個類似的方法被用來做圖片的描述，給定一張照片，輸出相對應的文字描述，此例輸出為『A cat sitting on a chair』。(此例以Alet net為範例)

- CNN做預訓練，去除最後的softmax unit
 - 4096維向量來表示這隻貓
 - 以此代表圖片的編碼網路(encoder network)
- RNN_decoder network
 - 產生圖片的描述

在描述不是很長的情況下，這個模型也非常有效果。

3-2.Picking the most likely sentence



deeplearning.ai

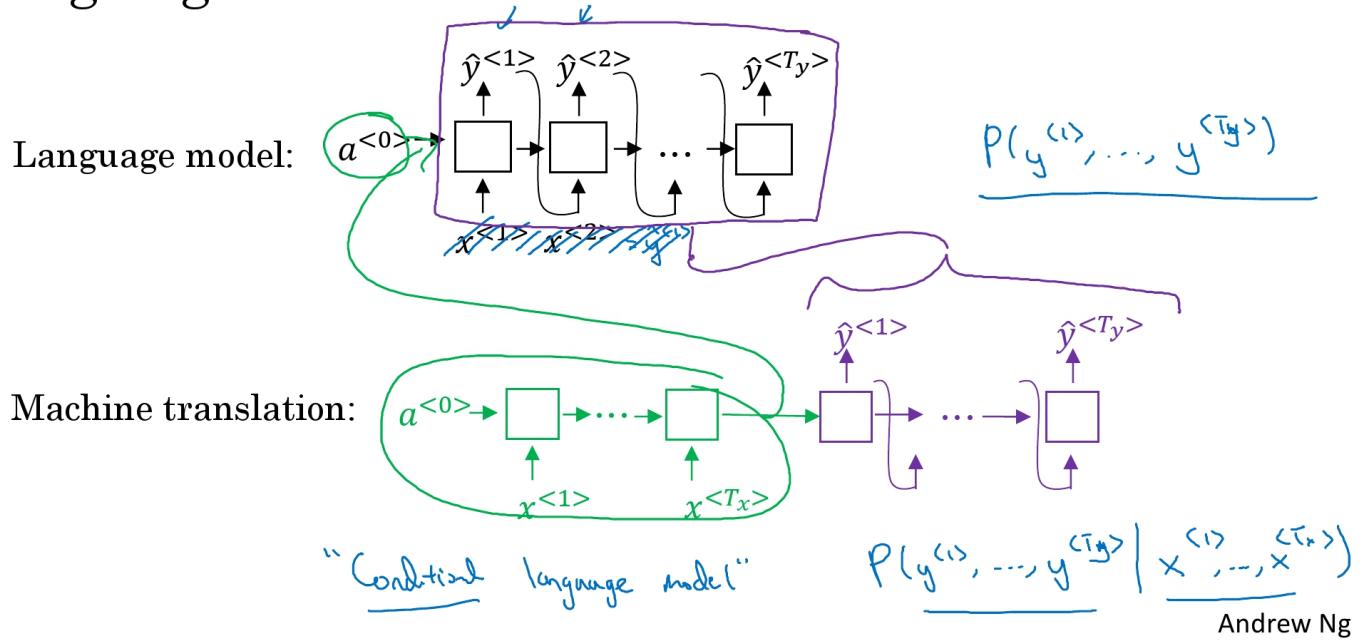
Sequence to sequence models

Picking the most likely sentence

課程說明：Picking the most likely sentence

Machine translation as building a conditional language model

Machine translation as building a conditional language model



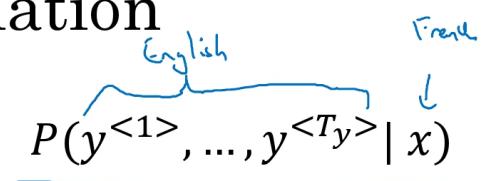
可以將機器翻譯想成是建立一個條件語言模型

- language model:
 - $P(y^{<1>} \dots y^{<T_y>})$
 - 估計句子的可能性
- machine translation:
 - $P(y^{<1>} \dots y^{<T_y>} | x^{<1>} \dots x^{<T_x>})$
 - encoder to decoder
 - decoder與language model幾乎一樣
 - 不同在於language model是以0向量開始，而encoder會計算一系列向量再提供decoder

Finding the most likely translation

Finding the most likely translation

Jane visite l'Afrique en septembre.



- Jane is visiting Africa in September.
- Jane is going to be visiting Africa in September.
- In September, Jane will visit Africa.
- Her African friend welcomed Jane in September.

$$\arg \max_{y^{<1>} \dots y^{<T_y>}} P(y^{<1>} \dots y^{<T_y>} | x)$$

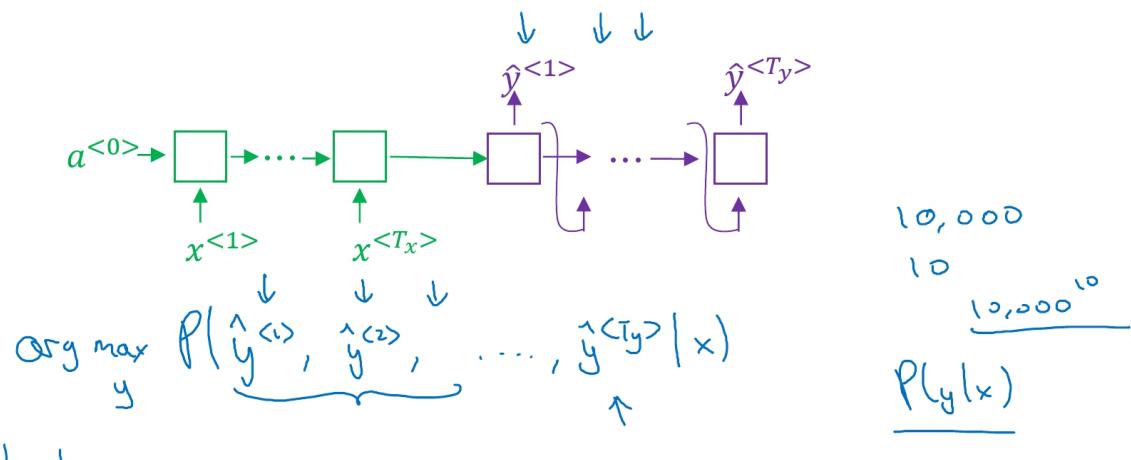
Andrew Ng

如果想透過模型做語言翻譯(此例為法、英)，透過輸入法文(x)，模型會回饋各種英文的翻譯(y)對應機率，這可能會有很多種不同的可能(如條件四項)，所以我們並不是要從資料分佈中做隨機取樣，而是要做條件機率最大化，其中最常見的演算法即為beam search(下節課)。

Why not a greedy search?

Why not a greedy search?

$P(\hat{y}^{<1>} | x)$



→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

$$P(\text{Jane is going } | x) > P(\text{Jane is visit } | x)$$

Andrew Ng

Greedy Search(貪心搜索)，取得 $P(\hat{y}^{<1>} | x)$ 之後，再取下一個單字，再下一個再下一個，一個一個取得，我們不使用greedy search的原因在於，我們要一次性的將整個單字序列取出，來最大化機率 $P(y^{<1>} \dots y^{<T_y>} | x)$ 。

以範例來看，如果採用greedy search，Jane is going to 出現的機率是遠大於Jane is visiting，這看起來就有點冗長。

3-3. Beam search



deeplearning.ai

Sequence to sequence models

Beam search

課程說明 : Beam search

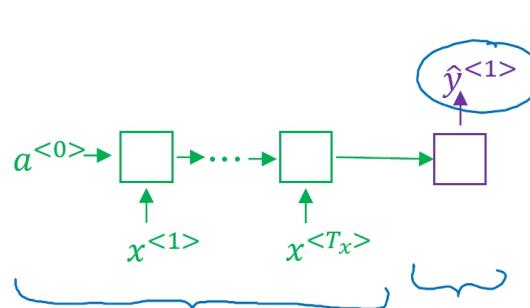
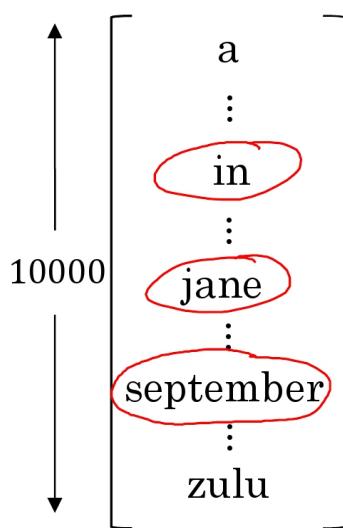
Beam search algorithm

Beam search algorithm

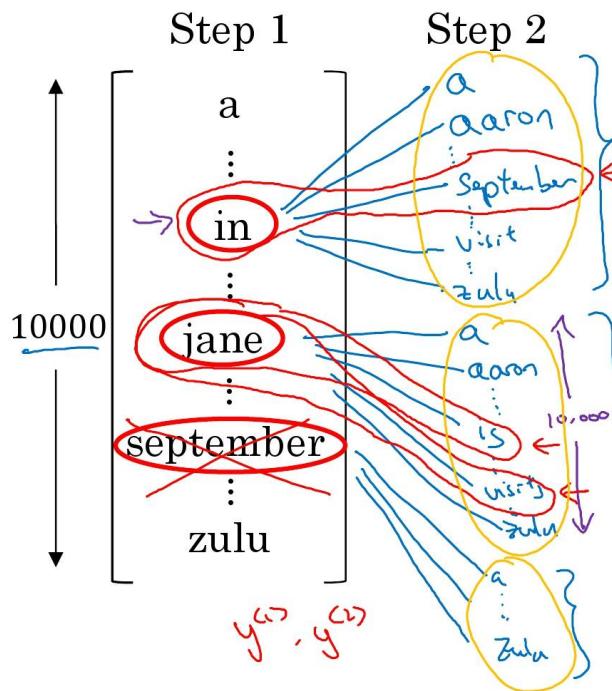
B = 3

(beam width)

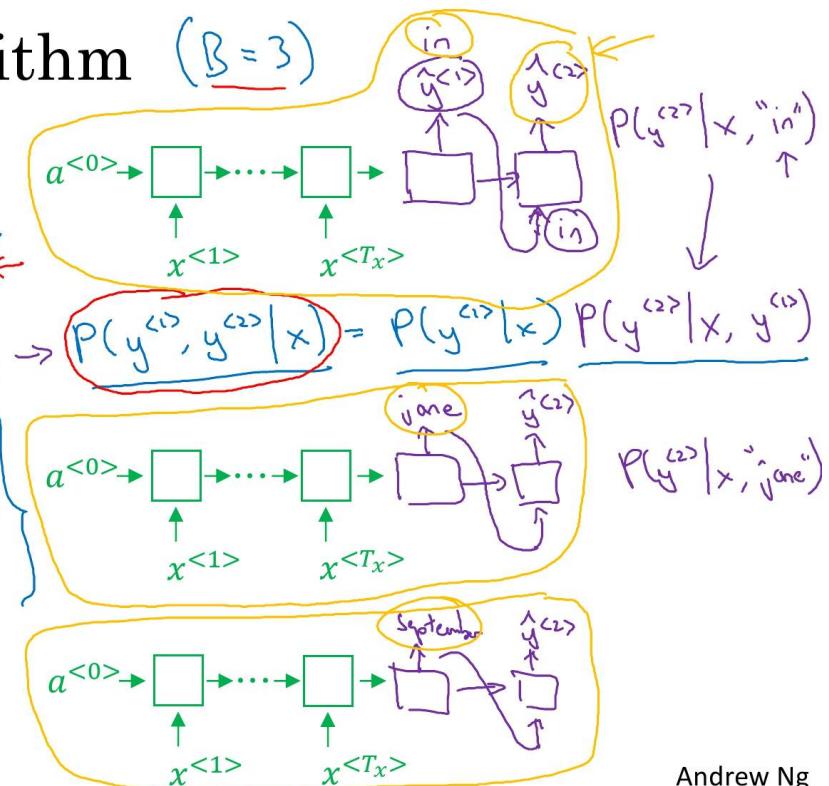
Step 1



Beam search algorithm

(B = 3)

Step 2



Andrew Ng

Andrew Ng

以此週課程中一直用的法語句子為例『Jane visite l'Afrique en Septembre』，我們希望將句子翻譯成『Jane is visiting Africa in September』，假設詞匯表有10000個單字，忽略大小寫。

Beam search步驟如下(綠色代表encoder，紫色為decoder)

- 評估第一個單字的概率值
 - $P(y^{<1>} | x)$

- 如果是greedy search只會計算第一個單字的機率，而beam search會考慮多單字
 - 透過超參數B(bean width)來控制考慮幾個單字
 - 此例為3，代表一次考慮3個單字
 - in , jane , september
- 輸出10000個概率值，並取前三個單字保存
- 得到in , jane , september
 - 針對上面三個單字各自考慮第二個單字
 - $P(y^{<2>}|x, 'in')$ (jane,september相同作法)
 - 我們關心的是第一個與第二個單字中有最大機率的組合
 - $P(y^{<1>}, y^{<2>}|x) = P(y^{<1>}|x) \cdot P(y^{<2>}|x, y^{<1>})$
 - 因為我們設置B=3，最後會有30000個可能
 - 從30000個可能中選出前3(B)個
 - in september, jane is, jane visits
 - 注意到目前選擇的前三個詞對已經沒有september，因為我們的B選擇為3，所以會複製三個相同的網路來選擇部份句子與最後結果。

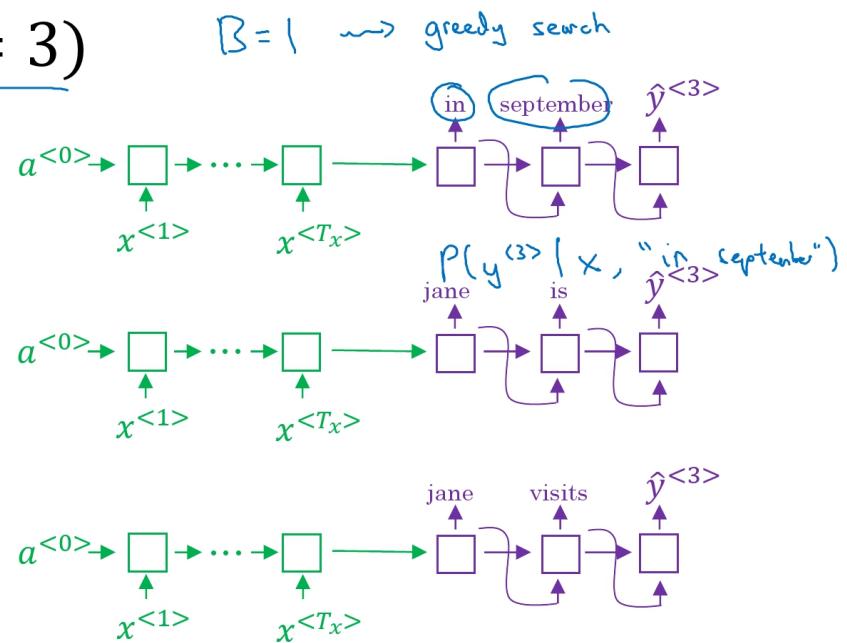
Beam search(B=3)

Beam search (B = 3)

in september
 ↗ aaron
 ↗ jane
 ↗ zuhu

jane is
 ↗ a
 ↗ visits
 ↗ zuhu

jane visits
 ↗ a
 ↗ africa
 ↗ zuhu



$$P(y^{<1>}, y^{<2>} | x)$$

jane visits africa in september. <EOS>

Andrew Ng

在第二步的時候已經選擇出三個詞對，每一個輸出都會是下一個輸入， $P(y^{<1>}, y^{<2>} | x)$ ，一樣的方式各別選擇在10000個詞彙內最高機率的第三個單字 $P(y^{<3>} | x)$ (x為前兩個單字)。從30000個詞對中選擇出前三個最高機率單字，最後找到jane visits africa in september.<EOS>
 如果B的設置為1，那就是每次選擇一個，這種模型就是Greedy search。

3-4. Refinements to beam search



deeplearning.ai

Sequence to sequence models

Refinements to beam search

課程說明：優化beam search

Length normalization

Length normalization

$$\arg \max_y \prod_{t=1}^{T_y} P(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>})$$

$\log P(y^{<1>} | x) = \log P(y^{<2>} | x, y^{<1>}) + \dots + \log P(y^{<T_y>} | x, y^{<1>}, \dots, y^{<T_y-1>})$

$$\arg \max_y \sum_{t=1}^{T_y} \log P(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>})$$

$\log P(y^{<1>} | x) \leftarrow$
 $P(y^{<1>} | x) \leftarrow$

$T_y = 1, 2, 3, \dots, 30.$

$$\rightarrow \frac{1}{T_y^\alpha} \sum_{t=1}^{T_y} \log P(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>})$$

$\alpha = 0.7$
 $d = 1$
 $d = 0$

Andrew Ng

beam search所做就是最大化每一步的概率，上圖式子的連續乘積即為

$P(y^{<1>} \dots y^{<t>} | x) = P(y^{<1>} | x)P(y^{<2>} | x, y^{<1>}) \dots P(y^{<T_y>} | x, y^{<1>} \dots y^{<T_y-1>})$ ，每一個概率值都是小於1，很多小於1的數連乘之後會造成數值下溢(numerical underflow)，這會導致電腦的浮點數精度不足，因此實務上我們會取log，乘積的log就會變成log的求和。log是一個單調的遞增函數，所以最大化 $\log P(y|x)$ 跟最大化 $P(y|x)$ 是一樣的。

此方式的求和項會造成很多的小於1的數值連乘，會造成短句的機率高於長句，因此我們不再取最大值，而是改取平均，取概率對數值的平均(除輸出句子長度 T_y)，以此減少對輸出長度較長句子的懲罰，實務上會加入超參數 α 來調控分母，當 $\alpha=1$ 即標準Length normalization，為 $\alpha=0$ 即不做Length normalization。

Beam search discussion

Beam search discussion

Beam width B?

$| \rightarrow 3 \rightarrow 10, \quad 100, \quad 1000 \rightarrow 3000$

large B: better result, slower
small B: worse result, faster

Unlike exact search algorithms like BFS (Breadth First Search) or DFS (Depth First Search), Beam Search runs faster but is not guaranteed to find exact maximum for $\arg \max_y P(y|x)$.

Andrew Ng

B的設置愈大，選擇愈多，找到的句子就可能越好，但相對的計算成本也高，實務上較常見的設置為10。不同於DFS與BFS的精度搜尋，Beam search不保證能找到最好的。

3-5.Error analysis on beam search



deeplearning.ai

Sequence to sequence models

Error analysis on beam search

課程說明：beam search誤差分析

Example

Example

Jane visite l'Afrique en septembre.

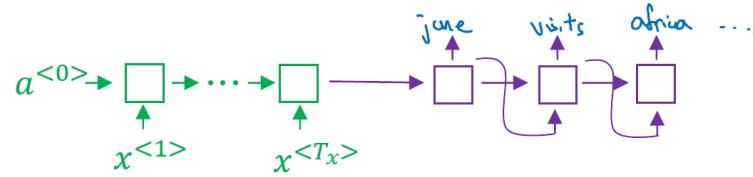
$\rightarrow RNN$
 \rightarrow Beam Search

B↑

Human: Jane visits Africa in September. (y^*)

Algorithm: Jane visited Africa last September. (\hat{y}) ←

$$RNN \text{ computes } P(y^*|x) \geq P(\hat{y}|x)$$



Andrew Ng

『Jane vistie l'Afrique en septembre.』

假設訓練資料集中有這麼一筆人工翻譯資料『Jane visits Africa in September.』，標記為(y^*)，而利用演算法執行之後的翻譯為『Jane vistis Africa last September.』，標記為(\hat{y})，這已經偏離原意了。

神經網路的模型為RNN，seq2seq，encoder與decoder，並且超參數為B的Beam search執行，我們需要知道的是，誤差來自於RNN還是來自於Beam search，加入數據集與增加B都是一個方式，但都可能無法得到期望的回應。

RNN計算著 $P(y|x)$ ，我們能做的分析即是，計算 $P(\hat{y}|x)$ 與 $P(y^*|x)$ ，比較兩個值的大小，以此判斷誤差原因。

Error analysis on beam search

Error analysis on beam search

Human: Jane visits Africa in September. (y^*)

$$P(y^*|x)$$

$$P(\hat{y}|x)$$

Algorithm: Jane visited Africa last September. (\hat{y})

Case 1: $P(y^*|x) > P(\hat{y}|x)$ ←

$$\arg \max_y P(y|x)$$

Beam search chose \hat{y} . But y^* attains higher $P(y|x)$.

Conclusion: Beam search is at fault.

Case 2: $P(y^*|x) \leq P(\hat{y}|x)$ ←

y^* is a better translation than \hat{y} . But RNN predicted $P(y^*|x) < P(\hat{y}|x)$.

Conclusion: RNN model is at fault.

Andrew Ng

- $P(y^*|x) > P(\hat{y}|x)$
 - 這代表beam search選擇了 $P(\hat{y}|x)$ ，也代表了beam search無法給你一個能使 $P(y|x)$ 最大化的值。
- $P(y^*|x) < P(\hat{y}|x)$
 - 這代表RNN出了問題，才會選擇 $P(\hat{y}|x)$
 - 這邊沒有考慮Length normalization

Error analysis process

Error analysis process

Human	Algorithm	$P(y^* x)$	$P(\hat{y} x)$	At fault?
Jane visits Africa in September.	Jane visited Africa last September.	<u>2×10^{-10}</u>	<u>1×10^{-10}</u>	(B)
...	...	—	—	(R)
...	...	—	—	B R R R ...

Figures out what fraction of errors are “due to” beam search vs. RNN model

Andrew Ng

此圖表方式記錄Beam Search與RNN之間的差異比，來了解調整優化的方向，如果是Beam search，那就可以加大B，如果是RNN就可以判斷是加入正規化，或是加大數據、...調校方式。

3-6.Bleu score(optional)



deeplearning.ai

Sequence to sequence models

Bleu score
(optional)

課程說明 : Bilingual evaluation understudy 評估機器翻譯系統

Evaluating machine translation

Evaluating machine translation

French: Le chat est sur le tapis.

Bleu
bilingual evaluation understudy

Reference 1: The cat is on the mat. ↪

Reference 2: There is a cat on the mat. ↪

MT output: the the the the the the.

Precision:

Modified precision:

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Andrew Ng

給定一句法語『Le chat est sur le tapis.』，文本集中可能有不同人翻譯的不同英文，但意涵相同，Bleu score做的就是替機器翻譯打分數，衡量好壞。只要機器翻譯出來的句子跟人工翻譯的愈接近就會得到一個高的分數。bleu即是用來代理確認機器翻譯的句子是否曾出現於人工標記文本，再予以確認衡量。

假設上面句子被機器翻譯為THE THE THE THE THE...，這很明顯的是一個非常糟的翻譯，如果以精準度來看the都出現在了句子內，準確度高達100%，但事實並非如此。但以改良後的精確度來看，我們會取該單字於參考句中出現的最高次數為主，意思是，the在參考一中出現兩次，在參考二出現一次，取最高次數，這句子得分為 $\frac{2}{7}$ ，但這只是針對單一單字的計算，我們如果希望針對多詞出現計算該如何處理。

Bleu score on bigrams

Bleu score on bigrams

Example: Reference 1: The cat is on the mat. ↪

Reference 2: There is a cat on the mat. ↪

MT output: The cat the cat on the mat. ↪

	Count	Count _{clip}	
the cat	2 ↪	1 ↪	
cat the	1 ↪	0	
cat on	1 ↪	1 ↪	
on the	1 ↪	1 ↪	
the mat	1 ↪	1 ↪	

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Andrew Ng

上面句子來看(MT output指機器翻譯)，我們從機器翻譯取得多組詞對(不重覆)，並計算各詞對出現次數，再各別比對出現於參考翻譯的最高次數之後加總，分母即為從機器翻譯中取得的詞對總數。

Bleu score on unigrams

Bleu score on unigrams

Example: Reference 1: The cat is on the mat.

$$P_1 \cdot P_2 = 1.0$$

Reference 2: There is a cat on the mat.

→ MT output: The cat the cat on the mat. (↑)

$$P_1 = \frac{\sum_{\text{unigram} \in \hat{y}} \text{count}_{\text{clip}}(\text{unigram})}{\sum_{\text{unigram} \in \hat{y}} \text{count}(\text{unigram})}$$

↑
Unigram

$$P_n = \frac{\sum_{n\text{-gram} \in \hat{y}} \text{count}_{\text{clip}}(n\text{-gram})}{\sum_{n\text{-gram} \in \hat{y}} \text{count}(n\text{-gram})}$$

↑
n-gram

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Andrew Ng

$$p_n = \frac{\sum_{n\text{-gram}} \text{Countclip}(n\text{-gram})}{\sum_{n\text{-gram}} \text{Count}(n\text{-gram})}$$

分母為於機器翻譯中取的的詞對總數，分子為出現在參考文本出現的最高次數，n代表計算幾個單字組成的詞對。

Bleu detail

Bleu details

p_n = Bleu score on n-grams only

P_1, P_2, P_3, P_4

Combined Bleu score: $\text{BP} \exp\left(\frac{1}{4} \sum_{n=1}^4 p_n\right)$

BP = brevity penalty

$$\text{BP} = \begin{cases} 1 & \text{if } \text{MT_output_length} > \text{reference_output_length} \\ \exp(1 - \text{MT_output_length}/\text{reference_output_length}) & \text{otherwise} \end{cases}$$

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Andrew Ng

p_n 即為計算出n元詞組改後后的精準度，如果n取1到4，那就會加總之後平均， $\text{BP} \exp\left(\frac{1}{n} \sum_1^n p_n\right)$ ，BP指簡短懲罰(brevity penalty)，在短句子也許各別單字都會出現在參考中，因此利用BP來做一個調整因子，調整較短翻譯可能得到較高的得分問題。

3-7.Attention model intuition



deeplearning.ai

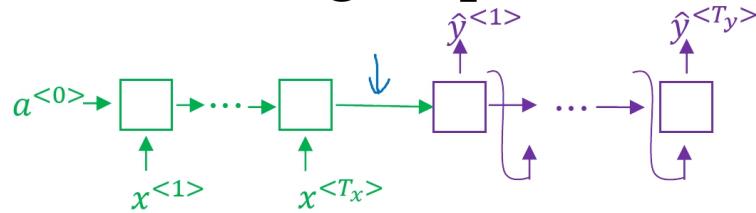
Sequence to sequence models

Attention model intuition

課程說明 : Attention model

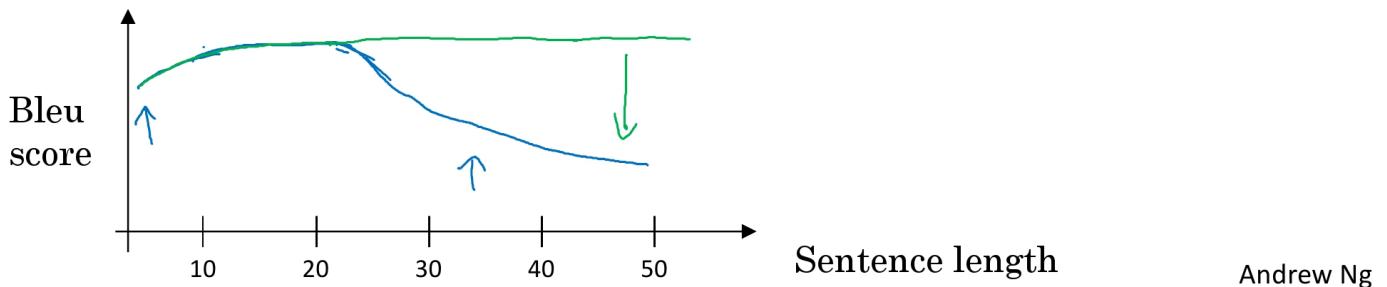
The problem of long sequences

The problem of long sequences



Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

Jane went to Africa last September, and enjoyed the culture and met many wonderful people; she came back raving about how wonderful her trip was, and is tempting me to go too.

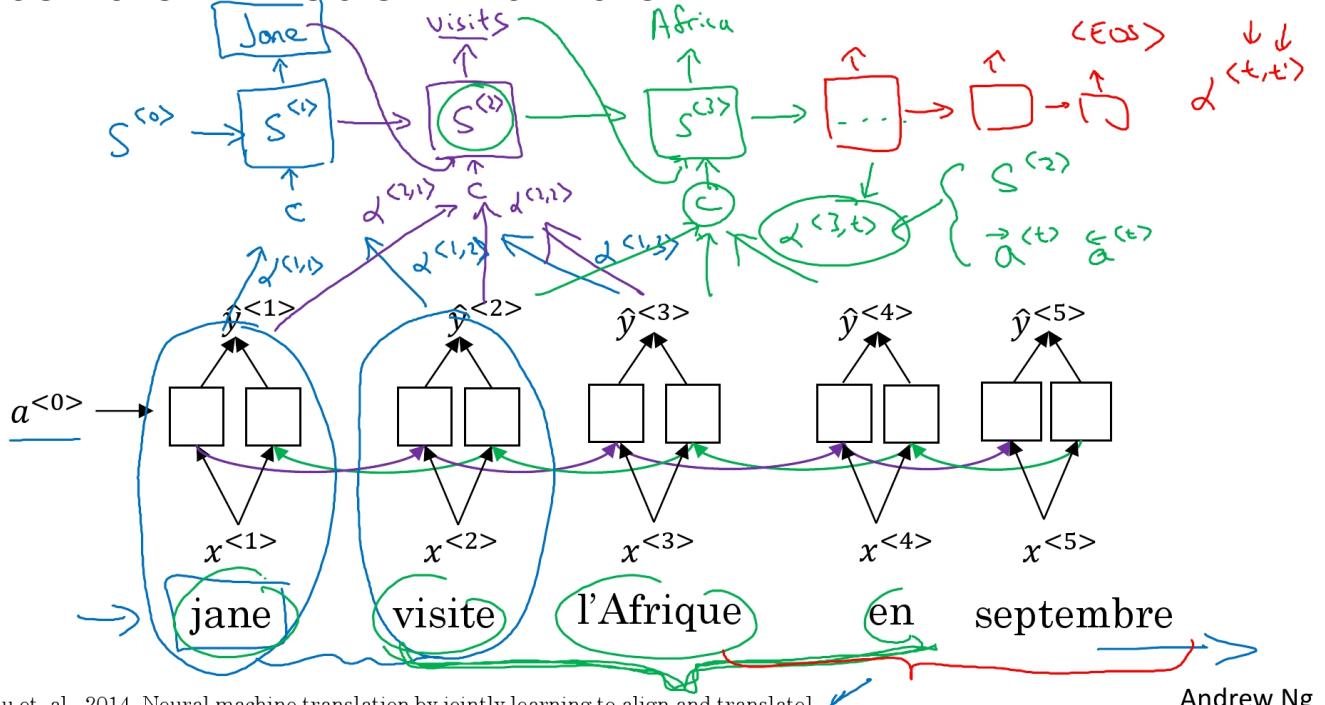


給定一段很長的句子，綠色區塊(encoder)要做的就是將整段句子讀入並記憶，再傳遞給紫色區塊(decoder)，人工翻譯並不會這麼做，他可能讀某一小段，翻一小段。

以bleu score來看，走勢會隨著長度增加而得分一直下降，這是因為在神經網路中記憶長句子是非常困難的。透過attention model，會讓模型效能呈現水平走勢，不會再是一個下降趨勢。

Attention model intuition

Attention model intuition



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

Andrew Ng

用一個簡短句子為例『jane visite l'Afrique en Septembre』，以BRNN計算每個輸入單字的特徵，接著利用另一個RNN來生成英文翻譯(以 $s^{<0>}$, $s^{<1>}$...表示)，我們希望第一個單字是『jane』，相對應的，我們應該先看原文(法文)的那一個部份？

注意力模型(Attention model)會計算權重(attention weights)，以 $\alpha(1, 1)$ 表示生成一個詞時應該放多少注意力在第一個輸入(jane)，以 $\alpha(1, 2)$ 表示生成第一個詞時應該放多少注意力在第二個輸入(visite)...這是RNN的第一步。

第二步， $s^{<2>}$ 會有新的注意力權重 $\alpha(2, 1)\alpha(2, 2)$ ，在計算 $s^{<2>}$ 的同時， $s^{<1>}$ 會是輸入，這就會是一個上下文區間，更細部的資訊下一節會說明，

3-8. Attention model

Sequence to sequence models

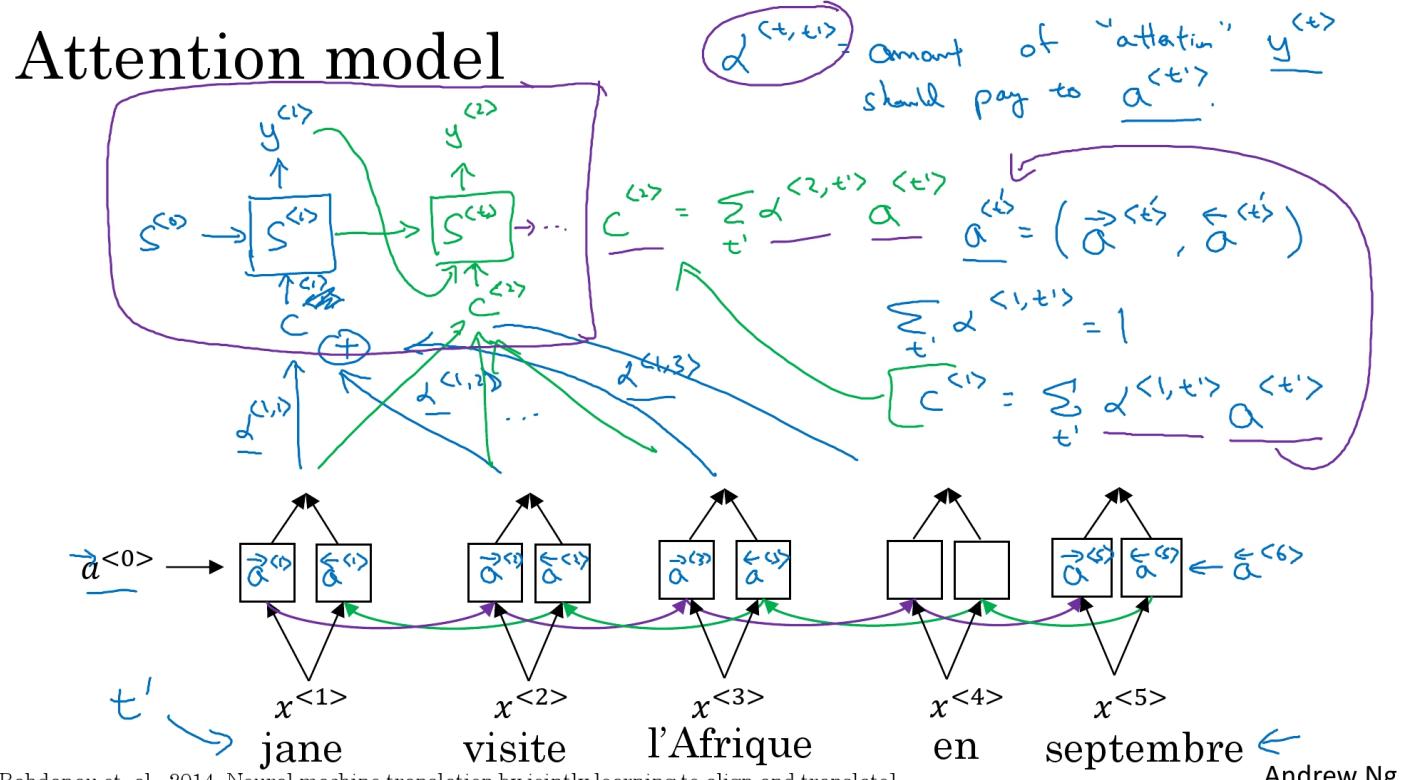


deeplearning.ai

Attention model

課程說明：細部探討注意力模型

Attention model



與上節相同案例，有一個BRNN(或GRU、LSTM也可以)去計算每個詞的特徵，因為是雙向，每個輸入都會有兩個輸出(前向、反向)，統一以 $a^{<t'>}$ 來表示。

s 用以表示輸出的翻譯， $\alpha(1, 1)$ 表示注意力，輸出得到 c (上下文來自不同時步被注意力權重除權的特徵值)。

$c^{<1>} = \sum_{t'} \alpha^{<1, t'>} a^{<t'>} \quad , \quad \alpha^{<t, t'>} \text{即是總合 } y^{<t>} \text{花在 } a^{<t'>} \text{上的注意力} \quad \text{也就是當你在時步 } t \text{生成輸出詞的時候，應該花多少注意力在 } t'.$

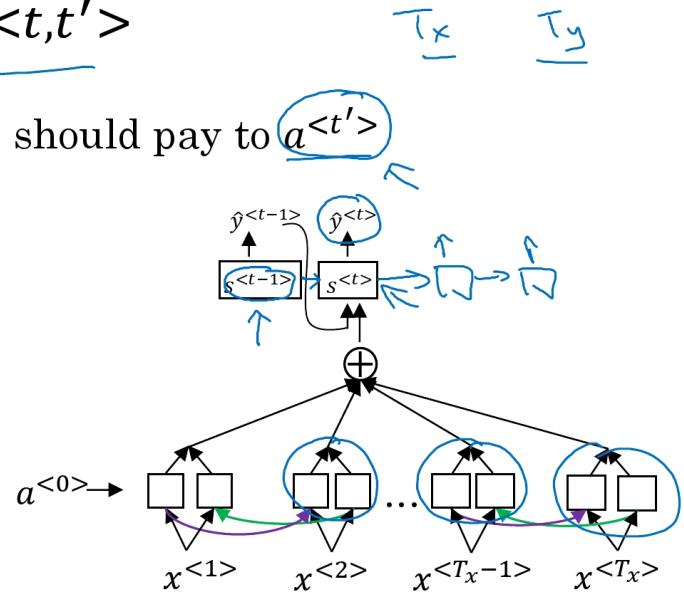
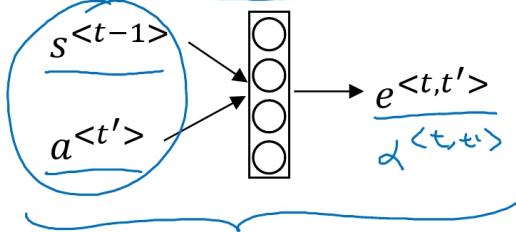
第二個詞相同的，有屬於它的上下文，並且以上一個輸出為輸入...以此循環至最後

Computing attention $\alpha^{<t, t'>}$

Computing attention $\underline{\alpha^{<t, t'>}}$

$\alpha^{<t, t'>} = \text{amount of attention } y^{<t>} \text{ should pay to } a^{<t'>}$

$$\alpha^{<t, t'>} = \frac{\exp(e^{<t, t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t, t'>})}$$



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate] ↩
 [Xu et. al., 2015. Show, attend and tell: Neural image caption generation with visual attention] ↩

Andrew Ng

$\alpha^{<t, t'>}$ 是 $y^{<t>}$ 應該花費在 $a^{<t'>}$ 上的注意力

$$\alpha^{<t, t'>} = \frac{\exp(e^{<t, t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t, t'>})} \quad (\text{式子內的關鍵在於softmax，確保權重加總為1})$$

實作的部份可以利用一個小型的神經網路，以 $s^{<t-1>}$ (上一個時步的隱藏狀態)與 $a^{<t'>}$ 為輸入，讓它去學習參數，得到輸出，事實證明，學習之後的效果是不錯的。

這個方式的缺點在於，時間複雜度為 $O(n^3)$ ，如果你有 T_x 個輸入單字與 T_y 個輸出單字，注意力參數就會是兩者相乘。(注意力模型亦應用在照片加標題的領域。)

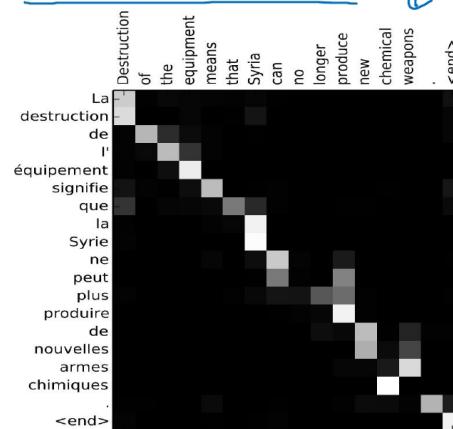
Attention examples

Attention examples

July 20th 1969 → 1969 – 07 – 20

23 April, 1564 → 1564 – 04 – 23

Visualization of $\alpha^{<t,t'>}$:



Andrew Ng

可以利用注意力模型來將日期轉換為標準化格式，也可以嚮式將注意力模型可視化。

3-9. Speech recognition



deeplearning.ai

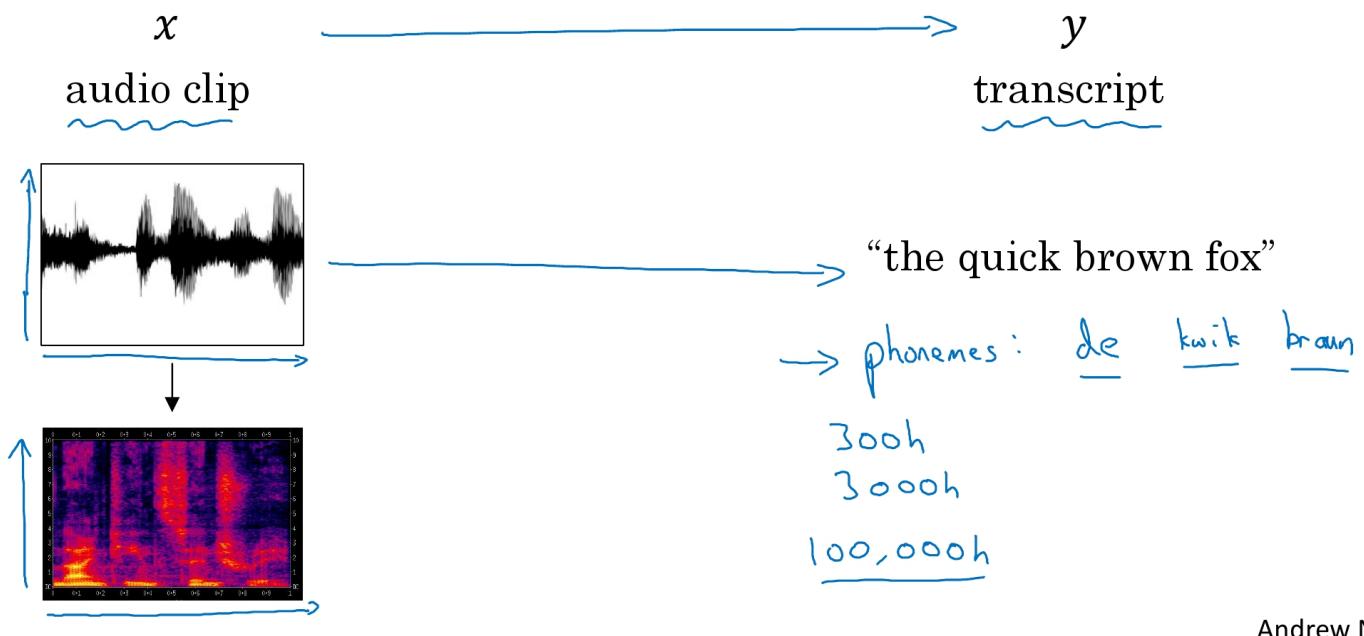
Audio data

Speech recognition

課程說明：seq2seq in 語音辨視

Speech recognition problem

Speech recognition problem

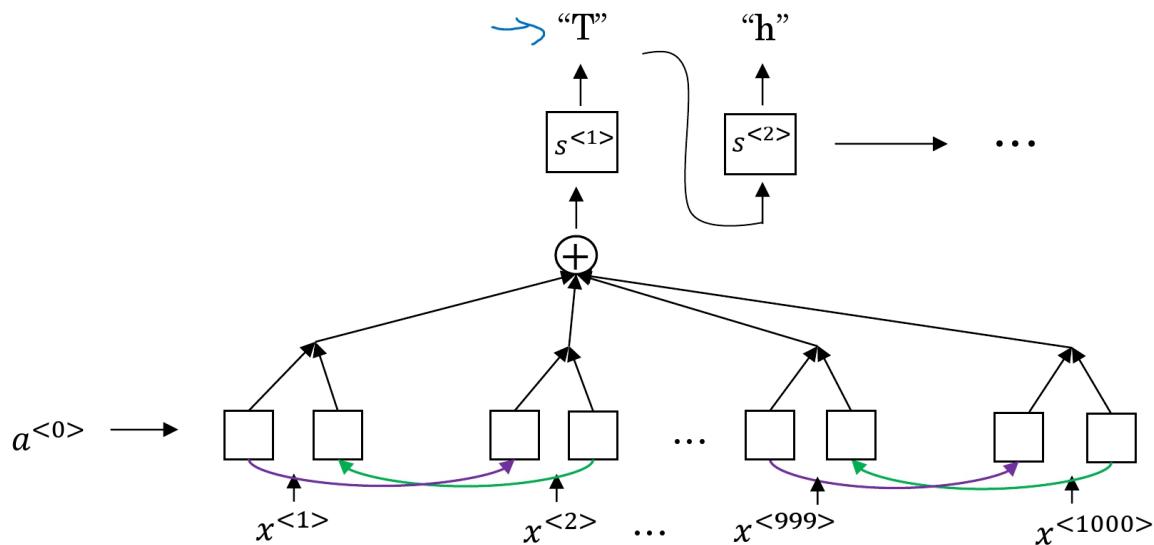


語音辨視問題是給定一段語音之後生成一個相對應的文本，音譜橫軸是時間，縱軸是頻率，不同的顏色顯示了聲波能量的大小。

早前語音辨視是以音位(phonesmes)來構建，即人工設置的基本單元，但在end-to-end模型中，這種表示法已經不必要了。輸入音頻之後直接輸出文本，不需要再提取音位。這麼做的前提就是需要有一個非常大的數據庫。(300hr或是3000hr，商業系統已經超過10000hr，持續加大中)

Attention model for speech recognition

Attention model for speech recognition



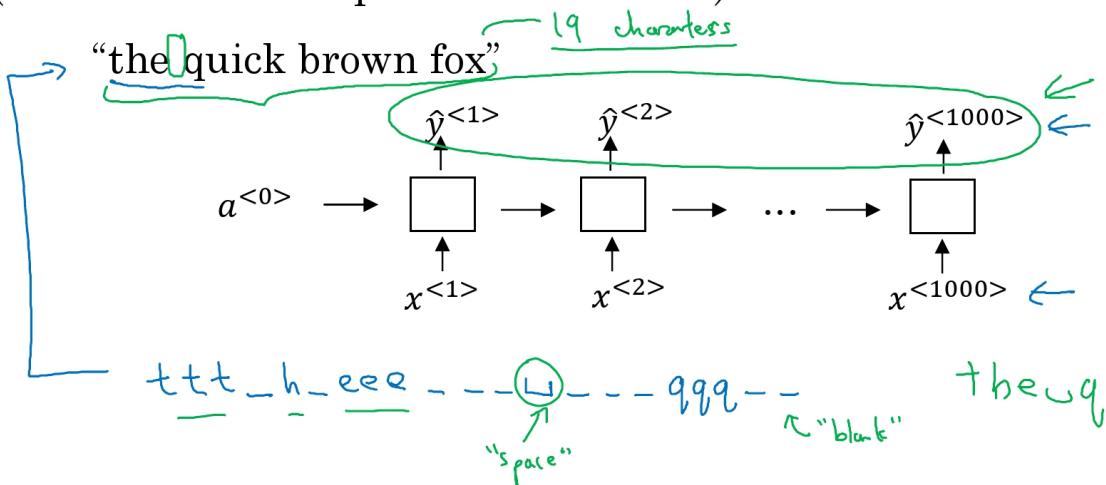
Andrew Ng

一直在討論的注意力模型也可以應用在語音辨視上

CTC cost for speech recognition

CTC cost for speech recognition

(Connectionist temporal classification)



Basic rule: collapse repeated characters not separated by “blank” ↴

[Graves et al., 2006. Connectionist Temporal Classification: Labeling unsegmented sequence data with recurrent neural networks] Andrew Ng

也可以用CTC(Connectionist Temporal Classification)損失函數來做語音辨視，假定有人說了一句『the quick brown fox』，注意到時步的部份有1000，語音辨視中，輸入的時步數量要比輸出的時步數量多很多，一段10秒的音頻，並且特徵是100赫茲(每秒100個樣本)，就有1000個輸入了。CTC允許字段上的重覆出現，讓輸出跟輸入的數量相等，再透過收折，讓重覆字段消失來產生最後的輸出。

3-10.Trigger word detection



deeplearning.ai

Audio data

Trigger word
detection

課程說明：觸發喚醒系統(你好 百度)

What is trigger word detection?

What is trigger word detection?



Amazon Echo
(Alexa)



Baidu DuerOS
(xiaodunihao)



Apple Siri
(Hey Siri)



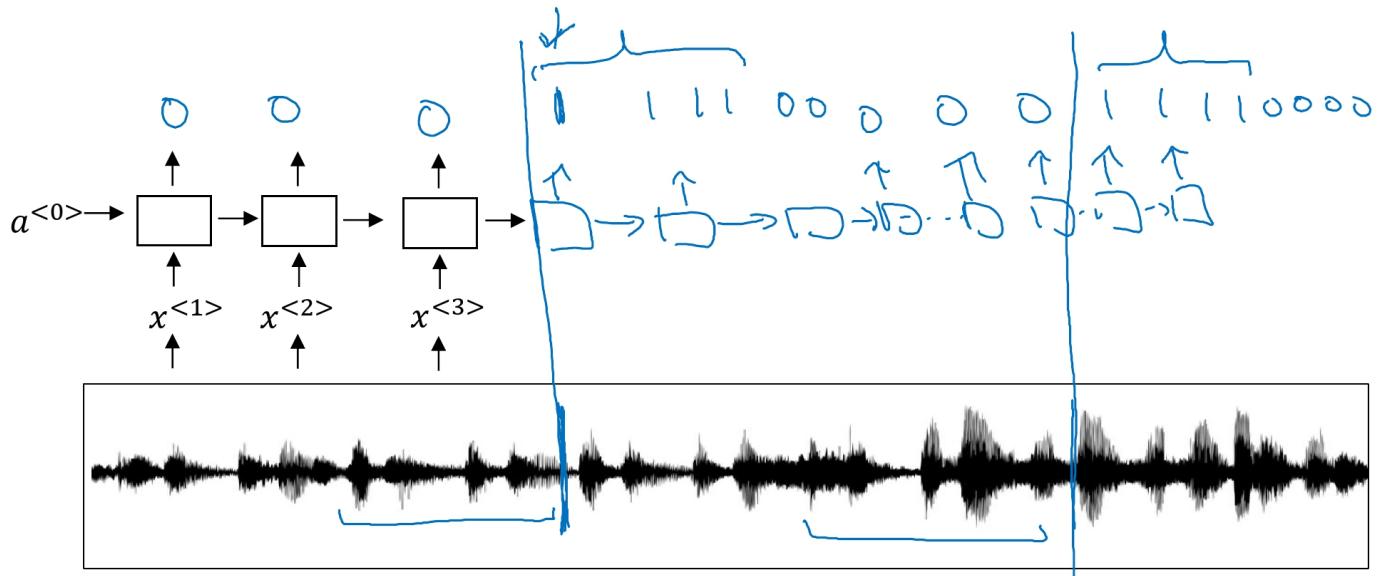
Google Home
(Okay Google)

Andrew Ng

alex、百度、google這類以一句話喚醒系統，即為trigger word detection system，

Trigger word detection algorithm

Trigger word detection algorithm



Andrew Ng

目前這類演算法尚未成熟，我們要做的就是將一句話(輸入音頻)計算出它的聲譜圖特徵，得到特徵向量，將之丟入RNN。假設音頻中有一段關鍵字(你好 百度，或OK GOOGLE)，那就將該時步區間外輸出為0，區間內為1，缺點是建構了一個不平衡的訓練集，0的數量比1多太多了。

3-11.



deeplearning.ai

Conclusion

Summary and thank you

Specialization outline

1. Neural Networks and Deep Learning
2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
3. Structuring Machine Learning Projects
4. Convolutional Neural Networks
5. Sequence Models

Andrew Ng

Deep learning is a super power



Please buy this
from shutterstock

and replace in
final video.



www.shutterstock.com · 331201091

Andrew Ng

Thank you.

- Andrew Ng