

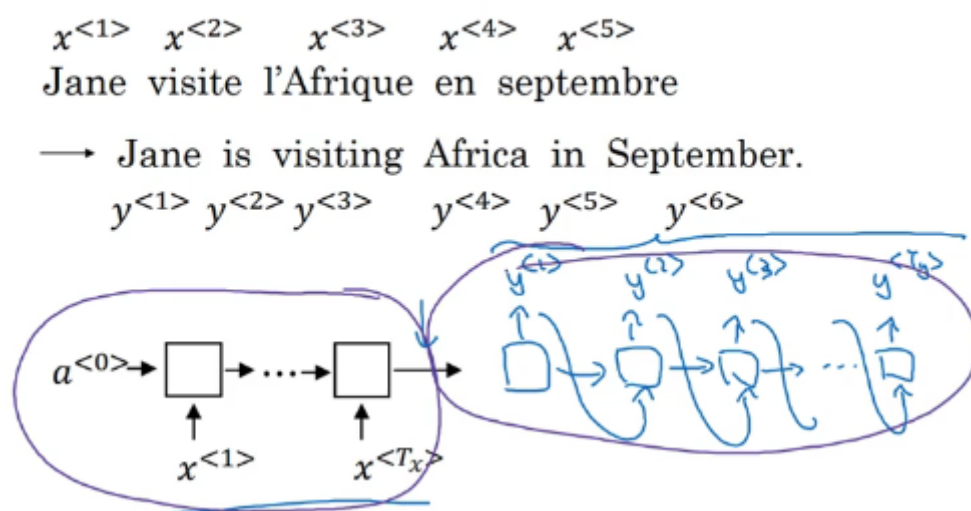
1. Basic model

1.1 Sequence to sequence model

该模型常用于机器翻译

- Input: $x^{<1>}, x^{<2>}, \dots, x^{<T_x>}$ 。每个 $x^{<t>}$ 代表法语句子中的单词。
- Output: $y^{<1>}, y^{<2>}, \dots, y^{<T_y>}$ 。每个 $y^{<t>}$ 代表法语句子中的单词。
- 对于机器翻译使用的seq-to-seq模型，如果有大量的语料库，则可以得到一个有效的翻译模型。
- 模型的前半部分使用一个编码网络对法语进行编码，后半部分使用一个解码网络生成对应的英文翻译。

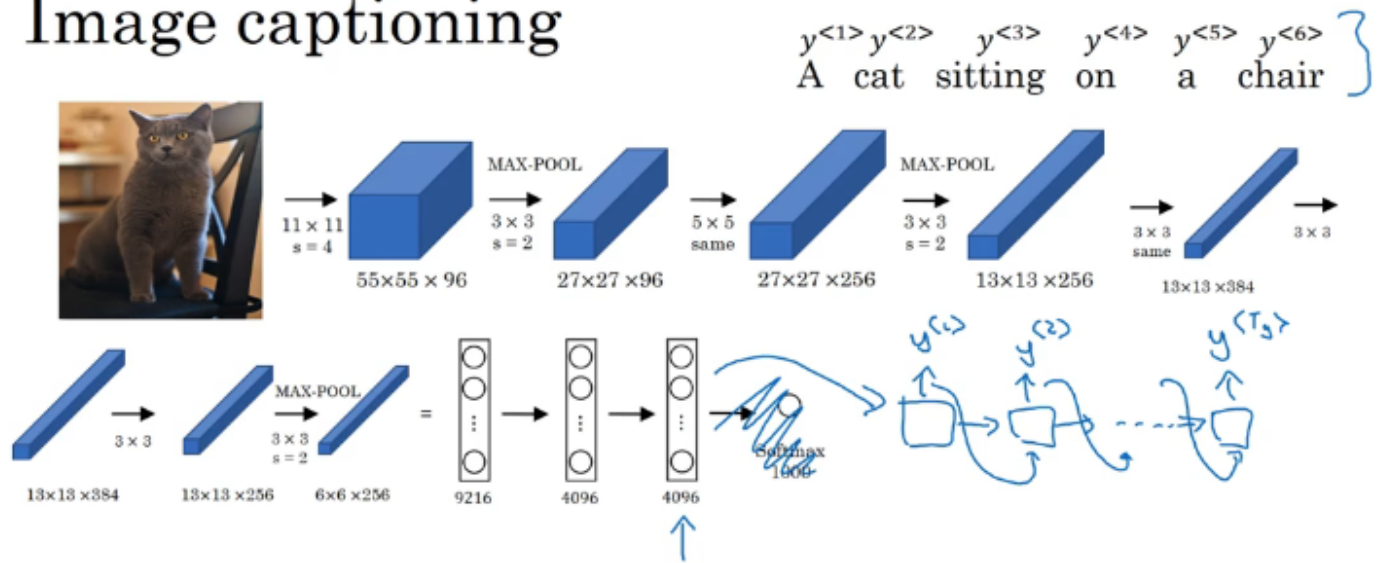
Sequence to sequence model



1.2 Image to sequence model

- Input: 图像
- Output: 描述图像的句子
- 利用AlexNet学习图像的编码向量，利用RNN输出对应的句子。

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]

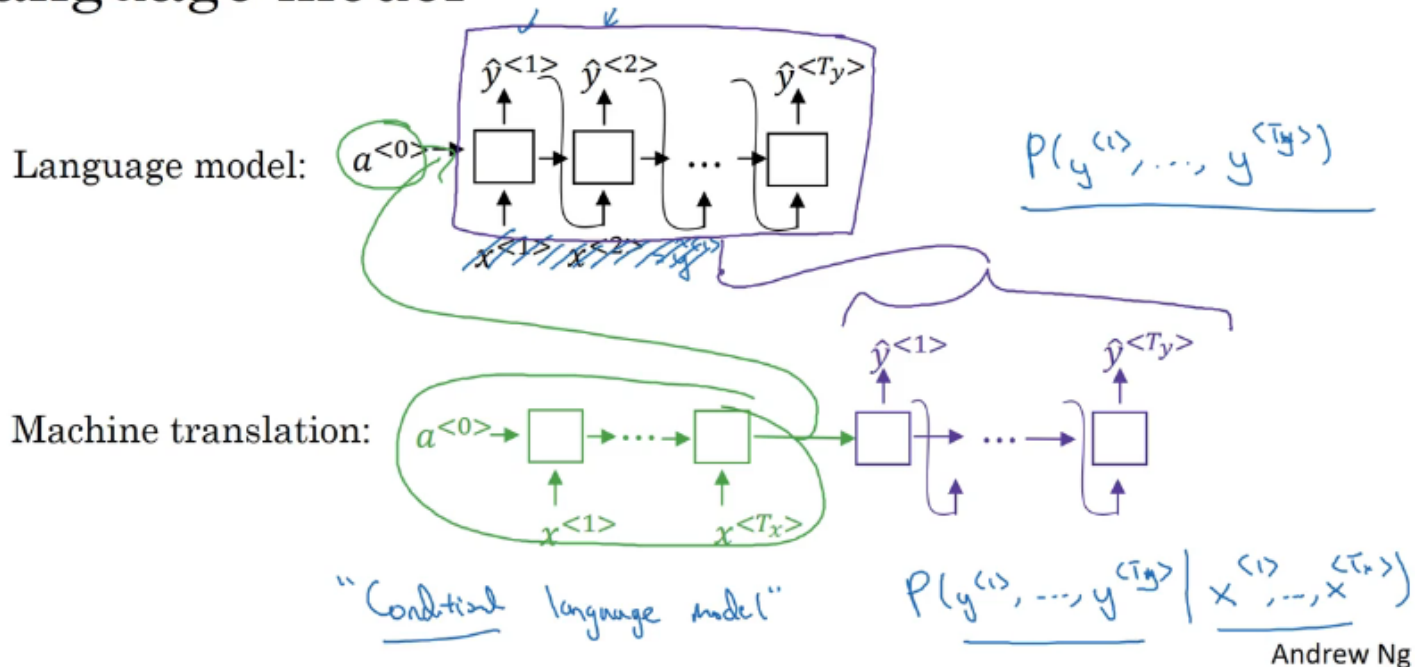
Andrew Ng

2. 选择最可能的句子

2.1 Machine translation as building a conditional language model

- 语言模型：通过评估输出的概率来生成新的句子。总是以零向量为第一个时间步的输入。
- 机器翻译：包含了编码网络和解码网络。解码网络与语言模型的结构是相似的。因此相比语言模型来说，机器翻译可以称作条件语言模型，其输出句子的概率是相对于输入的条件概率。

Machine translation as building a conditional language model



2.2 Finding the most likely translation

对于各种翻译的结果，我们要找到一个条件概率最大的英文句子作为输出，而不是对得到的分布进行随机采样。因此一个最重要的步骤就是设计一个找到条件概率最大的结果的算法。目前常用的是束搜索 (Beam search)。

Finding the most likely translation

Jane visite l'Afrique en septembre.

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

English

French

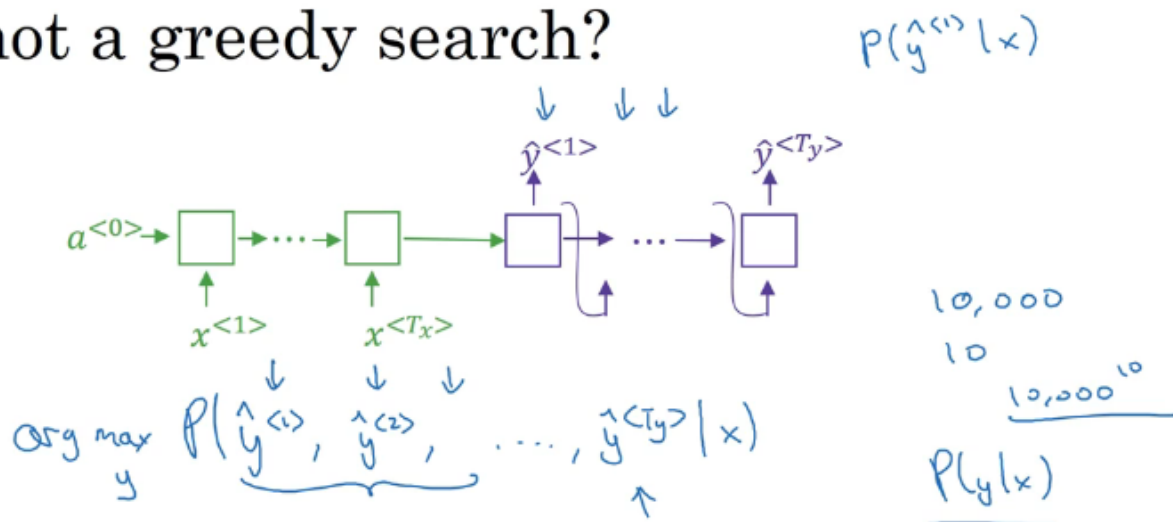
- Jane is visiting Africa in September.
- Jane is going to be visiting Africa in September.
- In September, Jane will visit Africa.
- Her African friend welcomed Jane in September.

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

2.3 Why not a greedy search?

使用贪心算法，在生成第一个词的分布后，贪心搜索会挑选出最有可能输出的第一个词语，在跳出第二个最有可能的词语。而对于我们的机器翻译模型来说，我们需要通过模型一次性挑选出整个序列，使得整体的条件概率最大化。而且如果单词库的词汇很多（=10000），去计算每一种单词的组合显然不现实。因此贪心算法在机器翻译中是不可行的。

Why not a greedy search?



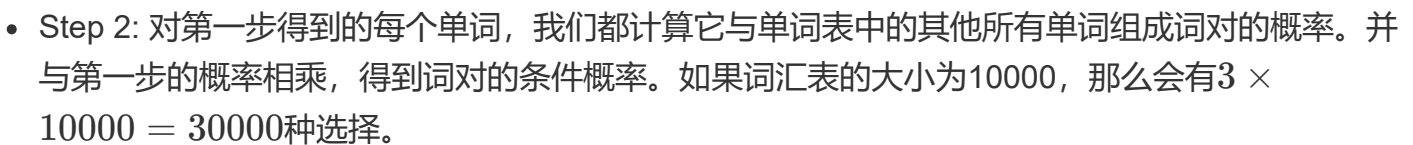
- Jane is visiting Africa in September.
- Jane is going to be visiting Africa in September.
 $P(\text{Jane is going} | x) > P(\text{Jane is visiting} | x)$

Andrew Ng

3. Beam search

- Step 1: 设置集束宽度的大小为3，取前3个最大输出概率的单词并保存。

$B = 3$ (beam width)



Beam search algorithm ($B=3$)

Step 1

10000

a
⋮
in
⋮
jane
⋮
~~september~~
⋮
zulu

$y^{(1)}, y^{(2)}$

Step 2

a
aaron
september
visit
zulu

a
aaron
is
visit
zulu

a
⋮
Zulu

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

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

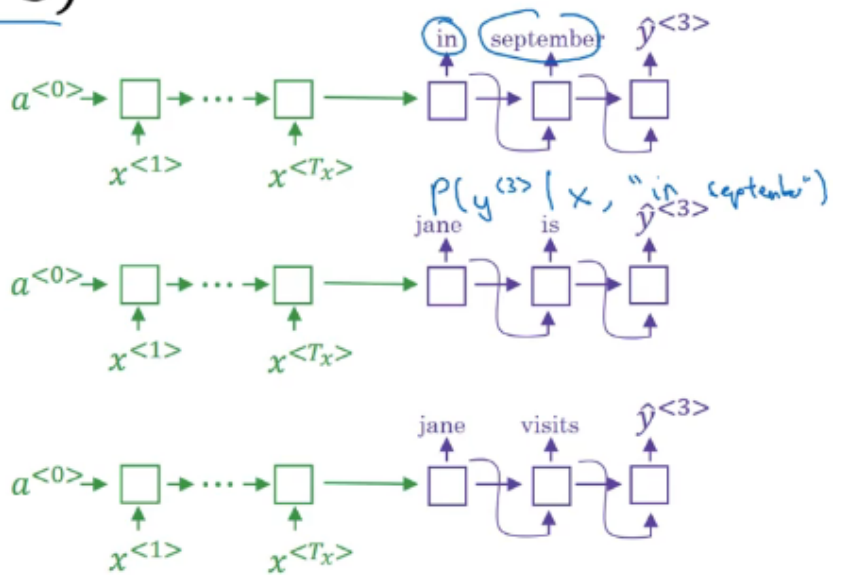
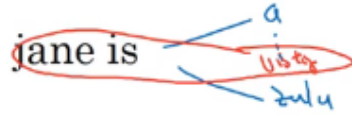
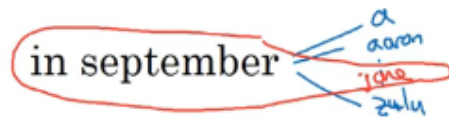
$P(y^{(2)} | x, y^{(1)})$

Andrew Ng

- Andrew Ng

Beam search ($B = 3$)

$B=1 \rightarrow$ greedy search



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

jane visits africa in september. <EOS>

4. Refinements to beam search

4.1 Length normalization

我们的优化目标:

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

这个结果可能导致最后的值太小，因此我们通常取log:

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

这个优化目标可能会使模型倾向于输出更短的句子，对结果产生影响。我们可以进行归一化，相比于直接除以句子的长度，我们可以加上一个参数 α ，使输出更加柔和。

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

4.2 Beam search discussion

- Beam width: 宽度越大，考虑的情况越多，但是计算成本也随之增加。常见的设置为10。
- 相比于精确的搜索算法，如BFS或者DFS，集束搜索的运行速度很快，但不能保证找到准确的最大值。

5. Error analysis on beam search

通过我们的模型，计算人类翻译的概率 $P(y^*|x)$ 和模型翻译的概率 $P(\hat{y}|x)$ 。比较二者的大小，便能知道是因为集束搜索算法的问题还是RNN模型的问题。

Example

Jane visite l'Afrique en septembre.

→ RNN

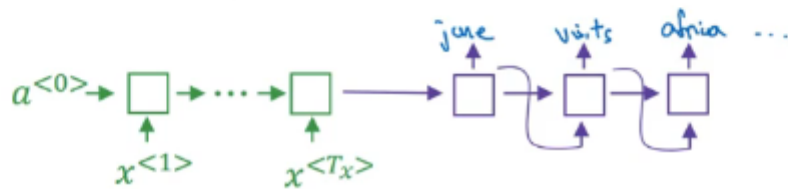
→ Beam Search

BT

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

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

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



- $P(y^*|x) > P(\hat{y}|x)$: 这意味着集束搜索并没有给你带来能使 $P(y|x)$ 最大化的 y 值。
- $P(y^*|x) \leq P(\hat{y}|x)$: 人类翻译的结果应该是比机器翻译更好的，但是RNN模型却预测 $P(y^*|x) < P(\hat{y}|x)$ 。

Error analysis on beam search

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

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

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

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

Case 1: $P(y^*|x) > P(\hat{y}|x) \leftarrow$

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

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

Conclusion: Beam search is at fault.

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

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

在dev集上对各个句子进行检测，得到每个句子对应的错误情况。

Error analysis process

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

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

Andrew Ng

6. Bleu score

6.1 评估机器翻译

机器翻译的结果常常有多种正确且何使的翻译，但是我们很难对这些结果进行评估。引入Bleu score (Bleu, bilingual evaluation understudy) 来评估翻译的结果。

Ref 1和Ref 2都是非常好的翻译结果。Bleu score的理念就是观察机器翻译结果种每一个词是否出现在至少一个人工翻译结果中。

Evaluating machine translation

French: Le chat est sur le tapis.

→ Reference 1: The cat is on the mat. ← 2 appears

→ Reference 2: There is a cat on the mat. ←

→ MT output: the the the the the the the.

Precision: $\frac{7}{7}$ Modified precision: $\frac{2}{7}$ ← Count_{clip}("the") ← Count("the")

Bleu bilingual evaluation understudy

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

Andrew Ng

- Precision: 观察输出的每个词是否出现在参考中。但是对于图中糟糕的翻译结果，精确度却非常高。
- Modified precision: 每个单词设置得分的上限。

6. 二元组的Bleu score

我们可以以两个相邻的单词作为二元组 (bigram)来进行评估，得到改进的精确度。

对于不同的n-gram，改良的精确度为：

$$P_1 = \frac{\sum_{n\text{-grams} \in \hat{y}} \text{Count}_{\text{clip}}(n\text{-grams})}{\sum_{n\text{-grams} \in \hat{y}} \text{Count}(n\text{-grams})}$$

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

6.3 Bleu details

- 得到每种n-gram的Bleu score。
- 组合Bleu score: $BP \exp(\frac{1}{4} \sum_{n=1}^4 P_n)$

其中BP (brevity penalty)是简短惩罚，作为一个调节因子，对太短的翻译结果的翻译系统进行惩罚。

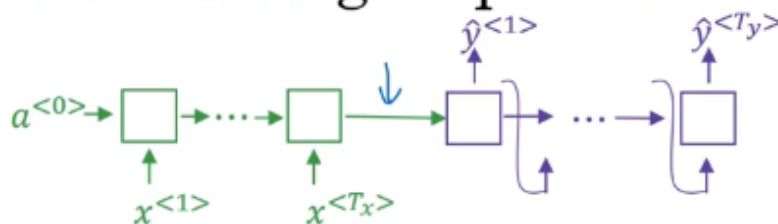
$$BP = \min(1, \exp(1 - \frac{reflength}{MTlength}))$$

7. 注意力模型

7.1 长句子存在的问题

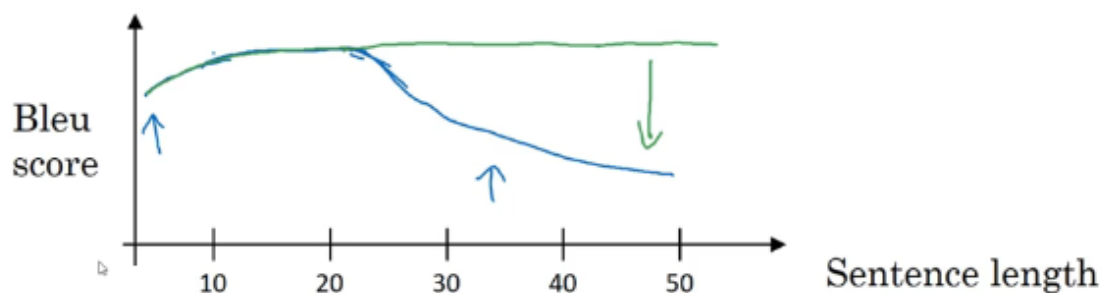
对于短句子来说，RNN模型能够获得非常好的结果。但是如果是长句子，翻译的结果就会变差。随着句子长度的增加，Bleu score会在超过一定值时下降。

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.

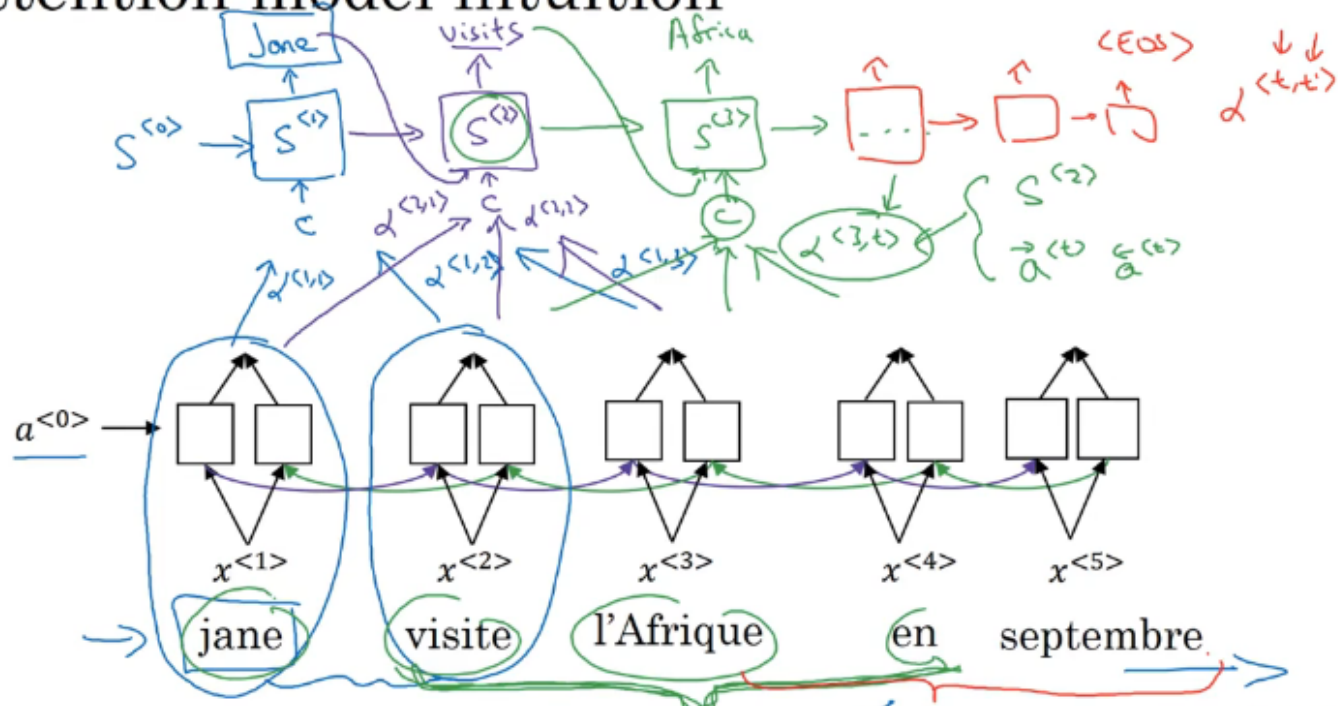


Andrew Ng

7.2 注意力模型介绍

对于机器翻译来说，对每个单词的输出影响较大的单词应该集中在附近几个单词。注意力机制会计算应该在一个RNN单元上花费多少注意力。

Attention model intuition



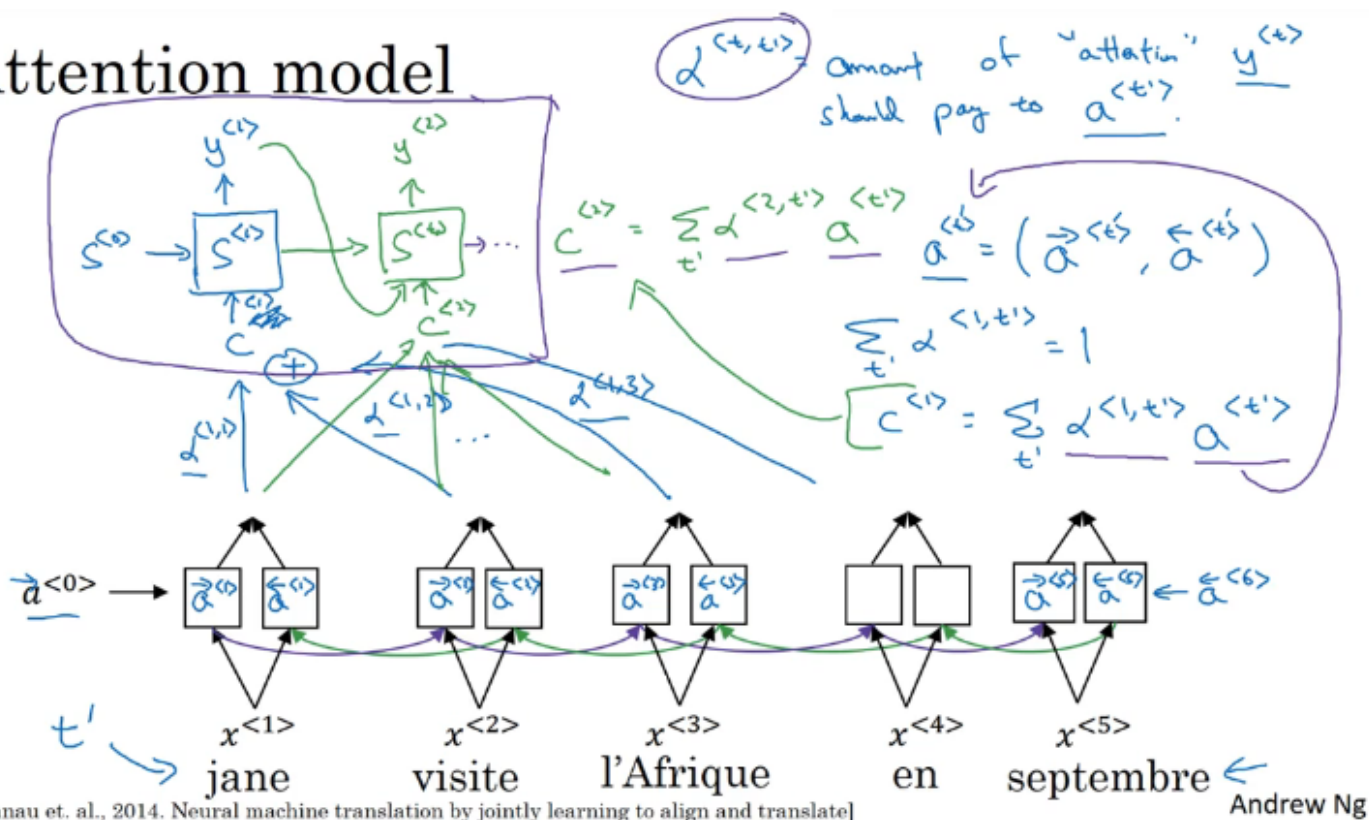
[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

Andrew Ng

我们以一个双向RNN模型翻译法语句子为例，其中每个cell可以是LSTM或者GRU。通过前向和后向传播，可以得到两个激活值 $\vec{a}^{<t'>}$ 和 $\overleftarrow{a}^{<t'>}$ 。使用 $a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$ 来表示。

对于英文输出，使用另一个RNN模型来构建。邻近单词的注意力权重为 $\alpha^{<t, t'>}$ ，表示英文单词 t 应该对法语单词 t' 花费多少关注度。可以得到输出单词的有关context的加权和 $C^{<t>}$ ，其中 $\sum_{t'} \alpha^{<t, t'>} = 1$ 且 $C^{<t>} = \sum_{t'} \alpha^{<t, t'>} a^{<t'>}$ 。

Attention model



7.3 计算注意力

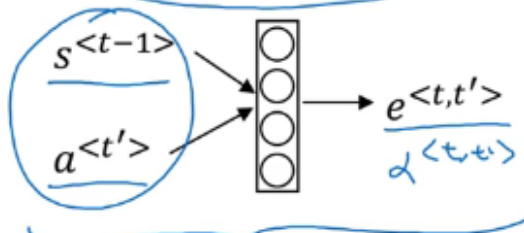
$\alpha^{(t,t')}$ 表示英语输出 $y^{(t)}$ 需要对法语单词的激活值 $a^{(t')}$ 付出的注意力。

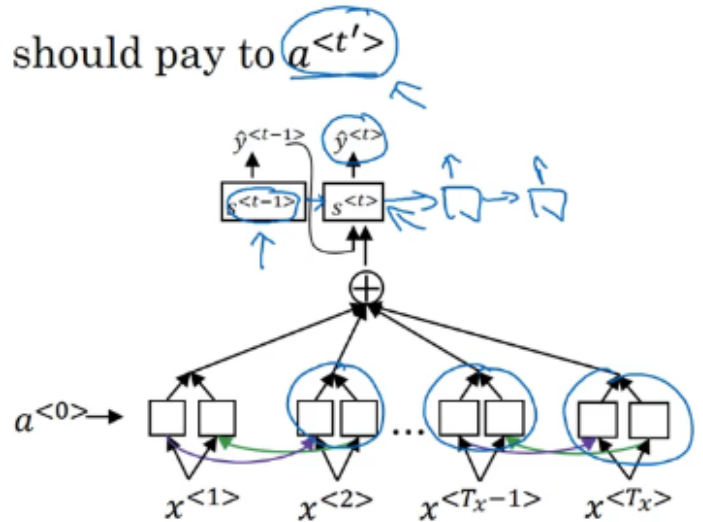
$$\alpha^{(t,t')} = \frac{\exp(e^{(t,t')})}{\sum_{t'=1}^{T_x} \exp(e^{(t,t')})}$$

$e^{(t,t')}$ 是通过一个浅层神经网络计算得到的，其值取决于输出RNN的前一步激活值 $s^{(t-1)}$ 以及输入RNN的当前步的激活值 $a^{(t')}$ 。

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

$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ 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

缺点就是时间复杂度高，具有 $O(T_x T_y)$ 的复杂度。但是机器翻译输入和输出句子一般不会太长，这样的复杂度还能接受。

7.4 注意力机制的例子

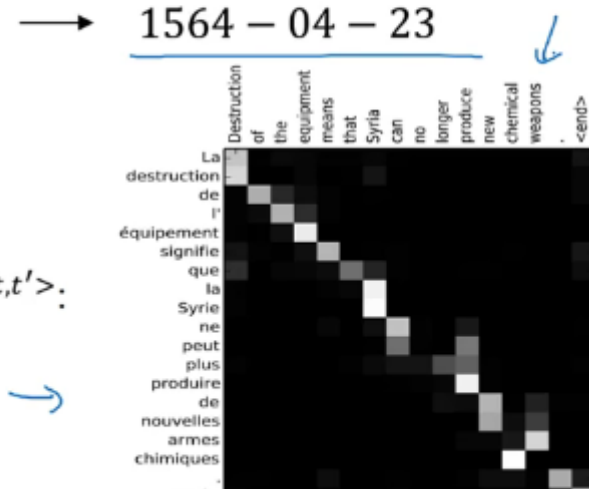
- 将不标准的时间格式转化为统一的时间格式。
- 注意力权重进行可视化。

Attention examples

July 20th 1969 → 1969 - 07 - 20

23 April, 1564 → 1564 - 04 - 23

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



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

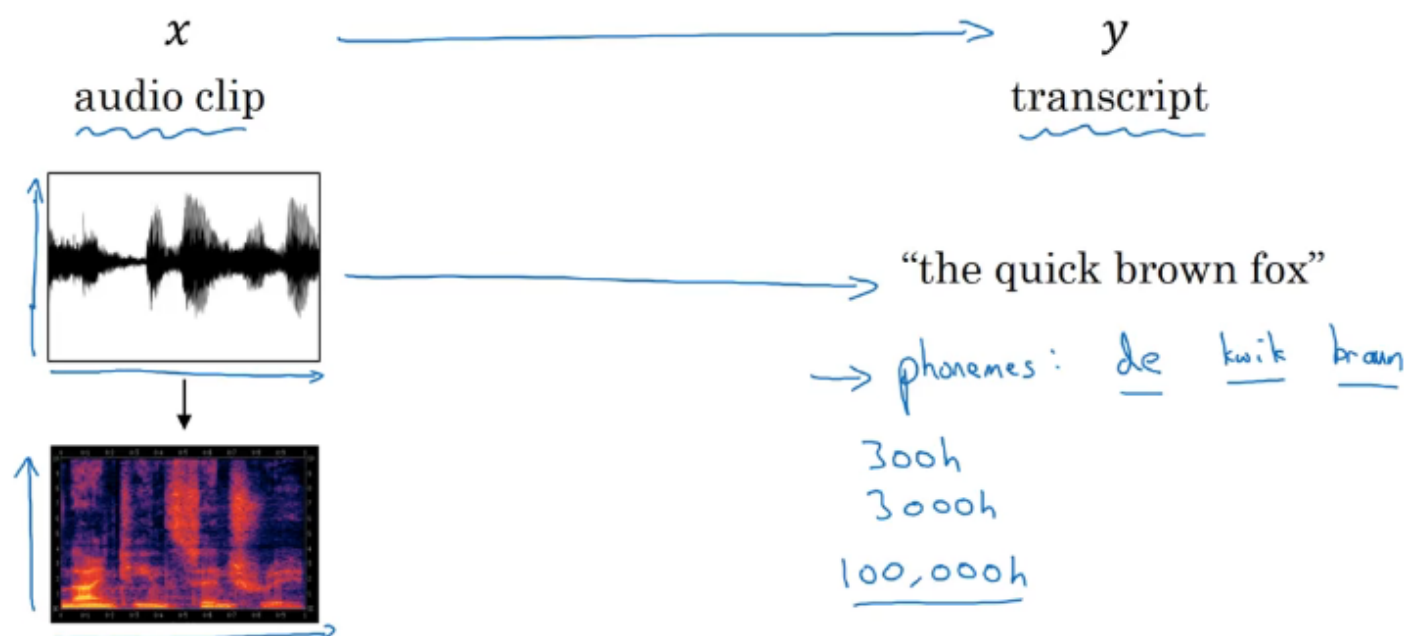
8. 语音识别

语音识别是将一段音频转化为相应的文本信息。

8.1 语音识别问题

借鉴了人耳的处理过程，将音频信号转化为声谱图，再将声谱图作为特征送到相应的算法中进行处理。在以前的语音识别系统中，语言学家构造了音位来学习模型。但随着DL的发展，在end-to-end模型中，这种音位的表示法已经不重要了。完全可以做到从语音信号直接转化为文本信息，不需要人工特征的设计。

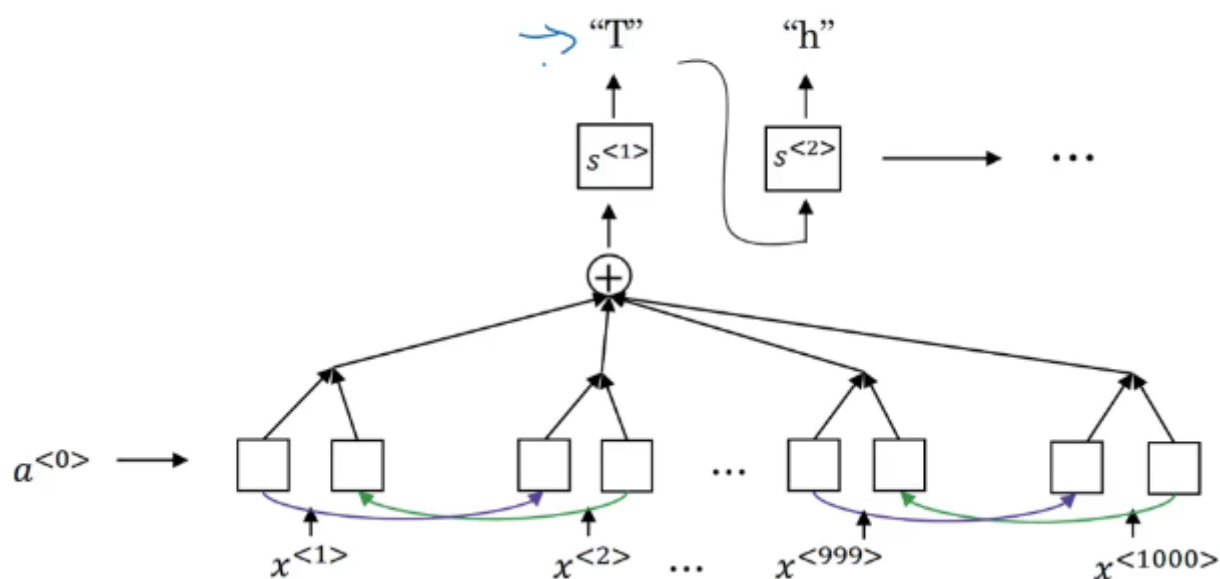
Speech recognition problem



Andrew Ng

8.2 注意力模型的语音识别系统

Attention model for speech recognition



8.3 CTC损失函数

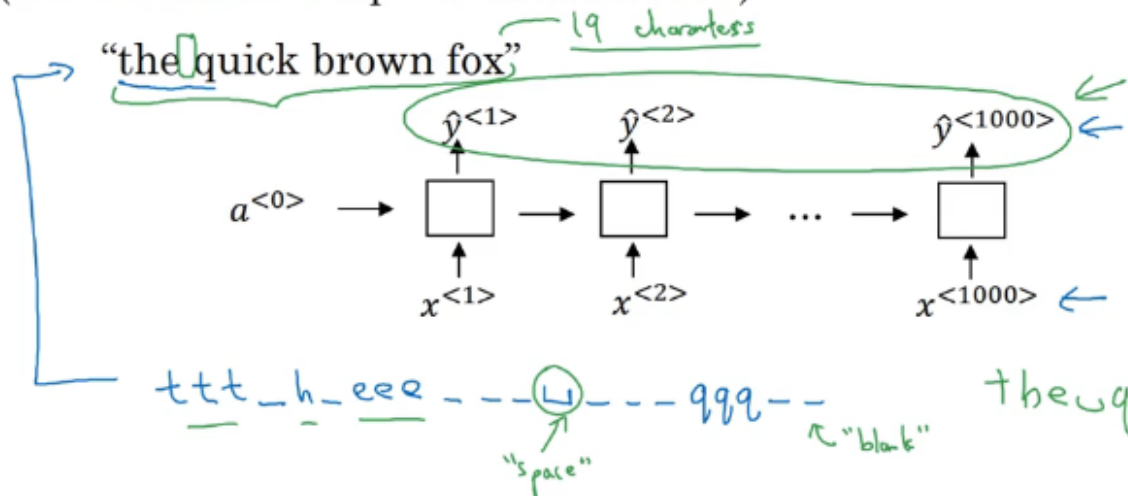
CTC (Connectionist temporal classification) 损失函数的语音识别模型也有良好的效果。

对于音频信号，我们以较短的时间间隔进行采样。一个10s的语音片段可能得到1000个特征输入片段，但是我们的输出往往只是几个单词。

在CTC损失函数中，允许我们的输出有重复的字符和空白符，强制使输入输出的大小一致。

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

9. 触发字检测

随着深度学习的发展，越来越多的设备需要通过一些关键语音进行唤醒。这些系统称作触发字检测系统。

What is trigger word detection?



Amazon Echo
(Alexa)



Baidu DuerOS
(xiaodunihao)



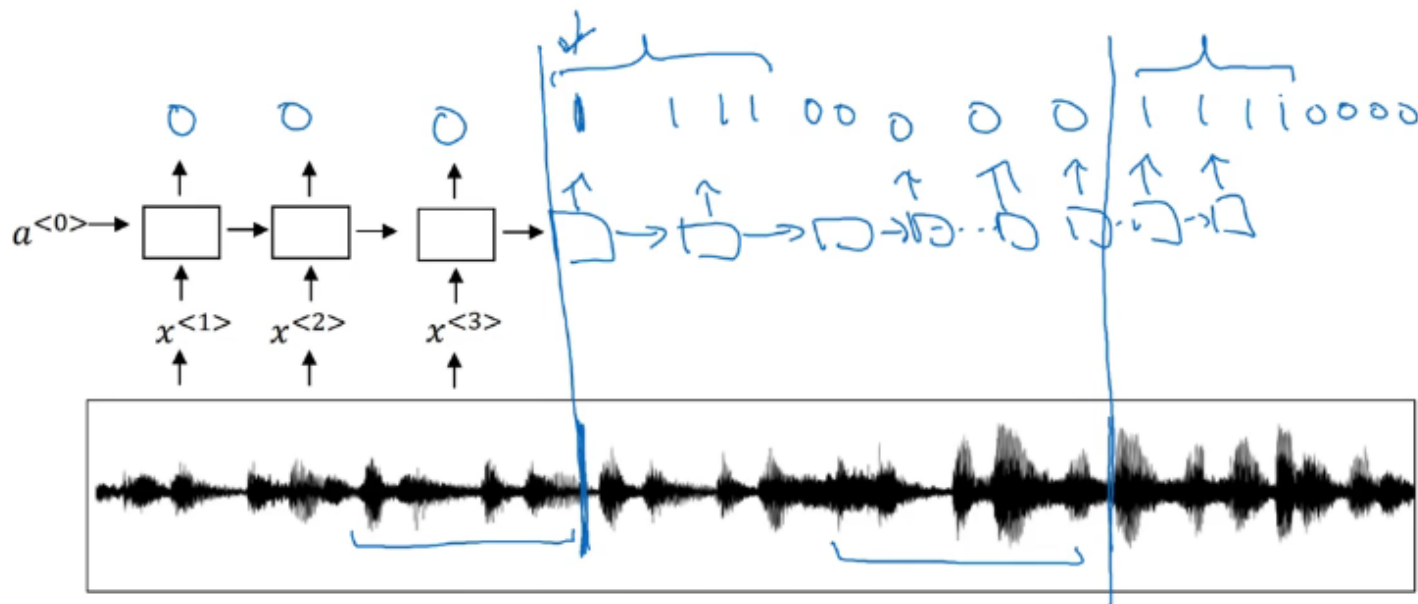
Apple Siri
(Hey Siri)



Google Home
(Okay Google)

一种简单的触发字检测算法就是使用RNN模型，将音频信号进行声谱图转化得到特征如数到RNN中。在输出的标签中，我们可以使触发字前的输出都标记为0，触发字的输出则标记为1。

Trigger word detection algorithm



这个算法的缺点是构建了不平衡的训练集。简单的方法就是在触发字后的多个目标都标记为1。