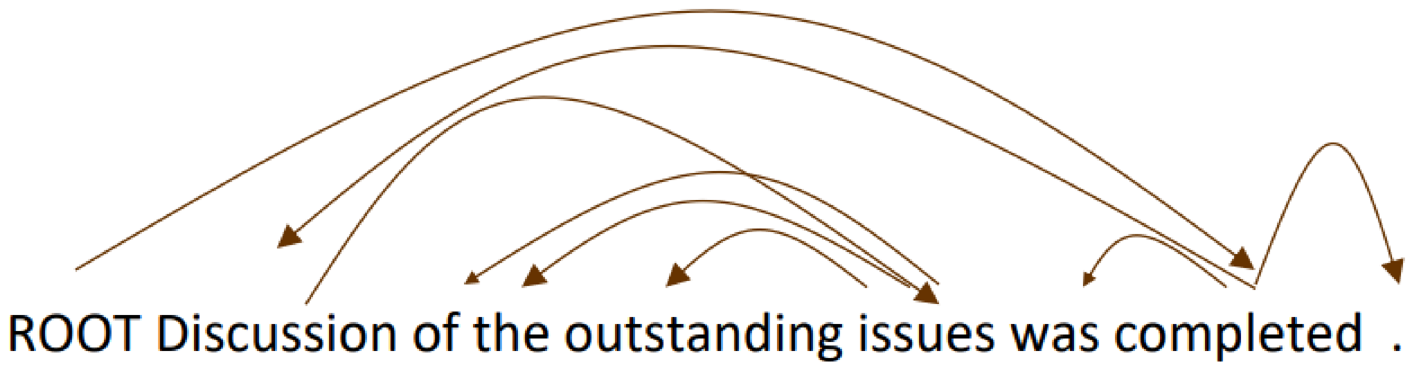
斯坦福大学 CS224n：使用深度学习做自然语言处理

- 第5课：语言学结构：依存解析 (Linguistic Structure: Dependency Parsing)
- 第6课：语言模型LM和RNN
- 第7课：梯度消失和更好的RNN
- 第8课：机器翻译，Seq2Seq，注意力模型(Attention)
- 第9课、第10课：大作业项目引导

第5课：语言学结构：依存解析 (Linguistic Structure: Dependency Parsing)

语言学结构的两个角度

- **构成关系**（constituency） = 短语结构语法（phrase structure grammar） = 上下文无关语法（context-free grammars, CFG）
- **依存结构**（dependency structure）



依存语法与依存结构

- 标注数据：Universal Dependencies treebanks [Universal Dependencies; cf. Marcus et al. 1993, The Penn Treebank, Computational Linguistics]
- 传统算法：动态规划 (Eisner, 1996)
- 传统算法：图算法 (McDonald et al.’s , 2005)
- 传统算法：约束满足问题 (Karlsson, 1990)
- 算法：基于转移的解析，或者是确定性依存解析

基于转移的解析

- Greedy transition-based parsing [Nivre 2003]
- MaltParser [Nivre and Hall 2005]

基于神经网络的解析

- A neural dependency parser [Chen and Manning 2014]
- A Neural graph-based dependency parser [Dozat and Manning 2017; Dozat, Qi, and Manning 2017]

第6课：语言模型LM和RNN

语言模型 Language Models (LM)：根据前文预测下一个词

$$
\begin{aligned}
P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}) &= P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} \mid \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} \mid \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(1)}) \\
&= \prod_{t=1}^{T} \underbrace{P(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t-1)}, \ldots, \boldsymbol{x}^{(1)})}_{\text{This is what our LM provides}}
\end{aligned}
$$

n-gram 语言模型

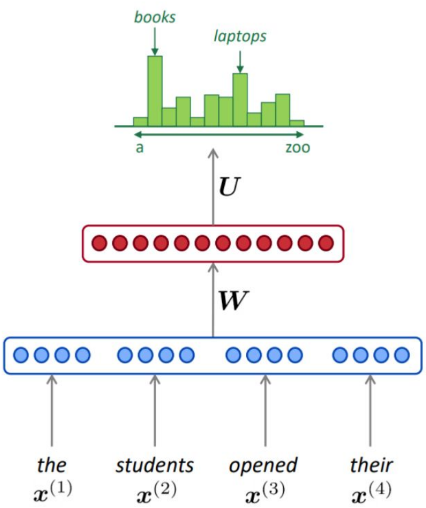
- 根据前面n-1个词来预测下一个词
- 稀疏性问题
- 词组必须见过，存储量太大
- 语法连贯，但不符合实际含义。

$$P(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)},\dots,\boldsymbol{x}^{(1)}) = P(\boldsymbol{x}^{(t+1)}|\overbrace{\boldsymbol{x}^{(t)},\dots,\boldsymbol{x}^{(t-n+2)}}^{n\text{-1 words}})$$

$$= \frac{\overbrace{\boxed{P(\boldsymbol{x}^{(t+1)},\boldsymbol{x}^{(t)},\dots,\boldsymbol{x}^{(t-n+2)})}}^{\text{prob of a n-gram}}}{\underbrace{\boxed{P(\boldsymbol{x}^{(t)},\dots,\boldsymbol{x}^{(t-n+2)})}}_{\text{prob of a (n-1)-gram}}}$$

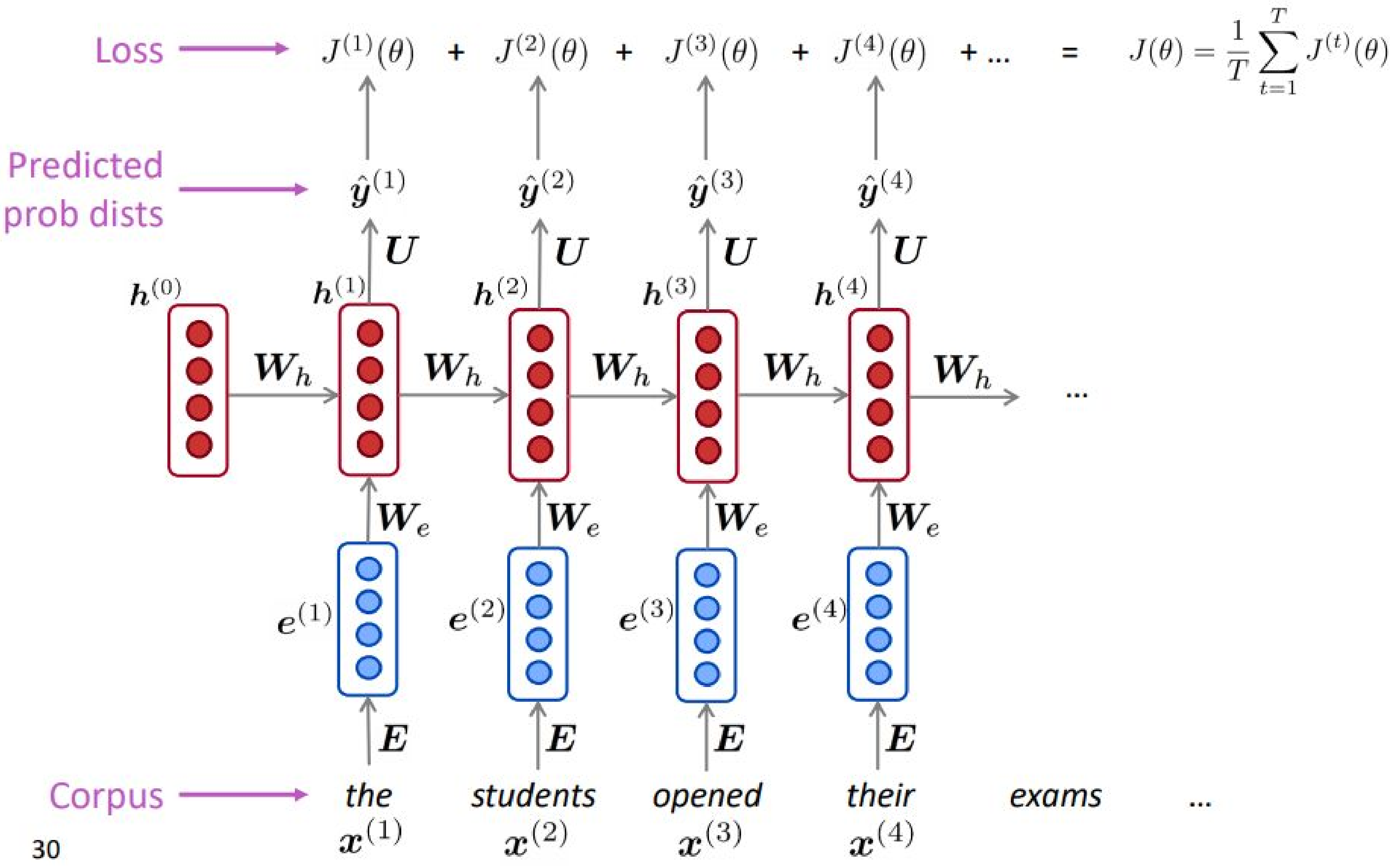
固定窗口神经网络语言模型（fixed-window neural language model）

- 解决稀疏性、存储问题
- window考虑范围过小
- 不同位置的变量完全不同，损失了对称性

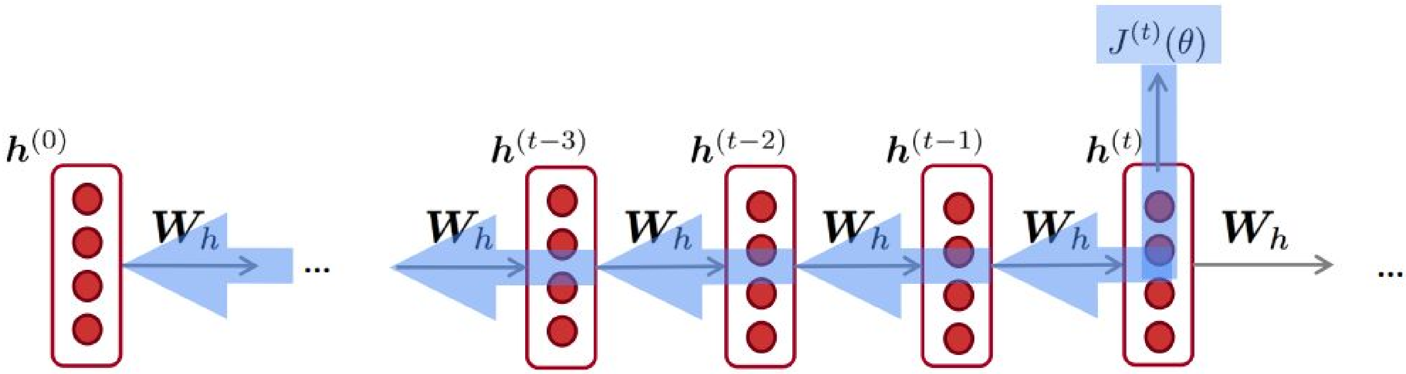


RNN系列语言模型

- 可以处理任意长度
- 变量共享，具备对称性
- 问题：计算太慢、很难记得很早之前信息
- 优化：采用SGD小批量梯度下降



RNN的后向传播（**backpropagation through time**）：累加共享变量在各个time step的导数



评价方法：**困惑度Perplexity**，越低越好

$$\text{perplexity} = \underbrace{\prod_{t=1}^{T} \left(\frac{1}{P_{\text{LM}}(\boldsymbol{x}^{(t+1)} | \, \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})} \right)}_{\text{Inverse probability of corpus, according to Language Model}}{}^{1/T}$$

Normalized by number of words

RNN-LM可以用于解决各类问题：命名实体识别、情感识别、问答、机器翻译、摘要

第7课：梯度消失和更好的RNN

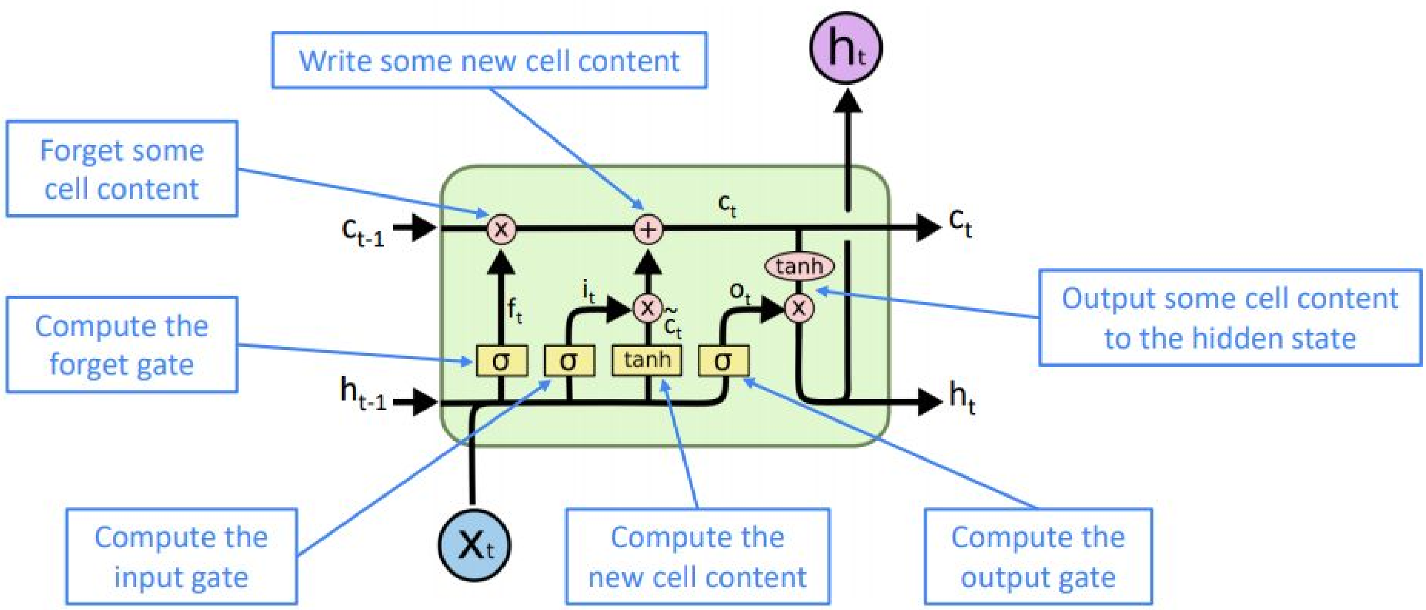
RNN的**梯度消失/梯度爆炸**问题：造成长距离信息无法携带，造成梯度溢出

$$\begin{aligned}\frac{\partial J^{(i)}(\theta)}{\partial \boldsymbol{h}^{(j)}} &= \frac{\partial J^{(i)}(\theta)}{\partial \boldsymbol{h}^{(i)}} \prod_{j<t\leq i} \frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} \\ &= \frac{\partial J^{(i)}(\theta)}{\partial \boldsymbol{h}^{(i)}} \boxed{\boldsymbol{W}_h^{(i-j)}} \prod_{j<t\leq i} \operatorname{diag}\left(\sigma'\left(\boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_x \boldsymbol{x}^{(t)} + \boldsymbol{b}_1\right)\right)\end{aligned}$$

If W_h is small, then this term gets vanishingly small as *i* and *j* get further apart

Long Short-Term Memory(**LSTM**) -- Hochreiter and Schmidhuber in 1997

- Hidden State h(t)
- Cell State c(t)：长距离信息
- 控制门：Forget gate、Input gate、Output gate
- 在2013~2015，LSTM是显著有效的方法。而如今（2019年）Transformers等更加流行。



Forget gate: controls what is kept vs forgotten, from previous cell state

Input gate: controls what parts of the new cell content are written to cell

Output gate: controls what parts of cell are output to hidden state

Sigmoid function: all gate values are between 0 and 1

$$\boldsymbol{f}^{(t)} = \boxed{\sigma}\left(\boldsymbol{W}_f \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_f \boldsymbol{x}^{(t)} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{W}_i \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_i \boldsymbol{x}^{(t)} + \boldsymbol{b}_i\right)$$

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{W}_o \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_o \boldsymbol{x}^{(t)} + \boldsymbol{b}_o\right)$$

New cell content: this is the new content to be written to the cell

Cell state: erase ("forget") some content from last cell state, and write ("input") some new cell content

Hidden state: read ("output") some content from the cell

$$\tilde{\boldsymbol{c}}^{(t)} = \tanh\left(\boldsymbol{W}_c \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_c \boldsymbol{x}^{(t)} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{f}^{(t)} \circ \boldsymbol{c}^{(t-1)} + \boldsymbol{i}^{(t)} \circ \tilde{\boldsymbol{c}}^{(t)}$$

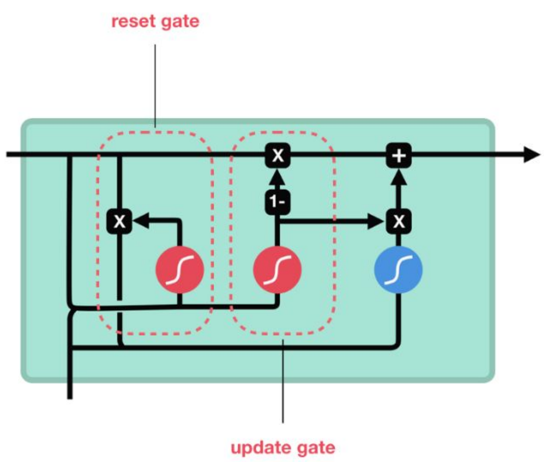
$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} \circ \tanh \boldsymbol{c}^{(t)}$$

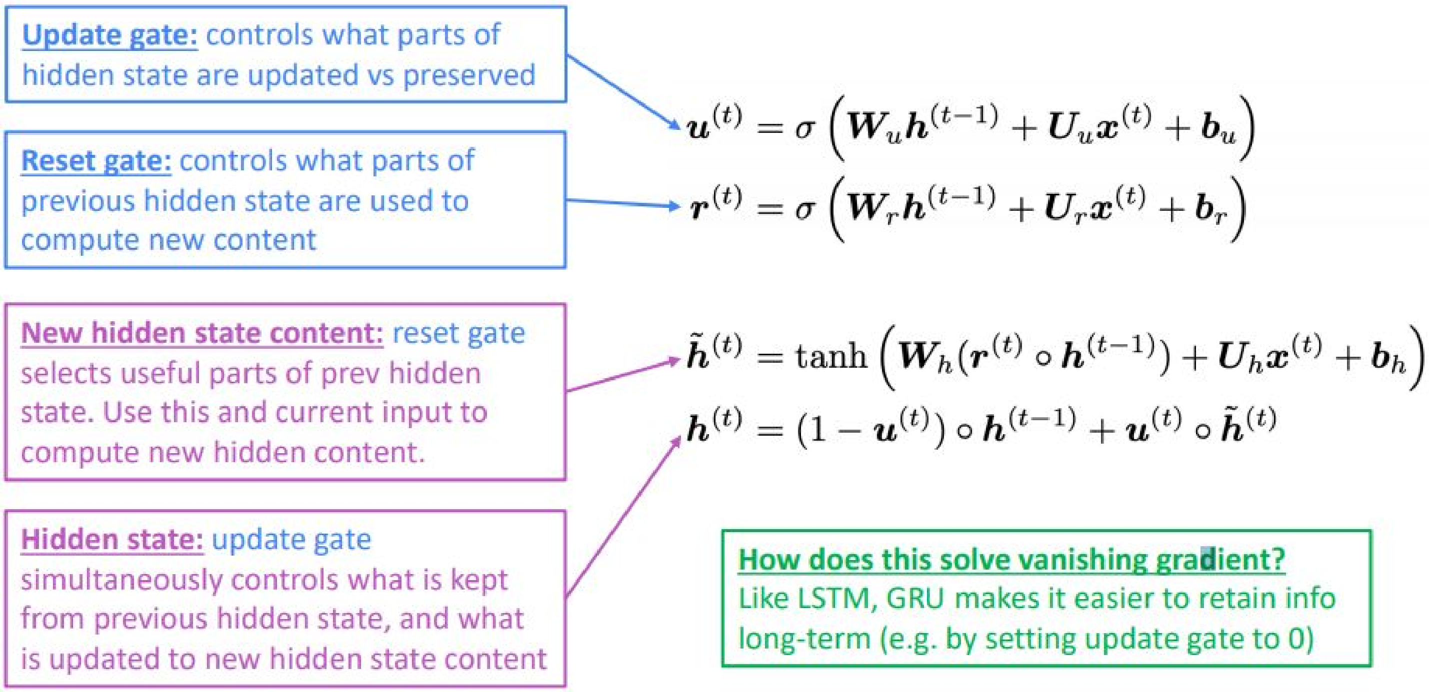
All these are vectors of same length n

Gates are applied using element-wise product

23

Gated Recurrent Units(**GRU**)





LSTM与GRU对比

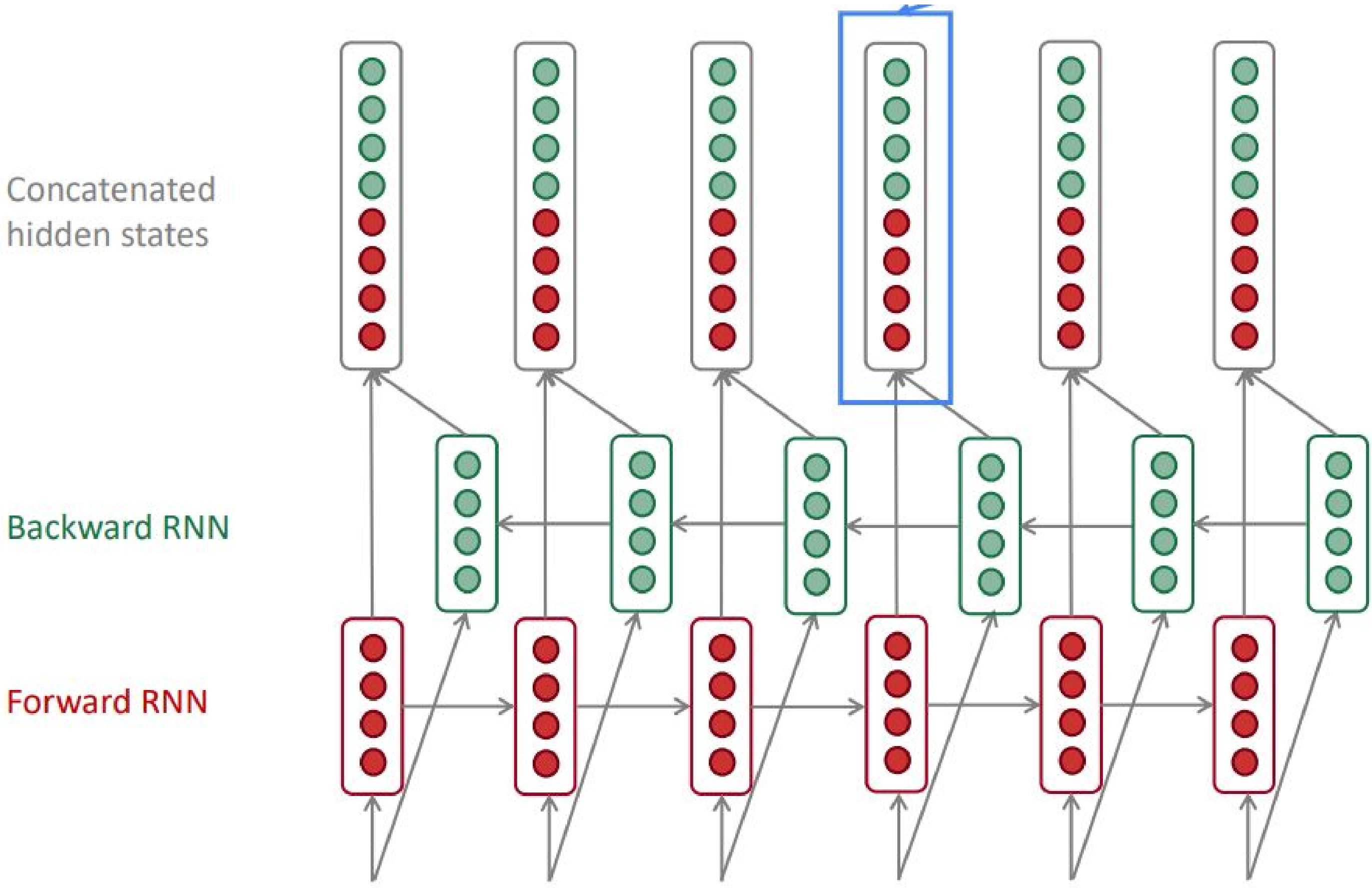
- GRU参数更少，计算更快
- 两者没有明显效果上的差别
- LSTM仍然是首选

梯度消失和梯度爆炸问题对LSTM、GRU，以及任何深度的神经网络都仍然存在

- 通过一些跨层直连（Highway）来解决，比如ResNet、DenseNet

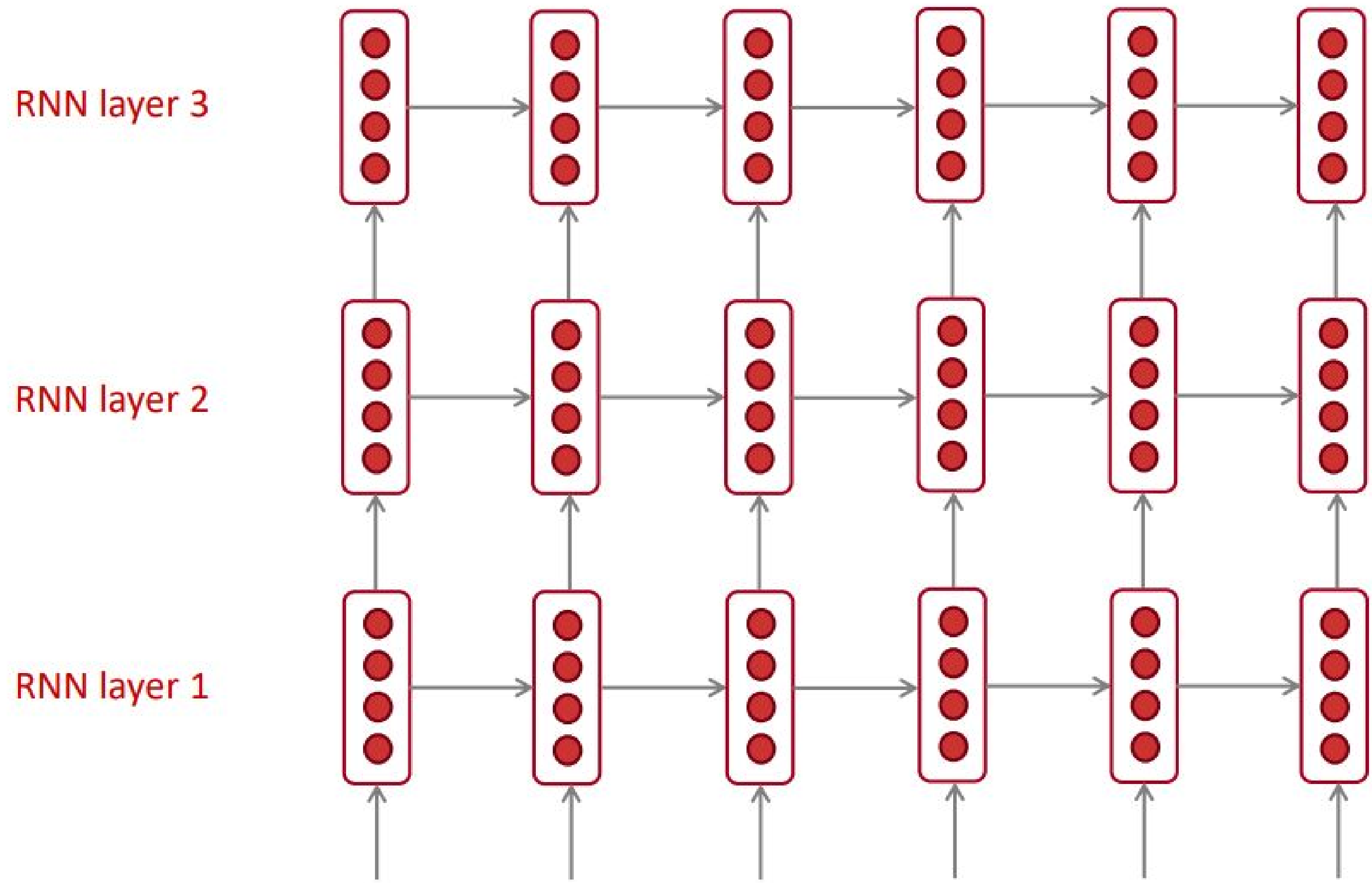
双向RNN（Bidirectional RNNs）

- 不适用于Language Model，只适用于了解完整句子信息。
- 当已知完整句子时，优先使用。BERT就是基于它。



多层RNN（Multi-layer RNNs）

- 良好表现的RNN通常是多层的
- 2017 Britz 的机器学习模型，encoder采用2~4层，decoder采用4层
- 基于Transformer的网络，可以高达24层



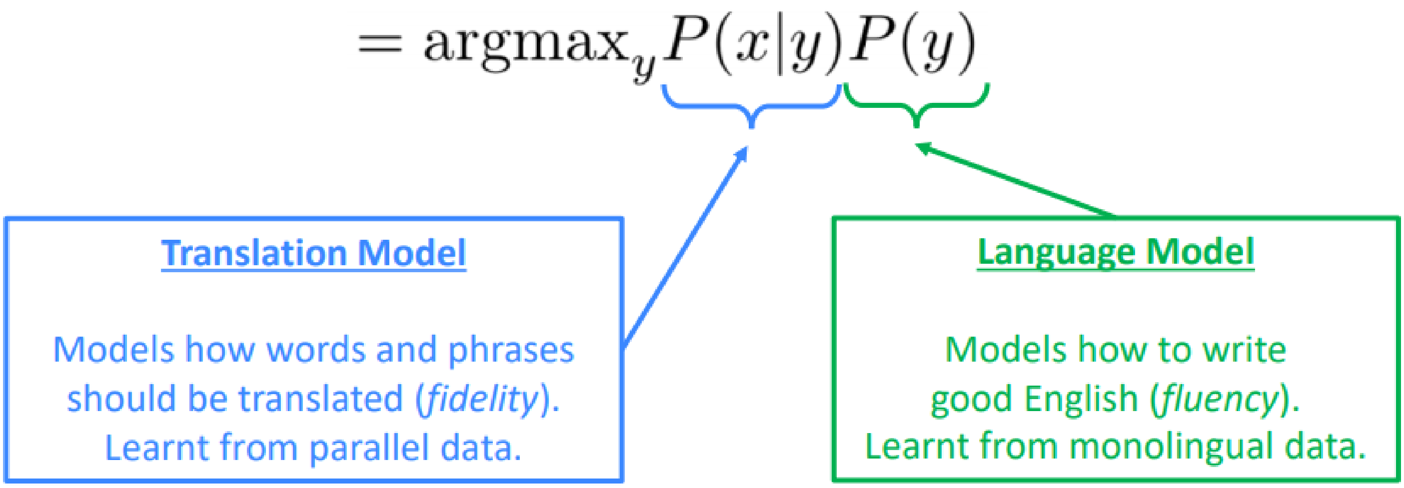
第8课：机器翻译，Seq2Seq，注意力模型(Attention)

> 机器翻译的历史：1950s~1980s基于规则，1990s~2010s基于统计，2014~神经网络

统计机器翻译 Statistical Machine Translation (**SMT**)

- 根据提供的源句x，寻找最好的翻译句y

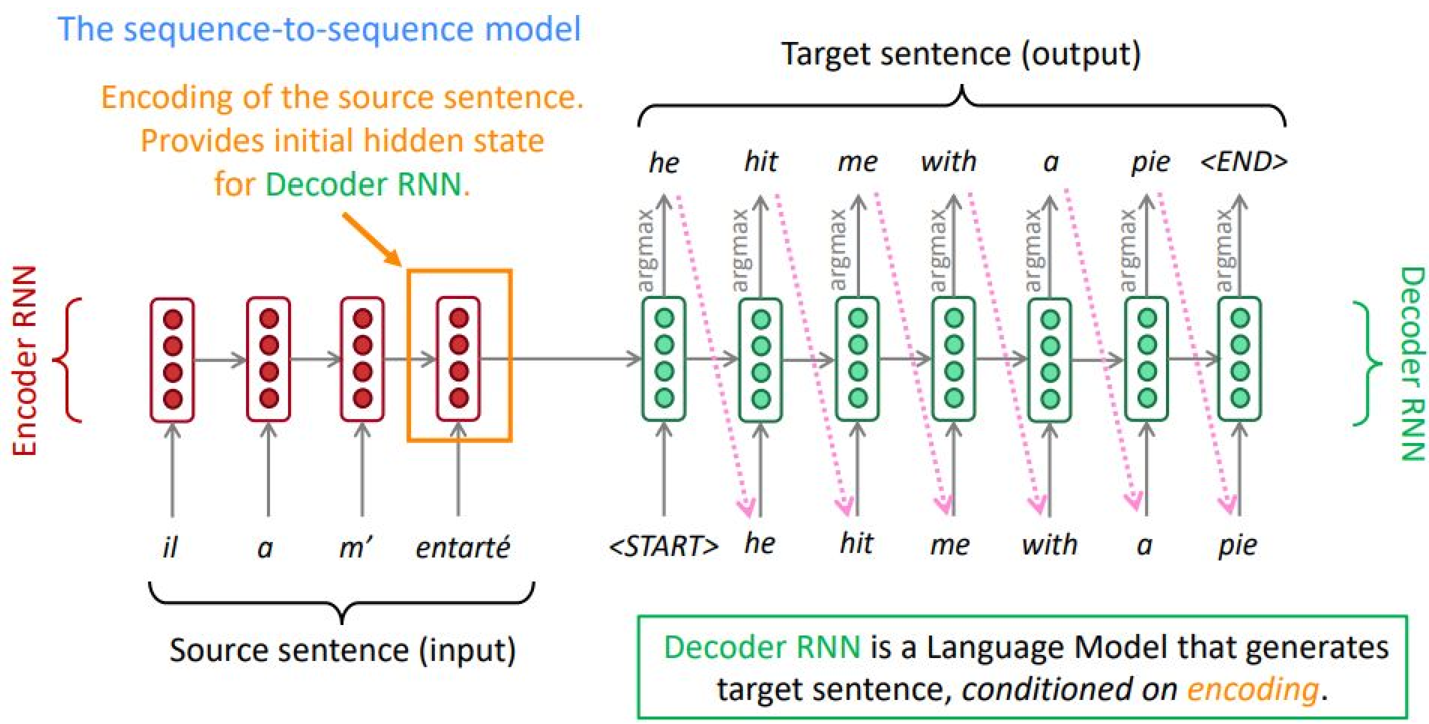
$$\operatorname{argmax}_y P(y|x)$$



- 预测y的同时，还要预测Alignment对齐（词语一对一、一对多、多对一、多对多、交叉）
- 采用启发式搜索，模块设计、人工干预、特征工程、计算量都非常复杂

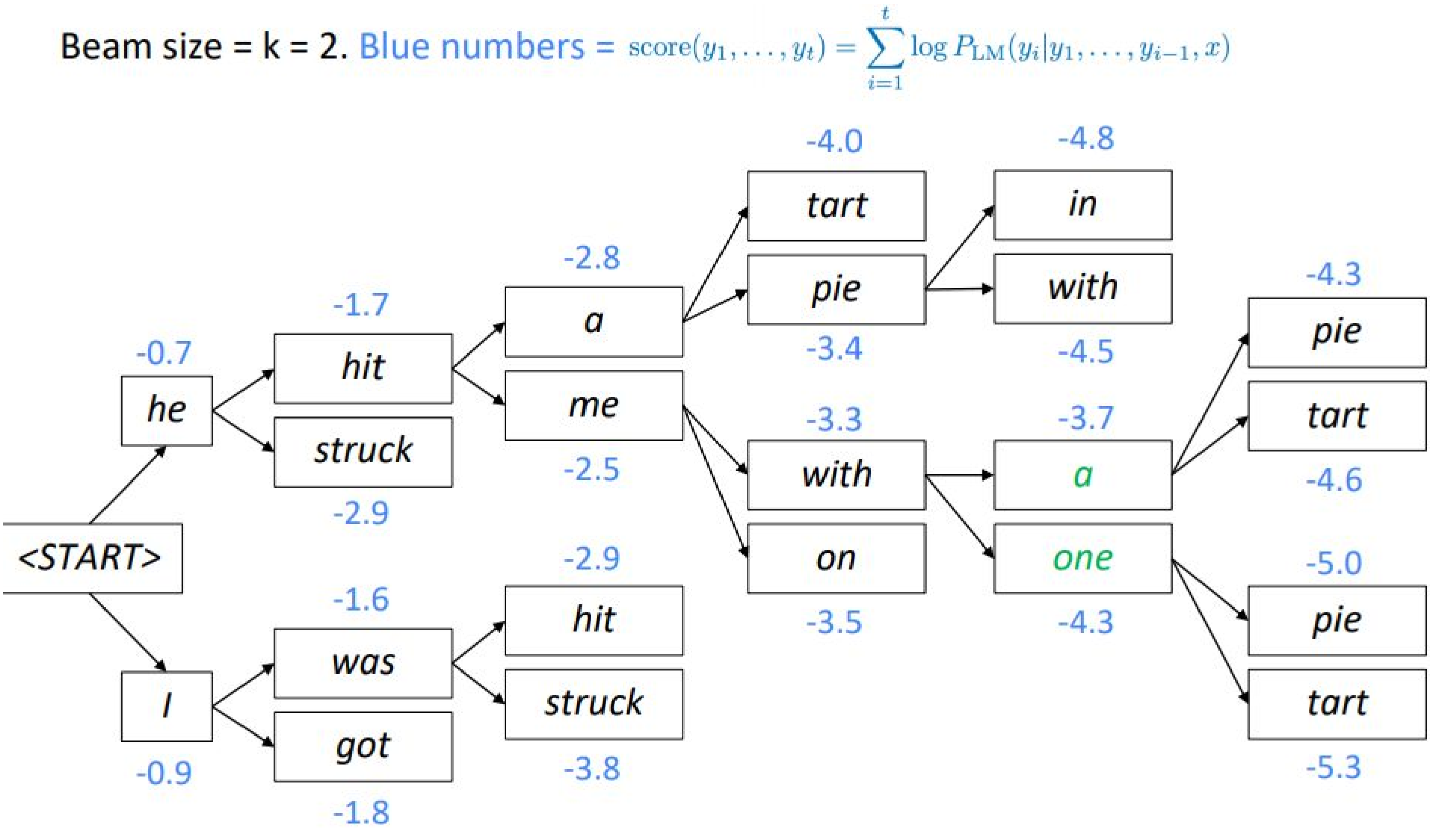
神经网络机器翻译 Neural Machine Translation (**NMT**) （2014年核弹级的诞生！）

- seq2seq不仅可以用于机器翻译，还可以用于摘要、对话、解析、代码生成
- [Sequence to Sequence Learning with Neural Networks]



NMT的搜索方式优化

- Greedy Search Decoding：默认每次都取最优选项，可能会造成整体最终句子并非最优
- Exhaustive Search Decoding：搜索所有可能情况
- Beam Search Decoding：每一步都只保留目前top k（k一般5~10）分数的句子，然后延伸下一步：



NMT vs SMT

- 效果更好：更流畅、更好使用上下文、更好使用词组相似性
- 端到端训练，无需子模块和特征工程
- 不可解释，不易调试，难以控制

机器翻译评估方式：**BLEU** （aclweb.org/anthology/P0...）

- 对比机器翻译结果和一种或多种人工翻译结果，按照n-gram比例+过短惩罚计算相似性

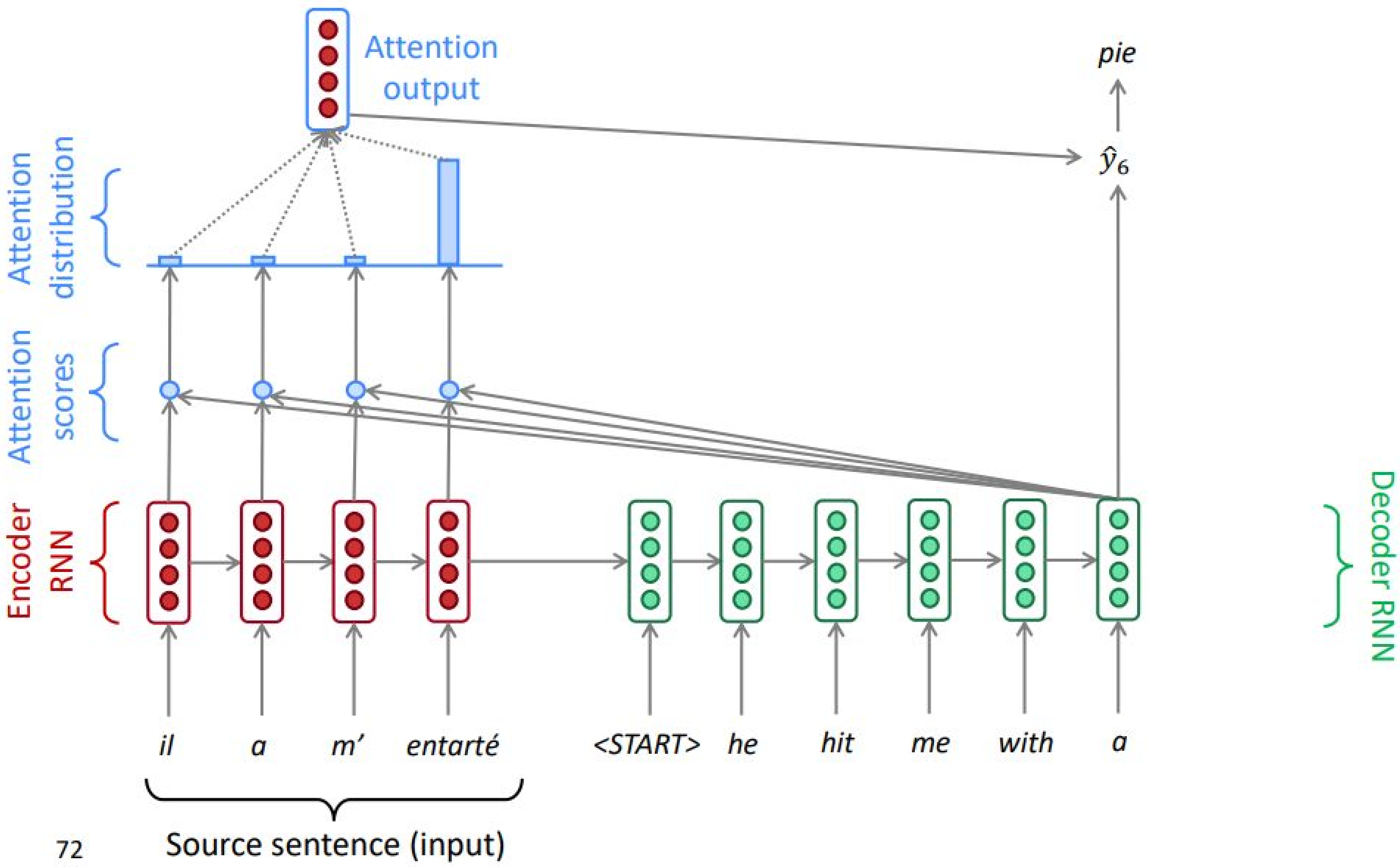
NMT的遗留问题： （更多：skynettoday.com/editori...）

- 未收录词
- 训练数据和测试数据领域不同
- 长文本的上下文维护
- 少资源的语言对
- 缺乏常识
- 从训练数据学习到了偏差
- 不可解释的系统经常出现奇怪的情况

Attention

- seq2seq的瓶颈问题：encoder和decoder之间的向量很难存储所有信息
- Attention能够解决seq2seq的瓶颈问题，让decoder直接连接encoder，并且关注在一部分之中。

- 基本原理：对encoder产生的隐层状态h_n和decoder产生的隐层状态s_t进行点乘形成标量，用标量作为比例乘上encoder隐层状态并累加为a_t，然后拼接a_t和s_t进行后续y_t计算。



Attention的优势

- 提升NMT效果，解决seq2seq瓶颈问题
- 减少梯度消失问题
- 具备一定的解释性，学习到了alignment
- Attention是普适的，不仅seq2seq，attention也可以用于其他架构和其他任务

Attention的变种

- 根据value和query隐层状态，计算attention scores，即e

- Basic dot-product attention: $e_i = \boldsymbol{s}^T \boldsymbol{h}_i \in \mathbb{R}$
  - Note: this assumes $d_1 = d_2$
  - This is the version we saw earlier
- Multiplicative attention: $e_i = \boldsymbol{s}^T \boldsymbol{W} \boldsymbol{h}_i \in \mathbb{R}$
  - Where $\boldsymbol{W} \in \mathbb{R}^{d_2 \times d_1}$ is a weight matrix
- Additive attention: $e_i = \boldsymbol{v}^T \tanh(\boldsymbol{W}_1 \boldsymbol{h}_i + \boldsymbol{W}_2 \boldsymbol{s}) \in \mathbb{R}$
  - Where $\boldsymbol{W}_1 \in \mathbb{R}^{d_3 \times d_1}, \boldsymbol{W}_2 \in \mathbb{R}^{d_3 \times d_2}$ are weight matrices and $\boldsymbol{v} \in \mathbb{R}^{d_3}$ is a weight vector.
  - d_3 (the attention dimensionality) is a hyperparameter

- 计算softmax，获得attention分布
- 采用attention分布作为权重，计算value隐层加和

第9课、第10课：大作业项目引导

寻找问题：

- Look at ACL anthology for NLP papers: https://aclanthology.info
- Also look at the online proceedings of major ML conferences: NeurIPS, ICML, ICLR
- Look at past cs224n project
- Look at online preprint servers, especially: https://arxiv.org
- 寻找arxiv预发表文章 arxiv-sanity.com/
- 寻找各项任务目前业界最优实践 paperswithcode.com/sota

获得数据

- Linguistic Data Consortium catalog.ldc.upenn.edu/
- Machine translation http://statmt.org
- Dependency parsing: Universal Dependencies https://universaldependencies.org
- machinelearningmastery.com...
- github.com/niderhoff/nl...

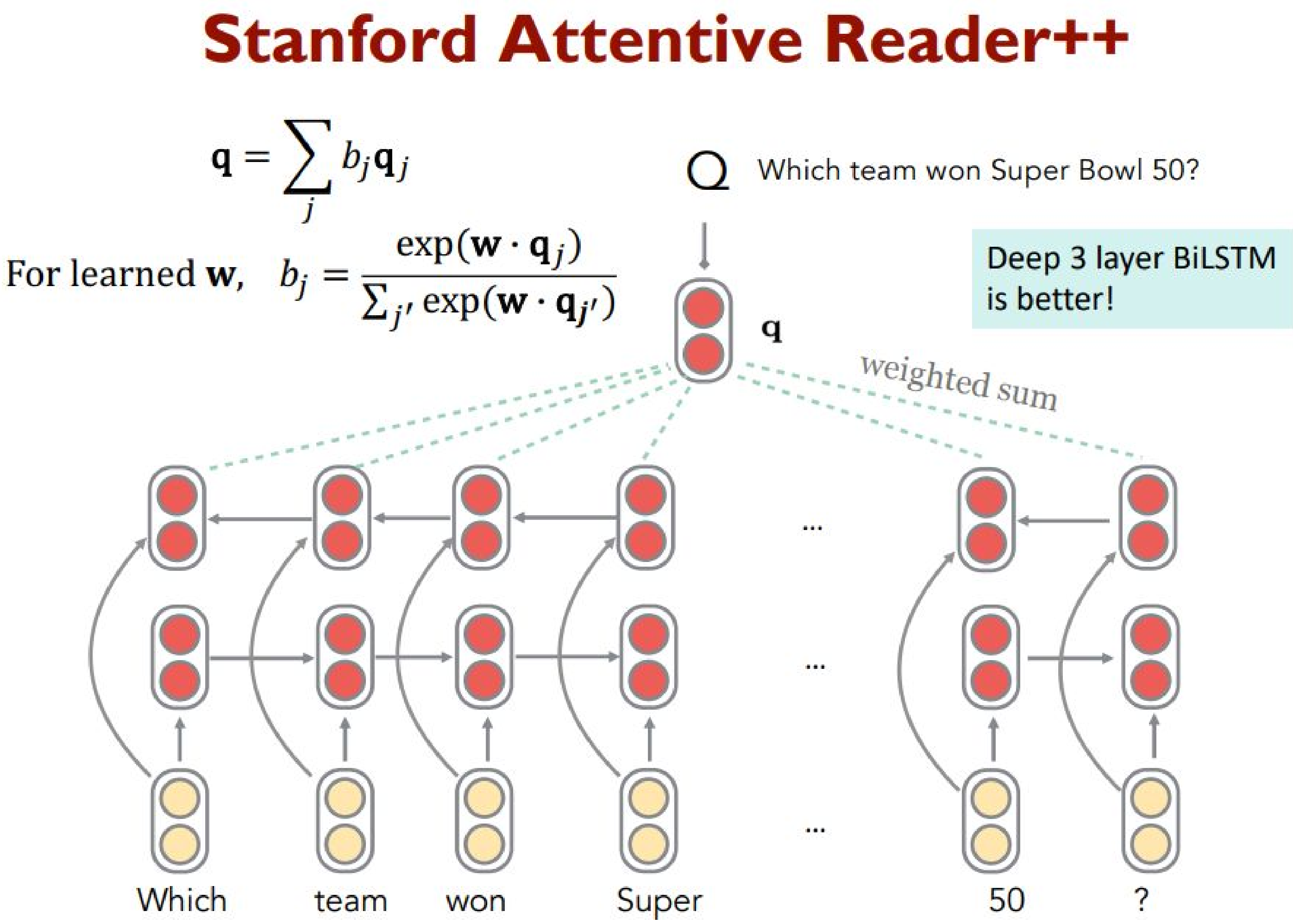
机器翻译评估标准

- BLEU Evaluation Metric
- 其他： TER, METEOR, MaxSim, SEPIA, RTE-MT

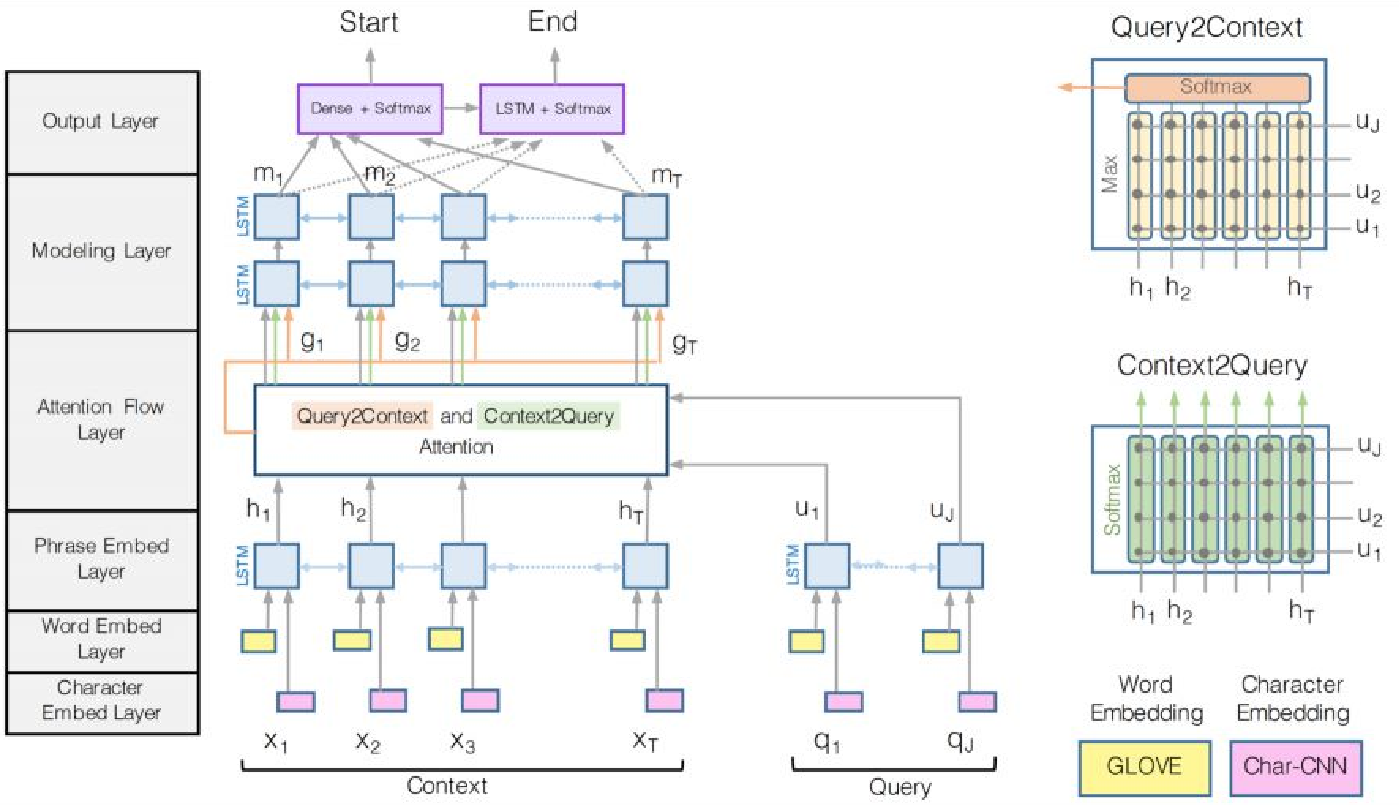
训练的一些tips

SQuAD 2.0 问答数据集

QA的一些前沿尝试：



5. BiDAF: Bi-Directional Attention Flow for Machine Comprehension
(Seo, Kembhavi, Farhadi, Hajishirzi, ICLR 2017)



DrQA: Open-domain Question Answering
(Chen, et al. ACL 2017) https://arxiv.org/abs/1704.00051

Q: How many of Warsaw's inhabitants spoke Polish in 1933?