RNN

[1. 参考资料 2](#_Toc490929102)

[2. 专有名词 2](#_Toc490929103)

[3. RNN 循环网路 2](#_Toc490929104)

[4. GRU 3](#_Toc490929105)

[5. LSTM 4](#_Toc490929106)

[6. RvNN 递归网络 5](#_Toc490929107)

[6.1. BTS训练RvNN 5](#_Toc490929108)

[6.2. 常用RvNN结构与场景 5](#_Toc490929109)

[6.2.1. Standard RNNs： Paraphrase Detection 5](#_Toc490929110)

[6.2.2. Matrix-Vector RNNs： Relation Classification 5](#_Toc490929111)

[6.2.3. Recursive Neural Tensor Network： Sentiment Analysis 6](#_Toc490929112)

[6.2.4. Tree LSTMs： Phrase Similarity 6](#_Toc490929113)

[7. Encoder Decoder 7](#_Toc490929114)

[8. CNN in NLP 8](#_Toc490929115)

[9. NLP网络模型比较 8](#_Toc490929116)

[10. NN in Speech Recognition 9](#_Toc490929117)

[10.1. HMM-DNN acoustic modeling 9](#_Toc490929118)

[10.2. Moden HMM-DNNs 10](#_Toc490929119)

[10.3. HMM-free RNN recognition 10](#_Toc490929120)

[11. NMT Neural Machine Translation 10](#_Toc490929121)

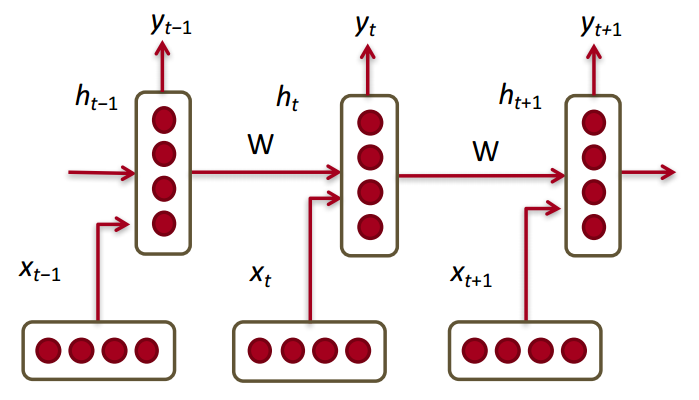
[11.1. MT 演化 10](#_Toc490929122)

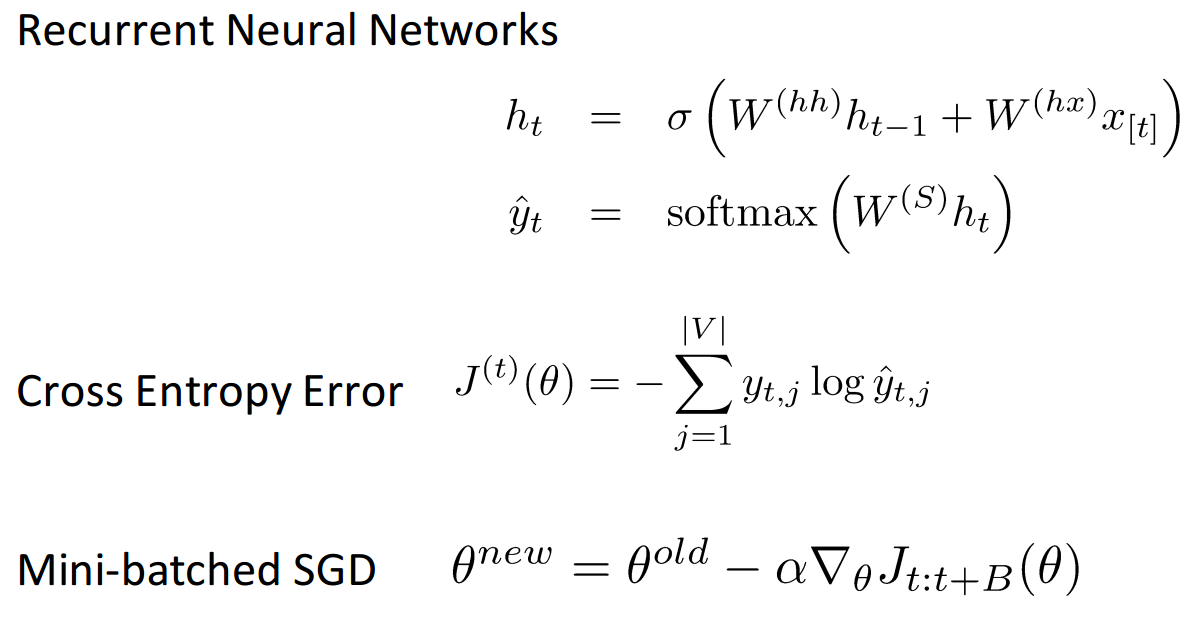
1. 参考资料
2. [http://cs224d.stanford.edu](http://cs224d.stanford.edu) 基本来自该课程2015年slides
3. 专有名词
4. hyperparameter 超参数 简单说就是参数的参数，模型或框架的参数具有参数分布，其参数分布的参数算是超参数，DL中一般是人工设定，然后不断学习更新。
5. GMM Gaussian Mixture Model 高斯混合模型
6. HMM Hidden Markov Model 隐马尔可夫模型
7. RNN 循环网路

Recurrent Neural Network

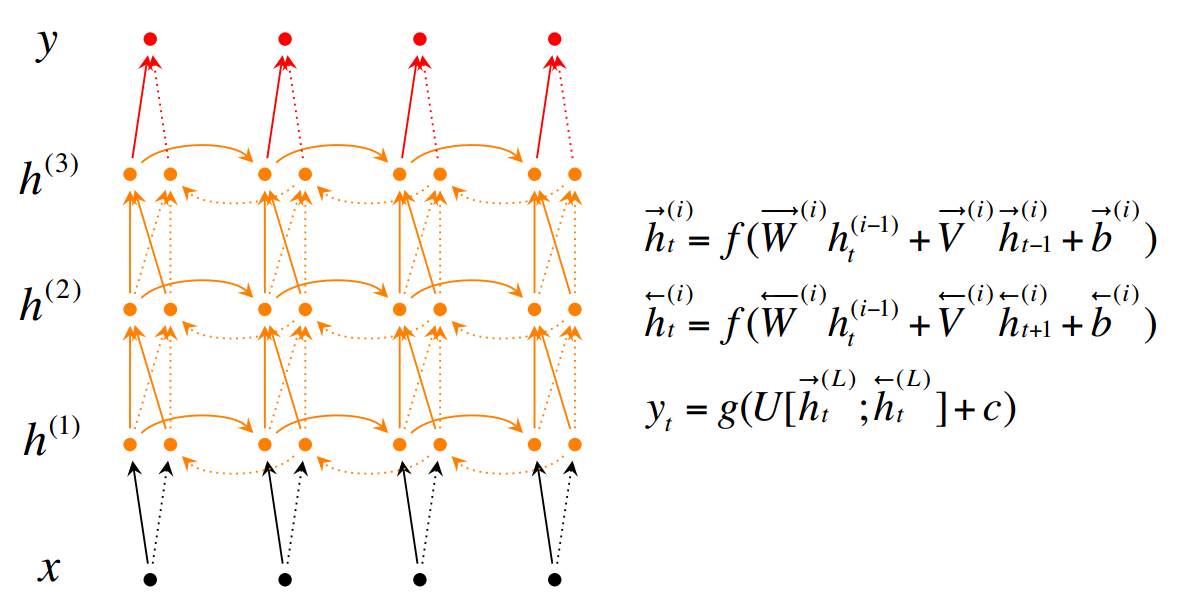
RNN 本身适合对序列化的输入进行建模，如句子和声音，但是传统的RNN存在长期依赖问题：后续的输入会覆盖掉前面的内容，导致后面序列的产生只受前一个时刻输入的影响，更早时刻的输入信息不能被保留下来。更加数学化的解释是传统的 RNN的激活函数使用的是 sigmod 或者tanh，模型的训练过程中参数的更新是通过目标函数误差反向传播进行的，其实也就是梯度的反向传播。由于 sigmod和 tanh 的导数在 0 到 0.25 之间，当网络层数比较多时，或者说当输入序列比较长时，后面层的梯度每往前传一层，都需要乘上一个 0 到 0.25 之间的小数，因此梯度是呈指数递减的，前面层的参数很难更新，这也就导致了长期依赖问题。为了解决长期依赖的问题，可以使用LSTM、GRU等网络。

简单的RNN又被称为Vanilla RNNs，vanilla表示naïve、normal，普通的RNN。





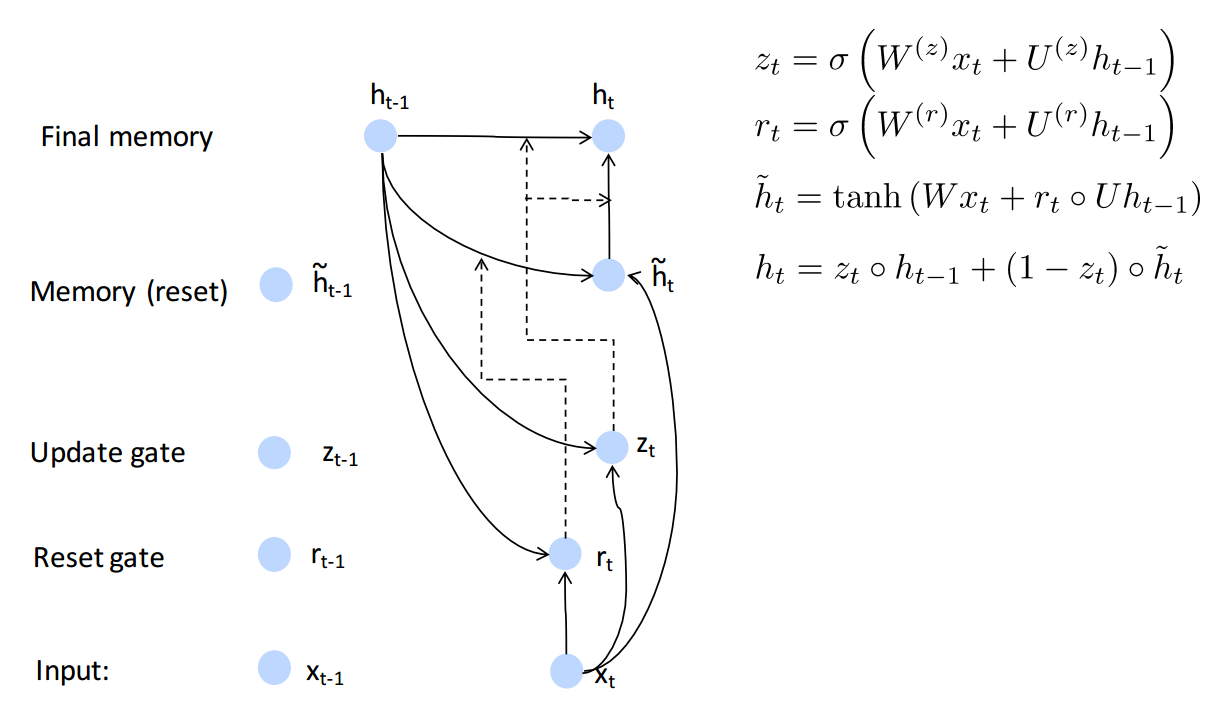
多个RNN cell



Deep Bidirectional RNNs

1. GRU

2014 Gated Recurrent Units，GRUs



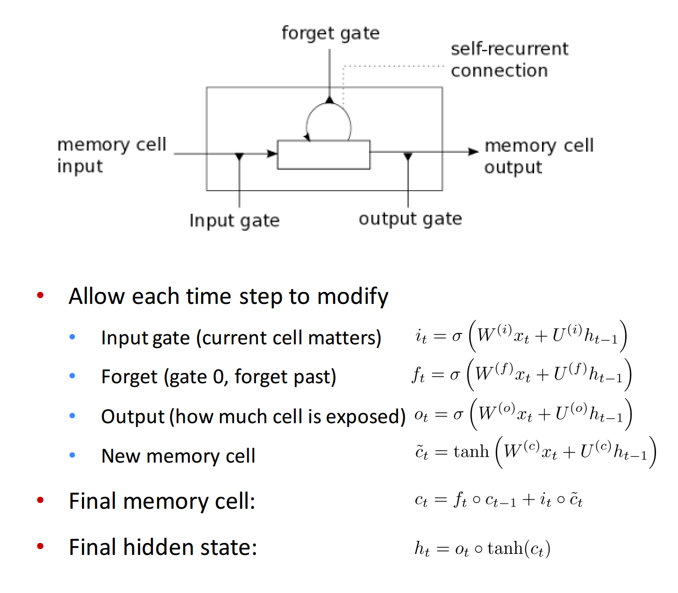
GRU单元

可以看出，每个GRU单元包括：上一个传递进来的hidden state、本次输入、本次输出、本次传递给下一次的输出，只不过传递信号用的就是输出信号。

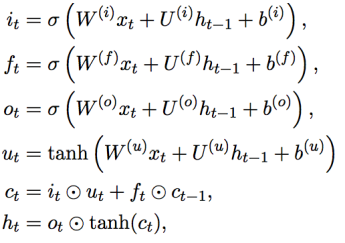
大致流程为：根据传递信号和输入信号，计算输入门Zt和遗忘门Rt；Memory根据遗忘门和上次输出信号决定是否遗忘上一次输入，并加入本次输入信号；输出根据输入门决定上一次输入和Memory信号的输出比例。

1. LSTM

1997 Long Short Term Memories



LSTM cell有输入门、遗忘门、输出门



bias写全的公式

LSTM单元包括：上一cell的hidden state、上一cell的输出、本次输入、本次输出、本cell的hidden state。

LSTM内部含有3个gate门限：输入门、遗忘门、输出门，都是关于输入和上一隐藏态的连接，由输入和上一隐藏态生成新的信号。

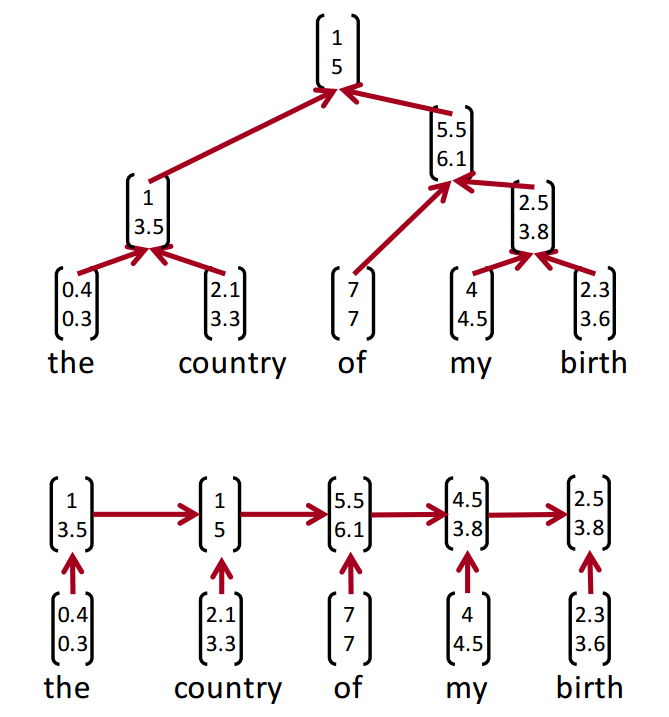
本输出= 遗忘门\*上一输出+输入门\*新信号。

本隐藏态= 输出门\*本输出。

1. RvNN 递归网络

Recursive Neural Network，本文简写为RvNN。

RvNN常用于构建短语、文本的语义表示，如何表示一句话的语义？如何衡量两句话的语义相似度？

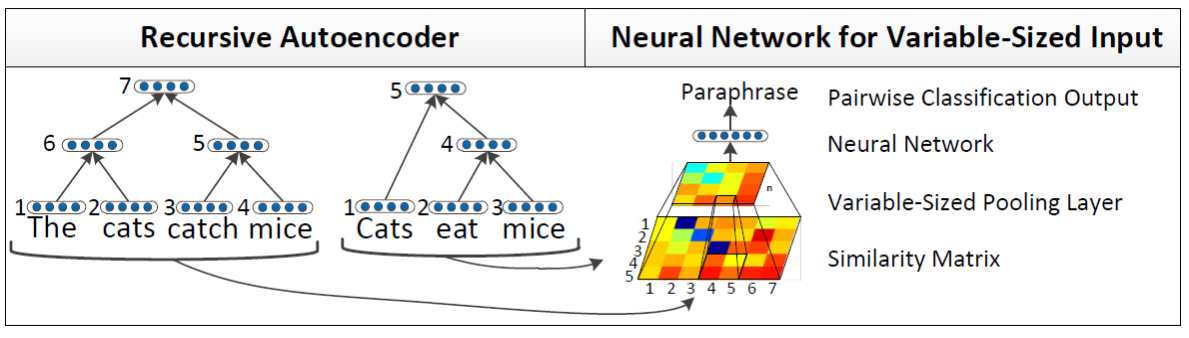


RvNN versus RNN示例：递归生成的句子语义不一定比循环的语义差，二者并没有证明

* 1. BTS训练RvNN

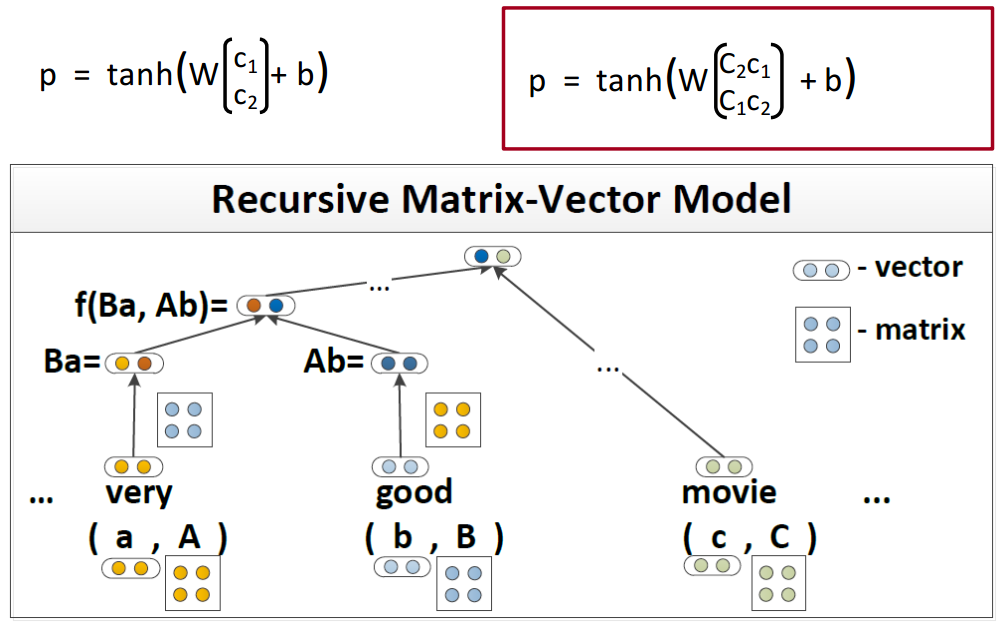
1996 BTS Backpropagation Through Structure进行BP训练，步骤为：

1. Sum derivatives of W from all nodes
2. Split derivatives at each node
3. Add error messages from parent + node itself
   1. 常用RvNN结构与场景
      1. Standard RNNs： Paraphrase Detection



Socher et al. NIPS 2011 使用RvNN实现释义检测（两句话语义相似度）

* + 1. Matrix-Vector RNNs： Relation Classification

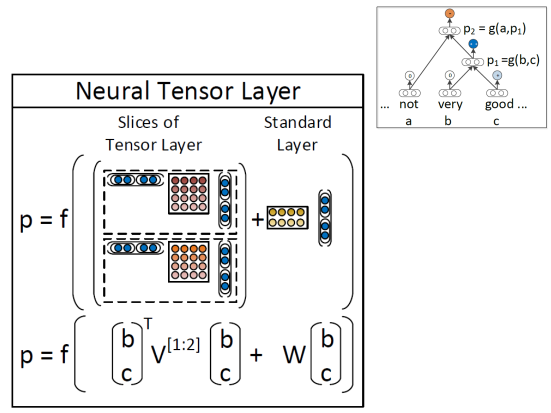


Matrix-Vector RNNs示例，可以使用不同词序C2C1提高语义效果

* + 1. Recursive Neural Tensor Network： Sentiment Analysis

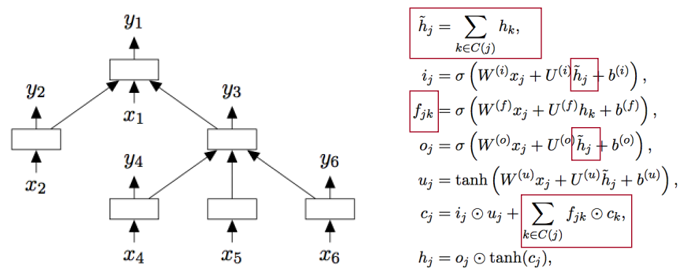
常用的方法，比如bag of words/TFIDF + linguistic features/processing/lexica （语言特征/预处理/词典）等都不能有效表示语义。

比如<white blood cells destroying an infection>与<an infection destroying white blood cells>分别表示<白血球破坏感染><感染破坏白血球>，用基于词频、特征、词典等方法很难有效区分。



Socher et al. 2013 在RvNN单元中添加更多的连接

* + 1. Tree LSTMs： Phrase Similarity



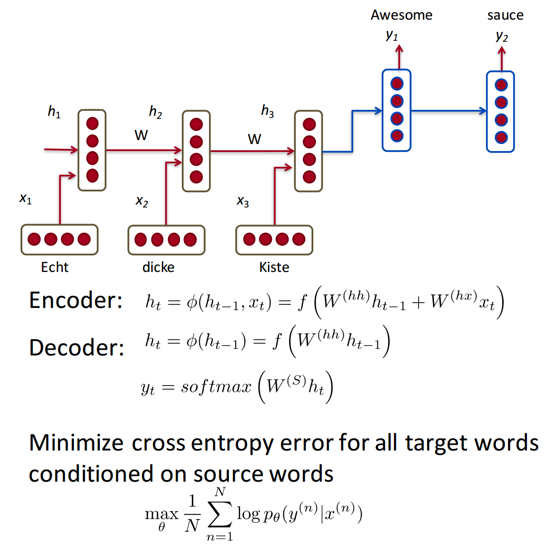
Tai et al. 2015 将LSTMs构造成Tree，先对子节点的向量求和，各子节点有独立的遗忘门

1. Encoder Decoder

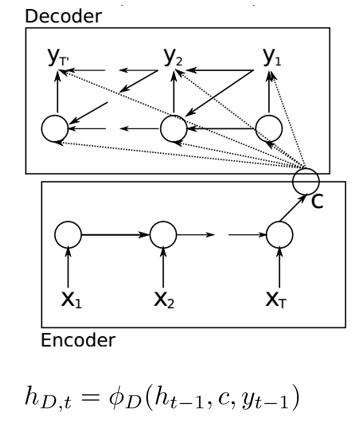
编码-解码器常用于MT (机器翻译Machine Translation)。

MT模型及改进：

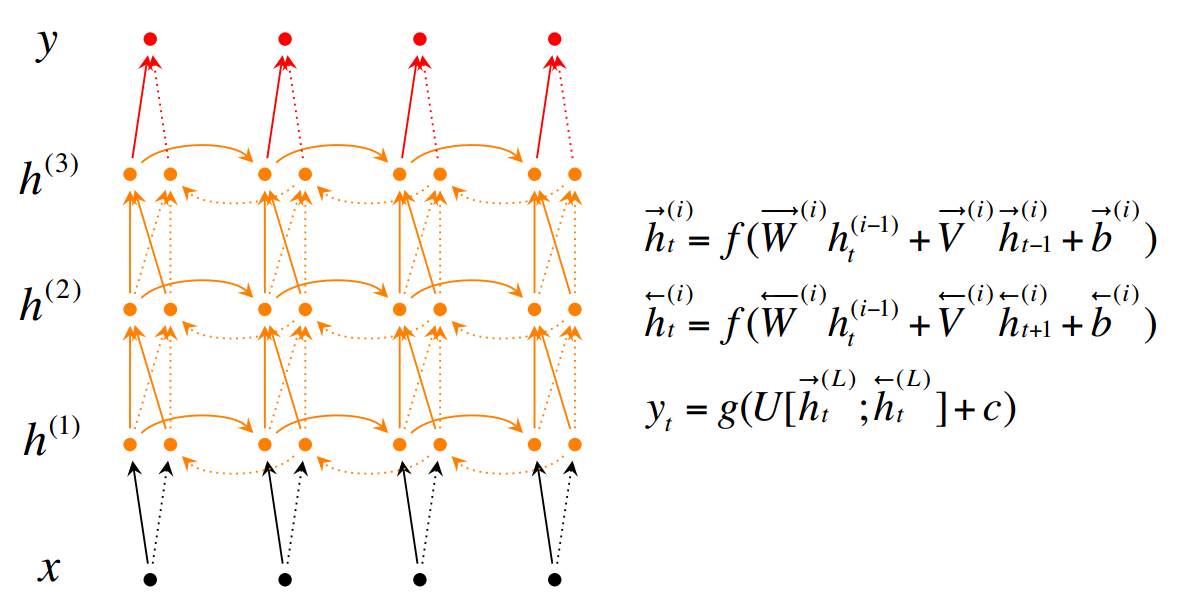
1. 简单的就是分别用一个RNN cell做encoder和decoder。
2. 分别用一层RNN做encoder和decoder，但是各个cell的参数不同。
3. Decoder不仅和encoder的c有关，而且和decoder上一个预测、上一个状态都有关。
4. 使用多层RNN构成Deep RNN作为encoder和decoder。
5. Encoder使用双向RNN，如BiRNN、BiLSTM。
6. RNN换用LSTM、GRU等。



简单的RNN MT



Decoder每一个输出不仅和语句语义C有关，而且和上一次输出和状态有关

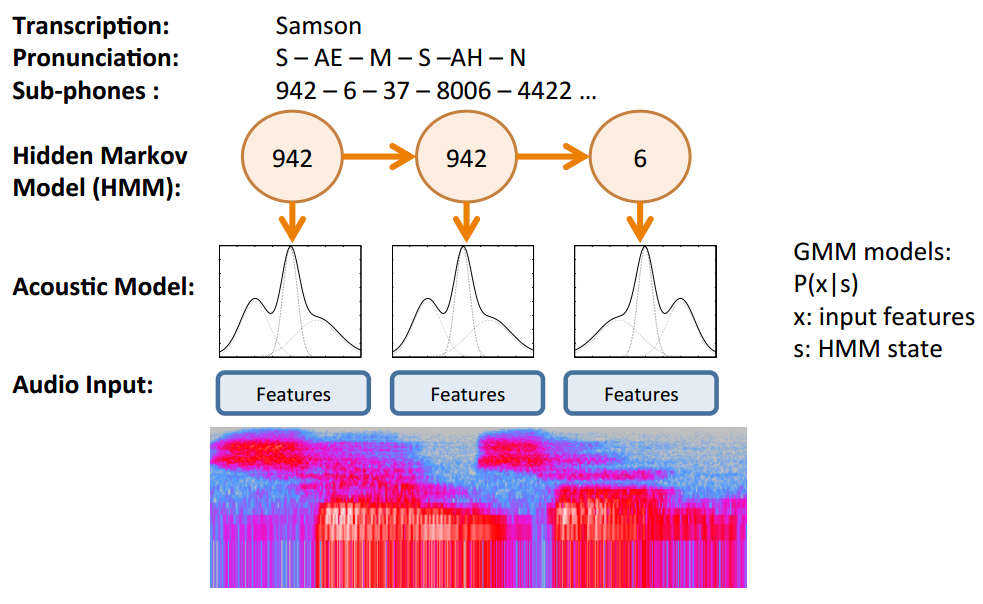


Encoder使用Bidirectional、多层结构

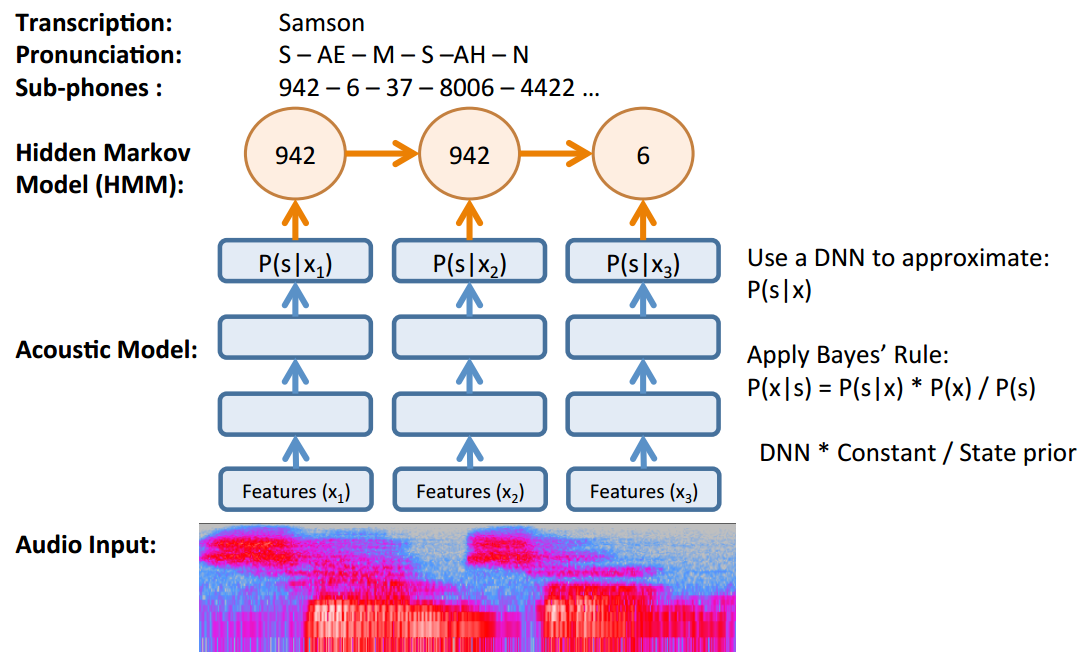
1. CNN in NLP
2. RvNN需要设计parser构造递归结构树。
3. RNN需要依靠前面的语境，而且偏后的词语输出比例较高。
4. CNN能够抽取更多的局部特征，使用更多的filter。
5. NLP网络模型比较
6. **Bag of Vectors:** Surprisingly good baseline for simple classification problems. Especially if followed by a few layers!
7. **Window Model:** Good for single word classification for problems that do not need wide context
8. **CNNs:** good for classification, unclear how to incorporate phrase level annotation (can only take a single label), need zero padding for shorter phrases, hard to interpret, easy to parallelize on GPUs
9. **Recursive Neural Networks:** most linguistically plausible, interpretable, provide most important phrases (for visualization), need parse trees
10. **Recurrent Neural Networks:** Most cognitively plausible (reading from left to right), not usually the highest classification performance but lots of improvements right now with gates (GRUs, LSTMs, etc).
11. Best but also most complex models: **Hierarchical recurrent neural networks with attention mechanisms and additional memory**
12. NN in Speech Recognition

More on spoken language understanding: [cs224s.stanford.edu](cs224s.stanford.edu)

* 1. HMM-DNN acoustic modeling



使用GMMs进行语音建模

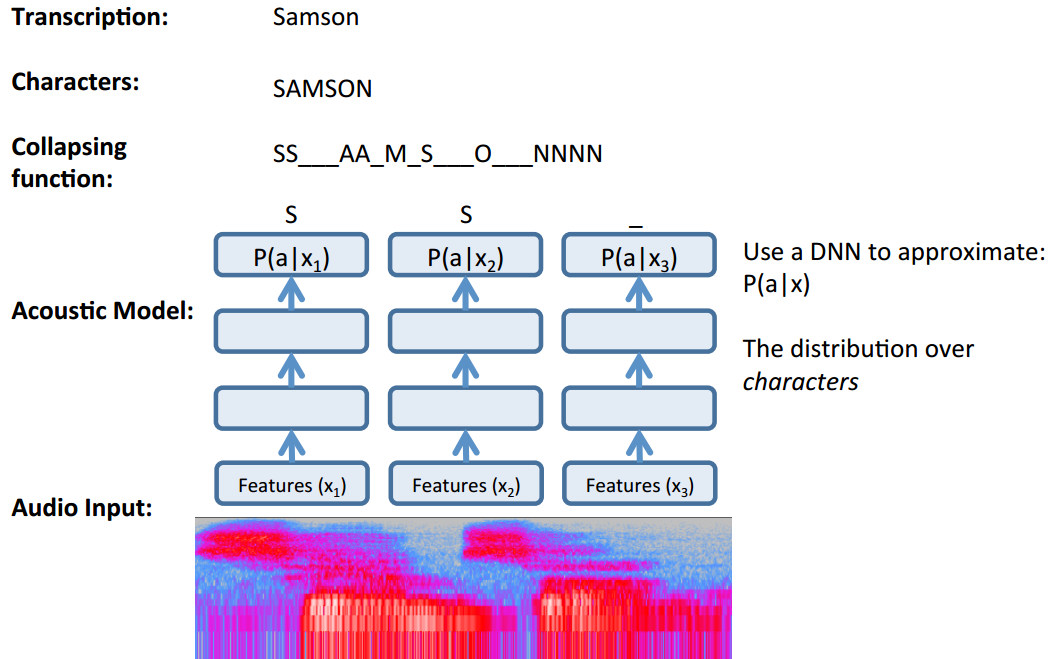


使用HMM-DNN hybrid混合建模

* 1. Moden HMM-DNNs

更深的网络，更多的参数，网络模型改进（使用tanh和relu等替换sigmod）等等。

* 1. HMM-free RNN recognition



Graves 2014 使用HMM-Free recognition 其中DNN可以用RNN

1. NMT Neural Machine Translation
   1. MT 演化
2. Phrase-based MT 基于短语的MT。句子先分成短语块，再进行：

Translation model：查找短语对应的翻译，translate locally

Language model：多个短语联合建立语言模型，LM uses only target words

1. Joint Neural Language Model 联合神经语言模型

Conditioned on source words (Devlin et al. 2014), still translate locally.

1. Neural Machine Translation NMT 神经网络机器翻译

Sequence-to-sequence: translate globally

End-to-end: simple & generalizable