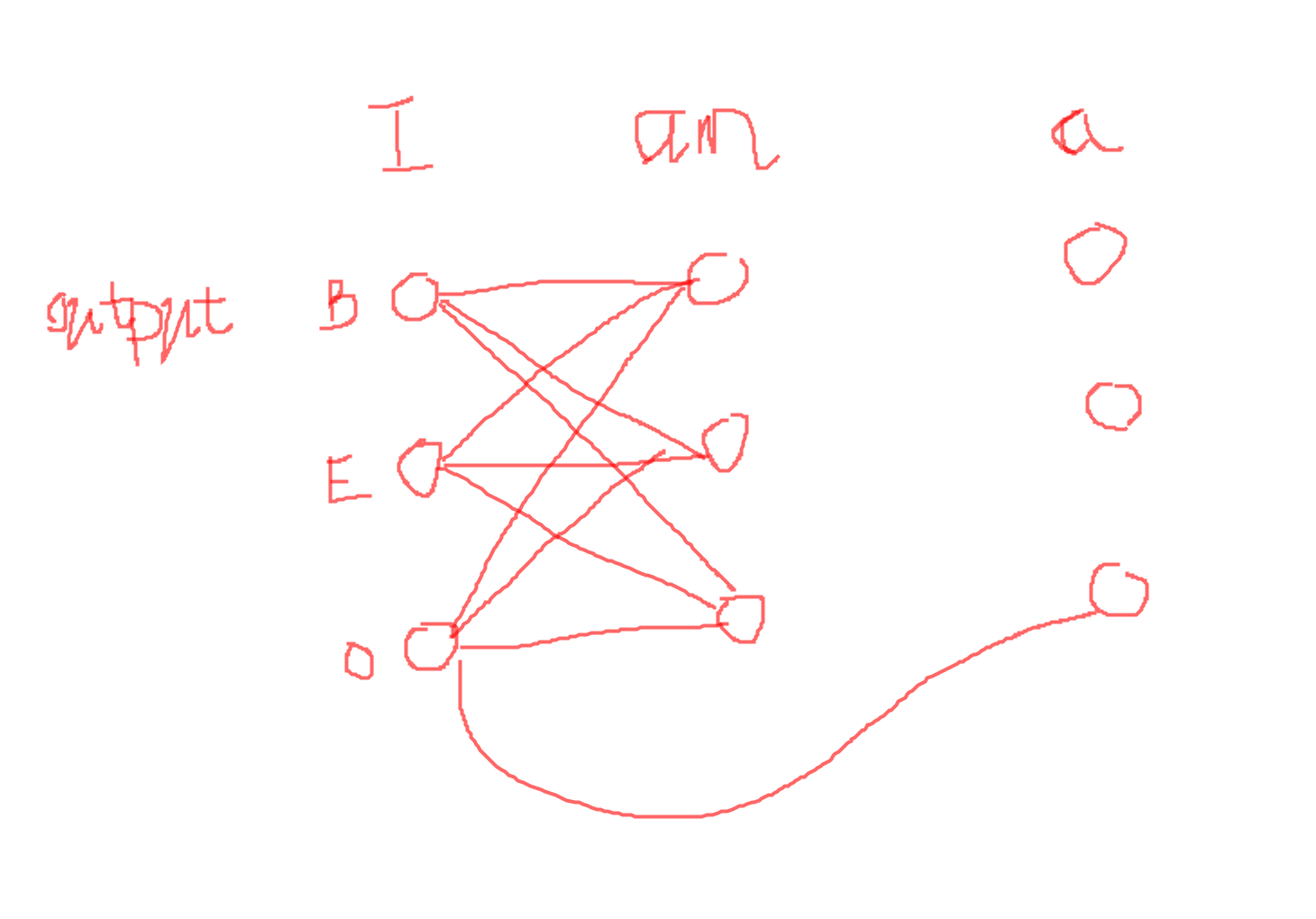
看了源码，对crf++特征函数理解有更正

1. 不能叫做“特征与输出通过模板对应”，应该是在当前位置通过特征模板的行列变换锁定特征函数，遍历所有可能的输出，若特征与输出对应则return 1，N行L输出，所有总数为L\*N。相应的，bigram也同样锁定特征，遍历所有可能输出的组合（构造方式：特征模板来确定标签相对位置，如%x[0,1]为考虑上一个输出标签）进行判定，为L\*L\*N个
2. 之前我误以为若不设置，bigram是默认在unigram的基础上提取上一个输出进行组合（官方给出的example都没有bigram模板），其实是要自己设置的，而我实验的样本也同样没有使用bigram，原因是语料库只有字特征，没有隐状态特征（即没有类似词、词分类、词性的对应），测试时需自己填充相同的隐变量，相应的，unigram模板的列全为0，关注的是当前字与周围一个或某几个字的对应关系
3. 源码中设置了node、path，即crf公式展开为状态函数和转移函数，node对应unigram，path对应bigram，若设置bigram模板为%x[0,1]（考虑上一个输出标签），则图示和多层感知机相同，如下图我画的示意图：

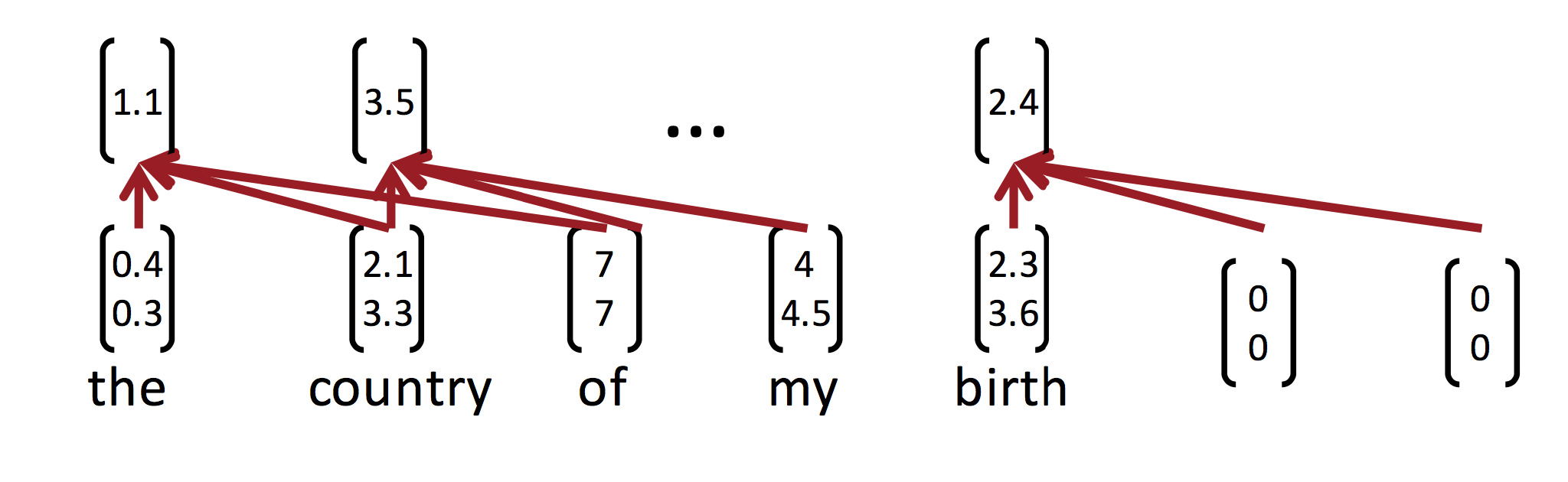


下面那条长线是bigram模板为%x[0,2]，考虑上上个输出

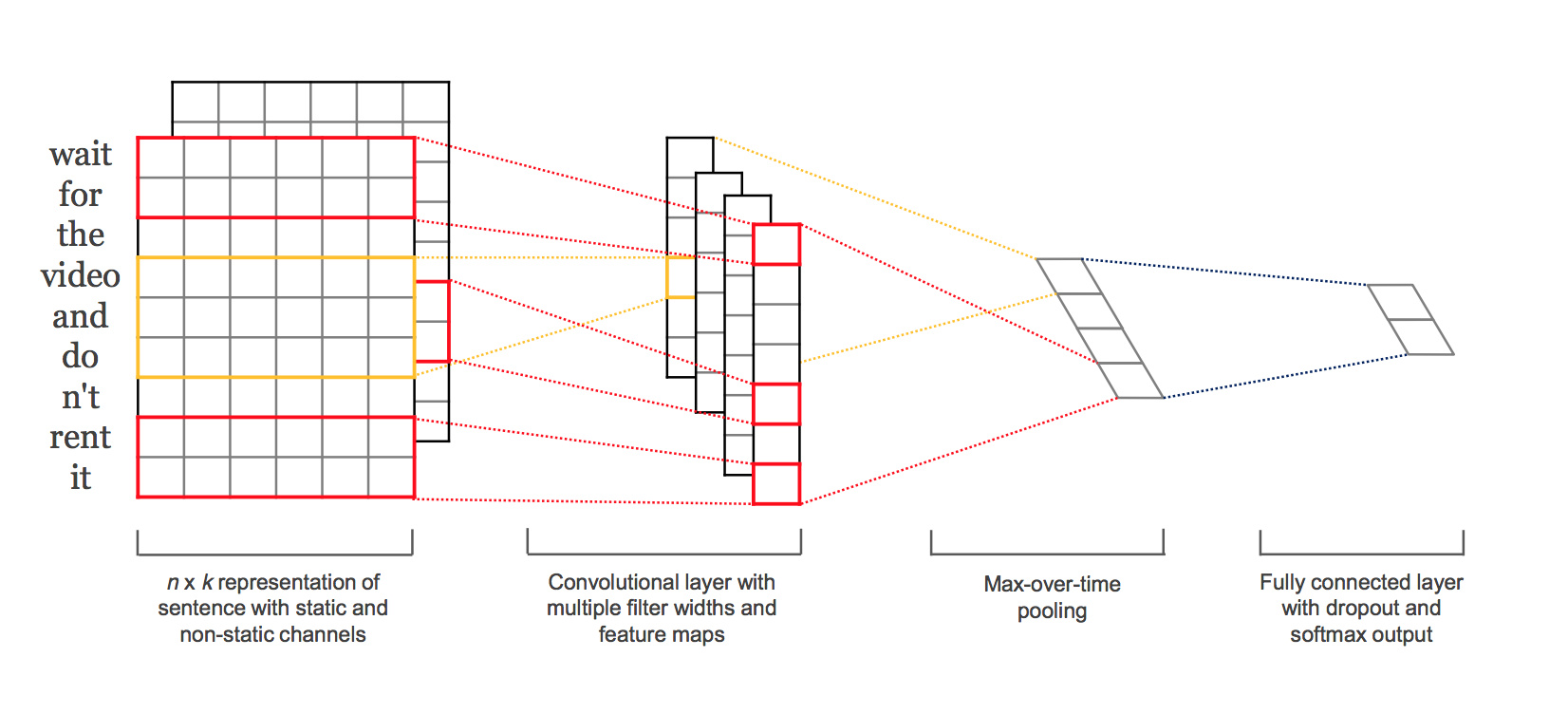
cs224n

Lecture 14 cnn

1. 从RNN到CNN
2. RNN无法用未来特征预测当前单词
3. 最后一个单词影响过大
4. 分析单一词向量较难，RNN只捕捉从左到右的“上文”信息
5. 卷积：与代表特定特征的卷积核通过叠加运算的匹配过程
6. 池化：提取最重要的特征，一个trick是为防止词向量随分类目标而变化丢失泛化能力，可以设置两份相同词向量，一份可变一份不变，将两个通道卷积结果输入到池化层中
7. NLP中单层CNN如图，最后会补0



1. 总结：一张漂亮的图



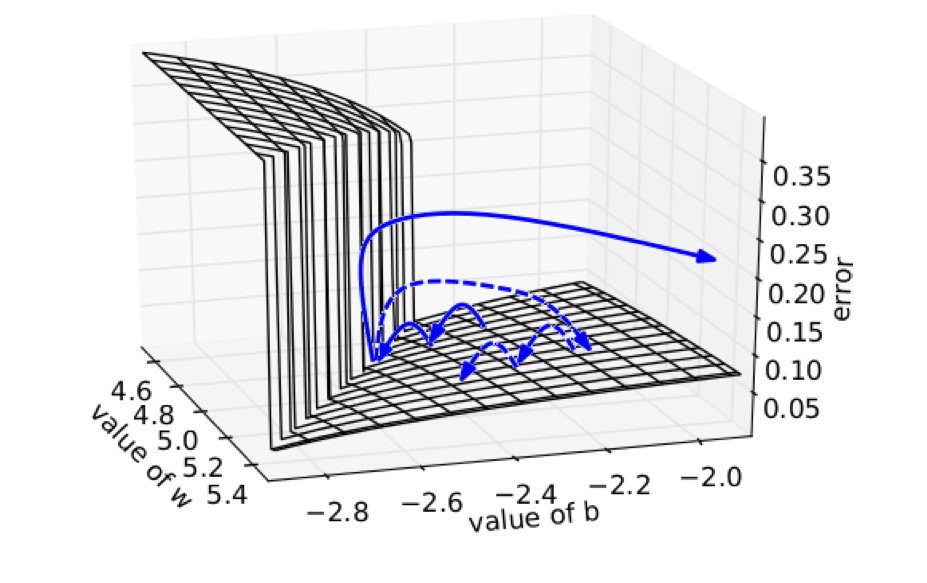
从左到右：双通道词向量、卷积核得到的特征表（红色是bigram，橙色是trigram）、池化提取最终特征、softmax输出

1. 提到的一些改进模型：
2. dropout：本质为通过伯努利分布缩放，在数据上采样子集，最后平均起来
3. 简单提到可用cnn做encoder、rnn做decoder来进行机器翻译，以及Quasi-RNN，在时序数据上做卷积

Lecture 8 rnn（无lstm）

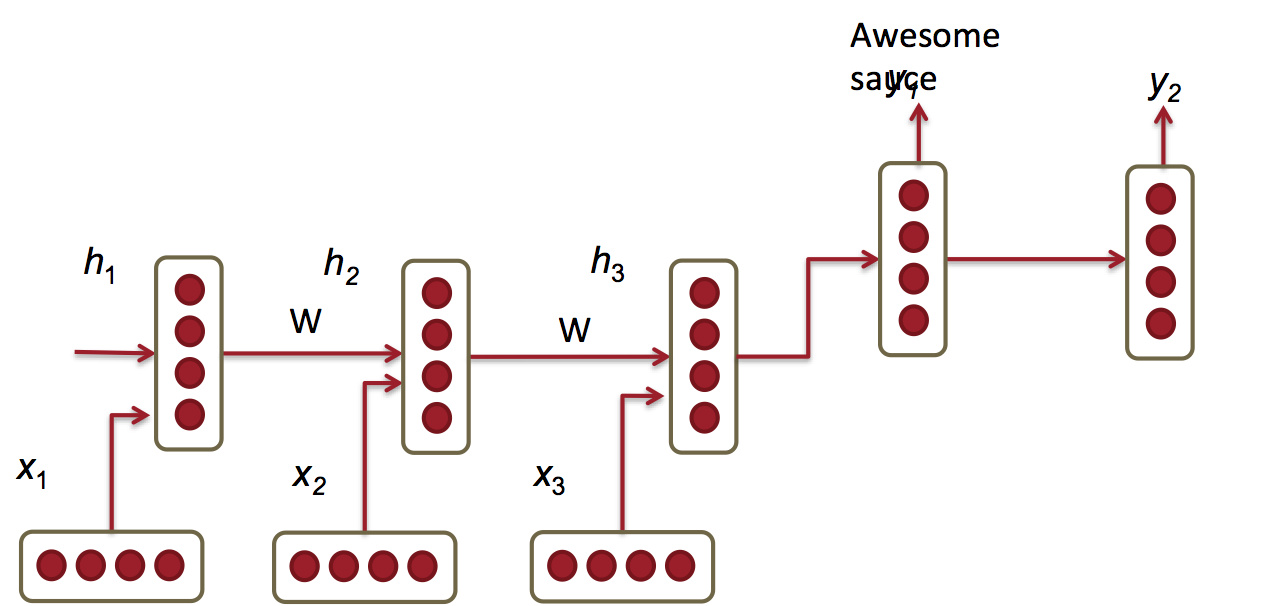
之前已整理内容的一些补充：

1. 参数矩阵初始化为单位矩阵，用ReLU激活函数，step多了之后，梯度消失的现象相对会减缓
2. 防止梯度爆炸的探索：梯度长度过大时自动缩放，如下图遇到梯度爆炸墙时缩放长箭头的梯度跳跃，但这个trick无法推广到梯度消失



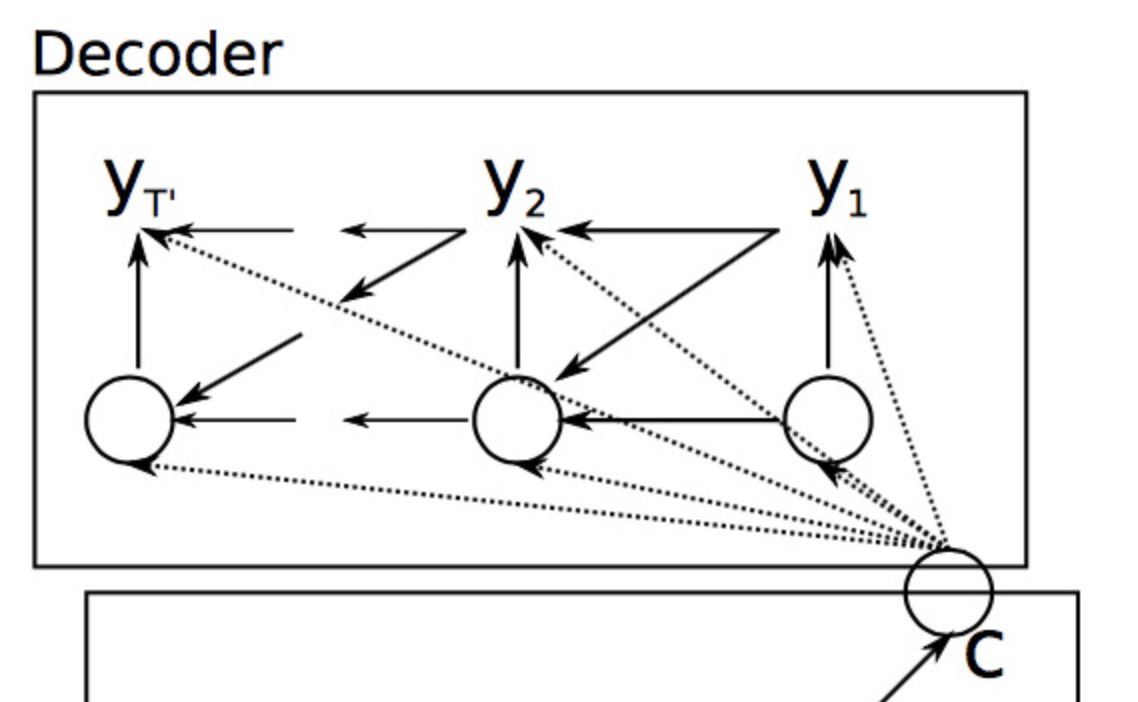
1. 词表太大，softmax分母项很麻烦，可以先预测词语分类（比如按词频分），再在分类中预测词。但词语分类过多会影响速度。
2. RNN机器翻译引入（下一节详细介绍）

基础模型：左encoder 右decoder



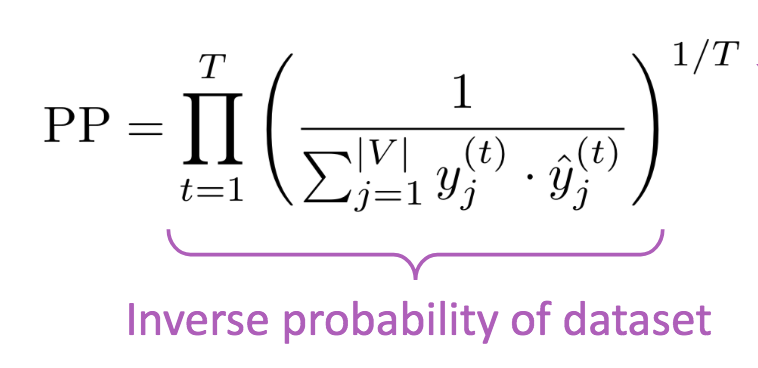
改进方法：encoder和decoder权值矩阵不同，

decoder输入拓展为前一时刻输出、隐藏层、encoder的组合如下图



还有使用更深的RNN、使用bi-encoder等等

1. 评估语言模型常用perplexity（困惑度）



越低越好，和最小化loss function是等价的