文章主题

句子分类（Convolutional Neural Networks for Sentence Classification）

1.使用的模型：
CNN-rand（初始数据参数是随机的，在训练期间修改）；
CNN-static（使用word2vec预训练的向量，但训练期间不对这些初始化参数进行修改）；
CNN-non-static（在训练中进行微调）；
CNN-multichannel（用静态加非静态的通道，好处是可以同时利用预训练的通用特征和任务特定的特征，提高模型的性能）
2.使用的数据集：
电影评论1(一句话，对应积极消极)；
评论2(对应更多的情绪，多加入了非常积极，中性，非常消极等)；
评论3(对于评论2移除中性标签)；
主观性数据集(对应主客观)；
问题数据集(问题是否关于人、地点、数字信息等)；
客户对各种产品的评论（对产品的正负面评价）
3.结论：
（1）CNN-rand随机化的效果差，加入word2vec预训练模型初始化的数据在效果上有了很大的提高，再加入微调可以使得效果更上一步
（2）多通道模型并没有明显优于单通道模型。
（3）非静态的微调比静态效果更好（没微调bad 跟 good相似，可能是因为在语法上一样），但在微调后可以修改，对于不在预训练向量中的，微调可以更好的学习。
（4）dropout是一个很好的正则化工具（在训练比较大的模型的时候）
4.展望
随机初始化不在word2vec中的单词时，本文的方法可能可以进一步改进
用Collobert等人(2011)在维基百科上训练的公开可用的词向量比word2vec效果差，尚未知道

序列到序列（Seq2Seq）机器翻译（Sequence to Sequence Learning with Neural Networks）

1.使用方法：
（1）使用了两个不同的LSTM:一个用于输入序列，另一个用于输出序列（因为这样做可以在忽略不计的计算成本下增加数量模型参数，并且可以很自然地同时在多个语言对上训练LSTM）；
（2）同时使用4层的lstm（发现深层的比浅层的效果好）
（3）将输入句子的单词顺序颠倒
2.使用的数据集：
（1）在一个由348M个法语单词和304M个英语单词组成的12M个句子子集上训练（这个是公开的，别人在这个数据集上做过实验）

（2）WMT ' 14英语到法语数据集（作为测试集）
源语言中使用了16万个最常见的单词，在目标语言中使用了8万个最常见的单词。每一个词汇外的单词都被一个特殊的“UNK”标记所取代（由于神经网络中每个单词都要对应一个向量）
（3）用BLEU分数和LSTM的测试困惑度来作为比较的参数
LSTM的测试困惑度（test perplexity）：是指在测试集上使用LSTM模型进行预测时，模型对于测试集中的文本的困惑度。困惑度是一种度量语言模型预测能力的指标，它表示模型对于给定文本序列的预测概率的倒数。困惑度越低，表示模型对于给定文本序列的预测越准确，LSTM模型在测试集上的困惑度从5.8降低到了4.7，这表明模型的预测能力得到了显著的提升。
3.结论：
（1）在几乎没有对问题结构做任何假设，可以在大规模MT任务上优于基于smt的标准系统
（2）把源句子倒过来发现效果很好。（找到一个具有最多短期依赖项的问题编码是很重要的，因为它们使学习问题变得更简单）
（3）此模型在长的句子上翻译的效果表现得很好。

序列建模实证（Empirical Evaluation of Gated Recurrent Neural Network on Sequence Modeling）

1.使用方法：GRU， lstm，tanh之间的比较
使用了20个分量的混合高斯函数作为输出层（"混合高斯函数"是指一种由多个高斯分布组成的概率密度函数。在序列建模任务中，可以使用混合高斯函数作为输出层来建模序列的概率分布。每个高斯分布代表了序列中的一个可能的状态或观测，而混合高斯函数则通过对这些高斯分布进行加权组合来表示整个序列的概率分布。目的：这意味着他们将序列建模任务转化为了一个混合高斯模型的参数估计问题，通过最大化给定训练序列的对数似然来学习混合高斯函数的参数。通过这种方式，他们可以对序列数据进行建模，并使用学习到的模型进行预测和生成）
2.使用的数据集：
使用复调音乐建模和语音信号建模来进行模型的比较
人为的选择每个模型的大小，以便每个模型具有大约相同数量的参数。有意使模型足够小，以避免过度拟合。
文章使用负对数概率进行比较，概率越低表示模型拟合的越好
又使用了学习曲线来证明门控系统比传统的好（学习曲线下降是指在训练过程中，模型的性能逐渐提升，损失函数的值逐渐减小的过程。在图中，学习曲线下降表示模型在训练集和验证集上的负对数似然损失逐渐减小。这意味着模型在学习过程中逐渐提高了对数据的拟合能力，捕捉到了数据中的模式和规律。学习曲线下降越快，表示模型学习得越快，性能提升得越好）
3.结论
在实验中GRU-RNN在除了Nottingham数据集之外的所有数据集上表现最好（LSTM-RNN和tanh-RNN）
虽然可以得到门控系统比传统的tanh表现的好，但对于CRU 和 lstm哪个比较好，本文还没法给出结果。

端到端的序列标记（End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF）

1.使用方法：
（1）双向LSTM-CNNs-CRF（使用字符嵌入的方法，减少对手工特征和数据预处理的依赖，同时可以提高模型对单词的学习效果）
（2）双向lstm基本思想是将每个序列向前和向后呈现为两个独立的隐藏状态，分别捕获过去和未来的信息。然后将两个隐藏状态连接起来形成最终输出
（3）首先是把词嵌入向量给CNN进行字符级别的表示学习（之前先用dropout层），在给双向lstm进行上下文建模，最后使用了一个顺序条件随机场（CRF）来联合解码整个句子的标签
（4）CRF是联合解码（比如形容词后面更有可能跟着名词而不是动词），这个方法效果好。
（5）优化方法：使用小批量随机梯度下降法(SGD)进行参数优化；使用梯度裁剪；对学习率也有一定的设置（选择初始学习率为η0 (POS标记为η0 = 0.01, NER为0.015，见3.3节)，并且学习率在训练的每个历元上更新为ηt = η0/(1 + ρt)，衰减率ρ = 0.05, t为完成的历元数）；使用早停（本文中最好的系数出现在50轮）；微调（对字符嵌入参数进行微调）；用dropout层放过拟合。
2.使用的数据集：使用两个序列标记任务- Penn Treebank WSJ词性标记(POS)语料库和CoNLL 2003命名实体识别(NER)语料库来验证模型的准确性
3.结论
（1）使用glove词嵌入比其他词嵌入的预训练集效果好，使用预训练集又比随即嵌入效果好
（2）使用dropout层确实对防止过拟合有效果
（3）文中提到的模型比先前的模型效果都好。
（4）文中考虑了词嵌入和训练集中没有出现的词，对这些词的处理该模型的处理效果会是如何的，结果当然模型对这些词处理效果优秀。

文章分类（Hierarchical Attention Networks for Document Classification）

1.使用方法：层次注意网络（HAN）
（1）HAN是由word encoder->word attention->sentence encoder->sentence attention
其中encoder是用GRU来进行编码，由于在每篇不同的文章中相同的词的权重不一样，所以本文引入了注意力机制，最后通过从词到句子再到文章，最后通过softmax来输出文章分类的结果。
（2）文中用了Linear methods（用了逻辑回顾）；SVMs；Neural Network methods（其中包括CNN，lstm，Conv-GRNN and LSTM-GRNN）来与HAN进行比较
（3）HN-{AVE, MAX, ATT}中又分为了AVE MAX ATT（HN-AVE：使用平均汇总的方式，将每个句子的注意力权重与其对应的单词的注意力权重相乘，然后对所有句子进行平均，得到文档级别的表示。 - HN-MAX：使用最大汇总的方式，将每个句子的注意力权重与其对应的单词的注意力权重相乘，然后选择最大的权重作为文档级别的表示。 - HN-ATT：使用层次注意力机制，通过计算每个句子的注意力权重和每个单词的注意力权重，将句子级别的表示和单词级别的表示进行加权汇总，得到最终的文档级别表示）

2.使用的数据集：用Yelp reviews；IMDB reviews；Yahoo answers等数据评估
3.结论
（1）HN-ATT效果最好
（2）文中可视化了该模型会选择高权重的单词比如delicious, amazing, terrible。又可视化了在低评分中，注意力机制对good的注意力权重集中在低端，表明了在不同的文章中，相同的

词语在该文章中的权重不同。又可视化了模型在文章中对非形成性的句子和单词的选择权重，模型会抛弃这些不重要的单词（非形成性的句子和单词是指在文本分类任务中，一些句子和单词可能对于分类结果没有太大的贡献，它们在模型中被认为是非信息性的）