RNN

RNN

引入语言模型
- 语言模型任务：根据已有单词预测将要出现的单词
- 前馈神经网络不能很好处理时间序列数据
- CBOW目的：从上下文预测出目标词
  作为副产品，获得了编码了单词含义的分布式表示
- 使用概率评估单词序列发生的可能性
  按照“作为句子是否自然”对候选句子排序
  条件语言模型：以目标词左侧全部单词作为上下文的概率
  P(wt|(w1,w2,…,wt-1))：n-1阶马尔可夫链
  - 若以CBOW为例：输入忽略上下文顺序，
    而拼接会使参数随上下文成比例增加
  - 如何解决：Recurrent NN
    RNN：无论上下文多长，都能将上下文记住
    能处理任意长度时序数据
    （Recursive NN递归神经网络，处理树结构）

结构
- 层内循环：多个RNN在同一层
- 具有状态：隐藏状态ht
- BPTT：基于时间的反向传播
  时序跨度增大，消耗资源成比例
  - 分治：水平方向截块计算
    Truncated BPTT
- 正向传播之间有关联：按顺序输入

语言模型的评价：困惑度（分叉度）
概率的倒数，概率约大，困惑都越小，模型越好

Gated RNN

- RNN无法解决长期依赖，梯度消失、爆炸
  GRU与LSTM效果相似，但更易于计算
- RNN每次矩阵乘积使用相同权重Wh，
  梯度大小随时间步长指数级增加
  - >1梯度爆炸，对策：梯度裁剪，设置阈值
  - <1梯度消失，改变结构：加入记忆单元c
    仅在LSTM内部接收和传递数据

门：阻止和释放水流，
控制开合程度
- 输出门：管理隐藏状态ht的输出
  - tanh[-1,1]:信息强弱程度
    sigmoid[0,1]:数据流出比例
- 遗忘门：ct-1：忘记不必要记忆，添加新的tanh记忆
- 输入门：添加加权后的新信息

为什么不会梯度消失：
- 梯度消失：RNN相同权重重复计算
- LSTM反向传播不是矩阵乘积，
  基于不同门值进行对应元素乘积

进一步改进：RNNLM
- 多层化：加深，叠加多个LSTM层
- 过拟合
  - 增加数据
  - 降低模型复杂度
    - 正则化
    - dropout：
      产生噪声会随时间成比例积累
      因此不要在时间轴上而是在垂直方向插入
      （在时间上变分dropout：同一层使用相同
      mask决定是否传递数据使得信息损失方式
      固定，避免常规指数级信息损失）
- 权重共享
  Embeding和Affine层
  - 减少学习参数数量
  - 抑制过拟合

seq2seq
从时序到时序

使用语言模型进行文本生成：
根据概率分布进行选择下一个单词，
概率高的更容易被选到

原理
- 组合RNN
- Encoder：将任意长度文本转换为固定长度向量
- Decoder：与编码层结构相同，
  唯一微小改变：接受向量h
  编码器与解码器的桥梁

改进
- reverse：反转数据，反转后梯度传播更平滑
- peeky：共享LSTM中重要信息h

当前编码器将所有信息转化为固定向量
（最后时刻的隐藏状态）
有用信息会溢出
需要改进seq2seq

Attention
注意力机制

编码
- 使用各个时刻LSTM隐藏状态构成hs矩阵

关注必要信息进行时序转换
输入和输出哪些单词相关：对齐
问题：这种选择操作无法微分
解决：全选，另行计算各单词贡献度权重a

解码
- a的求解方法：
  使用内积计算两个向量相似度

双向LSTM：添加反方向LSTM，
拼接各时刻两个LSTM层的隐藏状态

加深层表现力更强
残差连接（skip connection）
跨层连接，加法原样传播梯度。

Transformer
- 问题：不能在时间方向并行计算RNN
- 由Attention构成，使用self-Attention
  以时序数据为对象，观察一个时序数据中
  每个元素与其他元素的关系
- 控制计算量，充分利用并行计算

NTM（Neural Turing Machine）神经图灵机：
使用可微分的计算构建内存操作