RNN循环神经网络

潜变量自回归模型
x_t和它的潜变量h_t和上一个数据x_{t-1}相关

有隐状态的循环神经网络

假设在时间步`t`有小批量输入$X_t \in \mathbb{R}^{n\times d}$，对于有n个序列样本的小批量，$X_t$的每一行对应来自该序列的时间步t处的一个样本。在MLP的基础上增加了隐变量H_{t-1}并且引入了一个新的权重$W_{hh} \in \mathbb{R}$，以此来描述当前时间步中使用前一个时间步的隐变量。当前时间步的隐变量由当前时间步的输入和前一个时间步的隐变量得出：

$$H_t = \phi(X_tW_{xh} + H_{t-1}W_{hh} + b_h)$$

$$o_t = \phi(W_{oh}h_t + b_o)$$

增添的一项$H_{t-1}W_{hh}$保留了相邻时间步之间隐变量的关系，保留了当前时间步下的历史信息，例如当前时间步下神经网络的状态和记忆，因此这样的隐藏变量被称为隐状态。
输入是更新前的隐藏状态H_{t-1}，输出是更新后的隐藏状态H_t和输出o_t

由公式定义可以得到输出o_t是根据本时间步的输入x_t和隐变量h_{t-1}来预测的，最后计算损失的时候是比较o_t和真实标签的损失

衡量一个语言模型好坏可以使用平均交叉熵

$$\pi = \frac{1}{n}\sum_{i=1}^{n} -log\, p(x_t|x_{t-1},\dots)$$

`p`是语言模型预测概率，x_t是真实词
NLP使用困惑度$\exp(\pi)$来衡量，平均每次可能选项，1代表完美，无穷大是最差情况

梯度剪裁

每次迭代的时候要计算T个时间步的梯度，反向传播的时候要产生长度为O(T)的矩阵乘法链，导致数值不稳定
梯度剪裁能够有效预防梯度爆炸
如果梯度长度超过θ，那么拖回长度θ
$g \leftarrow min(1, \frac{\theta}{||g||})g$

由于RNN的参数共享机制导致RNN并不能很好地学习到前面的内容

RNN应用

文本生成one to many 给定起始词输出后续句子
文本分类many to one 对文本进行类别划分
问答、机器翻译many to many 前面输入句子后对其进行输出
tag生成 many to many 对每个词进行输出

门控循环单元GRU

能关注的机制(更新门)：

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r)$$

能遗忘的机制(重置门)：

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

候选隐状态：

$$\hat{H}_t = \tanh\left(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h\right)$$

假设R_t里面的元素全部等于0相当于重新初始化起始状态，如果全部等于1则相当于全部历史信息保留，就变为RNN

真正的隐状态

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \hat{H}_t$$

假设$Z_t = 1$则不更新历史信息，忽略掉当前状态，忽略当前输入X_t的影响，假设$Z_t = 0$则不看过去的状态，就看当前更新后的状态

其中候选隐状态是在新加入X_t之后进行更新后的状态，H_{t-1}则是上一时间步的隐状态，最后得到当前的隐状态，其中重置门相当于对过去隐状态和候选隐状态进行一个权重相加

R_t的作用是挑选过去的哪些历史信息是对现在有用的，Z_t的作用则是挑选现在的条件进一步进行筛选与更新，保留下来有用的