CS224N-2021 Assignment 3:Dependency Parsing

Jingwei Huang

2022-06-16

1 Machine Learning & Neural Networks

1.1 Question (a):Adam Optimizer

1.1.1 i:momentum

该方法为带动量的 SGD（SGD with Momentum）。

当参数当前时刻的梯度与历史时刻梯度方向相似时，参数更新步长为两梯度矢量之和，当前时刻的参数更新步长则会加强；反之，参数更新步长为两梯度矢量之差，则当前时刻的参数更新步长减弱，可帮助模型摆脱平坦区，使它更不容易陷入局部最小值。

1.1.2 ii:adaptive learning rates

自适应梯度（AdaGrad）根据梯度的平方和的倒数的平方根来衡量每个参数的学习速率。该方法将稀疏梯度方向上的梯度放大，从而使得稀疏特征的收敛速度更快。

1.2 Question (b):Dropout

参数说明

* $\mathbf{h}$：模型网络层。
* D_h：h 网络层的神经元总数。
* p_{drop}：随机失活的神经元在该网络层神经元总数的占比。

1.2.1 i:what

当模型在训练阶段使用 dropout 技术，当网络层 $\mathbf{h}$ 的输入为 h_i 时，神经元的输出期望为：

$$\begin{aligned}\mathbb{E}_{p_{\text{drop}}}^{\text{trian}}[\mathbf{h}] &= \sum_{i=0}^{D_h}(1-p_{\text{drop}}) * h_i + p_{\text{drop}} * 0 \\ &= \sum_{i=0}^{D_h}(1-p_{\text{drop}})h_i \\ &= (1-p_{\text{drop}}) * \mathbf{h}\end{aligned}$$

由于测试阶段会关闭 dropout，对于同样的输入，神经元此时的输出期望是 $\mathbb{E}_{p_{\text{drop}}}^{\text{test}}[\mathbf{h}] = \mathbf{h}$。

应对网络层输入进行 resale 操作确保神经元在训练阶段和测试阶段的输出期望保持一致，既 $\gamma * \mathbb{E}_{p_{\text{drop}}}^{\text{trian}}[\mathbf{h}] = \mathbb{E}_{p_{\text{drop}}}^{\text{test}}[\mathbf{h}]$。

令 $\gamma * (1-p_{\text{drop}}) * \mathbf{h} = \mathbf{h}$，得 $\gamma = \frac{1}{1-p_{\text{drop}}}$。

1.2.2 ii:Why

Q1:Why should dropout be applied during training?

A1: 训练的时候 dropout 的作用就是通过引入噪声，防止模型过拟合。dropout 导致两个神经元不一定每次都在一个 dropout 网络中出现。这样权值的更新不再依赖于有固定关系的隐含节点的共同作用，阻止了某些特征仅仅在其它特定特征下才有效果的情况

Q2:Why should dropout NOT be applied during evaluation?

A2: 因为 dropout 会对网络层进行随机失活，因此不关闭 dropout 的模型近似于随机模型。给定 x，模型的单次预测值 y 是从相应的数据分布中进行随机采样。模型的最终预测值是多次采样后 y 的均值，但是多次采样会带来额外的计算开销。基于一个假设，平均参数（关闭 dropout）得到的模型的预测值近似等于上面说的多次采样的平均值。因此，模型的测试阶段需要关闭 dropout。

2 Neural Transition-Based Dependency Parsing

2.0.1 refresher

我们构造一个三元组，分别是 Stack、Buffer 和一个 Dependency Set。

1. Stack 最开始只存放一个 Root 节点；
2. Buffer 则装有我们需要解析的一个句子；
3. Set 中则保存我们分析出来的依赖关系，最开始是空的。

我们要做的事情，就是不断地把 Buffer 中的词往 Stack 中推，跟 Stack 中的词判断是否有依赖关系，有的话则输出到 Set 中，直到 Buffer 中的词全部推出，Stack 中也仅剩一个 Root，就分析完毕了。

基于转移的依存分析包含以下操作（action）：

1. 移进 SHIFT：Stack 词汇无依存关系，将 Buffer 左顶词汇移入 Stack。
2. LEFT-ARC：左依存，箭头向左指，移除 Stack 从属词。
3. RIGHT-ARC：右依存，箭头向右指，移除 Stack 从属词。

2.0.2 Question (a)

Stack	Buffer	New dependency	Transition
[ROOT]	[I, parsed, this, sentence, correctly]		Initial Configuration
[ROOT, I]	[parsed, this, sentence, correctly]		SHIFT
[ROOT, I, parsed]	[this, sentence, correctly]		SHIFT
[ROOT, parsed]	[this, sentence, correctly]	I←parsed	LEFT-ARC
[ROOT, parsed,this]	[sentence, correctly]		SHIFT
[ROOT, parsed,this,sentence]	[correctly]		SHIFT
[ROOT, parsed,sentence]	[correctly]	this←sentence	LEFT-ARC
[ROOT, parsed]	[correctly]	parsed→sentence	RIGHT-ARC
[ROOT, parsed,correctly]	[]		SHIFT
[ROOT, parsed]	[]	parsed→correctly	RIGHT-ARC
[ROOT]	[]	ROOT→parsed	RIGHT-ARC

2.0.3 Question (b)

总操作步数为 2n。因为对于一个词汇数量为 n 的句子，我们需要对每个单词进行一次 SHIFT 和一次 ARC 操作，则操作总数为 2n。

2.0.4 Question (c)

done!

2.0.5 Question (d)

done!

2.0.6 Question (e)

2.0.7 Question (f)