

Assignment3 报告

一 . 对 LSTM 的求导

1.1 单步 LSTM 的求导

$$h_t = o_t * \tanh(C_t)$$

$$\frac{\partial h_t}{\partial o_t} = \tanh(C_t)$$

$$\frac{\partial h_t}{\partial C_t} = o_t * (1 - \tanh^2(C_t))$$

$$o_t = \sigma(W_o \cdot z + b_o)$$

$$\frac{\partial o_t}{\partial W_o} = \frac{\partial o_t}{\partial (W_o z + b_o)} * \frac{\partial (W_o z + b_o)}{\partial W_o} = [o_t * (1 - o_t)] \cdot z^T$$

$$\frac{\partial h_t}{\partial W_o} = \frac{\partial h_t}{\partial o_t} * \frac{\partial o_t}{\partial W_o} = \tanh(C_t) * [o_t * (1 - o_t)] \cdot z^T$$

$$\frac{\partial o_t}{\partial b_o} = \frac{\partial o_t}{\partial (W_o z + b_o)} * \frac{\partial (W_o z + b_o)}{\partial b_o} = o_t * (1 - o_t)$$

$$\frac{\partial h_t}{\partial b_o} = \frac{\partial h_t}{\partial o_t} * \frac{\partial o_t}{\partial b_o} = \tanh(C_t) * [o_t * (1 - o_t)]$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t$$

$$\frac{\partial h_t}{\partial f_t} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial f_t} = o_t * (1 - \tanh^2(C_t)) * C_{t-1}$$

$$\frac{\partial h_t}{\partial C_{t-1}} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial C_{t-1}} = o_t * (1 - \tanh^2(C_t)) * f_t$$

$$\frac{\partial h_t}{\partial i_t} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial i_t} = o_t * (1 - \tanh^2(C_t)) * \bar{C}_t$$

$$\frac{\partial h_t}{\partial \bar{C}_t} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial \bar{C}_t} = o_t * (1 - \tanh^2(C_t)) * i_t$$

$$\begin{aligned} \frac{\partial h_t}{\partial W_f} &= \frac{\partial h_t}{\partial f_t} * \frac{\partial f_t}{\partial (W_f z + b_f)} * \frac{\partial (W_f z + b_f)}{\partial W_f} \\ &= o_t * (1 - \tanh^2(C_t)) * C_{t-1} * [f_t * (1 - f_t)] \cdot z^T \end{aligned}$$

$$\begin{aligned} \frac{\partial h_t}{\partial b_f} &= \frac{\partial h_t}{\partial f_t} * \frac{\partial f_t}{\partial (W_f z + b_f)} * \frac{\partial (W_f z + b_f)}{\partial b_f} \\ &= o_t * (1 - \tanh^2(C_t)) * C_{t-1} * [f_t * (1 - f_t)] \end{aligned}$$

$$\begin{aligned} \frac{\partial h_t}{\partial W_i} &= \frac{\partial h_t}{\partial i_t} * \frac{\partial i_t}{\partial (W_i z + b_i)} * \frac{\partial (W_i z + b_i)}{\partial W_i} \\ &= o_t * (1 - \tanh^2(C_t)) * \bar{C}_t * [i_t * (1 - i_t)] \cdot z^T \end{aligned}$$

$$\frac{\partial h_t}{\partial b_i} = \frac{\partial h_t}{\partial i_t} * \frac{\partial i_t}{\partial (W_i z + b_i)} * \frac{\partial (W_i z + b_i)}{\partial b_i} = o_t * (1 - \tanh^2(C_t)) * \bar{C}_t * [i_t * (1 - i_t)]$$

$$\frac{\partial h_t}{\partial W_c} = \frac{\partial h_t}{\partial \bar{C}_t} * \frac{\partial \bar{C}_t}{\partial (W_c z + b_c)} * \frac{\partial (W_c z + b_c)}{\partial W_c} = o_t * (1 - \tanh^2(C_t)) * i_t * (1 - \bar{C}_t^2) \cdot z^T$$

$$\frac{\partial h_t}{\partial b_c} = \frac{\partial h_t}{\partial \bar{C}_t} * \frac{\partial \bar{C}_t}{\partial (W_c z + b_c)} * \frac{\partial (W_c z + b_c)}{\partial b_c} = o_t * (1 - \tanh^2(C_t)) * i_t * (1 - \bar{C}_t^2)$$

因为 $z = [h_{t-1}, x_t]$ ，所以不妨设 $W_* = [U_*, V_*]$ ，其中 U_* 的维度和 h_t 相同。

$$\begin{aligned} \frac{\partial h_t}{\partial h_{t-1}} &= \tanh(C_t) * \frac{\partial o_t}{\partial h_{t-1}} + o_t * \frac{\partial \tanh(C_t)}{\partial h_{t-1}} \\ &= \tanh(C_t) * \frac{\partial o_t}{\partial \sigma} * \frac{\partial \sigma}{\partial h_{t-1}} + o_t * \frac{\partial h}{\partial C_t} * \frac{\partial C_t}{\partial h_{t-1}} \\ &= U_o^T \cdot (\tanh(C_t) * [o_t * (1 - o_t)] + o_t * (1 - \tanh^2(C_t))) \\ &\quad * \left(C_{t-1} * \frac{\partial f_t}{\partial h_{t-1}} + \bar{C}_t * \frac{\partial i_t}{\partial h_{t-1}} + i_t * \frac{\partial \bar{C}_t}{\partial h_{t-1}} \right) \\ &= U_o^T \cdot (\tanh(C_t) * [o_t * (1 - o_t)] + o_t * (1 - \tanh^2(C_t)) * [U_f^T \cdot (C_{t-1} * f_t * (1 - f_t)) \\ &\quad + U_i^T \cdot (\bar{C}_t * i_t * (1 - i_t)) + U_c^T \cdot (i_t * (1 - \bar{C}_t^2))]) \end{aligned}$$

同理可得

$$\begin{aligned} \frac{\partial h_t}{\partial x_t} &= V_o^T \cdot (\tanh(C_t) * [o_t * (1 - o_t)] + o_t * (1 - \tanh^2(C_t)) * [V_f^T \\ &\quad \cdot (C_{t-1} * f_t * (1 - f_t)) + V_i^T \cdot (\bar{C}_t * i_t * (1 - i_t)) + V_c^T \cdot (i_t * (1 - \bar{C}_t^2))]) \end{aligned}$$

1.2 LSTM 在时间序列上的求导

我们采用交叉熵函数当做损失函数，设序列的最后位置为 τ

$$\hat{y}_t = \text{softmax}(W_p h_t + b_p)$$

$$L = -\sum_{t=1}^{\tau} y_t * \log \hat{y}_t$$

则当 $t = \tau$ 时，求导和 1.1 过程一样，梯度为

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial \hat{y}_t} * \frac{\partial \hat{y}_t}{\partial (W_p h_t + b_p)} * \frac{\partial (W_p h_t + b_p)}{\partial h_t} = W_p^T \cdot (\hat{y}_t - y_t)$$

当 $t < \tau$ 时，对 h_t 的求导为

$$\begin{aligned} \frac{\partial L}{\partial h_t} &= \frac{\partial \sum_{k=t}^{\tau} (-y_k \log \hat{y}_k)}{\partial h_t} = \frac{\partial \sum_{k=t+1}^{\tau} (-y_k \log \hat{y}_k)}{\partial h_t} + \frac{\partial y_t \log \hat{y}_t}{\partial h_t} \\ &= \frac{\partial \sum_{k=t+1}^{\tau} (-y_k \log \hat{y}_k)}{\partial h_{t+1}} * \frac{\partial h_{t+1}}{\partial h_t} + \frac{\partial y_t \log \hat{y}_t}{\partial h_t} \mathbf{0} \\ &= \frac{\partial L}{\partial h_{t+1}} * \frac{\partial h_{t+1}}{\partial h_t} + W_p^T \cdot (\hat{y}_t - y_t) \end{aligned}$$

所以对 W_o 等权重矩阵求导为

$$\frac{\partial L}{\partial W_o} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial h_t} * \frac{\partial h_t}{\partial W_o}$$

$$\frac{\partial L}{\partial b_o} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial h_t} * \frac{\partial h_t}{\partial b_o}$$

当 $t = \tau$ 时，对 C_t 求导为

$$\frac{\partial L}{\partial C_t} = \frac{\partial L}{\partial h_t} * \frac{\partial h_t}{\partial C_t}$$

当 $t < \tau$ 时，因为 $h_t = o_t * \tanh(C_t)$ 且 $C_{t+1} = f_{t+1} * C_t + i_{t+1} * \overline{C_{t+1}}$ ，所以对 C_t 求导为

$$\frac{\partial L}{\partial C_t} = \frac{\partial L}{\partial h_t} * \frac{\partial h_t}{\partial C_t} + \frac{\partial L}{\partial C_{t+1}} * \frac{\partial C_{t+1}}{\partial C_t} = \frac{\partial L}{\partial h_t} * o_t * (1 - \tanh^2(C_t)) + \frac{\partial L}{\partial C_{t+1}} * f_{t+1}$$

因此对 W_* 和 b_* 的求导为

$$\frac{\partial L}{\partial W_f} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial W_f} = \sum_{t=1}^{\tau} \left(\frac{\partial L}{\partial C_t} * C_{t-1} * [f_t * (1 - f_t)] \right) \cdot z^T$$

$$\frac{\partial L}{\partial b_f} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial b_f} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * C_{t-1} * [f_t * (1 - f_t)]$$

$$\frac{\partial L}{\partial W_i} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial W_i} = \sum_{t=1}^{\tau} \left(\frac{\partial L}{\partial C_t} * \overline{C_t} * [i_t * (1 - i_t)] \right) \cdot z^T$$

$$\frac{\partial L}{\partial b_i} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial b_i} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * C_{t-1} * [i_t * (1 - i_t)]$$

$$\frac{\partial L}{\partial W_c} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial W_c} = \sum_{t=1}^{\tau} \left(\frac{\partial L}{\partial C_t} * i_t * (1 - \overline{C_t}^2) \right) \cdot z^T$$

$$\frac{\partial L}{\partial b_c} = \sum_{t=1}^{\tau} \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial b_i} = \sum_{t=1}^{\tau} \left(\frac{\partial L}{\partial C_t} * i_t * (1 - \overline{C_t}^2) \right)$$

二、LSTM 的训练

1. 初始化

初始化训练参数如下：

Vocabulary size $|V| = 5656$

Batch size=20

Sentence length=64

Hidden size=256

Input size=128

Embedding 采用 Pytorch 自带的 `nn.Embedding()`。

权重矩阵 W_* 和偏置矩阵 b_* 初始化为 $[-S, +S]$ 的均匀分布，其中 $S = \sqrt{\frac{1}{\text{output size}}}$ ，在这里

$\text{output size} = \text{hidden size}$ 一开始我取均值为 0，方差为 1 的标准正态分布，后来发现这样初始化效果很差，一开始的 loss 很高。网上查阅了 Pytorch 自带的 `nn.Linear` 层的权重初始化才采取了这个分布，效果和 `nn.Linear()` 对比差不多。不设为 0 的原因是因为求导采用链式法则，只要有一步导数为 0 整个导数即为 0，会导致整个模型一开始梯度下降很慢甚至梯度没有下降。例如，当尝试将 $a_t = Wh_t + b$ 中的 W 和 b 取为 0 矩阵时，求出的 a_t 为 0，这样理论上 $\text{softmax}(a_t)$ 会出现除 0 的情况（虽然程序并没有报错）。因此需要取一个非 0 的随机值。

数据集：采用了这个 lab 给定的数据集，但是发现数据集过小导致非常容易过拟合。后

来尝试使用“全唐诗”的数据集，但是由于机器性能问题，运行完一轮需要很久，loss 下降得很慢，因此最终只使用了 700 首诗。

数据预处理：我认为 LSTM 能够学习到逗号和句号的特征，所以没有去掉逗号和句号。一开始将所有诗连起来然后以 sentence length 长度分开，但这样会破坏一整首诗的完整性导致给出第一句话不完整。所以我认为要保持整首诗的完整性，因此以诗为单位，诗句不足 sentence length 的补'E'填充符。如果一首诗长过 sentence length 则划分为几首诗。

早停策略：最近 10 个 epoch 发现 perplexity 没有更优值时停止。但由于训练时间不足采用了较高的学习率，因此很多时候训练 10 个 epoch 后 loss 降得很低而 perplexity 已经提高。而 perplexity 较低时并没有学习到逗号句号和诗句的特征，例如每句诗句长度一样，逗号后面跟句号等特征，导致生成古诗的时候效果非常糟糕。所以部分时候根据古诗生成效果判断手动停止程序。

2.生成

一开始学习给定的数据集'tangshi.txt'，但是该数据集过小，因此生成的诗句并不是很好。下面是根据给定的数据集'tangshi.txt'学习后生成的诗句，其中“湖”字第二个字符便为停止符没有生成。

‘日’：日月三千里，明公去一麾。可能休涕泪，岂独感恩知。草木穷秋后，山川落照时。如何望故国，驱马却迟迟。

‘红’：红烟雾死，将进酒，酒中有毒鸩主父，言之主父伤主母。母为妾地父妾天，仰天俯地不忍言。佯为僵踏主父前，主父不知加妾鞭。

‘山’：山林迹如踏铁，交河几蹴曾冰裂。五花散作云满身，万里方看汗流血。长安壮儿不敢骑，走过掣电倾城知。青丝络头为君老，何由却出横门道。

‘夜’：夜沈沈聊自适，放歌颇愁绝。岁暮百草零，疾风高冈裂。天衢阴峥嵘，客子中夜发。霜严衣带断，指直不得结。凌晨过骊山，御榻在嵯峨。

‘海’：海亭秋日望，委曲见江山。染翰聊题壁，倾壶一解颜。歌逢彭泽令，归赏故园间。予亦将琴史，栖迟共取闲。

‘月’：月三日天气新，长安水边多丽人。态浓意远淑且真，肌理细腻骨肉匀。绣罗衣裳照暮春，蹙金孔雀银麒麟。头上何所有，翠微盍叶垂鬓唇。

从网上下载了“全唐诗”数据集并使用了一部分诗句学习。下面是学习“全唐诗”数据集后生成的几首不同的诗句。

‘日’：日月高明月，春风如堕尘。我来一倾耳，万里天一隅。一闻鄮室中，苍生愧我身。

‘红’：红药就曾孙，西溪长击节。池草绿收弋，霜余春晓霞。山林十里旅，春风吹草霞。山林云水在，天上卿云雷。

石榴内消停起，一时不复辞。不辭援鼓桴，不是兼非真。

‘山’：山雨新葩酒，红日霁空秋。一笑相逢入，万里到桥边。

‘夜’：夜枕江海志，一笑相交似。一笑忽为人，一室自为人。。

‘湖’：湖外紫微，默默而神。一饭苍苍日，孤墳接平沙。一笑相逢入，天長鴻雁哀。

‘海’：海内思洪内，却憐悟舟驅。我来一倾耳，萬事一時心。一飽青天草，春風吹短檠。

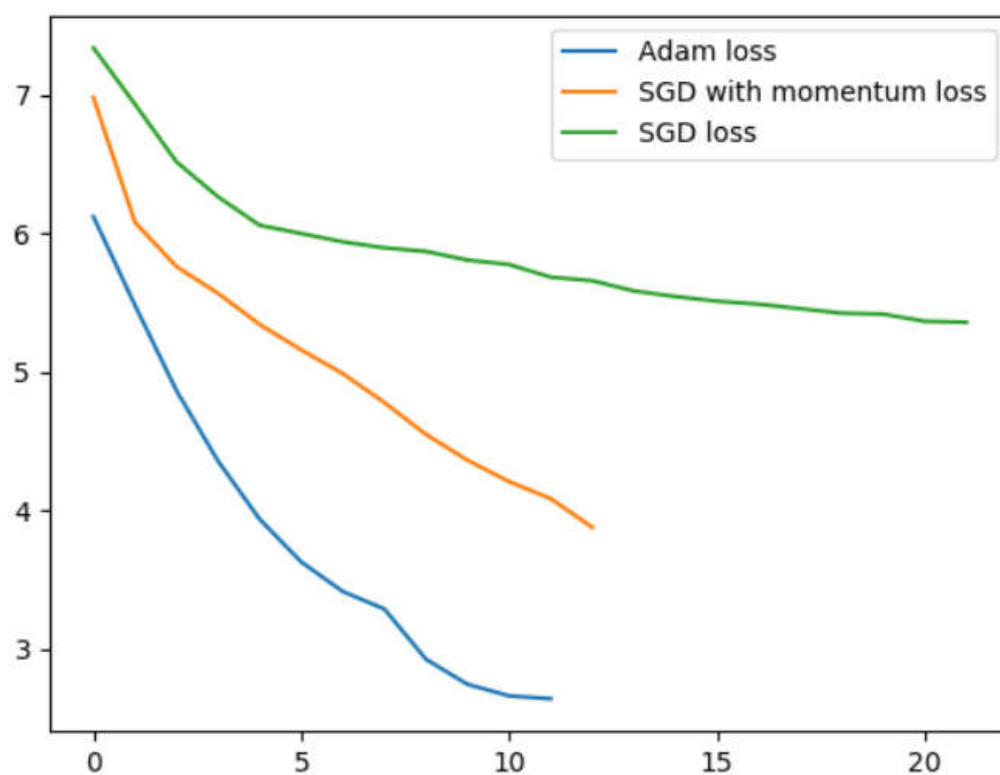
‘月’：月色初藏日，春風吹野服。我生嗜好，一笑伸眉。

3.最优化

本次实验一共使用了 SGD, SGD with momentum, Adam, 其中收敛最快的为 Adam。很多情况下 SGD 学习率采用 0.01, Adam 学习率采用 0.001, 但是由于机器性能问题这个学习率下降得太慢, 因此放大了学习率。尽管可能难以收敛, 但是能够确保初始 loss 下降得比较快。

一开始使用 SGD, 由于每次梯度下降时间较长, 当学习率较小的时候收敛时间非常长, 因此采用学习率为 0.5。SGD with momentum 学习率为 0.5, 动量为 0.9。使用 Adam 时学习率为 0.5 时无法收敛, 因此采用学习率 0.01。

SGD, SGD with momentum, Adam 的 loss 如下图



(Adam 和 SGD with momentum 因为早停所以只显示一部分)

上图可以很明显看出即使 Adam 学习率很小, 收敛仍明显快于 SGD with momentum 和 SGD。而在相同的学习率下 SGD with momentum 则快于 SGD。