基于LSTM的语言模型

人工智能1902宿希琳20195255

Abstract：我们可以通过三个门控-输入门，输出门，遗忘门来实现，相比于传统的RNN加入了门，其实是一种多层次的特征的选择方式。

1. 引言

1990年以来，循环神经网络（RNN）存在梯度爆炸和梯度消失等严重的问题使训练非常困难，应用十分受限。于是1997年人工智能研究所的主任Jurgen Schmidhuber 提出长短期记忆（LSTM），LSTM使用门控单元及记忆机制大大缓解了早期RNN训练的问题。于是构建了基于LSTM的语言模型。

1. 系统设计

LSTM是研究人员针对RNN记忆不足的问题改进而来，让网络能够自己决定是否记忆当前的信息，尤其是在长文本的任务上面表现更优。、那么，如何来实现基于LSTM的语言模型呢？

我们可以通过三个门控-输入门，输出门，遗忘门来实现，加入了门，其实是一种多层次的特征的选择方式。

输入门：控制输入给memory cell的值的大小打开意味着保留输入，关闭意味着丢弃输入。从完全打开到完全关闭对应值的变化从1到0。相当于给输入乘上一个权重矩阵，这个权重矩阵可以控制输入矩阵里每一个值的大小由网络本身根据输入门控制信号自动生成权重矩阵

输出门：控制LSTM cell输出到外面的值的大小，打开意味着保留输出，关闭意味着丢弃输出相当于一个每个值都是0~1之间的权重矩阵同样是网络自己通过输出门控制信号生成权重矩阵。

遗忘门：控制需要保留多少以前的记忆打开意味着保留记忆，关闭意味着遗忘记忆。还是一个每个值都是0~1之间的权重矩阵。权重值越小，遗忘得越多。

我们可以通过Sigmoid函数来挤压信号使信号在0-1之间达到生成合适的权重的效果，权重为0说明门被关闭，权重为1说明门被完全打开，实现控制的效果。每个门的控制信号都是根据下面这俩计算得到的

1.上一时刻隐藏层状态（即上一时刻cell的输出）

2.本时刻的输入

每个门的控制信号计算过程都是相同的：转换与相加

LSTM内部主要有三个阶段：

1. 忘记阶段。这个阶段主要是对上一个节点传进来的输入进行选择性忘记。简单来说就是会 “忘记不重要的，记住重要的”。

具体来说是通过计算得到的  （f表示forget）来作为忘记门控，来控制上一个状态的  哪些需要留哪些需要忘。

2. 选择记忆阶段。这个阶段将这个阶段的输入有选择性地进行“记忆”。主要是会对输入  进行选择记忆。哪些重要则着重记录下来，哪些不重要，则少记一些。当前的输入内容由前面计算得到的  表示。而选择的门控信号则是由  （i代表information）来进行控制。

将上面两步得到的结果相加，即可得到传输给下一个状态的  。也就是上图中的第一个公式。

3. 输出阶段。这个阶段将决定哪些将会被当成当前状态的输出。主要是通过  来进行控制的。并且还对上一阶段得到的  进行了放缩（通过一个tanh激活函数进行变化）。

与普通RNN类似，输出  往往最终也是通过  变化得到。

LSTM内部主要有三个阶段：

1. 忘记阶段。这个阶段主要是对上一个节点传进来的输入进行选择性忘记。简单来说就是会 “忘记不重要的，记住重要的”。

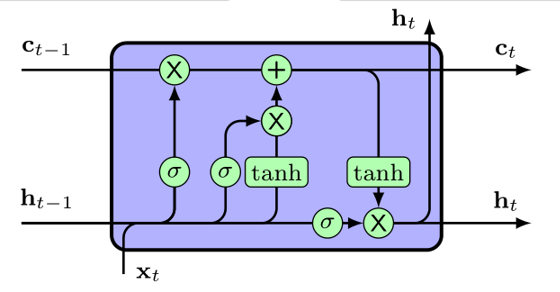
具体来说是通过计算得到的  Z^f（f表示forget）来作为忘记门控，来控制上一个状态的 ct-1哪些需要留哪些需要忘。

2. 选择记忆阶段。这个阶段将这个阶段的输入有选择性地进行“记忆”。主要是会对输入  进行选择记忆。哪些重要则着重记录下来，哪些不重要，则少记一些。

将上面两步得到的结果相加，即可得到传输给下一个状态的 ct.

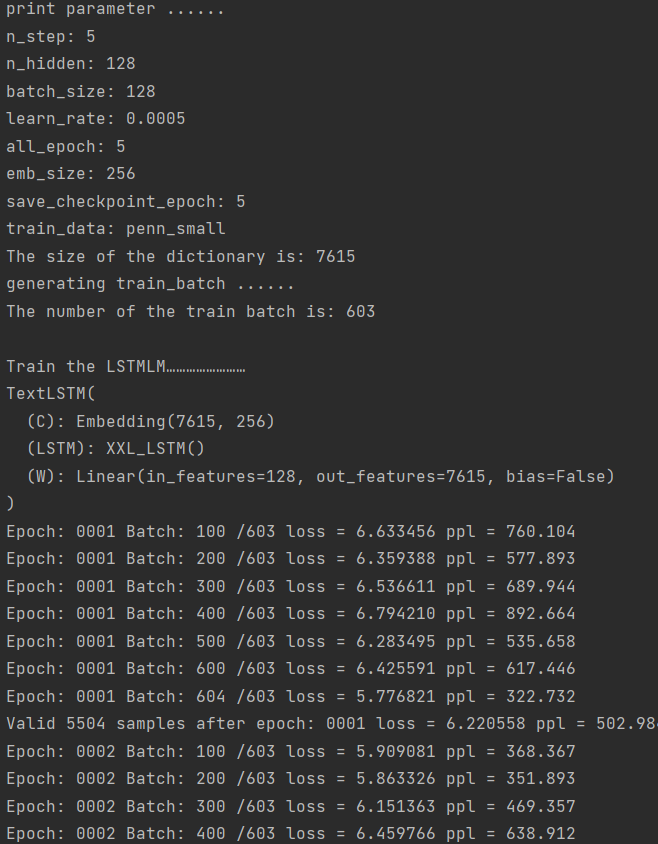
3. 输出阶段。这个阶段将决定哪些将会被当成当前状态的输出。主要是通过 z0来进行控制的。并且还对上一阶段得到的z0 进行了放缩（通过一个tanh激活函数进行变化）。

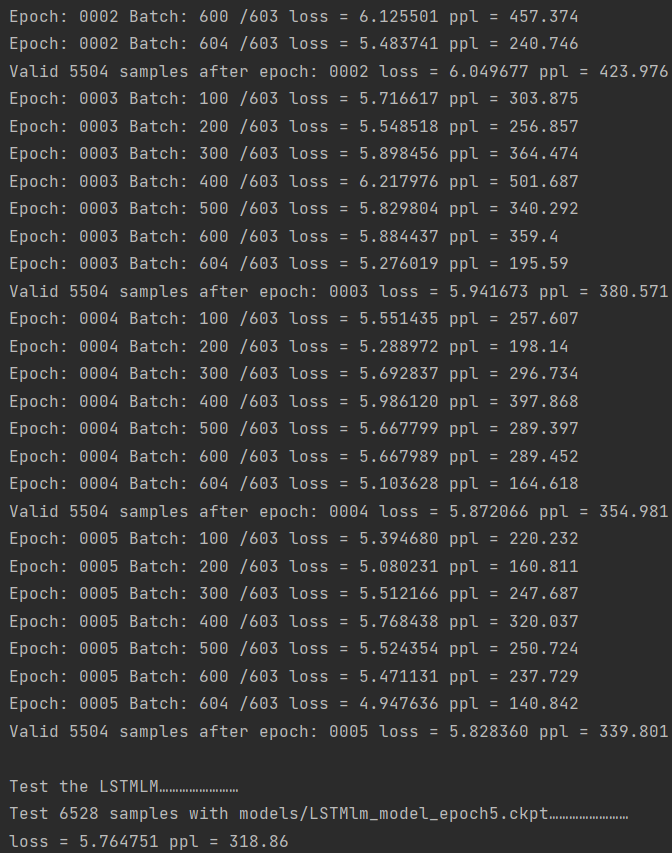
各门控组件之间以如下格式表示



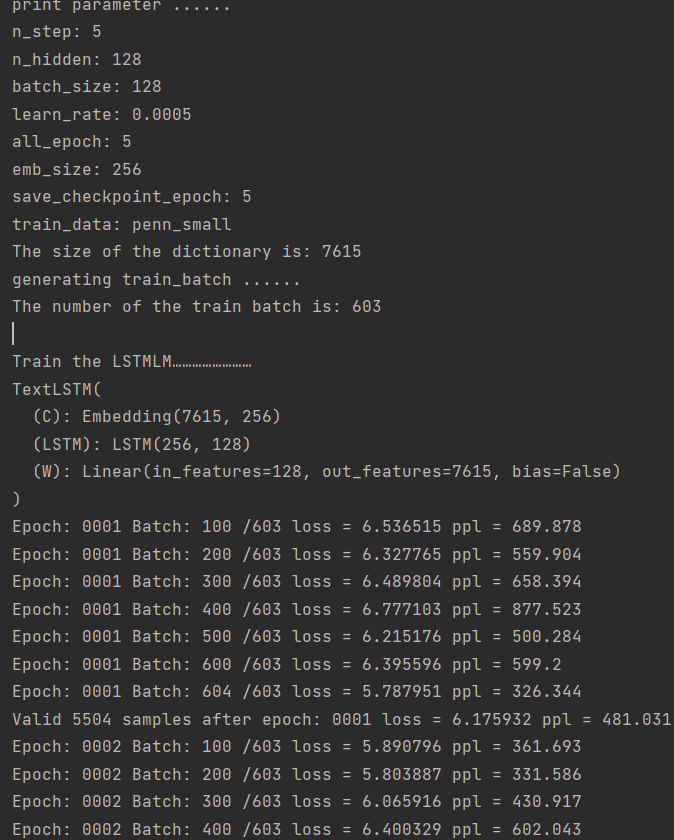
1. 实验数据

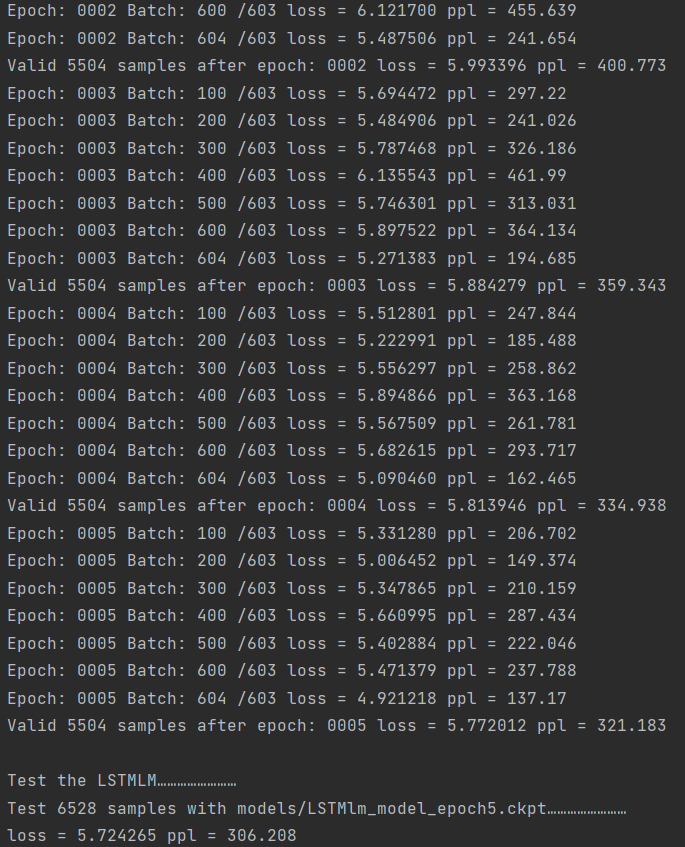
以下的数据为自己写的基于LSTM的语言模型的实验结果





以下的数据为pytorch的LSTM的语言模型的实验结果





1. 分析

通过相同的训练集以及测试集，发现pytorch的LSTM模型运行速度更快，损失也比自己写的基于LSTM的语言模型小，并且ppl也小于自己写的。

五．结论

通过实验结果来看，自己写的LSTM还有很多的改进空间。

1. 致谢

感谢肖桐老师及其研究生团队的教学和帮助。