自然语言处理实践报告

本学期学习了自然语言处理课程，老师从自然语言处理基础、人工神经网络基础、单词和单词向量、序列及语言模型和前沿模型介绍五个方面向我们介绍了自然语言处理发展历程、理论框架和前沿知识。同时开展了许多实践课程，让我们更加直观的理解课程上学习到的理论知识。

课程总结

在自然语言处理基础章节，老师通过什么是语言以及什么是自然语言处理两个问题向我们介绍了自然语言的特点和自然语言处理的基本任务，并简述了当前自然语言处理技术从统计学习到深度学习的转变。

在人工神经网络基础章节，老师向我们介绍了神经网络的原理与结构以及如何训练一个神经网络并简单讲述了人工神经网络的自然语言处理中的应用，即语言模型的神经网络。

在单词与单词向量章节，我们主要学习了如何通过计算机将单词表示出来。第一，我们学习了基于单词分布的表示：在这一小节我们学习了基于离散事件表示表示方法的独热表示，但由于不同的单词之间也有着一定的相关性，因此引出了基于上下文语境的单词表示，即基于共现矩阵的方法，并具体介绍了了单词-单词矩阵与单词-文本矩阵。第二，我们学习了基于单词分布式的表示，即单词嵌入或词嵌入，由于基于单词分布的表示并不能很好的表示词义或属性，因此基于单词分布式的表示有着更好的效果。老师也向我们介绍了那么如何训练得到一个词嵌入：n-gram、前馈神经网络、Word2Vec、CBOW、Skip-gram、FNNLM、GloVe。第三，我们学习了如何词嵌入的评价方法，内部任务评价方法、外部任务评价方法。

对于大部分自然语言任务来说，只使用单词是不好解决或者根本无法解决的，无法很好的利用单词的上下文。在序列与语言模型这一章节，我们将语言单位更进一级，使用由单词组成的序列来解决任务。首先，我们学习了基于循环神经网络的序列表示。RNN 模型可以较好的解决上述问题，但随着序列越来越长，RNN 的记忆信息的损失也越来越严重。为了解决找个问题，引出了长短时记忆模型，即 LSTM 模型。之后我们学习了 GRU 模型，并将 RNN 模型拓展为双节结构和多层结构。最后，我们学习了基于卷积神经网络网络的序列表示。

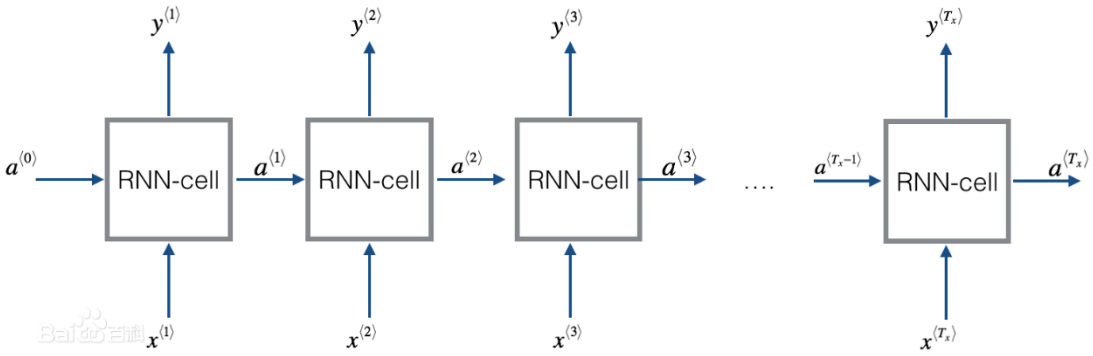
在最后几节课，老师向我们介绍了当前自然语言处理领域的前沿知识-自注意力机制、Transformer。

实践内容

本次课程实践内容为手动实现长短时记忆模型，即 LSTM 模型，助教学长向我们提供了使用 Pytorch 框架实现的 LSTM 处理文本全流程。我们只需将其中的 nn.LSTM()函数转换为手动实现的函数即可。

RNN

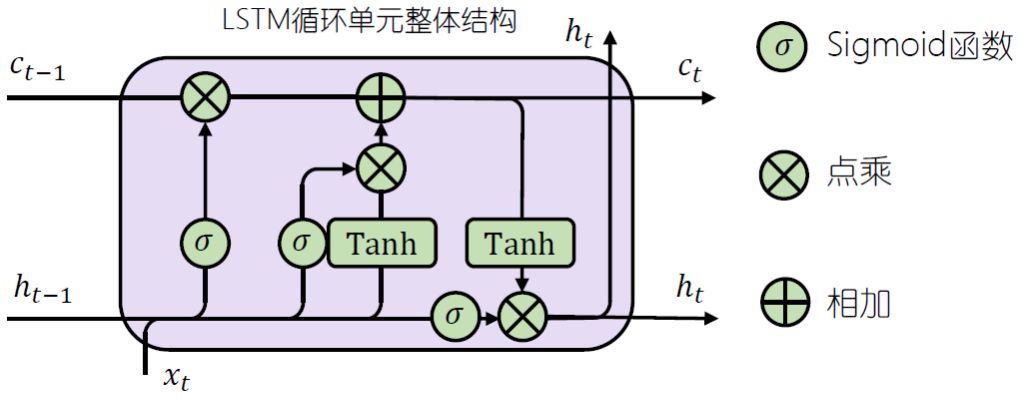
在自然语言中，前一个或多个单词对后续的单词有着一定的影响，所以为了解决这样的问题，更好的处理序列信息，RNN 模型诞生了。RNN 模型的核心原理为：设置隐藏状态（可以理解为记忆）每一次的输出不仅取决于当前的输出还取决于前一个隐藏状态，并更新隐藏状态。RNN 模型结构如图 1 所示



图表 1 RNN 模型结构

LSTM

RNN 结构在理论上可以编码无限长序列，但是随之也会产生一些问题：随着序列越来越长，在反向传播时循环神经网络会产生更多的局部梯度相乘计算，这会导致梯度消失或梯度爆炸问题，即 RNN 记忆信息的损失也越来越严重。为了解决找个问题，研究者提出了长短时记忆（ Long Short term Memory ）模型，即 LSTM 模型。LSTM 模型的创新主要体现在循环单元的设计上，引入了遗忘门、输入门、输出门来动态的选择遗忘和记忆多少之前的信息。同时，我们可以叠加多层的 LSTM 网络，将低层的输出作为高层的输入，以得到最终的结果，去解决更加复杂的问题。LSTM 循环单元整体结构如下图 2 所示

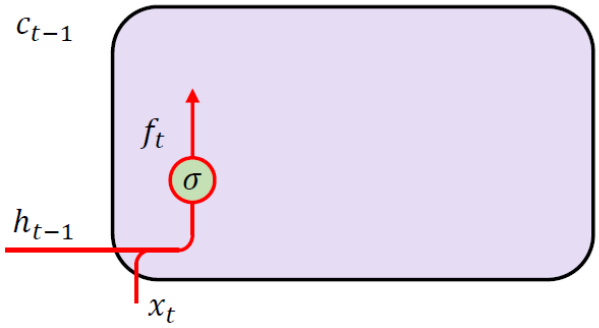


图表 2 LSTM 循环单元整体结构

下面是我在该次实践中，对门控单元的设计，在本次实践中，我参考了 Pytorch 框架内 LSTM 门控单元的设计和老师讲解内容

遗忘门

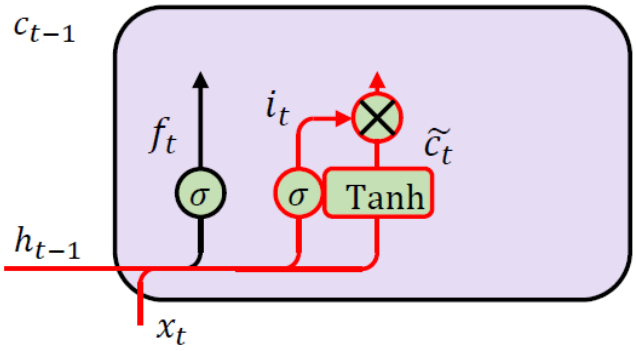
遗忘门结构如下图 3 所示



图表 3 遗忘门结构

输入门

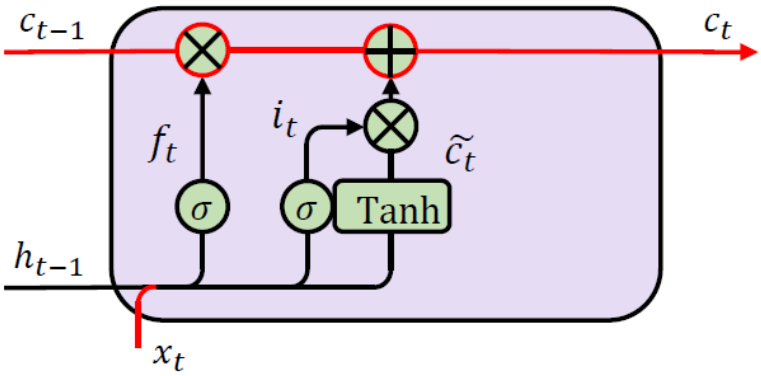
输出门结构如下图 4 所示



图表 4 输出门结构

记忆更新

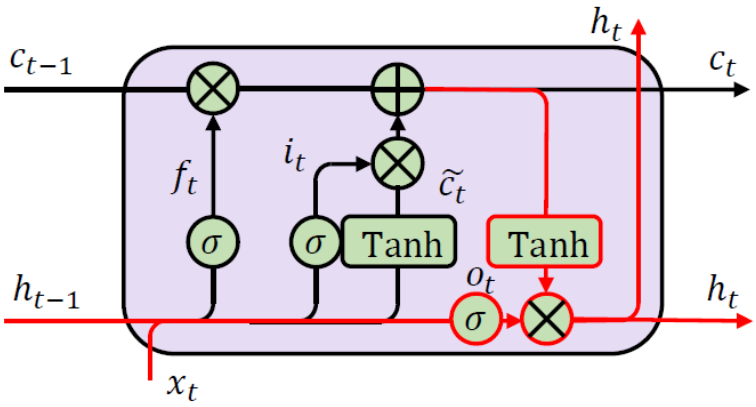
记忆更新部分如下图 5 所示



图表 5 记忆更新结构

输出门

输出门结构如下图 6 所示



图表 6 输出门结构

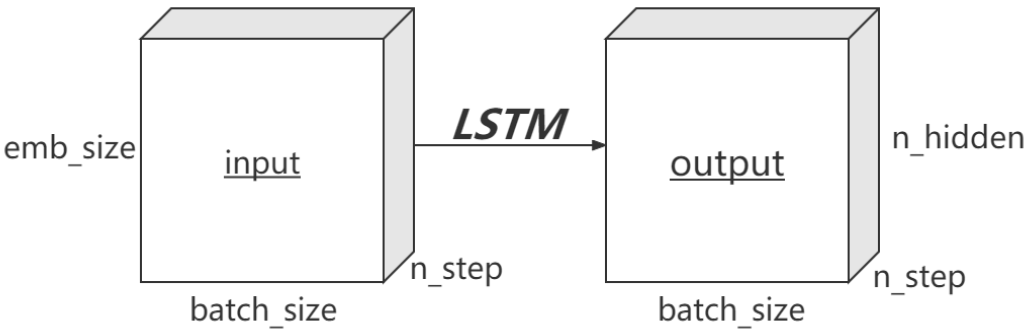
将上述的的门控单元转换为数学公式，如下图 7 所示

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

图表 7 LSTM 模型公式

问题与解决方法

在本次实践中主要的问题产生在在参数维度的计算上，经过对程序部分的仔细研究我总结出了 LSTM 模型的输入与输出，如下图 8 所示。



图表 8 LSTM 模型输入与输出维度示意图

心得体会

在经过这次实践活动后，我对自然语言处理所需要解决的问题与任务有了较为清晰的认识，同时也对上课中所学到的理论知识有了更加直观的理解与认识。在技术层面，我不仅对 Pytorch 框架有了一定程度上的掌握，同时也学会了 github 以及 markdown 文本的基本操作，希望日后还能参加一些这样的实践内容。

参考资料

1. VuePress (nlplab.com)
2. 人人都能看懂的 LSTM - 知乎 (zhihu.com)
3. 一文搞懂 RNN（循环神经网络）基础篇 - 知乎 (zhihu.com)