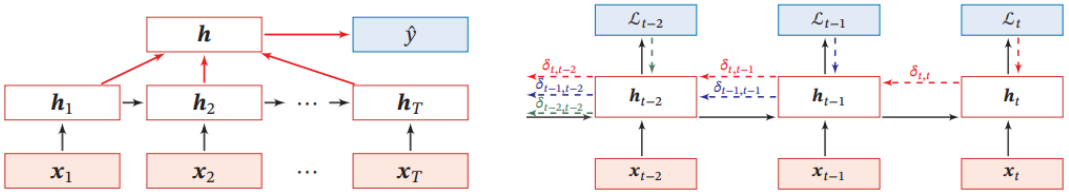
自然语言处理 Project1 实验报告

李菲菲 18307110500 2022/11/20

I. 背景介绍

词向量：代码中主要使用预训练的词向量 GloVe（向量维度 100）。GloVe 是相比 Word2Vec 的一种改进，仍然是一种静态的词向量嵌入。向量化的主要方法是训练词向量，以拟合统计的共现概率比值分布P_{ik}/P_{jk}，经过变形和近似推导后得到优化目标为$J=\sum_{i,k=1}^{V} f(X_{ik})\left(w_i^T\cdot\widetilde{w_k}+b_i+\tilde{b}_k-\log(X_{ik})\right)^2$，即带权重的平方差的形式。

循环神经网络：RNN (Recurrent Neural Network)，是一类具有短期记忆能力的网络。RNN 一般用于处理时序数据相关的问题。时序数据的主要特点有:(1)输入之间非独立：当前时刻的输出不仅和当前时刻的输入有关，还与历史的输入有关。(2)长度不固定：例如文本，不同的句子可以有不同个数的单词。因此，RNN 除了像前馈神经网络那样，接受来自其他神经元的信号，还会接收来自过去的自身的信号——将这一类信号建模为“隐变量”。学习参数时损失也需要沿着时间反向传播到 t0，即 BPTT(Backward Propagation Through Time).对于项目的文本分类任务，如 1 所示，可以使用隐藏状态 h 的序列，结合 attention 机制计算注意力，然后经过线性层输出分类结果。



(b) 按时间进行平均采样模式　　图 6.6　误差项随时间反向传播算法示例

1 左图为 RNN 用于时序数据分类示意图；有图为 BPTT 示意图。参考图来自 https://nndl.github.io/

BPTT 算法：时刻 t，损失 L 关于参数 U 的求导可以写为$\partial L_t/\partial U$；参数 U 与隐藏层在时刻 k 接收到的净输入$z_k=Uh_{k-1}+Wx_k+b$有关，于是有：

$$\frac{\partial L_t}{\partial u_{ij}}=\sum_{k=1}^{t}\frac{\partial^+ z_k}{\partial u_{ij}}\frac{\partial L_t}{\partial z_k},\text{其中：}\ \delta_{t,k}=\frac{\partial L_t}{\partial z_k}=\frac{\partial L_t}{\partial z_{k+1}}\frac{\partial z_{k+1}}{\partial h_k}\frac{\partial h_k}{\partial z_k}=diag\left(f'(z_k)\right)U^T\delta_{t,k+1};$$

$$\frac{\partial^+ z_k}{\partial u_{ij}}=[0,\dots,h_{k-1},\dots 0],\text{于是有：}\ \frac{\partial L_t}{\partial u_{ij}}=\sum_{k=1}^{t}\left[\delta_{t,k}\right]_i[h_{k-1}]_j.\text{最后写为矩阵形式：}\ \frac{\partial L_t}{\partial U}=\sum_{k=1}^{t}\delta_{t,k}h_{k-1}^T.$$

长短期记忆网络：LSTM，是 RNN 的一种变体. BPTT 对于过长的序列，由于 t 很大，隐藏状态不断传递，导致梯度消失或梯度爆炸的问题。除了神经元的隐藏状态$h_1,\dots,h_t$，LSTM 额外引入了新的**内部状态**建模隐藏状态的传递过程，存储历史信息，记为

$c_1, \ldots, c_t$，使用三个门决定下一时刻隐藏状态和内部状态的输出h_{t+1}, c_{t+1}：

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), \qquad c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \qquad h_t = o_t \odot \tanh(c_t).$$

其中：$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$为遗忘门，控制过去的内部状态$c_{t-1}$被遗忘的信息，

$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$为记忆门，决定候选状态$\tilde{c}_t$需要被记忆的信息

$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$ 为输出门，决定内部状态输出为隐藏状态h_t的信息.

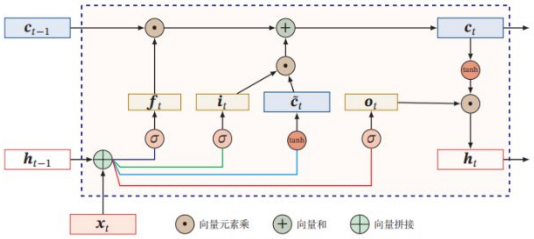


图 6.7 LSTM 网络的循环单元结构

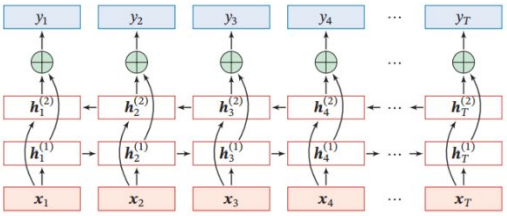


图 6.10 按时间展开的双向循环神经网络

双向长短时记忆网络：Bi-LSTM 包含两个消息传播方向不同的隐藏层，因此隐藏状态数会加倍。

注意力机制：对于输入 X，应用三个线性变换得到三个不同特征空间的表征Q, K, V，表示查询、键和值，其中QK^T求相似度矩阵，放缩和归一化后对V加权求和，得到注意力图。可以对于 Bi-LSTM 输出的结果（输出$\hat{y}$或隐变量h）作为 X 进行自注意力计算，得到关注全局信息的输出。

$$SetAttention(X) = Attention(Q, K, V), where$$
$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, Q = W^Q X, K = W^K X, V = W^V X.$$

Transformer：包括 encoder-decoder 架构。Encoder 包括对输入词嵌入、位置编码、多头注意力、和前馈网络，decoder 结构类似，也是由多头注意力模块和前馈网络组成。多头注意力机制即多个上述自注意力模块，自注意力模块输出需要经过线性层组成的前馈网络。文本分类任务可以使用其中的 encoder 进行特征学习，后接分类器进行文本分类。

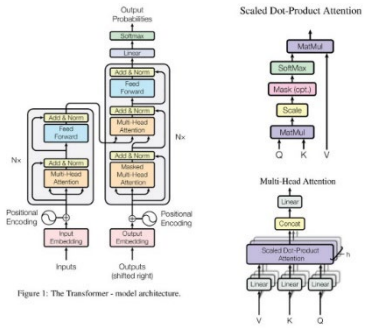


Figure 1: The Transformer - model architecture.

Adam 优化器：自适应调整学习率达到快速收敛的一种方法。通过计算梯度的（近似）一阶矩和二阶矩，经过修正后，对更新的步长进行动态的调整，具体如下：

一阶矩：$M_t = \beta_1 M_{t-1} + (1-\beta_1) g_t, M_0 = 0$；二阶矩：$G_t = \beta_2 G_{t-1} + (1-\beta_2) g_t \odot g_t, G_0 = 0$；修正项：$\hat{M}_t = \frac{M_t}{1-\beta_1^t}, \hat{G}_t = \frac{G_t}{1-\beta_2^t}$；更新公式：$\theta_t$：

$$= \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{G}_t + \epsilon}} \cdot \hat{M}_t$$

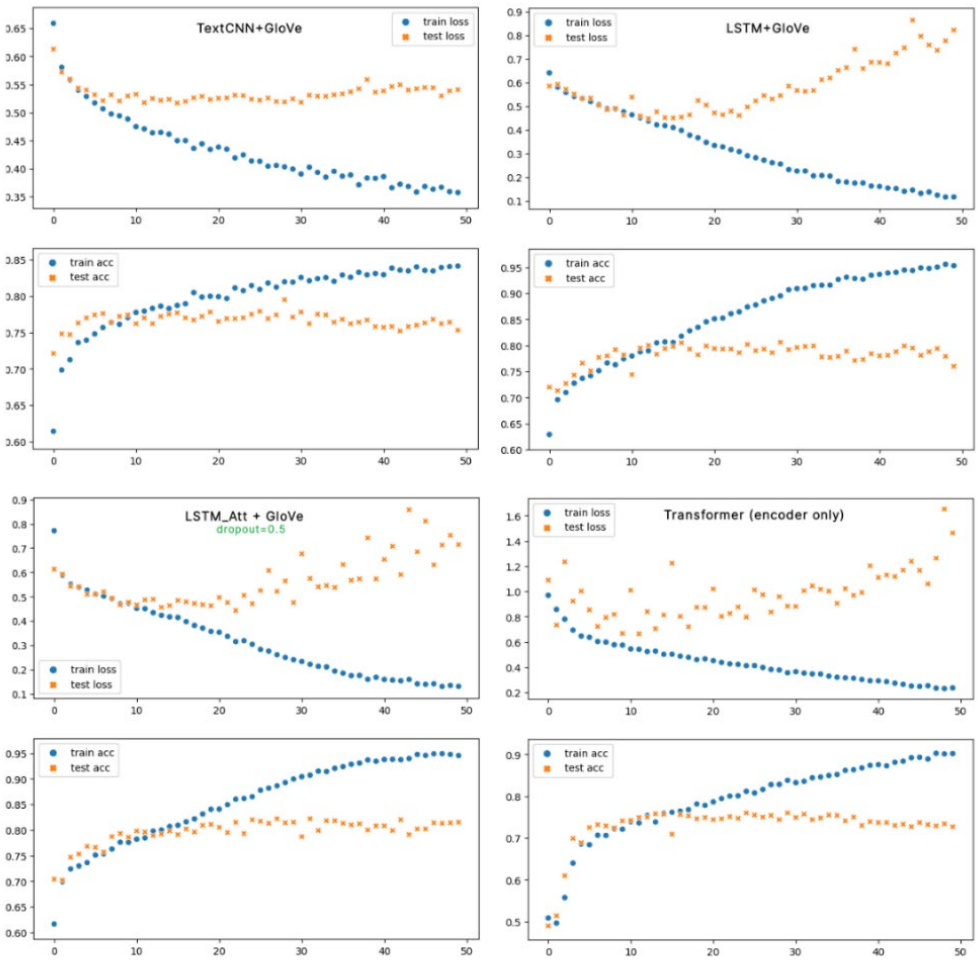
II. 实验内容

实验设置：数据集使用 fudan_nlp-movie_review 数据集(样本个数训练集 8596，验证集 1000，测试集 1066),标签为 positive 和 negative。词向量模型为 GloVe100.优化器使用 Adam，学习率 1e-3，权重衰减系数 1e-4.以下的实验默认使用 dropout=0.5，训练 epoch 为 50，随机数种子为 42.

不同架构实验结果：TextCNN 参考自 2014 年论文[1]Kim_CNN,；Bi-LSTM+Attention 参考自[3]，使用 LSTM 的隐状态经过 attention 后进行分类；Transformer-encoder 的实现参考论文[4]。最后提交的测试输出来自 BI-LSTM+ATT(DP=.7)。

架构	GLOVE100	注意力	(BEST)EVAL ACC	(BEST)TRAIN ACC	(BEST)TRAIN LOSS
TEXTCNN[1][2]	√		0.795	0.84121	0.35747
BI-LSTM	√		0.806	0.95556	0.11634
BI-LSTM+ATT[3]	√	√	0.815	0.93741	0.15921
BI-LSTM+ATT(DP=.7)	√	√	0.824	0.80991	0.40891
TRANSFORMER_ENC[4]	√	--	0.760	0.90251	0.23127

预处理操作：仅使用空格分词会得到比较多的 out of vocabulary 词，因此通过观察，进一步对单双引号、连字符、标点等进行了替换，将 oov 单词的数量从 2185 减少到了 736。



III. 总结

1. TextCNN 模型在 IMDB 数据集（训练集大小 25000，测试集大小 25000）上，容易得到 85%以上的测试准确率（5 个 epoch 之内）；而在给定的 nlp 数据集上（训练集大小 8500，验证集 1000）最多只能达到接近 80%的准确率。因此简单的模型需要更多的数据才能够达到更高的准确率。
2. 观察到明显的过拟合现象：在 dropout rate=0.5 的时候，模型很容易达到过拟合，即训练集上准确率接近 100%而验证集上准确率不到 80%。为了缓解过拟合，选择增加 dropout 大小为 0.7，理论上能够缓解过拟合；然而实际上应该是增加了方差，在迭代 80 次的实验中，更大的 dropout 难以给出真正的最优点。
3. 纯 Attention 模块的 Transformer 在小训练集上表现不如 LSTM；但是在 LSTM 的输出之后引入 Attention 有助于放大值得关注的部分，使得线性层分类器能得到更好的分类结果，更快的收敛速度。

IV. 参考

[1] Yoon Kim. “Convolutional Neural Networks for Sentence Classification.” Conference on Empirical Methods in Natural Language Processing (2014).

[2] kim_cnn/model.py at master • Impavidity/kim_cnn (github.com)

[3] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[4] Vaswani, Ashish, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. “Attention is All you Need.” ArXiv abs/1706.03762 (2017): n. pag.

[5] guocheng2018/Transformer-Encoder: Implementation of Transformer encoder in PyTorch (github.com)