文本表征学习 Lab 4 Transformer

PB22111599 杨映川

Pt.1 实验概述

使用IMDB情感分析数据集和Transformer模型训练句子向量并对其性能进行评估，对比分析不同设定的性能。使用配置如下：

- **向量维度设置为100，其它参数自定**
- **向量维度设置为200，其它参数与上个模型一致**

Pt.2 模型训练

2.1 参数选择

本次实验本人主要使用了`transformers`库和`torch.nn`库进行实现。为了能和要求的100维、200维向量适配，对`nhead`（注意力头个数，默认为8）、`dim_feedforward`（默认为2048）和`dropout`（默认为0.1）进行了调整。Transformer的具体参数如下：

参数名称	**取值**
d_model	100 or 200
nhead	10
num_encoder_layers	6
num_decoder_layers	6
dim_feedforward	800
dropout	0.5*
epochs	20

*关于dropout取值的选择：本人在具体运行代码时发现**dropout参数能比较显著地影响模型的性能**，过低或过高的dropout值都会导致模型过早收敛导致预测效果非常差（指准确率徘徊在0.5上下，与未训练的随机匹配无差别）。本人先后在100维的模型上分别测试了dropout=0.1，0.5，0.8三种情形，发现当**dropout=0.5**时模型能较好的进行学习。故dropout参数统一采用0.5。

2.2 数据预处理

首先将IMDB数据集加载出来，然后使用BertTokenizer进行处理，然后将其映射成pytorch能够理解的张量格式。

2.3 模型训练

以32条数据为1个batch（**batch_size=32**）进行训练，输出层使用平均池化获取固定大小的特征进行线性变换。训练完成后使用了Adam算法进行优化，因为Adam算法具有较好的可适应性；loss的测算选择交叉熵函

数，因为交叉熵的大小表示两个概率分布之间的差异，可以通过最小化交叉熵来得到目标概率分布的近似分布。

2.4 模型评估

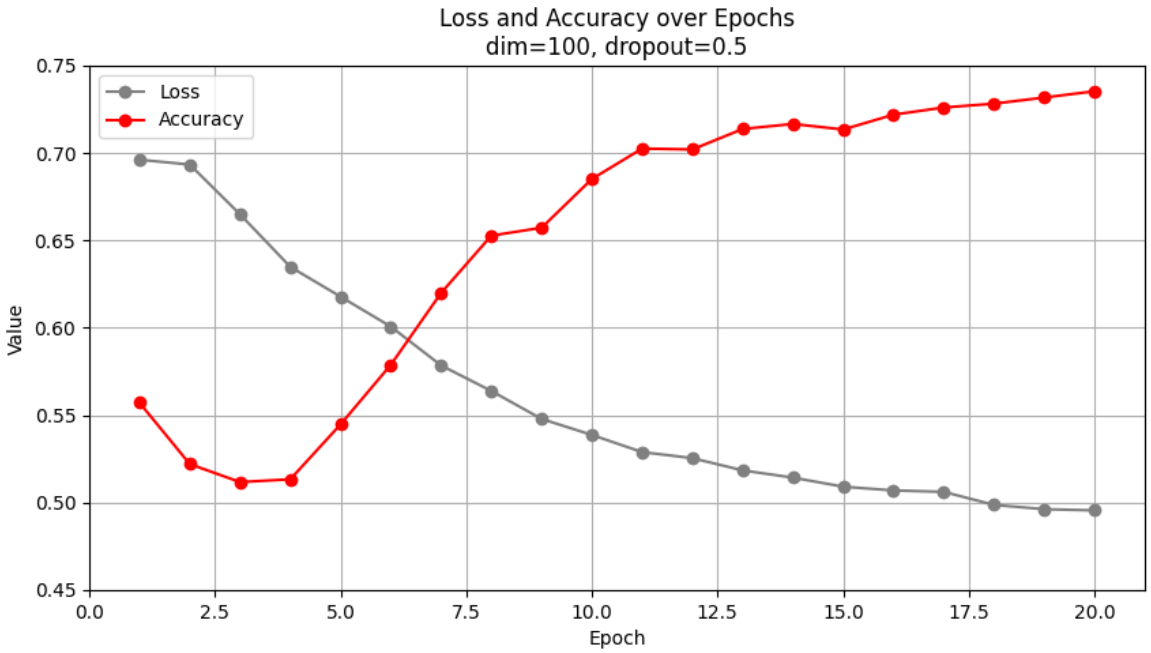
建立函数提取输出结果的特征，使用训练集训练好逻辑回归模型后，在测试集上基于存有特征信息的向量进行逻辑回归二分类，通过比对预测值和实际值计算准确率。使用的逻辑回归模型来自`sklearn.linear_model.LogisticRegression`并使用了默认参数。

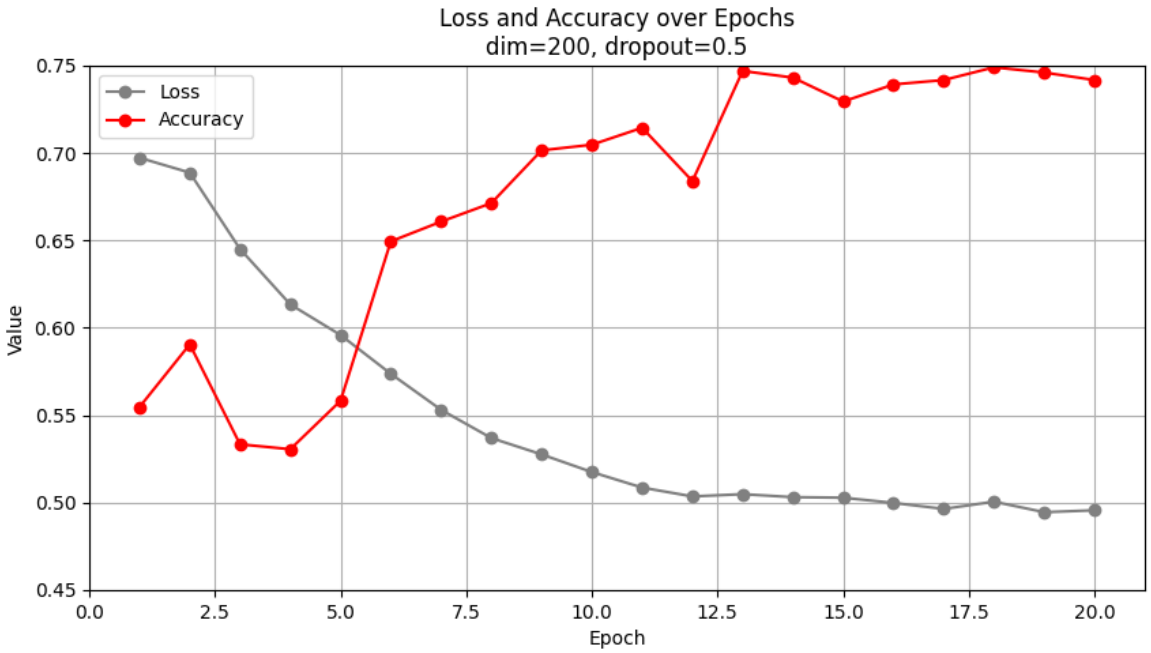
Pt.3 训练结果

100维的模型在大约第19次迭代时**loss值**逐渐收敛趋于稳定，200维的模型则约在第12次**loss值**逐渐收敛趋于稳定。模型的分类准确率分别取开始趋向稳定时的识别准确率。

向量维度	**分类准确率(%)**
dim=100	73.54
dim=200	74.70

以下为两个模型训练过程的具体变化





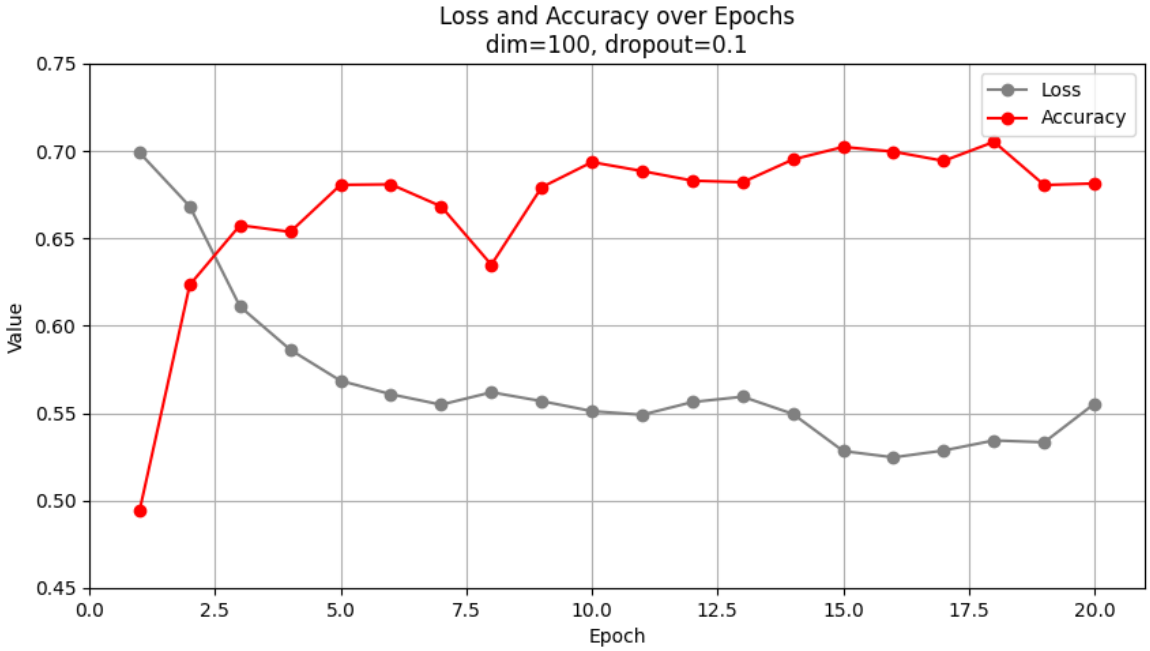
Pt.4 结果分析

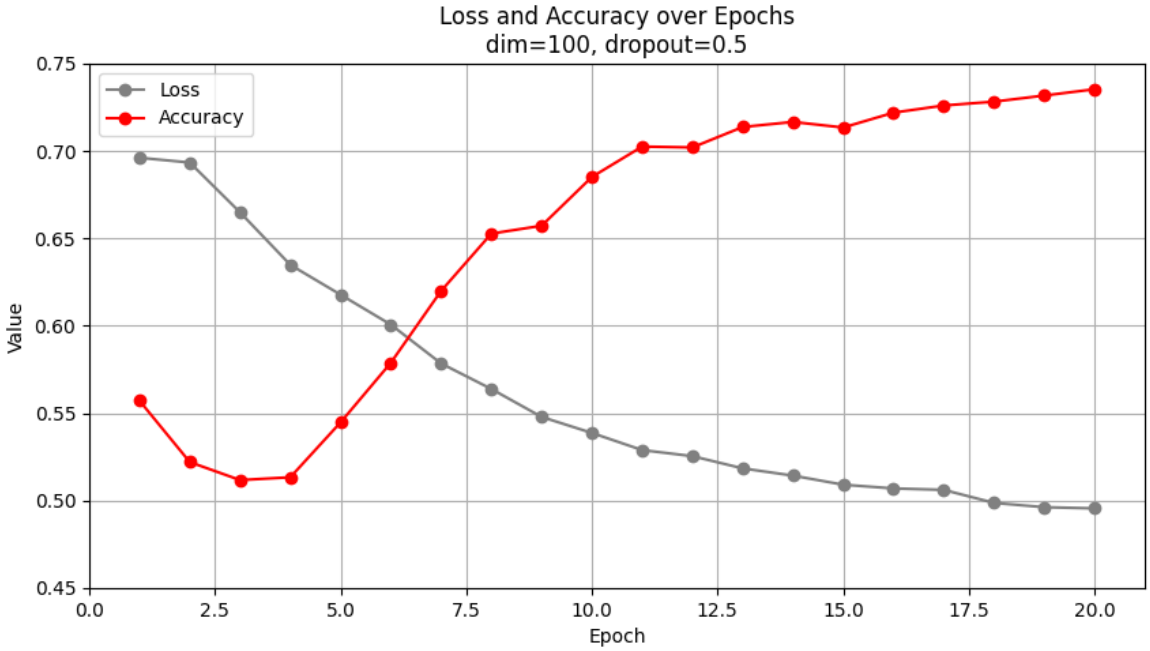
1. 200维模型的性能在较小的程度上优于100维模型，推测向量维度这一变量对识别准确率的影响不是很大；
2. 100维模型的loss收敛要出现的比200维模型更晚也更平缓，推测这可能与所选择的Adam优化算法有关；
3. 可以看到200维模型在训练后期，当loss继续减小时识别准确率反而出现了回降，推测此时模型过拟合；
4. 此外，其它参数对模型的影响也是较为显著的，比如dropout和dim_feedforward。

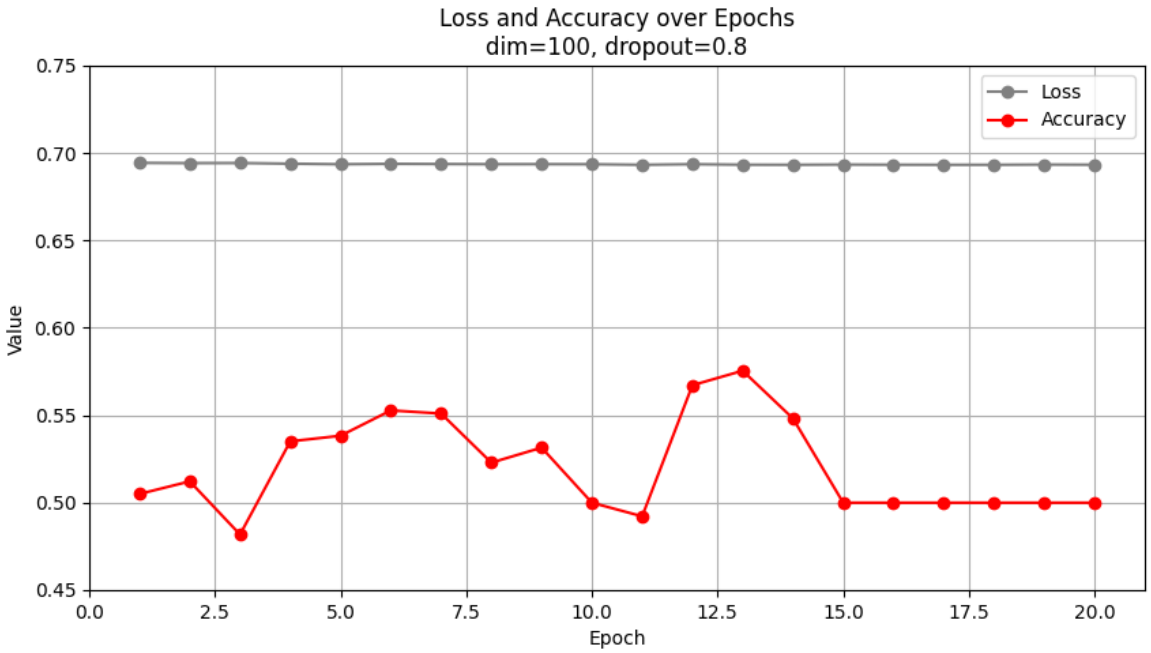
Pt.5 一些尝试

- 前文提到dropout参数对模型影响较为显著。合适的dropout可以减轻模型训练对数据的过拟合，提高模型的泛化性能。在参数与之前保持一致、向量维度选择100的情况本人对dropout的不同取值的模型性能进行了评估：

> - dropout=0.1
> - dropout=0.5
> - dropout=0.8


Loss and Accuracy over Epochs
dim=100, dropout=0.1
Loss
Accuracy
Value
Epoch
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Loss and Accuracy over Epochs
dim=100, dropout=0.5
Loss
Accuracy
Value
Epoch
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



> 可见，太小的dropout会导致模型过早收敛并过拟合，使模型效果较差；太大的dropout舍弃了过多的数据特征，导致模型无法进行正常训练学习。

Pt.6 Debugging Problems

1. 由于本人电脑性能有限，较大的batch_size无法被分配，且训练非常耗时——100维的模型在GPU上跑一次迭代约消耗20分钟，在CPU上跑一次迭代约消耗140分钟。故借助了**Kaggle**平台的云GPU，将一次迭代消耗时间缩短到了5分钟以内；
2. Fine-tuning is a painful task...