基于 BERT 预训练模型的情感分析

PB21511850 吴玄
PB22050937 洪柯昕

Contents

1 前言 **3**

2 主要工作 **3**

3 实验设定和结果 **3**

3.1 实验设定 3

3.1.1 IMDB 数据集 3

3.1.2 情感分类器 3

3.1.3 参数设定 3

3.2 实验结果 4

3.2.1 训练过程及测试结果 4

3.2.2 错误个例 5

4 分析讨论 **5**

4.1 实验分析 5

4.1.1 训练过程 5

4.1.2 测试结果 6

4.2 困难及解决办法 6

5 结论 **6**

6 组员分工及贡献情况说明 **7**

参考文献 **7**

1 前言

BERT[1] 是一种基于 Transformer 架构的预训练模型。由于 BERT 的双向注意力机制，它能够充分发掘 token 与上下文之间的关系，因此 BERT 在自然语言理解方面相比之前的模型有着显著优势。在实际应用中，BERT 常被用于文本分类、实体识别、问答系统等自然语言相关的任务。

我们的工作基于 BERT 展开，旨在研究 3 种 BERT 预训练模型[1] 在情感分类数据集上的性能，并将它们与微调之后的模型进行对比。

2 主要工作

我们的情感分类任务在“IMDB 情感分类数据集”[2] 上进行（下文简称为“IMDB”），选取 bert-base-uncased、bert-large-uncased、bert-base-multilingual-uncased 三个预训练模型进行实验（下文分别将其简称为“BASE”、“LARGE”、“MULTI”）。对于每个预训练模型，我们先冻结 BERT 的预训练参数，使用训练集训练出情感分类器，并在测试集上测试准确率和损失函数值。然后再使用模型中的所有参数参与训练，重复上一步操作。

3 实验设定和结果

3.1 实验设定

3.1.1 IMDB 数据集

IMDB 包括训练集、测试集、非监督数据三个部分，其中训练集和测试集各包括 25000 个文本段落与相应的情感标签，“0”、“1”分别代表负面情感和正面情感。非监督数据仅含文本，不包含标签。我们的实验使用训练集训练，在测试集上测试，未使用非监督数据。

3.1.2 情感分类器

我们构建的情感分类器包括两个部分，“BERT”和“分类器”。其中“BERT”即为预训练的 BERT 模型，“分类器”包括池化层、线性层 Layer1、ReLU、线性层 Layer2，它接收 BERT 输出结果中的“last_hidden_state”, 输出预测结果。

3.1.3 参数设定

实验中使用的主要参数如表1所示。

表 1: BERT 模型训练参数

参数	值
Learning Rate	1e-5
Batch Size	32
Number of Epochs	3
Optimizer	AdamW
Loss Function	CrossEntropyLoss
Dropout Rate	0.1
Layer1 output_size	256
Layer2 output_size	2

3.2 实验结果

3.2.1 训练过程及测试结果

对于每个预训练模型，我们训练的分类模型包括“PRETRAINED”和“FINETUNED”两种，前者表示冻结 BERT 部分的预训练参数，后者表示不冻结。图1和图2 呈现了各个模型在训练过程中损失函数的变化情况。为了让损失函数曲线更加清晰易分辨，我们将窗口为 50 个 batch 内的函数值进行了平滑。表2 和表3分别呈现了训练后各个分类模型在测试集上的准确率和损失函数值。

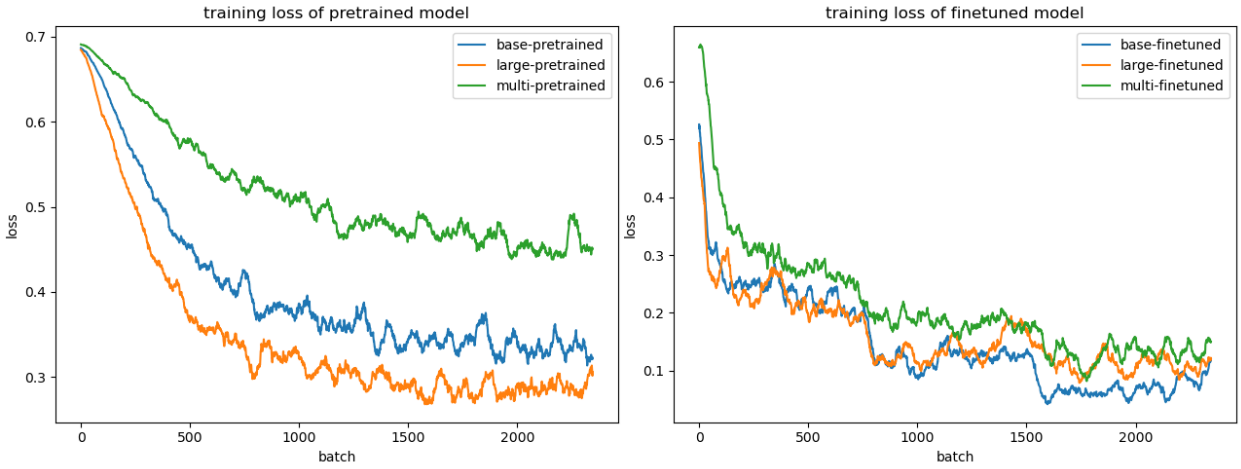

图 1: 三种预训练模型之间的训练过程对比

表 2: 各个模型在测试集上的准确率

	PRETRAIN			FINETUNE		
	EPOCH:1	EPOCH:2	EPOCH:3	EPOCH:1	EPOCH:2	EPOCH:3
BASE	84.56%	86.40%	87.01%	93.14%	93.84%	91.38%
LARGE	87.50%	88.89%	89.66%	93.76%	92.09%	93.63%
MULTI	75.96%	77.83%	79.20%	90.07%	91.39%	91.98%

图 2: 预训练和微调之间的训练过程对比

表 3: 各个模型在测试集上的损失函数值

	PRETRAIN			FINETUNE		
	EPOCH:1	EPOCH:2	EPOCH:3	EPOCH:1	EPOCH:2	EPOCH:3
BASE	0.379	0.325	0.310	0.176	0.193	0.230
LARGE	0.316	0.278	0.260	0.160	0.199	0.199
MULTI	0.520	0.471	0.449	0.234	0.220	0.234

4 分析讨论

4.1 实验分析

4.1.1 训练过程

由图1可知，使用预训练 BERT 训练分类模型时，梯度下降的速度为 LARGE>BASE>MULTI，并且最终的损失函数值也为 LARGE>BASE>MULTI。而在微调模型时，三种模型的训练速度和最终损失函数值都相近。这可能是因为在所有参数都参与训练时，BERT 在预训练时的差异在大量参数的优化下显得没那么重要了。

图2则显示三种模型在微调时的训练速度均快于冻结预训练参数的情形。

4.1.2 测试结果

表2的结果说明三种模型在微调后的准确率均高于冻结预训练参数时的准确率。如果取各自性能最好的 EPOCH 进行对比，则三种模型的性能满足：预训练：LARGE>BASE>MULTI；微调：BASE>LARGE>MULTI。

可以发现，微调模型在训练至 EPOCH2 时，出现了 LARGE 模型的准确率反而低于 BASE 模型准确率的情况。我们推测是由于 LARGE 模型参数过多，以至于在 EPOCH 不到 1 时就开始过拟合，导致在测试集上的表现反而不如 BASE 模型。而在 EPOCH3 时，LARGE 的准确率又高于 BASE，并且 LARGE 模型的损失函数值也低于 BASE 模型。这可能是因为较多的参数使得 LARGE 模型具有更好的“抗过拟合”能力，而 BASE 模型则由于参数过少导致其受训练样本影响较大。

我们着重考虑一下过拟合的情况。若认为在某个 EPOCH 之后，模型的准确率开始下降或者损失函数值开始上升，则认为模型发生了过拟合（我们认为这种假定是合理的，因为我们对每个 EPOCH 训练出的模型都会用测试集的全部数据进行测试，可以认为基本排除了偶然性的干扰）。由表2和表3的数据可以发现，三种模型在冻结预训练参数时，均没有在 3 个 EPOCH 之内发生过拟合。而在微调时，LARGE 和 BASE 模型存在过拟合的情况，MULTI 模型没有发生过拟合。这是由于 MULTI 和 BASE 两个模型的架构和参数数量是一致的，二者的区别在于预训练时使用的数据不同。因此，在微调了足够长的时间之后，可以认为这两个模型具有相近的性能。而 MULTI 模型之所以没有发生过拟合，是因为对于它而言，微调的 EPOCH 还不够多，没有达到符合自身参数数目的性能（即 BASE 模型的性能）。

4.2 困难及解决办法

我们实验中遇到的困难及解决办法如下：

- 冻结 BERT 参数时，使用逻辑回归分类效率低，收敛速度慢。解决办法：增加分类器的复杂度，使用两个线性层进行分类。
- 以 LARGE model 作为预训练模型时，GPU 显存不够。解决办法：使用混合精度 autocast() 存储中间数据。

5 结论

- 冻结预训练参数时，模型性能为 LARGE>BASE>MULTI。
- 微调预训练参数后，模型性能为 BASE>LARGE>MULTI。
- 冻结预训练参数时，模型不发生过拟合。
- 微调预训练参数时，MULTI 不发生过拟合，LARGE 最先发生过拟合但抗过拟合能力较强，BASE 发生过拟合且性能持续下降。

6 组员分工及贡献情况说明

- 吴玄：提出总体实验思路，负责实验的“微调”部分。
- 洪柯昕：负责实验的“预训练”部分，撰写实验报告。

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.