BERT 模型在 SQuAD 上的应用

江昊翰	3180101995	3180101995@zju.edu.cn
钟添芸	3180103009	3180103009@zju.edu.cn

摘　要：

在自然语言处理领域中，许多任务要求模型能够理解文字的语义（例如 SQuAD 任务）。在 Transformer 被提出后，Google 发布了一种双向 Transformer 的 Encoder 模型（BERT）。BERT 主要采用预训练的思想，在外部数据集中无监督训练并保存一个能够良好分析语义的模型，再进一步利用预训练的模型完成 SQuAD 任务。预训练中主要采用了 Masked-LM 以及语句预测的技巧。

关键词：

机器学习 自然语言处理 预训练模型 Transformer BERT SQuAD

第 1 章 绪论

1.1 SQuAD 课题背景

近年来，机器学习以一种极其猛烈的态势在各种行业、各种领域进行了广泛的应用。以需求进行区分，机器学习的主要任务可以分为视觉任务（CV）与自然语言处理任务（NLP）。在 NLP 领域，十分重要的一类任务是“序列到序列”(Seq2Seq)任务，这要求计算机能够将一串文本序列进行分析，并生成另一串序列作为答案。Seq2Seq 中有许多与我们生活紧密相关的部分，例如机器翻译、AI 聊天、文本摘要等。

Stanford Question Answering Dataset (SQuAD)是 Stanford 大学主办的 NLP 竞赛，在 NLP 领域久负盛名。该竞赛目前（2020.6）分为 SQuAD 1.1 与 SQuAD 2.0，二者都可以看作是 Seq2Seq 的任务。SQuAD 1.1 将给出一段阅读段落与题目文本，要求计算机能够在阅读段落中标记出题目的答案；**SQuAD 2.0** 则在此之上要求计算机能够判定阅读段落中是否含有题目答案，若否则应当放弃作答，这也是**本文所尝试解决的问题**。

SQuAD 的研究意义也不仅仅在于制作一个让计算机能够标注阅读段落中所需答案的学习模型。更进一步地，该模型得到的结果经过一些摘要或者扩充等操作，可以在 AI 聊天等领域产生重要作用。例如，用户在搜索引擎中键入问题后，搜索引擎可以通过类似 SQuAD 的分析在结果网页中自动选取、摘要用户真正需要的部分，节省大量翻阅的时间。因此，对于 SQuAD 的研究与竞赛是有着相当良好的应用意义的。

1.2 相关工作

本文所关注的 BERT 算法 [6] 是一种无监督的、预训练再微调的机器学习算法。因此，本节主

要针对该类算法的工作进行总结。

对于基于特征训练的无监督算法，Brown et al.[1]在 1992 年提出了非神经网络的实现；Tomas et al.[2]则在 2013 年提出了神经网络的实现模式。这类算法的主要特点是即使无监督也可以获取数据中的归类特征，整体上是对于数据集采用了巧妙的 trick 来自动生成无监督数据。缺点是一个问题需要一个新的网络，资源消耗大且不易调试。如果专注到 NLP 方向上的话，Peters et al. [3]提出的 ELMo 网络则是一种重要提升。ELMo 网络将每个 token 的上文与下文同时纳入考虑，进而很大地加强了模型对于 token 语义的感知能力与准确度。

进一步地，对于预训练微调的无监督算法，则出现得晚得多——Collobert 与 Weston[4]在 2008 年提出了对于 word embedding 进行预训练，再进一步应用于其他任务的模型思路。这项工作的意义是重大的，尽管其模型本身比较简易，但是其预训练的思想已经在机器学习的各个领域（尤其是 NLP）开花结果。Radford et al. [5]在 2018 年提出的 OpenAI GPT 模型在当时的许多 NLP 竞赛中获得了最佳成绩。而本文重点实现的 BERT 算法 [6]也在 2018 年的至少 11 项 NLP 顶级比赛中获得了最佳成绩。

第 2 章 BERT 算法实现

2.1 BERT 整体结构

Bidirectional Encoder Representation from Transformers（BERT）[6]，即双向 Transformer（Transformer 在课程中已经学习，此处不赘述）实现的 Encoder 算法，其与 OpenAI GPT 的整体架构对比图 [6]（去除后续任务部分）如下：

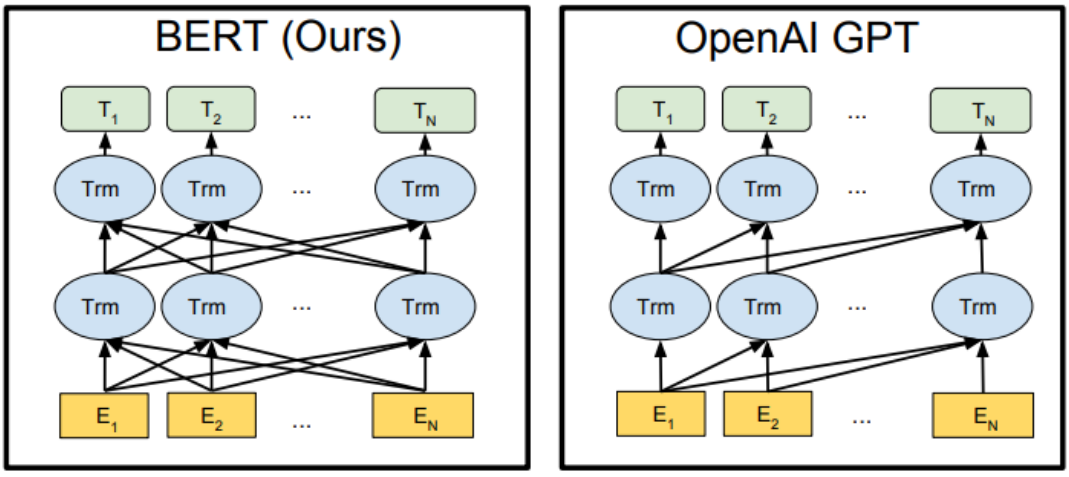

图 1 BERT 与 OpenAI GPT 整体架构对比图

从上图中可以看出，BERT 中 Transformer 是被所有前一层的 Transformer 所影响的，不论 Transformer 自身的位置；而 OpenAI GPT 则是 Transformer 只会被在自己位置前的 Transformer 所影响。由此可以发现，BERT 对于整体架构的创新点主要体现在其双向实现的 Transformer 架构，这也是为了能让模型同时感知到文本的上文与下文，从而更好地作出分析。

从整体上看，BERT 是利用 Masked LM 以及 Next Sentence Prediction 两种预训练任务，在完成实际任务前先训练出一个对于文本、句、词具有较好感知与预测能力的前置模型，再将

BERT 的输出交给后续网络（例如情感分类、SQuAD 等）。之所以该模型应当有效，是因为不论实际 NLP 任务是什么，句子构成、语义表达等特性应当是由语言本身所决定的，应当与 NLP 任务无关。
这种实现方式的优点是显而易见的——对于各种不同的 NLP 任务，我们只需要设计语义分析之后的部分，再把 BERT 与它连接即可。并且，在大部分情况下，BERT 部分只需要微调参数甚至完全不用改动，节省大量开发时间与计算时间。当然，BERT 也有着部分不足，这会在 2.3 节与 2.4 节中进行阐述。

2.2 Embedding

类似于大部分 NLP 的处理方式，BERT 需要对于文本做词嵌入（Embedding），即把文本映射到另一个多维空间中进行区分与联系。在 BERT 中，考虑到之后预训练任务需求（2.3 节与 2.4 节），我们需要同时考虑词本身、句子整体以及词在句子中的位置。因此，BERT 中的 embedding 是由三个 embedding 求和得到的：

$$Embedding = Token\_Embedding + Segment\_Embedding + Position\_Embedding$$

- $Token\_Embedding$ 表示传统的词向量，由词语本身映射得到，用于区分不同词语
- $Segment\_Embedding$ 表示句向量，由句中所有的词语计算得到，用于区分不同句
- $Position\_Embedding$ 表示位置向量，由该词语在句中位置计算得到

与传统 Transformer 略有不同的是，上述三种 embedding 均是需要经过训练的，而 Transformer 并不训练 $Position\_Embedding$（直接采用周期正弦函数），这一点上给予了模型稍大一些的自由度。

2.3 Masked-LM 预训练

BERT 需要能够给出一个良好的语言模型，而这就首先要求 BERT 能够良好地理解词语的语义。因此，Google 研究者[6]提出了一种 Masked Language Model (Masked LM or MLM)的训练方法，该任务只需要大量的有效文本即可进行（如 Wiki 文本）。我们采用无监督的方式来生成训练数据：将文本中约 12%的单词替换为[MASK]，约 1.5%的单词随机替换成文本中出现的其他单词，再令模型去预测这些[MASK]的单词原本是什么单词。另外，此处将 1.5%（这个数字来源于 Google 的测试结果）的单词替换为随机单词的原因是在实际文本中并不出现[MASK]，通过随机单词可以在一定程度上缓解[MASK]加入文本带来的语义误差。
例如：" It is sunny today. Let's go mountain climbing!" →
" It is [mask] today. Let's go mountain climbing!"
最后将结果与应有结果进行比较，计算 Cross Entropy，再将误差反向传播更新网络。

2.4 语句预测预训练

对于形如 SQuAD 这种任务，BERT 应当能够分析词与句的关系以及句与句的关系，因此我们需要在 MLM 的基础上进一步追加对于句子联系的训练——Next Sentence Prediction(NSP)。该预训练的设计比较简单，同样是无监督地生成数据 :从文本库中抽取一些语句对，使得 50%的语句对是连续的上下句，另外 50%不是。再将语句对交给模型，令其分辨该语句对是不是连续的上下句。最后将结果与应有结果进行比较，计算 Cross Entropy，再将误差反向传播更新网络。

2.5 微调实现 SQuAD

在 BERT 之后，我们添加了一层简单的预测网络，用于预测答案的起始位置 Pos_L 与终止位置 Pos_R。以起始位置为例，我们构造一个节点作为起始向量 L （结束向量为 R），下标为 index 的单词会成为答案起始单词的概率使用 softmax 计算：

$$P_{index} = \frac{e^{L \bullet T_{index}}}{\sum_j e^{L \bullet T_j}}$$

上式中 T_{index} 代表下标为 index 的 transformer 输出向量

同时，我们计算如下积分作为评估[i, j]作为答案范围的正确性：

$$Score(i, j) = L \bullet T_i + R \bullet T_j$$

最终，我们令模型在最大化上述分数情形取出最大概率的起始位置与终止位置

另外，对于 SQuAD 加入的无答案问题，我们则是采用额外训练一个阈值分数 $Score_{threshold}$，当最大化的 $Score_{\max} < Score_{threshold}$ 时，返回无答案

第 3 章 测试结果

3.1 预训练结果

由于 BERT 预训练计算资源消耗极其巨大（使用个人电脑 GPU 或者初级 TPU 几乎不可能完成）且调试不便，我们最终使用了 Google 训练完成的预训练模型（在代码中表现参数初值加载了已有的参数模型）。根据 Google 原论文的数据 [6]，该预训练模型在完成 MLM 与 NSP 预训练之后，其可以在 NSP 训练集上达到 97%~98%的准确率。

3.2 SQuAD 结果

我们起初希望按照 BERT 模型的框架直接用 torch.nn 里的库实现一个模型，但是在 debug 良久还是没有结果，并且能够实现的部分效果较差的情况下，我们认为直接使用预训练的结果可能更好。
我们使用 transformer 库导入预训练好的 BERT 模型，再通过自己规定 scheduler&optimizer 等情况下进行参数的调试，在运行的过程中碰到了相当多的问题，查询了 transformer 库的相关文档也不能得到解决，于是我们仿照 transformer 库的示例程序的流程，调用官方给出的接口，能得到和 transformer 库官方给出的预期的结果，其中 bert 的结果比 XLNet 的结果差了些许，bert 在 SQUAD1.1 上的表现是 81.4%的准确率，而 XLNet 模型在 SQUAD2.0 上的结果是 76%的有回答准确率和 84%的无回答准确率。

第 4 章 结语

我们小组在参考使用开源代码的基础上，经过学习、分析、调试，最终实现了 BERT 模型在 SQuAD 上的成功应用。尽管我们自己所撰写的代码并未跑出较好结果，但是我们也的确收获颇丰——至少我们详实地理解了 BERT 模型，也见识到了前沿机器学习任务的复杂性与精密性。当然，我们可能并没有很好地完成全部任务，仍需要加强自身的代码工程能力与模型构造能力。
愿有所广益。

附录

代码链接

github 链接：
https://github.com/Kizuna-AI/ZJUML4-2
注：请阅读 repo 的 readme.md

任务分配

姓名	学号	邮箱	任务贡献
江昊翰	3180101995	3180101995@zju.edu.cn	主要代码/次要报告
钟添芸	3180103009	3180103009@zju.edu.cn	主要报告/次要代码

参考文献

[1] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. Computational linguistics, 18(4):467–479.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc

[3] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In ACL.

[4] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM.

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

[6] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805 (2018)