

深度学习与自然语言处理第五次作业

李志渊
PT2200081

一．问题描述

任意选取 3~4 个目前前沿语言大模型（不限中文或者英文），通过提示工程的方法来检验和对比不同模型下游任务上的性能，可选择 3~5 个不同的自然语言下游任务来进行测试。

二．大语言模型

大型语言模型（LLM）是指包含数千亿（或更多）参数的语言模型，这些参数是在大量文本数据上训练的，例如模型 GPT-3、PaLM、Galactica 和 LLaMA。具体来说，LLM 建立在 Transformer 架构之上，其中多头注意力层堆叠在一个非常深的神经网络中。现有的 LLM 主要采用与小语言模型类似的模型架构（即 Transformer）和预训练目标。作为主要区别，LLM 在很大程度上扩展了模型大小、预训练数据和总计算量（扩大倍数）。他们可以更好地理解自然语言，并根据给定的上下文（例如 prompt）生成高质量的文本。这种容量改进可以用标度律进行部分地描述，其中性能大致遵循模型大小的大幅度增加而增加。然而根据标度律，某些能力（例如上下文学习）是不可预测的，只有当模型大小超过某个水平时才能观察到。

2.1 GPT-3 模型

GPT-3 是一种语言模型，它可以通过少量的样本进行学习，因此被称为“Few-Shot Learner”。和人类一样，GPT-3 不需要完全不看任何样例就能学习，只需要看一小部分样例就能学会更多的知识。GPT-3 的体量非常庞大，因此在下游任务中进行 fine-tune 的成本很高。为了解决这个问题，GPT-3 使用了“In-Context Learning”的方式，在不进行梯度更新或 fine-tune 的情况下，直接在上下文中进行学习。

GPT-3 虽然很强悍，但是仍旧有局限性：

- 1) 数据量和参数量的骤增并没有带来智能的体感。从参数量上看，从 GPT2 1.5B 到 GPT3 175B 约 116 倍参数量的增加，从数据量上看，GPT2 40G 到 GPT3 570G 近 15 倍训练数据增加，带来的“更”智能，或者简单点说“更 few/zero-shot”的能力。
- 2) GPT-3 的训练数据是从互联网上爬取的，因此可能存在一些错误或不准确的数据。
- 3) GPT-3 在处理某些任务时可能会出现错误或不准确的结果，以及不合理或不合逻辑的结果。

2.2 GPT3.5 模型

GPT-3 纵然很强大，但是对于人类的指令理解的不是很好，这也就延伸出了 GPT3.5 诞生的思路。在做下游的任务时，我们发现 GPT-3 有很强大的能力，但是只要人类说的话不属于 GPT-3 的范式，他几乎无法理解。

ChatGPT 是基于 GPT3.5 的基础模型框架，核心变化在于通过真实的调用数据以及人类反馈的强化学习进行训练。ChatGPT3.5 主要用于自然语言处理、机器翻译等任务，而 ChatGPT3.5-Turbo 拥有更强大的强度，可用于更复杂的语言分析，比如情感分析、语法结构分析。所以，ChatGPT 和 GPT3.5 是同一系列的产品，但 ChatGPT 是在 GPT3.5 的基础上进行了改进和优化。

2.3 BLOOM 模型

BLOOM (BigScience Large Open-science Open-access Multilingual Language Model) 是在 46 种自然语言和 13 种编程语言上训练的 1760 亿参数语言模型，其是由数百名研究人员合作开发和发布的。

2.4 BERT 模型

BERT 的全称是 Bidirectional Encoder Representation from Transformers，由谷歌开发的语言模型，BERT 的新语言表示模型，它代表 Transformer 的双向编码器表示。与最近的其他语言表示模型不同，BERT 旨在通过联合调节所有层中的上下文来预先训练深度双向表示。因此，预训练的 BERT 表示可以通过一个额外的输出层进行微调，适用于广泛任务的最先进模型的构建，比如问答任务和语言推理，无需针对具体任务做大幅架构修改。

三．下游任务

3.1 文本分类

任务：输入一条新闻，将其进行分类

例如：输入一条新闻标题，输出为该新闻的类别

3.2 问答系统

任务：回答用户提出的问题

例如：问“1996 年 NBA 总决赛 FMVP 获得者是谁”。答“迈克尔乔丹”

3.3 机器翻译

任务：将一段英文文本翻译为中文

例如：输入一段英文文本，输出对应的中文翻译

3.4 文本纠错

任务：将带有拼写错误的文本修正

例如：输入一段含有拼写错误的文本，输出修正后的文本

3.5 文本摘要

任务：生成一段新闻的简洁概要

例如：输入一篇新闻稿，输出其简介，能概括原文内容

四．大模型对比

	GPT3.5	BLOOMChat	文心一言
文本分类 问题：我国首艘国产大型邮轮出坞	根据您提供的信息，这则新闻似乎属于“经济”或“旅游”类别。该新闻报道了中国建造的首艘国产大型邮轮下水的消息，这反映了中国在船	船舶	我国首艘国产大型邮轮“爱达·魔都号”于 2023 年 6 月 6 日成功出坞，这标志着我国邮轮产业迈向新的里程碑。

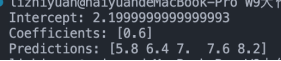
	<p>舶制造和旅游业方面的发展。请注意，我无法直接访问实时新闻或了解最新的事件，因此我的判断基于截至 2021 年的知识和常识。对于最新的新闻，请您参考可靠的新闻来源进行确认。</p>		
<p>问答系统</p> <p>1996 年 NBA 总决赛 FMVP 是谁</p>	<p>1996 年 NBA 总决赛的最有价值球员 (FMVP) 是迈克尔·乔丹 (Michael Jordan)。乔丹领导的芝加哥公牛队在 1996 年总决赛中以 4 比 2 击败西雅图超音速队，获得了他们的第四个 NBA 总冠军。乔丹在系列赛中展现了卓越的表现，场均得到 27.3 分、5.3 个篮板和 4.2 次助攻。他在总决赛期间的出色表现使他获得了该系列赛的最有价值球员荣誉。</p>	<p>1996 年 NBA 总决赛 FMVP 是迈克尔·乔丹 (Michael Jordan)。</p>	<p>1996 年 NBA 总决赛 FMVP 得主是迈克尔·乔丹。</p>
<p>机器翻译</p> <p>原文为 Re-thinking Federated Active Learning based on Inter-class Diversity 的摘要</p>	<p>尽管联邦学习取得了令人瞩目的进展，但大多数研究假设客户端的数据是完全标记的。然而，在真实世界的情况下，每个客户端可能有大量未标记的实例。在利用未标记数据的各种方法中，联邦主动学习框架已成为一种有希望的解决方案。在分散设置中，有两种可用的查询选择器模型，即“全局”和“仅本地”模型，但很少有文献讨论它们的性能优势及其原因。在这项工作中，我们首先证明了两种选择器模型的优越性取决于全局</p>	<p>翻译如下：虽然联邦学习取得了令人瞩目的进展，但在大多数研究中，都假设客户端的数据是完全带标签的。然而在现实世界场景中，每个客户端可能都包含大量的未标记实例。为了充分利用未标记的数据，一种联邦主动学习框架正在成为一种有前景的解决方案。在去中心化的场景下，有两种可用的“全局”和“局部”查询选择模型，但是很少有文</p>	<p>尽管联邦学习 awe-inspiring 进展，大多数研究都假设客户端的数据是完全标记的。然而，在现实世界的情况下，每个客户端都可能有一个显著数量的未标记实例。在利用未标记数据的各种方法中，联邦主动学习框架已成为一个有前途的解决方案。在分布式环境中，有两种类型的可用查询选择器模型，即“全局”和“仅本地”模型，但很少有文献讨论它们的性能优势及其原因。在这项工作中，我们首先证明了两种选择器模型的</p>

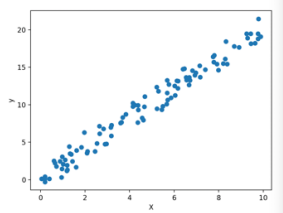
	<p>和本地类间多样性。此外，我们观察到全局和仅本地模型是解决双方不平衡问题的关键。基于我们的发现，我们提出了 LoGo，这是一种对不同本地异质性水平和全局不平衡比率具有鲁棒性的 FAL 抽样策略，通过两步主动选择方案集成了两种模型。在总共 38 个实验设置中，LoGo 在数量上一直优于六种主动学习策略。</p>	<p>献比较这两种模型的性能，以及造成这种差异的原因。在本文中，我们首先证明了这两种选择模型的优势，这取决于全局和局部之间的多样性。此外，我们观察到全局和局部模型是解决不平衡的关键。基于我们的发现，我们提出 LoGo，一种适用于各种局部异质性和全局不平衡的 FAL 采样策略，它结合了这两种模型，使用两个步骤中的选择性采样方案。LoGo 在 38 种实验设置中，始终优于现有的六种主动学习策略。</p>	<p>优越性取决于全局和局部类别多样性。此外，我们观察到全局和仅本地模型是解决双方不平衡的关键。基于我们的发现，我们提出了 LoGo，这是一种 FAL 抽样策略，能够很好地应对局部异质性和全局不平衡比例的变化，通过两种主动选择方案的整合这两个模型。在总共 38 个实验设置中，LoGo 一致地优于六种主动学习策略。</p>
<p>文本摘要</p> <p>对以下文本进行摘要处理：在污染治理和生态环境建设方面，上海成绩显著。截至 2022 年底，上海纳入排污许可管理的排污单位约 4.3 万家；生活垃圾回收利用率达 42%，生活垃圾焚烧能力达 2.8 万吨/日，湿垃圾集中处置能力超过 7000 吨/日；农村生活污水治理率达 93.3%，提前完成了“十四五”目标任务。此外，上海的森林覆盖率达到 18.51%。在产业领域，上海实施了 500 项市级产业结构调整项目，推动了市级绿色工厂和绿色供应链</p>	<p>上海在污染治理和生态环境建设方面取得了显著成绩。截至 2022 年底，上海纳入排污许可管理的排污单位约 4.3 万家。生活垃圾回收利用率达到 42%，生活垃圾焚烧能力达到 2.8 万吨/日，湿垃圾集中处置能力超过 7000 吨/日。农村生活污水治理率达到 93.3%，提前完成了“十四五”目标任务。此外，上海的森林覆盖率达到 18.51%。在产业领域，上海实施了 500 项市级产业结构调整项目，推动了市级绿色工厂和绿色供应链</p>	<p>在污染治理和生态环境建设方面，上海成绩显著。截至 2022 年底，上海纳入排污许可管理的排污单位约 4.3 万家；生活垃圾回收利用率达 42%，生活垃圾焚烧能力达 2.8 万吨/日，湿垃圾集中处置能力超过 7000 吨/日；农村生活污水治理率达 93.3%，提前完成“十四五”目标任务；森林覆盖率达 18.51%。在能源领域，全面启动百万千瓦海上</p>	<p>上海在污染治理和生态环境建设方面成绩显著，纳入排污许可管理的排污单位约 4.3 万家，生活垃圾回收利用率达 42%，农村生活污水治理率达 93.3%，森林覆盖率达 18.51%。在绿色低碳发展方面，上海实施了 500 项市级产业结构调整项目，创建了 66 家市级绿色工厂和 11 家市级绿色供应链，同时推进了金山二工区、星火开发区等重点区域的转型升级。在能源领域，上海全面启动百万千瓦海上风电建设，加</p>

<p>污水治理率达93.3%，提前完成“十四五”目标任务；森林覆盖率达18.51%。对于绿色低碳发展的执着，已渗透进各行各业，深入人心——产业领域，上海去年实施市级产业结构调整项目500项，推动创建66家市级绿色工厂、11家市级绿色供应链，有序推进金山二工区、星火开发区等重点区域的转型升级。在能源领域，全面启动百万千瓦海上风电建设，加快推进“光伏+”专项工程。在交通领域，去年新增或更新的公交车、出租车全部采用新能源汽车，建成各类充电桩近70万根，累计推广新能源汽车约100万辆，集装箱铁海联运完成约5万标准箱，同比增长40.15%。在建筑领域，上海绿色建筑</p>	<p>的创建，并有序推进了重点区域的转型升级。在能源领域，全面启动百万千瓦的海上风电建设，并加快推进了“光伏+”专项工程。在交通领域，新增或更新的公交车和出租车全部采用新能源汽车，建设了近70万个各类充电桩，累计推广了约100万辆新能源汽车，集装箱铁海联运完成了约5万标准箱，同比增长40.15%。在建筑领域，上海的绿色建筑规模已经达到3.2亿平方米。这些成就表明上海在绿色低碳发展方面的坚定承诺已经深入各行各业，广泛影响人们的心中。上海将继续推动污染防治和美丽中国建设，坚持降碳、减污、扩绿、增长的协同推进，采取精准、科学、依法治污的措施，推动绿色高质量发展。</p>	<p>风电建设，加快推进“光伏+”专项工程。在交通领域，去年新增或更新的公交车、出租车全部采用新能源汽车，建成各类充电桩近70万根，累计推广新能源汽车约100万辆，集装箱铁水联运完成约5万标准箱，同比增长40.15%。在建筑领域，上海绿色建筑规模已达3.2亿平方米。</p>	<p>快推进“光伏+”专项工程。在交通领域，新能源汽车占比逐年提高，建成近70万根充电桩，累计推广新能源汽车约100万辆。在建筑领域，上海绿色建筑规模已达3.2亿平方米。</p>
--	---	---	---

规模已达 3.2 亿平方米。			
数学推理 小华每天读 24 页书，12 天读完了《红岩》一书。小明每天读 36 页书，几天可以读完《红岩》？	小华每天读 24 页，12 天读完了《红岩》一书。那么小华总共阅读的页数为 $24 \text{ 页/天} \times 12 \text{ 天} = 288 \text{ 页}$ 。 现在我们来计算小明需要多少天才能读完《红岩》。 假设小明需要 x 天才能读完《红岩》。根据题目中的信息，小明每天读 36 页，所以小明总共需要阅读的页数为 $36 \text{ 页/天} \times x \text{ 天} = 36x \text{ 页}$ 。 根据题目要求，小明需要读完的总页数等于小华已经读完的总页数，即 $36x = 288$ 。 我们可以解这个方程来计算 x ： $36x = 288$ 解这个方程，可以得到： $x = 288 / 36$ $x = 8$ 所以，小明需要 8 天才能读完《红岩》。 正确	如果小明每天读 36 页书，12 天可以读完《红岩》。 错误	我们可以先求出小华读完《红岩》所需的天数，再根据小明的阅读速度计算小明读完《红岩》所需的天数。 已知小华每天读的页数：24 页 已知小华读完《红岩》所需的天数：12 天 根据页数=每天读的页数×天数，可计算《红岩》的总页数： $24 \times 12 = 288 \text{ 页}$ 已知小明每天读的页数：36 页 根据页数=每天读的页数×天数，可计算小明读完《红岩》所需的天数： $288 \div 36 = 8.0 \text{ 天}$ 所以，小明读完《红岩》需要 8.0 天。 正确
5 辆汽车 4 次可以运送 100 吨钢材，如果用同样的 7 辆汽车运送 105 吨钢材，需要运几次？	假设需要 x 次才能用 7 辆汽车运送 105 吨钢材。 根据题目中的信息，5 辆汽车 4 次可以运送 100 吨钢材，所以每次运送的量为 $100 \text{ 吨} / 4 \text{ 次} = 25 \text{ 吨}$ 。 我们可以建立一个比例关系来解决这个问题： $5 \text{ 辆汽车的运输量} / 4$	7 辆汽车运送 105 吨钢材需要运两次。 错误	我们可以先计算每辆汽车每次可以运送的钢材数量，然后计算 7 辆汽车每次可以运送的钢材数量，最后计算需要运几次。 已知原来有 5 辆汽车 已知原来需要运送 4 次 已知原来需要运送 100 吨钢材 根据每辆汽车每次可以

	<p>次 = 7 辆汽车的运输量 / x 次</p> <p>将已知值代入，得到：</p> $25 \text{ 吨} / 4 \text{ 次} = 105 \text{ 吨} / x \text{ 次}$ <p>通过交叉乘法计算：</p> $25 \text{ 吨} * x \text{ 次} = 105 \text{ 吨} * 4 \text{ 次}$ <p>得到：</p> $25x = 420$ <p>解这个方程，可以得到：</p> $x = 420 / 25$ $x \approx 16.8$ <p>因为次数必须是整数，所以我们可以向上取整，得到：</p> $x = 17$ <p>所以，用 7 辆汽车运送 105 吨钢材需要运 17 次。</p> <p>错误</p>		<p>运送的钢材数量=总钢材数量÷汽车数量÷运输次数，可计算每辆汽车每次可以运送的钢材数量：</p> $100 \div 5 \div 4 = 5.0 \text{ 吨}$ <p>已知现在有 7 辆汽车</p> <p>已知现在需要运送 105 吨钢材</p> <p>根据每辆汽车每次可以运送的钢材数量=总钢材数量÷汽车数量÷运输次数，可计算现在每辆汽车每次可以运送的钢材数量：</p> $105 \div 7 \div 4 = 3.75 \text{ 吨}$ <p>根据每辆汽车每次可以运送的钢材数量=总钢材数量÷汽车数量÷运输次数，可计算需要运几次：</p> $105 \div (7 \times 3.75) = 4.0 \text{ 次}$ <p>所以，需要运 4.0 次。</p> <p>错误</p>
写一个线性回归的代码，python	<pre>import numpy as np import matplotlib.pyplot as plt # 生成随机数据 np.random.seed(0) X = np.random.rand(100, 1) * 10 y = 2 * X + np.random.randn(100, 1) # 可视化数据 plt.scatter(X, y) plt.xlabel('X') plt.ylabel('y') plt.show() # 初始化模型参数 theta = np.zeros((2, 1)) # 参数向量 [theta0,</pre>	<p>Python 是一种解释型、高级、面向对象、动态数据类型的编程语言。它支持多种编程范式，其中包括函数式编程、命令式编程和面向对象编程。</p> <p>Python 具有丰富的库，能够支持多种开发任务，其中包括系统管理、网络通信、图形图像处理、数学计算和文本处理等。Python 的语法简洁清晰，具有很强的可读性。</p> <p>下面是一个线性回归的 Python 代码</p>	<pre>import numpy as np from sklearn.linear_model import LinearRegression # 创建训练数据 X_train = np.array([[1], [2], [3], [4], [5]]) y_train = np.array([2, 4, 5, 4, 5]) # 创建模型并拟合数据 model = LinearRegression() model.fit(X_train, y_train) # 打印模型的截距和</pre>

	<pre> theta1] lr = 0.01 # 学习率 num_ iterations = 1000 # 迭代次数 # 执行梯度下降算法 m = len(X) # 样本数量 X_b = np.c_[np.ones((m, 1)), X] # 添加偏置项, X_b 为增广矩阵 for iteration in range(num_ iterations): gradients = 2 / m * X_b.T.dot(X_b.dot(theta) - y) # 计算梯度 theta = theta - lr * gradients # 参数更新 # 打印训练得到的参数 print('theta0:', theta[0][0]) print('theta1:', theta[1][0]) # 可视化拟合曲线 X_new = np.array([[0], [10]]) # 预测输入 X_new_b = np.c_[np.ones((2, 1)), X_new] # 增广矩阵 y_predict = X_new_b.dot(theta) # 预测输出 plt.plot(X_new, y_predict, 'r-', label='Predictions') plt.scatter(X, y, label='Training Data') plt.xlabel('X') plt.ylabel('y') plt.legend() plt.show() </pre>	<p>示例：</p> <pre> import numpy as np # 创建一个线性回归模型 model = np.polyfit(x, y, 1) # 拟合曲线 plt.plot(x, y, 'ro', label='data') plt.plot(x, model[0] + model[1] * x, label='fit') plt.legend() plt.show() </pre> <p>这个示例使用 numpy 库创建了一个线性回归模型, 并使用 polyfit 函数拟合了 x 和 y 之间的曲线。然后, 使用 plot 函数绘制了数据点和曲线。最后, 通过 legend 函数添加了图例。</p> <p>无法正常运行</p>	<p>系数</p> <pre> print('Intercept:', model.intercept_) print('Coefficients:', model.coef_) # 创建测试数据 X_test = np.array([[6], [7], [8], [9], [10]]) # 预测测试数据的输出值 y_pred = model.predict(X_test) print('Predictions:', y_pred) </pre>  <pre> cizhiyuan@naliyandemacbook-Pro ~\$ python 1.py Intercept: 2.1999999999999993 Coefficients: [0.6] Predictions: [5.8 6.4 7. 7.6 8.2] </pre>
--	---	--	--

由于 Bard 目前支持的语言只有英语、日语和韩语，故没有进行测试。

五．总结

根据以上下游任务的对比，可以明显看到 GPT 所给出的回答是过于冗长的，但其在回答的完整性以及翻译的准确性上是有较大优势的。从数学推理题的角度，三者的表现都不是很好，这是还需要加强的部分，尤其是 BLOOMChat，它的回答不仅没有完全错误，而且毫无过程可言。

而使用模型生成代码时，可以看出 ChatGPT 生成的代码更有逻辑也更加完善，BLOOMChat 在同样问题下生成的代码无法运行，文心一言生成的代码是可运行的。