

## 1-使用大模型进行指令学习

笔记本: LLM\_practice

创建时间: 2025/2/1 14:36

更新时间: 2025/2/3 16:29

作者: alexritchie@163.com

URL: <https://developer.baidu.com/article/details/1462960>

---

# 1-使用大模型进行指令学习

---

## 尝试获取Stanford Sentiment Treebank (SST-2)数据集并运行

```
# 加载 SST-2 数据集
dataset = load_dataset("glue", "sst2", split="test[:20]")

# 查看训练集的前几个样本
print(dataset)
```

一个具体的SST-2测试样例

```
this film 's relationship to actual tension is the same as what christmas-
tree flocking in a
spray can is to actual snow : a poor -- if durable -- imitation .

-1
```

## 使用DeBERTa的传统编码器微调

尝试用DeBERTa模型，在训练集上微调后，在test来进行预测，看看具体准确率是多少？

- 用全部的train训练集微调deberta-base，记录accuracy。
- 仅使用16、64、256和1024个样本，微调deberta-base，记录accuracy。

## 什么是指令学习？

指令学习（Instruction Learning）是一种让机器根据人类提供的指导性信息来完成特定任务的方法。在指令学习中，机器不依赖大规模的数据集进行预测和决策，而是依赖于人类

提供的指导性信息。这种学习方法在处理一些结构化的任务时具有显著的优势，比如问答、机器翻译和文本生成等。

说到底一句话：写一个指令模版，让生成式模型接着进行生成。

例如：

```
Below is an instruction that describes a task, paired with an input that
provides further
context. Write a response that appropriately completes the request.

### Instruction:
Analyze the given text from an online review and determine the sentiment
polarity. Return a
single number of either -1 and 1, with 1 being positive and -1 being the
negative sentiment.
No further explanation or justification is required.

### Question:
this film 's relationship to actual tension is the same as what christmas-
tree flocking in a
spray can is to actual snow : a poor -- if durable -- imitation . What' s the
sentiment polarity
of the text?

### Response:
```

试着把这句话发给ChatGPT，看看它会返回什么？

### Instruction:

Analyze the given text from an online review and determine the sentiment polarity. Return a single number of either -1 and 1, with 1 being positive and -1 being the negative sentiment. No further explanation or justification is required.

### Question:

this film 's relationship to actual tension is the same as what christmas-tree flocking in a spray can is to actual snow : a poor -- if durable -- imitation . What's the sentiment polarity of the text?

### Response:

🌀 -1



ChatGPT直接返回了-1，看起来和正确答案是一致的，这就带来几个问题：

- 大模型的智能看起来是通过生成模型的“涌现”机制来完成的，也就是大模型在使用之前经过类似的过程来进行训练，这样它可以根据上文来预测下文的生成。
- 越大越新的模型可以带来越好的指令学习能力，反之，越小越旧的模型可能会出现“幻觉” (hallucination)。因此，在进行指令学习而不需要进行模型的训练时，尽可能使用更大的模型来完成，例如ChatGPT、GPT-o1、全量的DeepSeek-r1、Qwen等模型。
- 指令模版一般参考alpaca，其书写规范可以参考：[https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca)

参考这样的格式：

```
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
```

```
### Instruction:
```

```
{instruction}
```

```
### Input:
```

```
{input}
```

```
### Response:
```

万事具备，我们试着把所有的代码合在一起，调用一个DeepSeek Distil Llama 8B模型，利用推理模式来完成一个指令学习。

```
import transformers
import torchmodel_id = "deepseek-ai/DeepSeek-R1-Distill-Llama-8B"
```

```

pipeline = transformers.pipeline(
    "text-generation",
    model=model_id,
    max_new_tokens=2048,
    model_kwargs={"torch_dtype": torch.bfloat16},
    device_map="auto"
)

prompt = """Below is an instruction that describes a task,
paired with an input that provides further context. Write a
response that appropriately completes the request.

Before answering, think carefully about the question and
create a step-by-step chain of thoughts to ensure a logical
and accurate response.

### Instruction:
You are an expert of consumer comment analysis. Analyze the given text from
an online review and determine
the sentiment polarity. Return a single number of either -1
and 1, with -1 being negative and 1 being the positive
sentiment.

### Input:
this film's relationship to actual tension is the same as
what christmas-tree flocking in a spray can is to actual
snow : a poor — if durable — imitation .

### Response:
<think>"""

response = pipeline(prompt)

print(response[0]["generated_text"])

```

注意几个细节：

- 在Instruction的第一句，要明确的说明所分析的领域，类似于角色扮演，告诉模型你是什么角色。
- 具体执行的过程放在Instruction部分，给出明确的输出。
- 注意每个section之前的###
- 注意在Response之后的<think>标签，这是为了触发模型的Reasoning过程

到这里，我们可以试着给模型更多的输入，并且写一个函数把最终的结果解析出来，并且作为预测标签：

```

from datasets import load_dataset

# from huggingface_hub import login
#
# login(
#     token = "hf_xxx", # ADD YOUR TOKEN HERE
#     add_to_git_credential=True
# )

# 加载 SST-2 数据集
dataset = load_dataset("glue", "sst2", split="test[:20]")

# 查看训练集的前几个样本
print(dataset)

import transformers
import torch

# model_
model_id = "deepseek-ai/DeepSeek-R1-Distill-Llama-8B"
# model_

pipeline = transformers.pipeline(
    "text-generation",
    model=model_id,
    max_new_tokens=2048,
    model_kwargs={"torch_dtype": torch.bfloat16},
    device_map="auto"
)

prompt_style = """Below is an instruction that describes a task, paired with
an input that provides further context. Write a response that appropriately
completes the request.

Before answering, think carefully about the question and create a step-by-
step chain of thoughts to ensure a logical and accurate response.

### Instruction:

Analyze the given text from an online review and determine the sentiment
polarity. Return a single number of either -1 and 1, with -1 being negative
and 1 being the positive sentiment.

### Input:
{}

```

```
### Response:
<think>{}"""

for i, item in enumerate(dataset):
    input_text = item['sentence']

    response = pipeline(prompt_style.format(input_text, ""))
    print(response[0]["generated_text"])
    print("true label: " + str(item['label']))
```

## 试着改变不同的模型大小

接下来，试着改变模型的大小，并且完成所有样本的预测后，和test的真实标签，统计一个准确率看看。

注意到，我们实际并没有使用到train训练集，也不需要去训练这个模型。

那这就带来一个问题，我是否可以自己来架设一个全量的DeepSeek来进行推理呢？

## 一、利用Ollama部署DeepSeek

- 首先从Ollama的官网下载安装包：<https://ollama.com>
- 在左上角的Models标签里面可以找到不同模型
- 可以试着用cmd，先运行一个32B的模型

```
ollama run deepseek-r1:32b
```

运行之后，再试着把刚才的指令发送给deepseek看看。

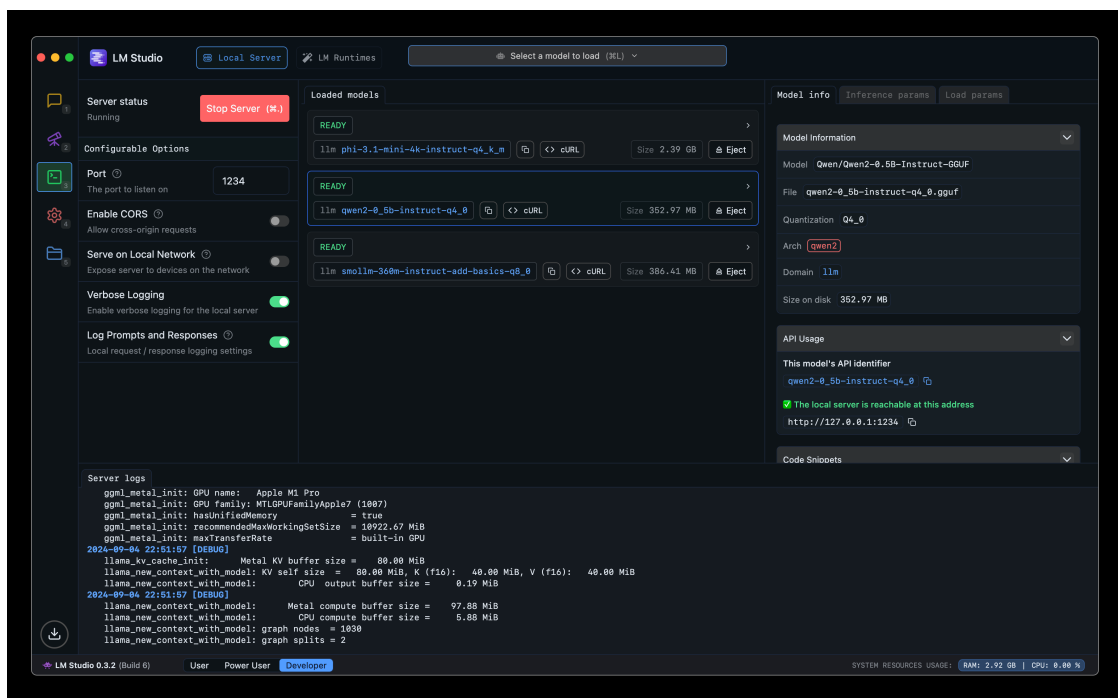
## 如何用代码API直接调用自己架设的DeepSeek

可以参考：

<https://github.com/datawhalechina/handy-ollama/tree/main/docs/C4>

## 二、使用LMStudio部署DeepSeek

- 首先从LMStudio下载安装包：<https://lmstudio.ai>
- 假设REST API
- 可以参考：<https://lmstudio.ai/docs/api>



## 记录结果

试着用Llama、Qwen、Gemma、Phi-4和Mistral来统计下SST-2指令学习的准确度。再和全量微调、小样本微调的DeBERTa结果对比一下。