



9月17日，Gitee Xtreme 极智AI重磅发布，来Gitee直播间一起探索AI时代的软件研发新模式



Tele-AI/T1

Watch 1 Star 0 Fork 0

代码

Issues 0

Pull Requests 0

Wiki

统计

流水线

服务

该仓库未声明开源许可证文件（LICENSE），使用请关注具体项目描述及其代码上游依赖。



简介

T1 模型是 TeleChat 系列专注于复杂推理的模型，由中国电信人工智能研究院基于国产算力研发训练。该系列模型借助先进的思维推理和批判纠错能力，在下游复杂任务中有很好的表现。本次我们开源了 T1-35B 和 T1-115B 两款不同尺寸的模型，与同尺寸模型相比都具有较好的效果表现。

暂无标签

README

0 Stars

1 Watching

0 Forks

发行版

暂无发行版

贡献者 (1)

全部



近期动态

姜 2个月前推送了新的提交到 master 分支，522b711...9de412e

姜 4个月前推送了新的提交到 aster 分支

姜 4个月前创建了分支

master 分支 1 标签 0

克隆/下载

姜卓 update README.md. 9de412e 2个月前 2次提交

README.md update README.md. 2个月前

README

AI 翻译由 translate.js 提供

Translate to English

T1

目录

模型介绍

- 训练策略
- 模型下载

效果评测

模型推理

- Transformers
- ModelScope
- vLLM 推理
- 离线推理
- OpenAI 兼容的 AP...
- 推理注意事项

国产化适配

声明、引用

- 声明
- 引用

T1

github · Hugging Face · ModelScope · gitee · WeChat

目录

- 模型介绍
- 效果评测
- 模型推理
- 国产化适配
- 声明、协议、引用

模型介绍

T1 模型是 TeleChat 系列专注于复杂推理的模型，由中国电信人工智能研究院基于国产算力研发训练。该系列模型借助先进的思维推理和批判纠错能力，在下游复杂任务中有很好的表现。本次我们开源了 T1-35B 和 T1-115B 两款不同尺寸的模型，与同尺寸模型相比都具有较好的效果表现。

训练策略

采用课程学习贯穿全流程的后训练方案，循序渐进提升模型效果。

- 微调阶段：将多任务数据集进行难度划分（根据模型推理正误比率判断），首先使用中低难度冷启动微调，然后使用RFT方式筛选中高难度数据进行持续微调进行效果提升；
- 强化学习阶段：首先对数理逻辑、代码能力进行提升，采用难度渐进式课程学习方案进行能力强化；然后，基于指令遵循、安全、幻觉、Function Call等10多种混合通用任务进行持续强化，全面提升模型效果；



模型版本	下载链接
T1-35B	modelscope
T1-115B	modelscope

效果评测

模型		MATH-500	AlignBench	BF
OpenAI o1-mini		90	7.91	-
DeepSeek-R1-Distill-Qwen-32B		94.3	7.42	76
QWQ-32B		96	7.97	83
Qwen3-32B（长推理）		93	8.27	86
T1-35B		90	7.93	80
T1-115B		94	8.22	83

模型推理

Transformers

T1 系列模型支持使用 `transformers` 库进行推理，示例如下：

```
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer,

tokenizer = AutoTokenizer.from_pretrained("T1/T1-35B", trust_r
model = AutoModelForCausalLM.from_pretrained(
    "T1/T1-35B",
    trust_remote_code=True,
    torch_dtype=torch.bfloat16,
    device_map="auto"
)
prompt = "生抽和酱油的区别是什么？"
messages = [{"role": "user", "content": prompt}]
text = tokenizer.apply_chat_template(messages,
    tokenize=False,
    add_generation_prompt=True
)
model_inputs = tokenizer([text], return_tensors="pt").to(model
generated_ids = model.generate(
    **model_inputs
)
generated_ids = [
    output_ids[len(input_ids):] for input_ids, output_ids in z
]
response = tokenizer.batch_decode(generated_ids, skip_special_
print(response)
```

► 推理结果

ModelScope

T1 系列模型支持使用 ModelScope 推理，示例如下：





```
from modelscope import AutoModelForCausalLM, AutoTokenizer, GenerationConfig
tokenizer = AutoTokenizer.from_pretrained('T1/T1-35BB', trust_remote_code=True)
model = AutoModelForCausalLM.from_pretrained('T1/T1-35BB', trust_remote_code=True, torch_dtype=torch.bfloat16)

prompt = "生抽与老抽的区别？"
messages = [{"role": "user", "content": prompt}]
text = tokenizer.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)
model_inputs = tokenizer([text], return_tensors="pt").to(model.device)
generated_ids = model.generate(model_inputs)

generated_ids = [
    output_ids[len(input_ids):] for input_ids, output_ids in zip(model_inputs.sequences_ids, generated_ids)
]
response = tokenizer.batch_decode(generated_ids, skip_special_tokens=True)
print(response)
```

vLLM 推理

T1 支持使用 vLLM 进行部署与推理加速，示例如下：

离线推理

```
from transformers import AutoTokenizer
from vllm import LLM, SamplingParams

tokenizer = AutoTokenizer.from_pretrained("T1/T1-35B", trust_remote_code=True)
sampling_params = SamplingParams(temperature=0.6, repetition_penalty=1.1, max_tokens=1024)
llm = LLM(model="T1/T1-35B", trust_remote_code=True, tensor_parallel_size=8)

prompt = "生抽和酱油的区别是什么？"
messages = [{"role": "user", "content": prompt}]
text = tokenizer.apply_chat_template(
    messages,
    tokenize=False,
    add_generation_prompt=True
)

outputs = llm.generate([text], sampling_params)
for output in outputs:
    prompt = output.prompt
    generated_text = output.outputs[0].text
    print(f"Prompt: {prompt!r}, Generated text: {generated_text!r}")
```

OpenAI 兼容的 API 服务

您可以借助 vLLM，构建一个与 OpenAI API 兼容的 API 服务。请按照以下步骤操作：

```
vllm serve T1/T1-35B \
  --trust-remote-code \
  --dtype bfloat16 \
  --disable-custom-all-reduce
```

然后，您可以与 T1 进行对话：

```
from openai import OpenAI
openai_api_key = "EMPTY"
openai_api_base = "http://localhost:8000/v1"
```





```
messages=[
    {"role": "user", "content": "生抽和酱油的区别是什么？"},
],
temperature=0.6,
max_tokens=8192,
extra_body={
    "repetition_penalty": 1.05,
    "skip_special_tokens": False,
    "spaces_between_special_tokens": False,
},
)
print("Chat response:", chat_response)
```

推理注意事项

- 1. T1 系列模型在 chat template 中加入了一些适配复杂推理模型的特
 - T1 系列模型在 chat template 中加入了 <think>\n 符号以确保过程。如果借助 transformers 库推理，并采用 apply_chat_t generation_prompt 设为 True，则将会在推理时自动拼接 <t vLLM 库推理，也会自动在推理起始拼接 <think>\n 符号。此的 <think>\n 符号。
 - T1 系列模型在进行多轮推理时不应传入之前轮次回答中的 <t 在 chat template 中已经实现了对多轮历史信息的自动处理。
- 2. T1 系列模型推理参数选择
 - 在推理数学、代码任务时，建议使用 repetition_penalty=1.0 p=0.95 的推理设置。
 - 在推理通用任务时，建议使用 repetition_penalty=1.05, temp 5 的推理设置，可以有效减少重复生成现象。

国产化适配

T1系列模型均进行了**国产化算力适配**，具体信息可见

- 1. [MindSpore-Lab/T1-35B](#)
- 2. [MindSpore-Lab/T1-115B](#)

声明、引用

声明

我们在此声明，不要使用 T1 系列模型及其衍生模型进行任何危害国家动。同时，我们也要求使用者不要将 T1 系列模型用于没有安全审查和们希望所有使用者遵守上述原则，确保科技发展在合法合规的环境下进

我们已经尽我们所能，来确保模型训练过程中使用的数据的合规性。然了巨大的努力，但由于模型和数据的复杂性，仍有可能存在一些无法预由于使用 T1 系列开源模型而导致的任何问题，包括但不限于数据安全或模型被误导、滥用、传播或不当利用所带来的任何风险和问题，我们

引用

如需引用我们的工作，请使用如下 reference：





```
author={Zihan Wang and Xinzhang Liu and Yitong Yao and C
year={2025},
eprint={2507.18013},
archivePrefix={arXiv},
primaryClass={cs.CL},
url={https://arxiv.org/abs/2507.18013},
}
```



北京奥思研人工智能科技有限公司版权所有

- | | | | |
|------------|------------|------------|------|
| Git 大全 | Gitee 封面人 | OpenAPI | 关于我们 |
| Git 命令学习 | 物 | MCP Server | 加入我们 |
| CopyCat 代码 | GVP 项目 | 帮助文档 | 使用条款 |
| 克隆检测 | Gitee 博客 | 在线自助服务 | 意见建议 |
| APP与插件下载 | Gitee 公益计划 | 更新日志 | 合作伙伴 |
| | Gitee 持续集成 | | |

-  client@oschina.cn
-  企业版在线使用: 400-606-0201
-  专业版私有部署: 13670252304
- 13352947997



技术交流QQ群




微信服务号

 开放原子开源基金会 合作代码托管平台

 违法和不良信息举报中心

京ICP备2025119063号

 简体 / 繁體 / English

