基于SAE的金融LLM可解释性研究

# 技术路线图



# 语料库的选择

## 东方财富

FinGPT，中文语料库的使用，[open-compass/OpenFinData (github.com)](https://github.com/open-compass/OpenFinData)

OpenFinData是由东方财富与上海人工智能实验室联合发布的开源金融评测数据集。该数据集代表了最真实的产业场景需求，是目前场景最全、专业性最深的金融评测数据集。它基于东方财富实际金融业务的多样化丰富场景，旨在为金融科技领域的研究者和开发者提供一个高质量的数据资源。

优点：数据集模块多，缺点：数量少，为json文件，压缩大小为400KB，总共为1500条数据记录

## 轩辕开源

[Duxiaoman-DI/FinCorpus · Datasets at Hugging Face](https://huggingface.co/datasets/Duxiaoman-DI/FinCorpus)

中文金融数据集FinCorpus，包含上市公司公告、金融资讯、金融新闻和金融试题四个模块



## 最终选择

Anthropic在进行大模型可解释性分析时，使用The Pile数据集，并使用四千万的tokens来生成八十亿的数据集。所有在金融大模型领域进行可解释时，其金融语料库的内容需要丰富并具大规模的特点。

其中轩辕大模型开源的FinCorpus语料库，其数据丰富，包括上市公司公告、金融资讯、金融新闻以及金融试题，满足构建用于SAE训练的数据集的要求。

# 金融大模型的选择

## 基于GPT2微调

MLP层神经元为768个。

## Fingpt

基于ChatGLM2-6B进行微调。其MLP层神经元为4096个

## 轩辕

从头开始训练，和Fingpt类似的结构。

## 最终模型的选取

需要考虑进行大模型可解释性分析时，模型结构对显存占用的消耗。

# SAE数据集的生成

如何生成用于训练SAE的数据集。

# SAE用于可解释性分析



## 占用显存分析

### SAE模型占用显存



M=神经元数量，N=特征数量，F=数据精度。

总的显存占用=(M \* B \* F + M \* N \* F \* 2 + M \* B \* F + N \* B \* F + M \* F + N \* F)/(1024\*\*3)GB

之下表格为B=1024，F=4时的情况。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| M/N | 2\*\*12(4K) | 2\*\*15(32K) | 2\*\*17(131K) | 2\*\*20(1M) | 2\*\*22(4M) | 2\*\*25(34M) |
| 512 | 0.03 | 0.25 | 1.00 | 8.00 |  |  |
| 768 | 0.04 | 0.31 | 1.25 | 10.00 | 40.02 | 320.13 |
| 1024 | 0.05 | 0.38 | 1.50 | 12.01 | 48.02 | 384.13 |
| 4096 | \ | 1.15 | 4.53 | 36.03 | 144.04 | 1152.15 |

### 数据shuffle占用显存

主要问题：

1. 目前没找到哪个变量影响buffer\_count
2. shuffle过程能否独立出来，并转入内存中，而不是显存。

buffer\_size=store\_batch\_size\_prompts \* (n\_batches\_in\_buffer / 2) \* context\_size \* d\_model

buffer\_count=?。

当d\_model为4096，context\_size为512，n\_batches\_in\_buffer为64，store\_batch\_size\_prompts为16，float32精度时，buffer\_size的大小为4GB。

当d\_model为4096，context\_size为256，n\_batches\_in\_buffer为32，store\_batch\_size\_prompts为8，float16精度时，buffer\_size的大小为0.25GB。

n\_batches\_in\_buffer为8，store\_batch\_size\_prompts为8，d\_model为768，context\_size为512，float32精度时：





### 实际消耗

n\_batches\_in\_buffer简称为nbib，expansion\_factor简称为ef

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| model | 大小 | d\_model | ef | 激活 | dtype | batchsize | nbib | GPU |
| tiny-stories-1L-21M | 13M | 1024 | 16 | Relu | float32 | 4096 | 64 | 13.2 |
| 13M | 1024 | 16 | topk | float32 | 4096 | 64 | 13.6 |
| 13M | 1024 | 16 | Relu | float16 | 4096 | 64 |  |
| 13M | 1024 | 16 | topk | float16 | 4096 | 64 | 9.7 |
| 13M | 1024 | 16 | topk | float16 会导致loss变为nan | 1024 | 64 | 9.8 |
| 13M | 1024 | 16 | topk | float32 | 4096 | 32 | 9.6 |
| gpt2-small | 85M | 768 | 16 | topk | float32 | 4096 | 64 | 18.2 |
| 85M | 768 | 16 | topk | float32 | 4096 | 32 | 14.1 |
| 85M | 768 | 16 | topk | float32 | 1024 | 64 | 18.3 |

## SAE的变体

### gated SAE

### TopK SAE

## SAE评价指标分析[1]

如何评估SAE，目前存在L0和Loss Recovered两个指标 。L0是SAE编码中非零元素的平均数。Loss Recovered是用重建的激活替换GPT或者其他大模型的原始激活并测量不完美重建的额外损失。这两个指标之间通常存在跷跷板效应，需要进行权衡。毕竟SAE会选择降低重建精度以增加稀疏性。

1. 手动检查：这些特征是否看起来可解释？
2. 特征密度：我们发现“实时”特征的数量和它们触发的标记百分比是极其有用的指南。（详情请参阅附录。）
3. 重构损失：自动编码器如何很好地重构 MLP 激活？最终目标是解释 MLP 层的功能，因此 MSE 损失应该很低。
4. 玩具模型：拥有玩具模型，可以在其中了解基本事实，从而可以清晰地评估自动编码器的性能。

# 特征可视化

## github开源代码1：

[GitHub - callummcdougall/sae\_vis: Create feature-centric and prompt-centric visualizations for sparse autoencoders (like those from Anthropic&apos;s published research).](https://github.com/callummcdougall/sae_vis)

[文件夹 - Google 云端硬盘](https://drive.google.com/drive/folders/1sAF3Yv6NjVSjo4wu2Tmu8kMh8it6vhIb)

包含feature-centric的可视化，复现Anthropic的可视化，即我们查看一个特定feature并查看哪些tokens在该功能上触发最强。

以及以prompt-centric的可视化，查看一次特定prompt并根据各种不同的指标查看哪些feature在该prompt上触发最强。

# 引导

# 参考文献

1. Bricken T, Templeton A, Batson J, et al. Towards monosemanticity: Decomposing language models with dictionary learning[J]. Transformer Circuits Thread, 2023, 2.