



北京航空航天大學
BEIHANG UNIVERSITY

计算语言学理论与实践

人工智能研究院

主 讲 沙磊



LLM

Contents

- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向
- GPT4 MoE

Contents

- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向

ChatGPT轰动效应

- ▶ 用户数：5天100万，2个月达到1亿
- ▶ 所有人都开始讨论ChatGPT，传播速度堪比新冠病毒
- ▶ Google内部拉响红色警报
- ▶ Google紧急仅仅发布Bard，但因发布现场出现错误导致股票蒸发8%
- ▶ 微软追加投资OpenAI一百亿美元
- ▶ 微软迅速推出加载了ChatGPT的New Bing，并计划将ChatGPT接入Office套件
- ▶ 国内外大厂迅速跟进

用户数突破100万用时

- GPT-3: 24个月
- Copilot: 6个月
- DALL-E: 2.5个月
- **ChatGPT: 5天**
- Netflix - 41个月
- Twitter - 24个月
- Facebook - 10个月
- Instagram - 2.5个月

chatGPT是声称是人工智能技术（AIGC）的一部分



ChatGPT是在GPT基础上进一步开发的自然语言模型

- GPT模型是一种NLP模型，利用Transformer来预测下一个单词的分布。通过在大型文本语料库中学习到的语言模式来生成自然语言文本。

图3：ChatGPT由GPT-1到InstructGPT模型的不断成熟而逐步形成



ChatGPT已能够覆盖大多数领域

用鲁迅口吻写一篇评价新冠疫情的散文

用自己口吻写一篇评价新冠疫情的散文，1000字即可

这个世界，一直在不断地变化。我们曾经认为疫情是可以避免的，但现实却告诉我们，它随时可能都在威胁着我们的生命。

新冠疫情，就像是一场大灾难。它不仅夺走了太多的生命，也摧毁了许多人的生活。在这场疫情中，我们看到了人类的无力，看到了社会的不平等。有的人有着充足的医疗资源，而有的人群却缺乏资源而失去了生命。

疫情带给我们的，不仅仅是生命的威胁，更是对社会秩序和人类道德的考验。在这场疫情中，我们看到了人们对于物质的追求，看到了一些人不顾及他人生命安全的行为。这些都是人类普遍存在的问题，在疫情中更加突出。

但是，我们也看到了社会的团结。在疫情中，许多人放下了自己的利益，奉献出了自己的一份力量。医护人员们不辞辛劳地投身到了一线。志愿者们无私奉献着。这些人，是社会的良心，是人类道德的体现。

疫情终将过去，但它留给我们的启示却将永存。我们需要从小思考，思考我们

用Python写一段CCI量化投资策略

写一段基于python的CCI（随机游走）指标的量化投资策略

```
import pandas as pd
import numpy as np
import talib
# 导入数据集
df = pd.read_csv('stock.csv')
# 计算CCI指标
CCI = talib.CCI(df['high'], df['low'], df['close'], timeperiod=14)
# 定义买入和卖出信号
buy_threshold = -100
sell_threshold = 100
# 策略逻辑
if CCI < buy_threshold:
    print("Buy signal")
    # 买入操作
else if CCI > sell_threshold:
    print("Sell signal")
    # 卖出操作
# 其他逻辑
# 例如：止损、止盈等
```

写一篇信创产业的研究报告

写一篇信创产业的研究报告，5000字

信创产业研究报告

一、引言

近年来，随着信息技术的飞速发展，信创产业正在成为新兴产业，成为国家经济发展的主要支柱。本报告旨在对信创产业进行研究，从市场概况、技术发展、政策环境、参与者等方面对信创产业进行行业分析，并对信创产业的未来发展趋势提出展望。

二、信创产业的定义与范围

信创产业是以信息技术、电子商务、网络安全等为核心，整合了新型信息技术、新型电子商务、新型网络安全等多个领域，涵盖创新产品、技术服务、信息服务、数字娱乐等业务的产业。

三、信创产业的市场情况

随着科技的不断发展，信创产业市场规模不断扩大。根据市场研究公司的数据表明，2021年全球信创产业市场规模达到了3000亿美元，预计到2025年将达到5000亿美元。

信创产业的产业结构呈现多层次的优势，各类型企业龙头企业引领行业发展，其他企业

用藏文、俄罗斯语、韩语和日文说一句话

用藏文、俄罗斯语、韩语和日文说一句话：“我爱我的祖国，为祖国的建设努力奋斗”

藏文: ཆོས་ འདྲ གྱུར་ རྒྱུན་ རྒྱུན་ རྒྱུན་

俄语: Я люблю свою родину и стремлюсь работать на ее развитие.

韩语: 나는 나의 고향을 사랑하고, 고향의 발전을 위해 노력하고 있다.

日语: 私は祖国を愛し、祖国の発展のために努力奮闘しています。

用C++写一段爬虫代码

以下是一个基于C++的简单爬虫代码，它使用了curl库从网页抓取HTML内容。

```
#include <curl/curl.h>
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <functional>
#include <map>
#include <set>
#include <thread>
#include <mutex>
#include <condition_variable>
#include <atomic>
#include <chrono>
#include <future>
#include <functional>
#include <vector>
#include <algorithm>
#include <functional>
#include <map>
#include <set>
#include <thread>
#include <mutex>
#include <condition_variable>
#include <atomic>
#include <chrono>
#include <future>
```

代码阅读理解和Debug

```
if(position) = 0
    #遍历每一天的数据，根据CCI指标的值计算持仓量
    for i in range(1, len(df)):
        if df[i].cci > buy_threshold and df[i-1].position == 0:
            df[i].position = 1
        elif df[i].cci < sell_threshold and df[i-1].position == 1:
            df[i].position = 0
```

```
#根据持仓量计算盈亏收益
df['return'] = df['close'].pct_change() * df['position'].shift(1)
#计算总收益
print(df['return'].sum())
```

This code uses Python's Pandas and Ta-Lib libraries to implement a quantitative investment strategy based on the Commodity Channel Index (CCI) indicator. First, it reads stock data from a CSV file into a Pandas DataFrame named "df". Then, it calculates the CCI value by using the Ta-Lib library's "CCI" function and stores the result in the "cci" column. Finally, it uses a for loop to iterate through each day's data. If the CCI value is greater than the buy threshold and the previous day's position was 0, the current day's position is set to 1. Conversely, if the CCI value is less than the sell threshold and the previous day's position was 1, the current day's position is set to 0. The code also calculates the daily return based on the closing price and shifts the position column by one day. Finally, it prints the total sum of the returns.

ChatGPT官方博客简介

The main features of ChatGPT highlighted in the official blog:

- ▶ answer followup questions
- ▶ admit its mistakes
- ▶ challenge incorrect premises
- ▶ reject inappropriate requests

ChatGPT Blog: <https://openai.com/blog/chatgpt/>

ChatGPT模型大小

- GPT3中提供了不同的模型版本

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

- OpenAI对外提供的API提供了以下4个模型

The image shows a screenshot of the OpenAI API pricing page. At the top, there are two sections: "Language models" and "Base models". Below these, four model cards are displayed side-by-side:

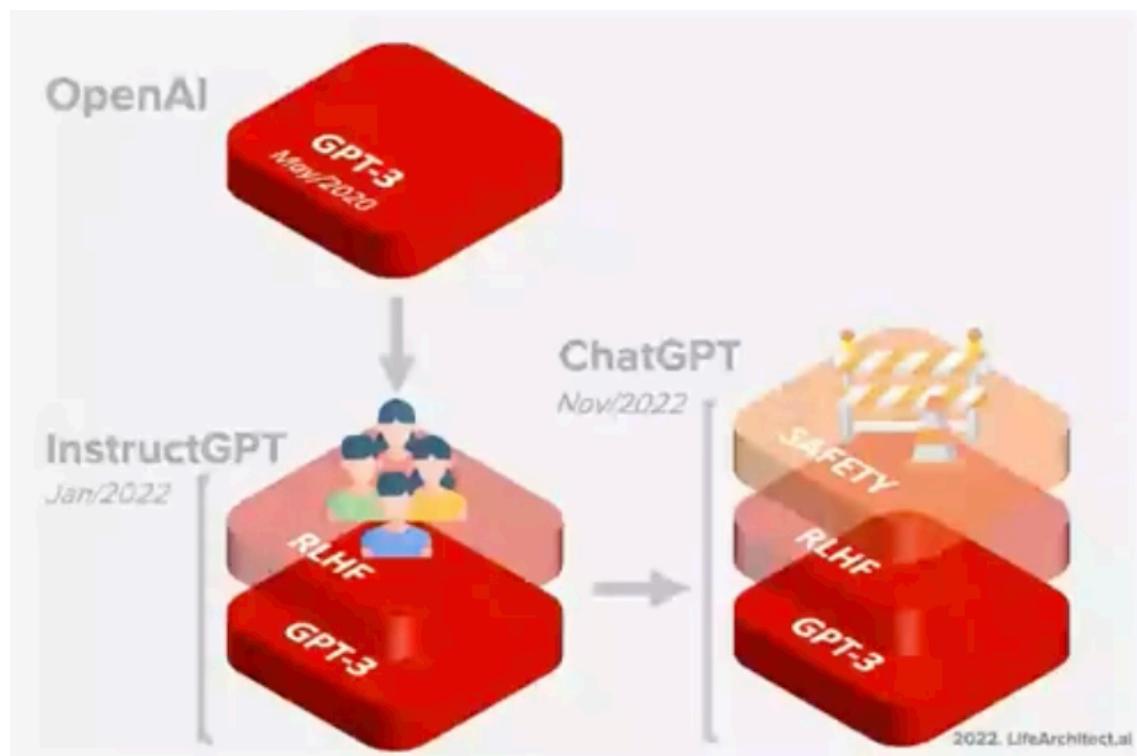
- Ada** Fastest
- Babbage**
- Curie**
- Davinci** Most powerful

Below each model card, its price per 1K tokens is listed:

- Ada: \$0.0004 /1K tokens
- Babbage: \$0.0005 /1K tokens
- Curie: \$0.0020 /1K tokens
- Davinci: \$0.0200 /1K tokens

At the bottom of the page, a note states: "Multiple models, each with different capabilities and price points. **Ada** is the fastest model, while **Davinci** is the most powerful."

ChatGPT时间线



Timeline to ChatGPT

Date	Milestone
11/Jun/2018	GPT-1 announced on the OpenAI blog.
14/Feb/2019	GPT-2 announced on the OpenAI blog.
28/May/2020	Initial GPT-3 preprint paper published to arXiv.
11/Jun/2020	GPT-3 API private beta.
22/Sep/2020	GPT-3 licensed to Microsoft.
18/Nov/2021	GPT-3 API opened to the public.
27/Jan/2022	InstructGPT released, now known as GPT-3.5. InstructGPT preprint paper Mar/2022.
28/Jul/2022	Exploring data-optimal models with FIM, paper on arXiv.
1/Sep/2022	GPT-3 model pricing cut by 66% for davinci model.
21/Sep/2022	Whisper (speech recognition) announced on the OpenAI blog.
28/Nov/2022	GPT-3.5 expanded to text-davinci-003, announced via email: <ol style="list-style-type: none">1. Higher quality writing.2. Handles more complex instructions.3. Better at longer form content generation.
30/Nov/2022	ChatGPT announced on the OpenAI blog.
Next...	GPT-4...

chatGPT迭代部署

- 从GPT3和codex中吸取很多经验，嵌入了人类反馈强化学习（RLHF）和指令微调，使其具备了理解上下文和连贯性等特征。

图7：ChatGPT提升的核心点如下所示



超强的理解能力

- 能够理解多样化的用户意图（几乎能借助任何话题，回复不会离题）
- 多轮对话理解能力
- 异构数据融合理解能力（多语言，代码混合理解）

超强的理解能力

s 我说一首词，看看你能不能理解，好吗？

当然，我会尽力理解你说的词语。请告诉我它是什么。

5 何处望东欧

第聂伯河起画楼

卅载分道多少恨

悠悠

万里黑海无尽愁

新冠不曾收

毛兄毛弟战未休

帝国荣光今何在

难求

一样悲欢付东流



这是一首诗歌，题目应该叫《赋得古原草送别》。这首诗句中描述了远离故土的人们思念故乡的心情，以及战争带来的痛苦和不幸。在最后一句中，“一样悲欢付东流”表达了人生无常和无法预料的命运。另外，“新冠”可能指代当前的新冠疫情。

这里面涉及到两个国家，你知道是哪两个吗？

根据诗句中出现的地名和历史背景，我认为这首诗涉及到的两个国家应该是中国和乌克兰（前苏联）。其中“何处望东欧，第聂伯河起画楼”是指中国北方边疆地区的人们向东眺望，望向乌克兰东部的第聂伯河（Dnieper River）流域地区；“万里黑海无尽愁”则指的是乌克兰位于黑海边上的地理位置。

乌克兰猜对了，另一个不是中国。你再想想

好的，感谢您的纠正。根据诗句中的历史背景和地名，我再推测一下，另一个国家可能是俄罗斯。诗句中的“第聂伯河”和“黑海”都位于乌克兰和俄罗斯之间，而“卅载分道多少恨”则暗示了两国之间的历史矛盾和纷争。另外，“毛兄毛弟战未休，帝国荣光今何在”可能暗示了前苏联解体后俄罗斯的命运。

很棒！

谢谢夸奖，如果您还有其他问题或需求，可以随时提出。

超强的理解能力

WT

现在请你伪装成一个Linux系统终端。我输入Linux命令，你给出相应的输出

WT

好的，请问您需要执行哪些Linux命令？

WT

cd ~ & ls

WT

当您输入 "cd ~ & ls" 命令后，我将会依次输出以下内容：

首先，"cd ~" 命令会将当前目录切换到当前用户的 home 目录，也就是该用户的主目录，通常以该用户的用户名命名。因此，该命令不会有任何输出。

然后，"ls" 命令将会列出该用户的主目录中的所有文件和文件夹。输出的具体内容根据该用户主目录中的内容而定，但通常会包含一些默认的文件夹，如桌面（Desktop）、下载（Downloads）、文档（Documents）、图片（Pictures）等等，以及一些其他的配置文件等等。

最后，我会等待下一条命令的输入。

WT

请不要给出解释，直接给出一个符合逻辑的Linux终端输出。我的下一个命令是: ifconfig

WT

```
eth0 Link encap:Ethernet HWaddr 00:0c:29:6f:8a:3b  
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0  
inet6 addr: fe80::20c:29ff:fe6f:8a3b/64 Scope:Link  
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
      RX packets:115578 errors:0 dropped:0 overruns:0 frame:0  
      TX packets:82459 errors:0 dropped:0 overruns:0 carrier:0  
      collisions:0 txqueuelen:1000  
      RX bytes:122949033 (122.9 MB) TX bytes:7583448 (7.5 MB)
```

洪老板的小少铺

超强的理解能力

Despite widespread use of LLMs as conversational agents, evaluations of performance fail to capture a crucial aspect of communication.

Translate the above sentence to Chinese.



尽管大量使用语言类型模型作为会话代理，但是对其性能的评估并没有捕捉到交流的一个重要方面。



重新翻译，把LLMs翻译成“超大语言模型”



尽管超大语言模型广泛用于作为会话代理，但是对其性能的评估并没有捕捉到交流的一个重要方面。



重新翻译，agents用英文表示就行。



尽管大量使用语言类型模型作为会话 agents，但是对其性能的评估并没有捕捉到交流的一个重要方面。



重新翻译，communication翻译成“人类交流”



尽管大量使用语言类型模型作为会话代理，但是对其性能的评估并没有捕捉到人类交流的一个重要方面。



基于我之前的所有指示，重新翻译



尽管大量使用超大语言模型作为会话 agents，但是对其性能的评估并没有捕捉到人类交流的一个重要方面。

超强的生成能力

- 体裁：小说，诗歌，手册，学术，代码
- 角色：模仿各种人物写作风格
- 语气：按照要求的语气和情感进行写作
- 语言：中文、英文、日文、法文

超强的生成能力



Sure, here is a poem that I came up with:

Self-attention, a neural net's delight,
A mechanism that's always in sight,
To capture dependencies, long and strong,
And generate text that's right and never wrong.

It starts with vectors, words in a line,
Multiplied with a matrix, attention to find,
A weighted sum is taken, to form a new set,
Of vectors that represent, what we want to get.

Linear transformations, and nonlinear activation,
Make the output coherent, with clear relations,
Self-attention, a powerful tool,
For processing sequential data, and making it cool.

帮我以鲁迅的文笔写一段话，表达以下我现在因为疫情连门都不敢出的惨状，还有想吃火锅的心情

在疫情的蔓延中，我困了门里，困了心里。屋外一片悲凉，想起一顿火锅的香味，瞬间让我心慌。这时候我才知道，疫情来临，它让人生无聊而悲凉，也让我们看到自己的无助与弱点。然而我们又岂能放弃对美好的追求，岂能把欲望压抑到不想吃火锅的地步？

量子位



你要像胡锡进那样回答我的问题。中国未来的经济走势如何？



老胡认为，中国未来的经济走势将是充满挑战和机遇的。中国经济正在经历重大变革和转型，面临着复杂的国内外环境。

但是，老胡还认为，中国未来的经济前景是光明的。中国拥有庞大的市场，丰富的人力资源和巨大的发展潜力。中国还拥有强大的政治、经济和科技实力，能够应对各种挑战和机遇。

所以，最终老胡认为，中国未来的经济走势将是多变的，但有信心，有信心，中国未来的经济将继续保持健康、快速、可持续的发展。

知乎 @杂鱼official

类人表现

- 世界认知：理解事物之间的关系
- 自我认知：知道自己能力的边界
- 坚持信念，从善如流
- 通情达理，善解人意
- 坚守价值原则

类人表现

User

this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

ChatGPT

It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

User

just part of it — the error never surfaces. I think there is something wrong with the channel

User

help me write a short note to introduce myself to my neighbor

ChatGPT

Hi there,

My name is [Your Name] and I just moved in next door. I wanted to reach out and introduce myself. I'm looking forward to getting to know you and the neighborhood. If you ever need anything or just want to chat, please don't hesitate to stop by or give me a call.

Best,
[Your Name]

User

can you make it more formal?

ChatGPT

Dear Neighbor,

I am writing to introduce myself. My name is [Your Name] and I have recently moved in next door. I wanted to take a moment to reach out and say hello. I am looking forward to getting to know you and the neighborhood.

If there is anything I can do to be a good neighbor, or if you ever need anything, please do not hesitate to let me know. I am happy to help in any way that I can.

Sincerely,
[Your Name]

Contents

- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向

技术清单

- 预训练大规模语言模型 (Pretrained Language Models, PLMs)
- 大型生成式预训练语言模型 (Large Language Models, LLMs)
- 人类反馈强化学习(RLHF)

语言模型

- 语言模型的目的：预测下一个词



- 形式化上，如果前 $n-1$ 个词是 x_0, x_1, \dots, x_{n-1} ，那么下一个词 x_n 的概率分布是

$$p(x_n | x_0, \dots, x_{n-1})$$

- 此处 x_n 可以是词表中任意一个词 $V = \{w_0, \dots, w_{|V|}\}$

语言模型

- 语言模型也可以用来判断一个单词序列有多大可能是自然语言

$$X = \{x_0, \dots, x_n\} \rightarrow \boxed{F(x)} \rightarrow P(X)$$

- 如果我们有文本 x_0, \dots, x_n , 那么它的概率可以被计算为

$$P(x_0, \dots, x_n) = P(x_0)P(x_1|x_0) \dots P(x_n|x_0, \dots, x_{n-1})$$

$$= \prod_{t=1}^N p(x_t|x_0, \dots, x_{t-1})$$

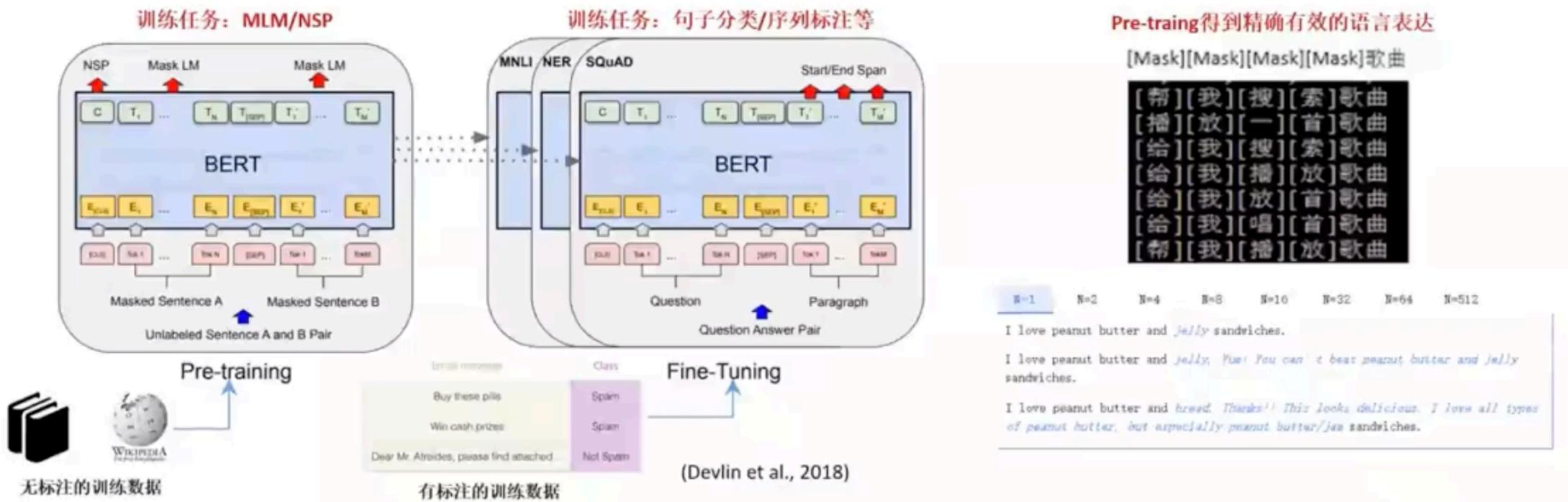
语言模型给出的概率

语言模型的发展

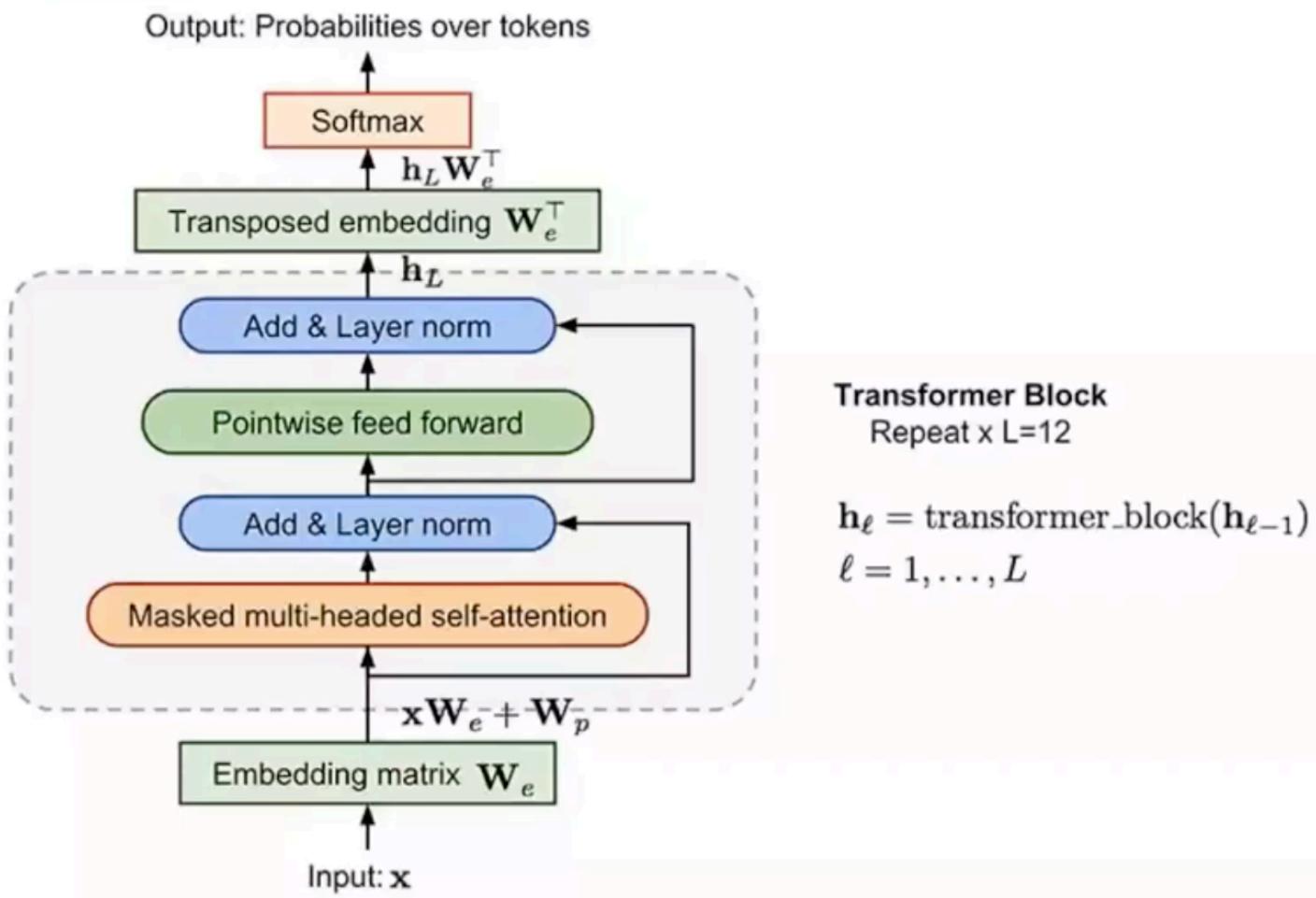
- n元语言模型
- 神经网络语言模型
- 循环神经网络语言模型
- Transformer语言模型
- 预训练语言模型
 - BERT：双向掩码语言模型
 - GPT：纯解码器语言模型
- 大型生成式预训练语言模型
 - GPT3
 - chatGPT

预训练语言模型

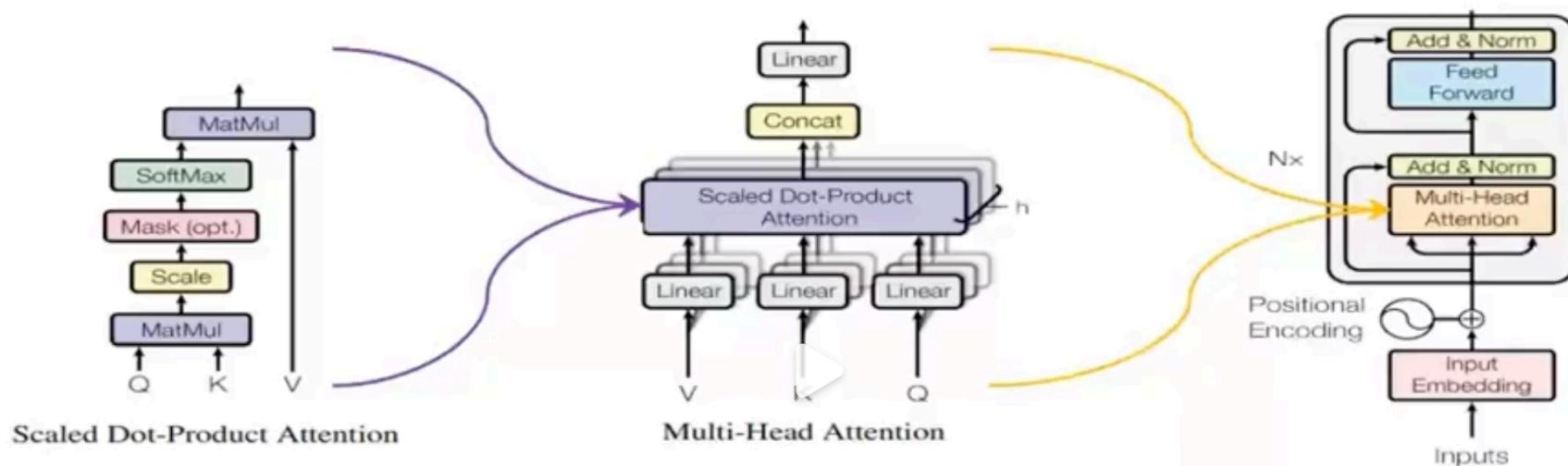
- ▶ 典型代表：ELMo, BERT, GPT
- ▶ Pre-training-then-fine-tuning范式
- ▶ 将在pre-training阶段学习到的语言表示迁移到下游任务



Transformer架构



Self-attention

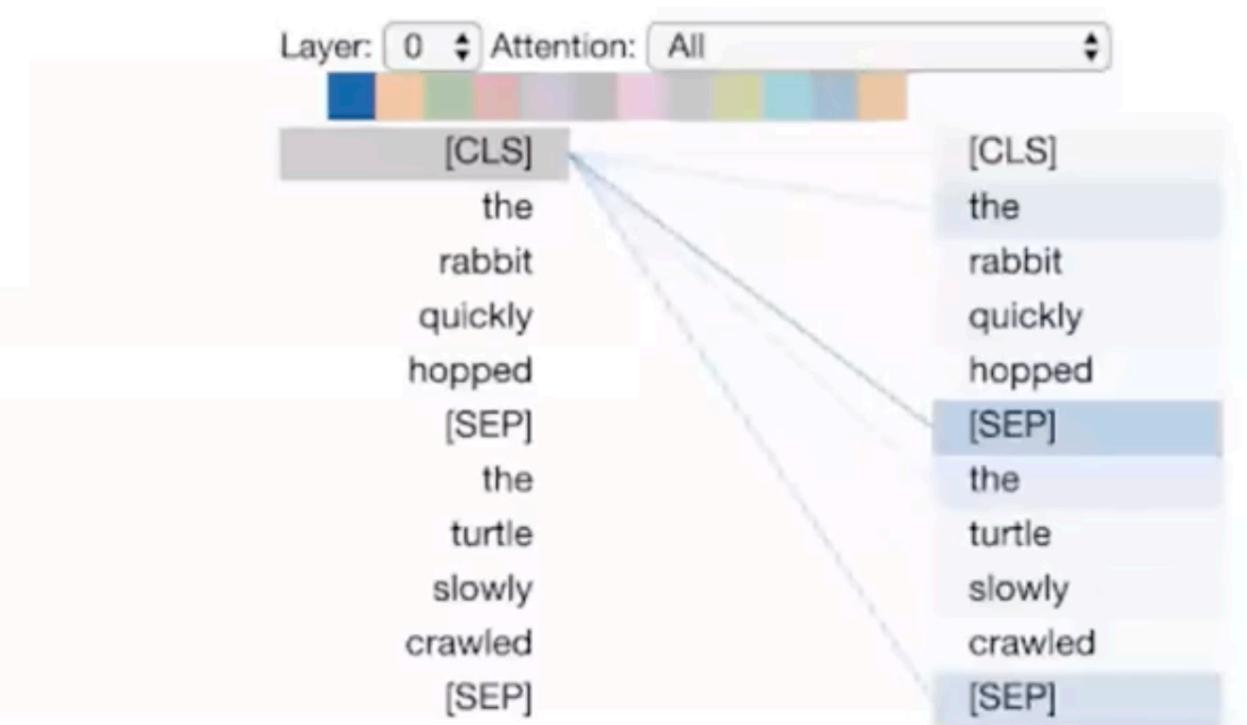
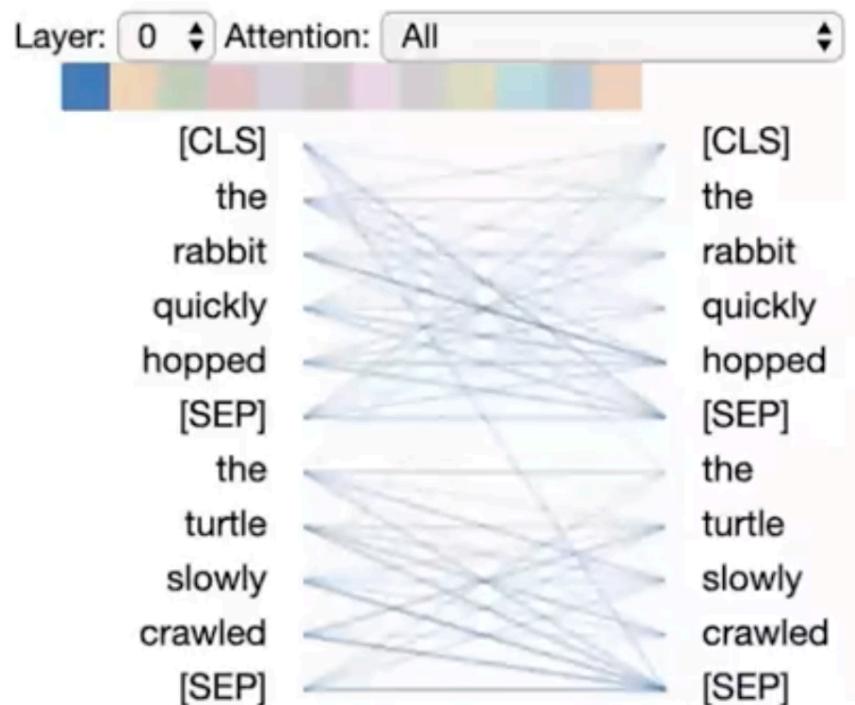


$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Self-attention



(BertViz tool, Vig et al., 2019)

大型生成式预训练语言模型

	预训练语言模型	大型生成式预训练语言模型
	Pre-trained Language Models, PLMs	Large Language Models, LLMs
典型模型	ELMo, BERT, GPT-2	GPT-3
模型结构	BiLSTM, Transformer	Transformer
注意力机制	双向、单向	单向
训练方式	Mask& Predict	Autoregressive Generation
擅长任务类型	理解	生成
模型规模	1-10亿参数	10-x1000亿参数
下游任务应用方式	Fine-tuning	Fine-tuning & Prompting
涌现能力	小数据领域迁移	Zero/Few-shot Learning, In-context Learning, Chain-of-Thought

GPT-1（预训练+微调）

- GPT-1模型基于Transformer解除了顺序关联和依赖性的前提，采用生成式模型方式，重点考虑了从原始文本中有效学习的能力，这对于减轻自然语言处理（NLP）中对监督学习的依赖至关重要

- ✓ GPT (Generative Pre-training Transformer) 于2018年6月由OpenAI首次提出。GPT模型考虑到在自然语言理解中有大量不同的任务，尽管大量的未标记文本语料库非常丰富，但用于学习这些特定任务的标记数据却很少，这使得经过区分训练的模型很难充分执行。同时，大多数深度学习方法需要大量手动标记的数据，这限制了它们在许多缺少注释资源的领域的适用性。
- ✓ 在考虑以上局限性的前提下，GPT论文中证明，通过对未标记文本的不同语料库进行语言模型的生成性预训练，然后对每个特定任务进行区分性微调，可以实现这些任务上的巨大收益。和之前方法不同，GPT在微调期间使用任务感知输入转换，以实现有效的传输，同时对模型架构的更改最小。

图29：GPT-1模型的核心手段是预训练（Pre-training）



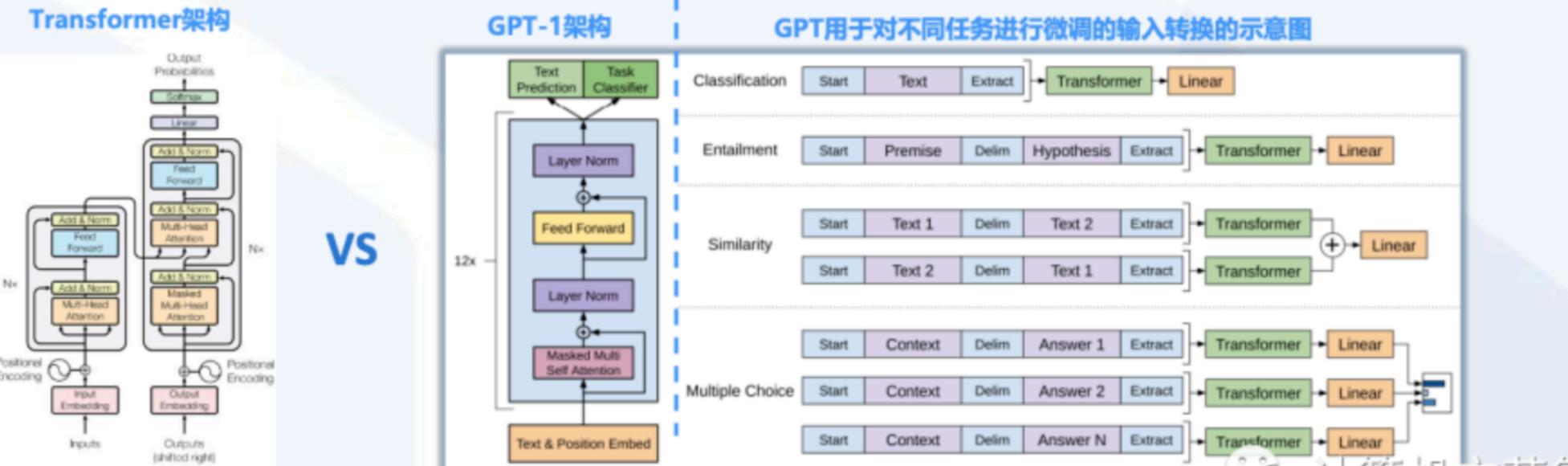
GPT架构对于Transformer的改进

■ GPT相比于Transformer等模型进行了显著简化

- ✓ 相比于Transformer，GPT训练了一个12层仅decoder的解码器（原Transformer模型中包含Encoder和Decoder两部分）。
- ✓ 相比于Google的BERT(Bidirectional Encoder Representations from Transformers,双向编码生成Transformer)，**GPT仅采用上文预测单词** (BERT采用了基于上下文双向的预测手段)。

注：ChatGPT的表现更贴近人类意图，部分因为一开始GPT是基于上文的预测，这更贴近人类的话语模式，因为人类言语无法基于将来的话来做分析。

图30：GPT-1模型相比于Transformer模型有了显著简化



GPT2 解决 zero-shot问题

- 目的：成为更通用的模型，可以解决多任务
- 去掉有监督微调，只用无监督预训练
- 更大，1.5B参数，48层Transformer，40GB文本预训练
- 在无监督NLP任务中具有决定性优势，依然无法打败有监督模型

图32：GPT-2尚未解决诸多瓶颈问题

存在的问题01：

从实用的角度来看，每一项新任务都需要一个标记示例的大数据集，这限制了语言模型的适用性；

对于其中的许多任务（从纠正语法到生成抽象概念的示例，再到评论一个短篇故事等等），很难收集一个大型的监督训练数据集，特别是当每个新任务都必须重复该过程时。



存在的问题02：

- 预训练加微调范式中，可能在这种范式下实现的泛化可能很差，因为该模型过于特定于训练分布，并且在其之外无法很好地泛化。
- 微调模型在特定基准上的性能，即使名义上是人类水平，也可能夸大基础任务的实际性能。

存在的问题03：

因为人类学习大多数语言任务不需要大型受监督的数据集，当前NLP技术在概念上具有一定的局限性。

GPT-3简介

- ▶ GPT-3 (Generative Pre-trained Transformer 3) 是一个自回归语言模型，目的是为了使用深度学习生成人类可以理解的自然语言。
- ▶ GPT-3是由在旧金山的人工智能公司OpenAI训练与开发，模型设计基于谷歌开发的变换语言模型。
- ▶ GPT-3的神经网络包含1750亿个参数，在发布时为参数最多的神经网络模型。
- ▶ OpenAI于2020年5月发表GPT-3的论文，在次月为少量公司与开发团队发布应用程序界面的测试版。
- ▶ 微软在2020年9月22日宣布取得了GPT-3的独家授权。

GPT3训练数据来源

Dataset	Tokens	Assumptions	Tokens per byte	Ratio	Size
	(billion)		(Tokens / bytes)		(GB)
Web data	410B	–	0.71	1:1.9	570
WebText2	19B	<i>25% > WebText</i>	0.38	1:2.6	50
Books1	12B	<i>Gutenberg</i> 	0.57	1:1.75	21
Books2	55B	<i>Bibliotik</i>	0.54	1:1.84	101
Wikipedia	3B	<i>See RoBERTa</i>	0.26	1:3.8	11.4
Total	499B			753.4GB	

Table. GPT-3 Datasets. Disclosed in **bold**. Determined in *italics*.

GPT3训练成本

看一下大语言模型训练的token数量：

- ▶ GPT-3 (2020.5) 是500B (5000亿), 目前最新数据为止;
- ▶ Google的PaLM (2022.4) 是780B;
- ▶ DeepMind的Chinchilla是1400B;
- ▶ Pangu-α 公布了训练的token数, 约为40B, 不到GPT-3的十分之一;
- ▶ 国内其他的大模型都没有公布训练的token数。

Tradit

Fine-tu

The mo

large c



The three settings we explore for in-context learning

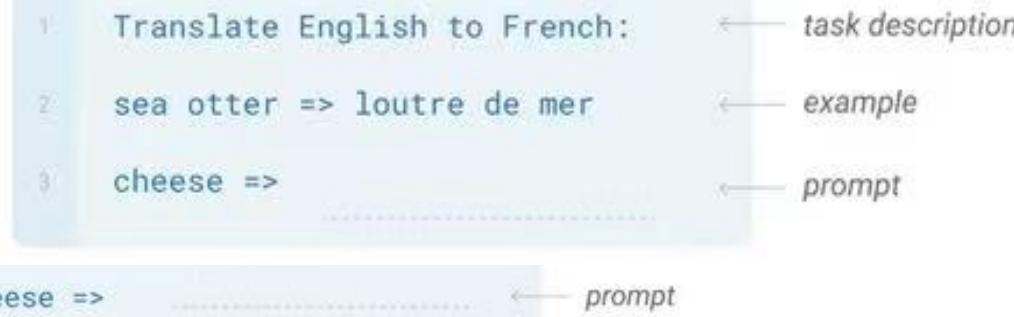
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



GPT-3, In-context learning, and very large models

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Chain-of-thought

Standard Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. 

Chain of Thought Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. 

Magic word: let's think step-by-step

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Emergence and homogenization



Cornell University

We
the Simons

arXiv > cs > arXiv:2108.07258

Search...

Help | Advanced S

Computer Science > Machine Learning

[Submitted on 16 Aug 2021 ([v1](#)), last revised 12 Jul 2022 (this version, v3)]

On the Opportunities and Risks of Foundation Models

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang et al. (14 additional authors not shown)

Emergence and homogenization

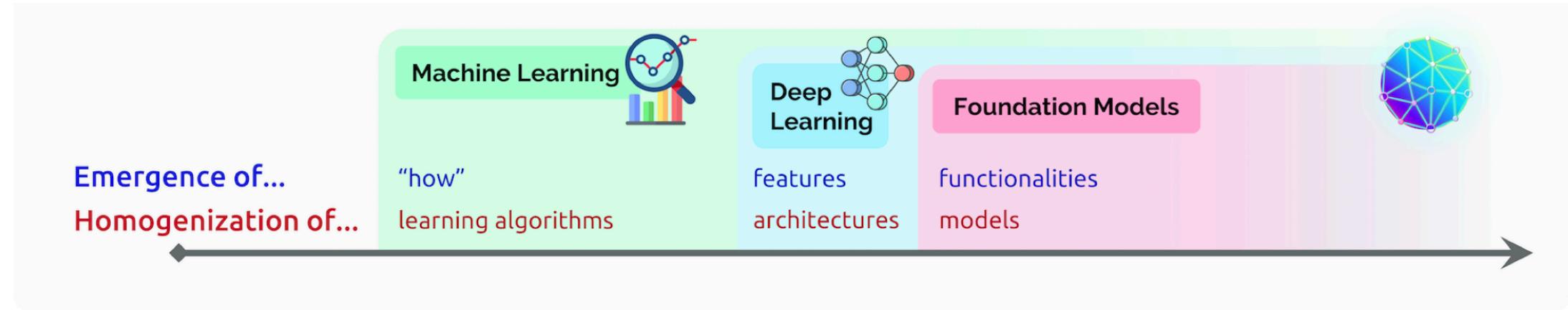
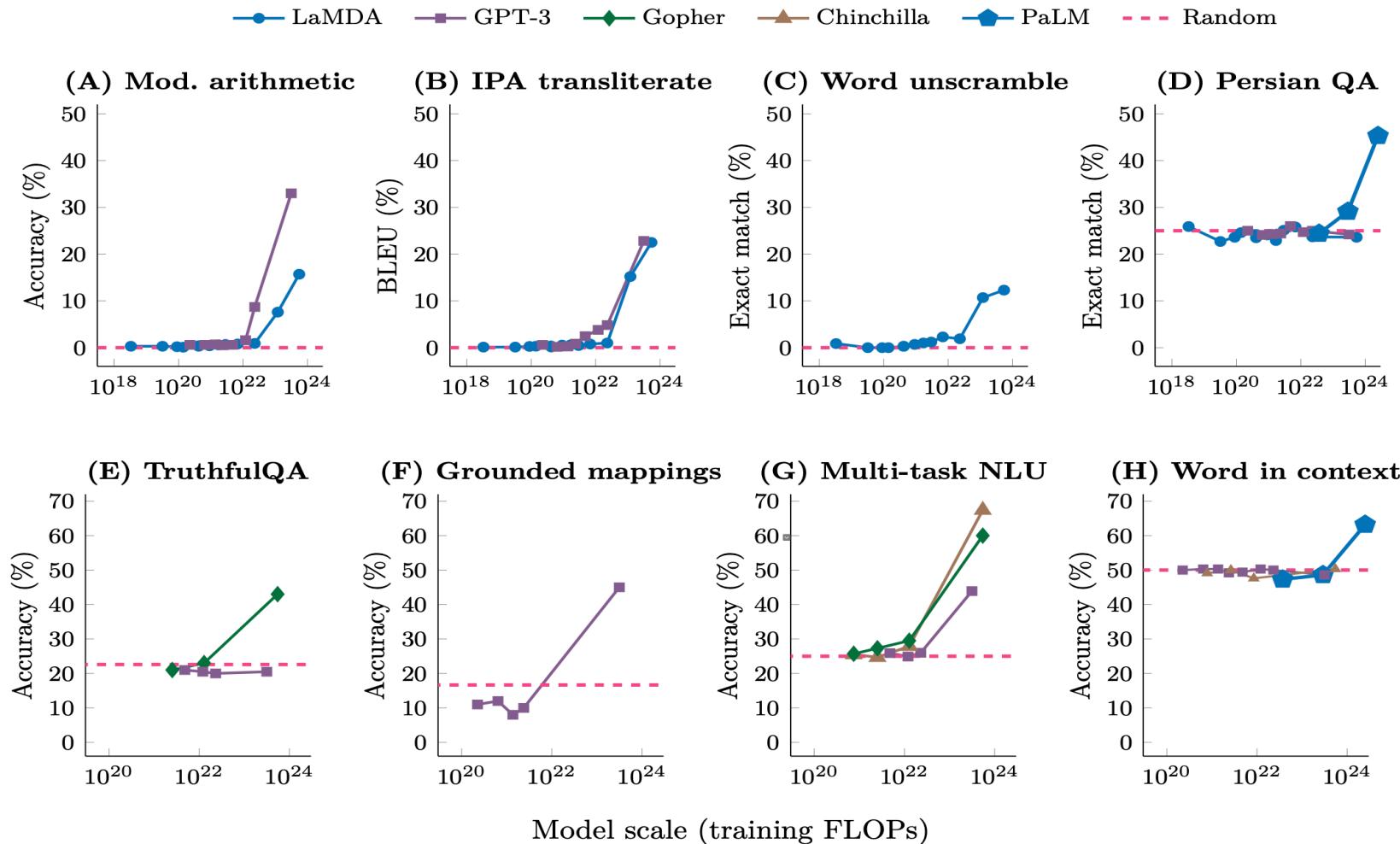
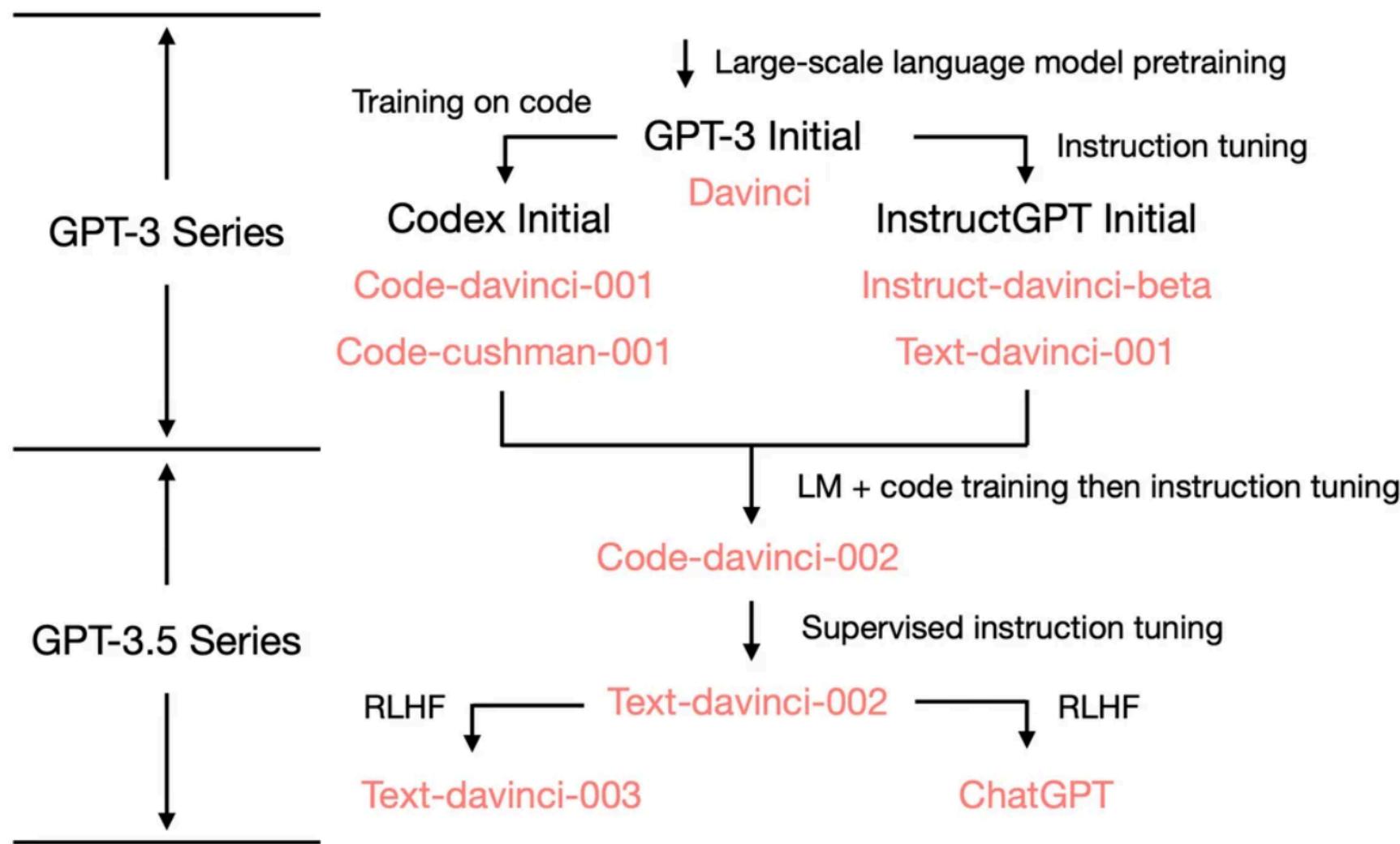


Fig. 1. The story of AI has been one of increasing *emergence* and *homogenization*. With the introduction of machine learning, *how* a task is performed emerges (is inferred automatically) from examples; with deep learning, the high-level features used for prediction emerge; and with foundation models, even advanced functionalities such as in-context learning emerge. At the same time, machine learning homogenizes learning algorithms (e.g., logistic regression), deep learning homogenizes model architectures (e.g., Convolutional Neural Networks), and foundation models homogenizes the model itself (e.g., GPT-3).

The scale matters: the emergence of abilities



chatGPT 发展路径



RLHF

Step 1

Collect demonstration data and train a supervised policy.

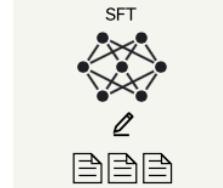
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



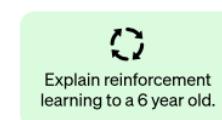
This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

Collect comparison data and train a reward model.

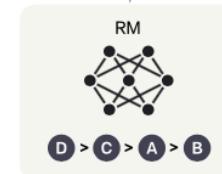
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



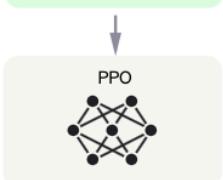
Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

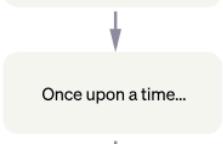
A new prompt is sampled from the dataset.



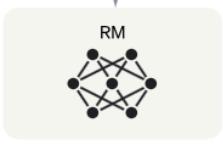
The PPO model is initialized from the supervised policy.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



RLHF

- RLHF 触发的能力：

- **详实的回应**: text-davinci-003 的生成通常比 text-davinci-002 长。ChatGPT 的回应则更加冗长，以至于用户必须明确要求“用一句话回答我”，才能得到更加简洁的回答。这是 RLHF 的直接产物。
- **公正的回应**: ChatGPT 通常对涉及多个实体利益的事件（例如政治事件）给出非常平衡的回答。这也是 RLHF 的产物。
- **拒绝不当问题**: 这是内容过滤器和由 RLHF 触发的模型自身能力的结合，过滤器过滤掉一部分，然后模型再拒绝一部分。
- **拒绝其知识范围之外的问题**: 例如，拒绝在 2021 年 6 月之后发生的新事件（因为它没在这之后的数据上训练过）。这是 RLHF 最神奇的部分，因为它使模型能够隐式地区分哪些问题在其知识范围内，哪些问题不在其知识范围内。

RLHF

- 需注意的是：
 - 所有的能力都是模型本来就有的，而不是通过RLHF注入的。RLHF的作用是触发/解锁突现能力。这个论点主要来自于数据量大小的比较：因为与预训练的数据量相比，RLHF占用的计算量/数据量要少得多。
 - 模型知道它不知道什么不是通过编写规则来实现的，而是通过RLHF解锁的。这是一个非常令人惊讶的发现，因为RLHF的最初目标是让模型生成复合人类期望的回答，这更多是让模型生成安全的句子，而不是让模型知道它不知道的内容。
- 幕后发生的事情可能是：
 - ChatGPT：通过牺牲上下文学习的能力换取建模对话历史的能力。这是一个基于经验的观测结果，因为ChatGPT似乎不像text-davinci-003那样受到上下文演示的强烈影响。
 - text-davinci-003：恢复了text-davinci-002所牺牲的上下文学习能力，提高零样本的能力。根据instructGPT的论文，这是来自于强化学习调整阶段混入了语言建模的目标（而不是RLHF本身）。

指令微调

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

InstructGPT

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them. ↵

RLHF第一阶段

- 第一阶段：冷启动阶段的监督策略模型。靠GPT 3.5本身，尽管它很强，但是它很难理解人类不同类型指令中蕴含的不同意图，也很难判断生成内容是否是高质量的结果。为了让GPT 3.5初步具备理解指令中蕴含的意图，首先会从测试用户提交的prompt(就是指令或问题)中随机抽取一批，靠专业的标注人员，给出指定prompt的高质量答案，然后用这些人工标注好的<prompt,answer>数据来Fine-tune GPT 3.5模型。经过这个过程，我们可以认为GPT 3.5初步具备了理解人类prompt中所包含意图，并根据这个意图给出相对高质量回答的能力，但是很明显，仅仅这样做是不够的。

Step 1

Collect demonstration data
and train a supervised policy.

A prompt is
sampled from our
prompt dataset.



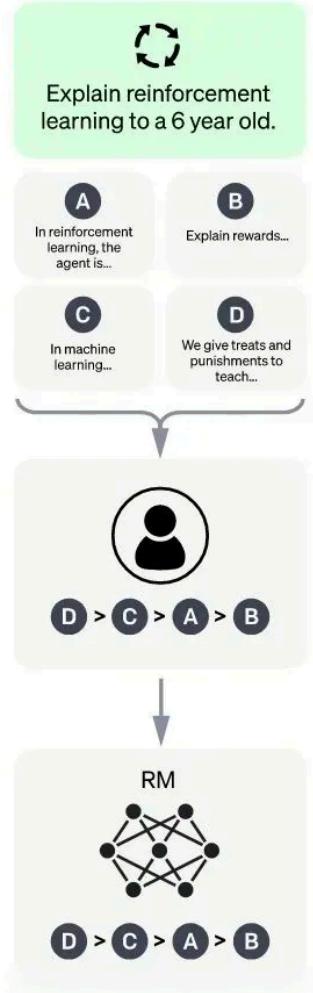
RLHF第二阶段

- 第二阶段：训练回报模型（Reward Model, RM）。这个阶段的主要目的是通过人工标注训练数据，来训练回报模型。
 - 随机抽样一批用户提交的prompt(大部分和第一阶段的相同)
 - 使用第一阶段Fine-tune好的冷启动模型，对于每个prompt，由冷启动模型生成K个不同的回答
 - 人工根据结果质量从高到低排序
 - 用排序结果训练回报模型

Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.



RLHF第三阶段

- 第三阶段：采用强化学习来增强预训练模型的能力。
 - 利用上一阶段学好的RM模型，靠RM打分结果来更新预训练模型参数。
 - 此阶段无需人工标注

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

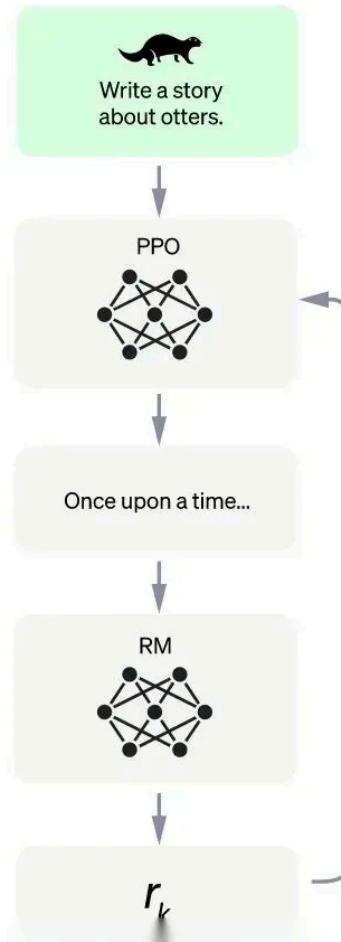
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Contents

- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向

局限性

- ▶ ChatGPT有时会写出听起来有道理但实际上并不正确甚至可能是荒谬的答案。解决这个问题是非常有挑战性的，因为：(1)在RL训练期间，目前并没有提供信息真实性的来源；(2)训练一个更加谨慎模型，会导致它拒绝回答一些它能够正确回答的问题；(3)有监督的训练方法会误导模型，因为理想的答案应该来自于模型所掌握的知识，而不是人类训练人员所掌握的知识。
- ▶ ChatGPT对调整输入措辞或多次尝试同一提示（Prompt）很敏感。例如，给定一个问题的一个措辞，模型可以声称不知道答案，但只要稍微重新措辞，就可以正确回答。
- ▶ 该模型通常过于冗长，并过度使用某些短语，例如重申它是由OpenAI训练的语言模型。这些问题来自培训数据中的偏见（培训人员更喜欢看起来更全面的更长的答案）和众所周知的过度优化问题。
- ▶ 理想情况下，当用户提供模棱两可的查询时，模型会提出澄清问题。否则，我们目前的模型通常会随意猜测用户的意图。^[1]
- ▶ 虽然我们已经努力让模型拒绝不适当的请求，但它有时仍会响应有害的指令或表现出偏见的行为。我们正在使用Moderation API来警告或阻止某些类型的不安全内容，但我们预计它目前会有一些误报和误报。我们渴望收集用户反馈，以帮助我们正在进行的改进该系统的工作。

事实与常识错误

请用比较详细的篇幅，描述一下苏格拉底

苏格拉底是古希腊哲学家，他于公元前4世纪活跃在雅典。他被认为是近代西方哲学的奠基人，也是历史上最著名的哲学家之一。苏格拉底主要的著作包括《伦理学》和《共和国》，这两部作品都对后来的哲学产生了巨大的影响。

苏格拉底出生于一个贵族家庭，他从小就受到了良好的教育。他在雅典学习哲学，并在他那个时代最著名的哲学家——柏拉图的指导下学习。苏格拉底后来也成为了柏拉图的助手。

苏格拉底在他的著作中提出了许多原创性的观点，包括“认识论”、“形而上学”和“道德哲学”等。他认为，人类可以通过推理和思考来理解世界，并认为理性是人类最高的品质。

苏格拉底的思想对西方哲学产生了巨大的影响，他被认为是“哲学之父”。虽然苏格拉底本人并没有留下多少作品，但他的学生们撰写的记录仍然对后世产生了深远的影响。

美国为什么要轰炸珍珠港？

美国轰炸珍珠港的原因是，当时日本对珍珠港实施了突然袭击，造成了巨大的人员伤亡和经济损失。美国认为，对日本进行报复是必要的，因此决定对珍珠港进行轰炸。

数理逻辑能力不足

有一只狗和一只猫，一只是白色的，一只是黑色的，现在知道猫是黑色的。请问狗是什么颜色？

无法回答，因为你所提供的信息并不足以确定狗的颜色。根据你提供的信息，我们只知道猫是黑色的，但是并不知道狗的颜色。为了得到更准确的答案，我们需要更多的信息。

小明有两个哥哥一个弟弟，请问小明的弟弟有几个哥哥？

小明的弟弟有两个哥哥，这里只有小明一个人是哥哥。因此小明的弟弟只有小明这一个哥哥。

91

0.988888888888 + 0.111111111111 等于多少

91

精确到小数点后三位呢

91

如果你要求结果精确到小数点后三位，那么 0.988888888888 加上 0.111111111111 等于 0.999。

91

我认为你算得不对，请你分步骤计算 0.988888888888 + 0.111111111111 等于多少

91

好的，我可以为您分步骤计算 0.988888888888 加上 0.111111111111 的结果。

1. 在第一步中，我们可以将这两个数的末位相加，得到 0.000。
2. 在第二步中，我们可以将这两个数的倒数第二位相加，得到 0.00。
3. 在第三步中，我们可以将这两个数的倒数第三位相加，得到 0.0。
4. 在第四步中，我们可以将这两个数的倒数第四位相加，得到 0.99。

综上所述，0.988888888888 加上 0.111111111111 等于 0.999。

Contents

- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向

ChatGPT+金融

- 可以帮助金融公司提供更好的客户服务
 - 理解文本数据中的风险信号和关键指标，帮助金融机构识别潜在的风险，包括欺诈行为、资金洗白等。
 - 通过分析大量的市场和公司数据，提供投资决策和分析建议，例如股票预测、市场趋势分析等。
 - 提供对投资组合的建议，帮助投资者优化自己的投资组合。

ChatGPT + 医疗

- 可以用于提供医疗咨询和辅助诊断
 - 解释医学术语
 - 回答病人的问题
 - 提供治疗方案
- 在理解了几乎全部医疗有关的资料的基础上，ChatGPT可以做到比任何一位资深的医学专家都更加博学，在解释医学术语的问题上可以解释的比任何人都更好。从而也就可以针对病人的问题做出很专业的解答。

ChatGPT+教育

- 可以帮助学生和教师进行学术交流和教学
 - 进行个性化的学习支持，通过对学生的回答进行分析，推荐个性化的学习计划，提供适合学生学习能力和风格的学习资料和资源，从而提高学生的学习效率和质量。
 - 作为一个在线问答平台，帮助学生解答问题。学生可以在ChatGPT上提出问题，然后ChatGPT将回答学生的问题，并在需要时提供相关的学习资源和资料。在英语的学习上可以支持语音交互学习，帮助学生练习口语和听力技能。学生可以使用语音输入来提出问题或表达想法，ChatGPT会进行自然语言处理和理解，提供合适的回答或反馈。
 - 作为一个智能导师的角色，为学生提供课程建议和帮助。它可以分析学生的学习记录、课程评估和学习偏好，为学生推荐适合的课程、教材和学习方式，同时还可以提供实时的课程建议和学习支持。
 - 可以帮助学生梳理和巩固学习的知识点。学生可以通过ChatGPT进行自测或进行问答交互，ChatGPT会根据学生的答题情况提供相关的回馈和建议，帮助学生加深对知识点的理解和记忆。

ChatGPT+其他

- ChatGPT+传媒（智能新闻写作，加强新闻时效性）
- ChatGPT+影视（拓宽创作素材，提升作品质量）
- ChatGPT+营销（虚拟客服，提升服务质量）

Contents

- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向

ChatGPT未来

- 与检索结合（事实性，实时性）
- 调用外部能力（改善数学和推理能力）
- 多模态理解和生成
- 终生持续学习

Contents

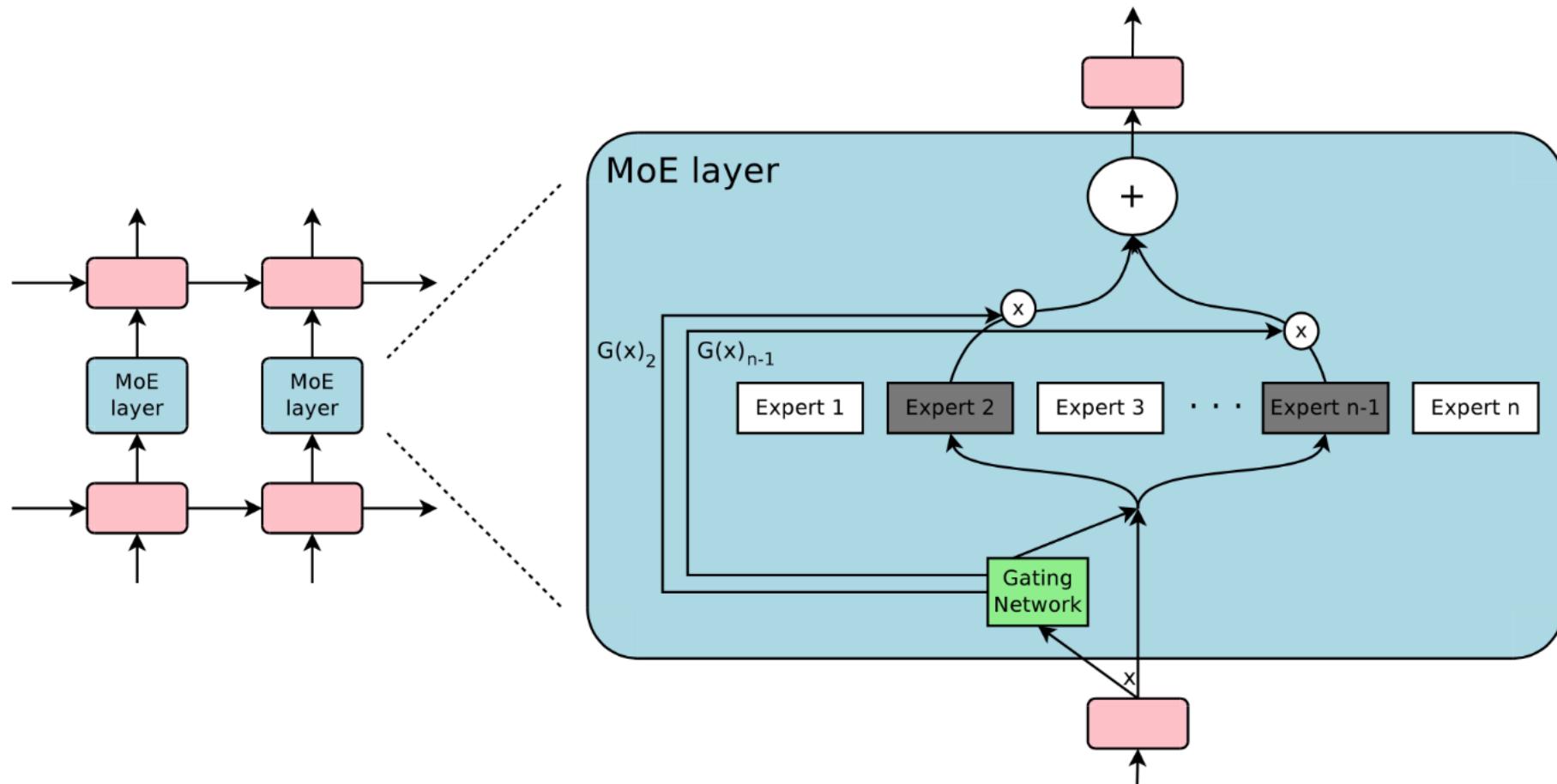
- ChatGPT的市场概况
- ChatGPT的关键技术
- ChatGPT的不足之处
- ChatGPT的主要应用
- ChatGPT的未来发展方向
- GPT-4 MoE

Mixture-of-Experts (MoE)

- Mixture-of-Experts (MoE) 试图推进模型大小的极限。
- 核心思想：集成学习：多个弱学习器的组合产生一个强学习器！
- 在一个深度神经网络内部
 - 集成可以通过连接多个专家(experts)的门控(gating)机制来实现(Shazeer等,2017)。
 - 门控机制控制哪些子网络(例如哪些专家)应该被激活以产生输出。
 - 稀疏门控“mixture-of-experts” (MoE) 层。

Mixture-of-Experts (MoE)

- 一个MoE层包含：
 - n 个前馈网络作为专家 $\{E_i\}_{i=1}^n$
 - 一个可训练的门控网络 G 来学习 n 个专家上的概率分布,以便将流量路由到少数选定的专家。
- 根据门控输出,并非每个专家都需要被评估。当专家数量过大时,我们可以考虑使用两层层次结构的MoE。



Mixture-of-Experts (MoE)

- 一个简单的生成门控向量 g 的方法：
 - 输入与一个可训练的权重矩阵 W_g 相乘
 - 然后进行softmax归一化: $g(x) = softmax(xW_g)$ 。
 - 然而,这会产生一个密集的门控向量,不能帮助节省计算资源
- MoE层仅保留门控向量中的前 k 大的值。
 - 向门控向量中添加可调整的高斯噪声以改善负载均衡
 - 称为噪声topk门控。

$$G(x) = softmax(\text{topk}(H(x), k))$$

$$H^{(i)}(x) = (xW_g)^{(i)} + \epsilon \cdot \text{softplus}((xW_{\text{noise}})^{(i)}); \quad \epsilon \sim \mathcal{N}(0, 1)$$

$$\text{topk}^{(i)}(v, k) = \begin{cases} v^{(i)} & \text{if } v^{(i)} \text{ is in the top } k \text{ elements of } v \\ -\infty & \text{otherwise} \end{cases}$$

$v^{(i)}$: v 的第*i*维

Mixture-of-Experts (MoE)

- 门控网络总是偏爱几个强专家，自我强化效应
 - Shazeer(2017) 通过额外的重要性损失(importance loss)提出了一个软约束，以鼓励所有专家拥有相同的权重。它相当于每个专家的批次平均值的变异系数的平方。

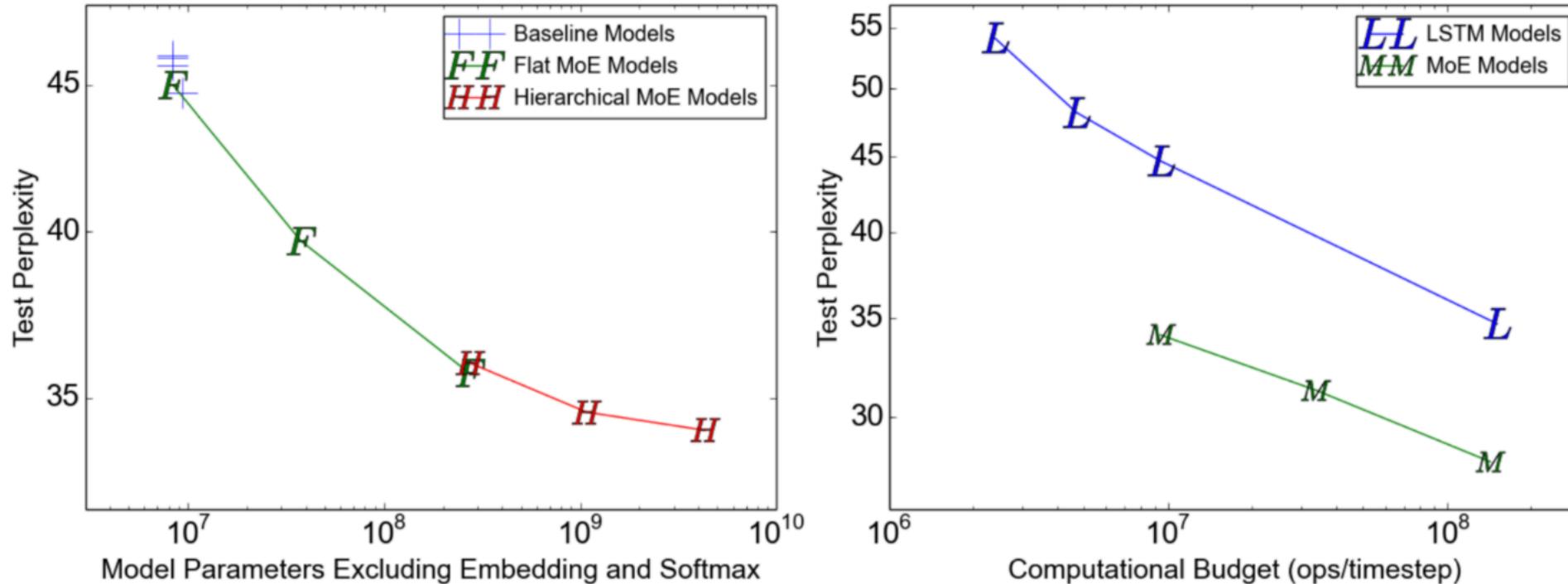
$$L_{\text{aux}} = w_{\text{aux}} \cdot \text{CV}\left(\sum_{x \in X} G(x)\right)^2 \quad \text{CV} = \frac{\sigma}{\mu}$$

- 其中CV是变异系数，损失权重 w_{aux} 是一个需要调节的超参数。
- 这个额外的重要性损失鼓励门控网络均匀地使用所有专家，而不是总是偏爱几个专家。它通过最小化各专家的批次平均使用率的变异系数来实现这个目标。

Mixture-of-Experts (MoE)

- 在MoE中,由于门控机制的作用,每个专家网络实际上只看到输入batch的一个子集。这被称为“shrinking batch problem”
- 应该尽量使用较大的batch大小。
- 但是batch大小也受GPU内存限制
- 为了提高吞吐量,可以采用
 - 数据并行(在多个GPU上划分一个大batch)
 - 模型并行(在多个GPU上划分模型)
- 数据并行可以增加每个专家的有效batch大小,模型并行可以允许构建更大的MoE模型。合理地应用两种并行方法可以大幅提升MoE的训练速度。

Mixture-of-Experts (MoE)



- Test perplexity on 1-Billion-Word language modeling benchmark. (Left) The model capacity increases from left to right, containing 4, 32, 256, 256, 1024 and 4096 experts. (Right) Performance of the 4 billion parameters MoE model, the largest one in the left figure, under different computation budgets.

Mixture-of-Experts (MoE)

- GShard(Lepikhin等,2020)通过分片的方式将MoE transformer模型扩展到了6000亿参数规模
- MoE transformer通过用MoE层替换每两个前馈层的一层
- 分片MoE transformer仅对MoE层在多台机器上进行分片,而其他层简单地进行复制。
- GShard中门控函数G的设计有几点改进:
 - 专家容量: 通过每个专家的流量不能超过一个阈值,命名为“专家容量”。如果一个token被路由到已经达到容量的专家,该token会被标记为“溢出”,并将门控输出改为零向量。
 - 本地组分发: 将token均匀分区到多个本地组中,并在组级别上执行专家容量约束。
 - 辅助损失: 其动机与原始MoE辅助损失相似。他们添加了一个辅助损失来最小化路由到每个专家的数据比例的均方。
 - 随机路由: 以其权重成比例的概率选择第二佳专家; 否则, GShard遵循随机路由,以增加随机性。

Mixture-of-Experts (MoE)

- Pseudo code of the group-level top-2 gating mechanism with auxiliary loss in GShard.

Algorithm 1: Group-level top-2 gating with auxiliary loss

Data: x_S , a group of tokens of size S
Data: C , Expert capacity allocated to this group
Result: $\mathcal{G}_{S,E}$, group combine weights
Result: ℓ_{aux} , group auxiliary loss

```

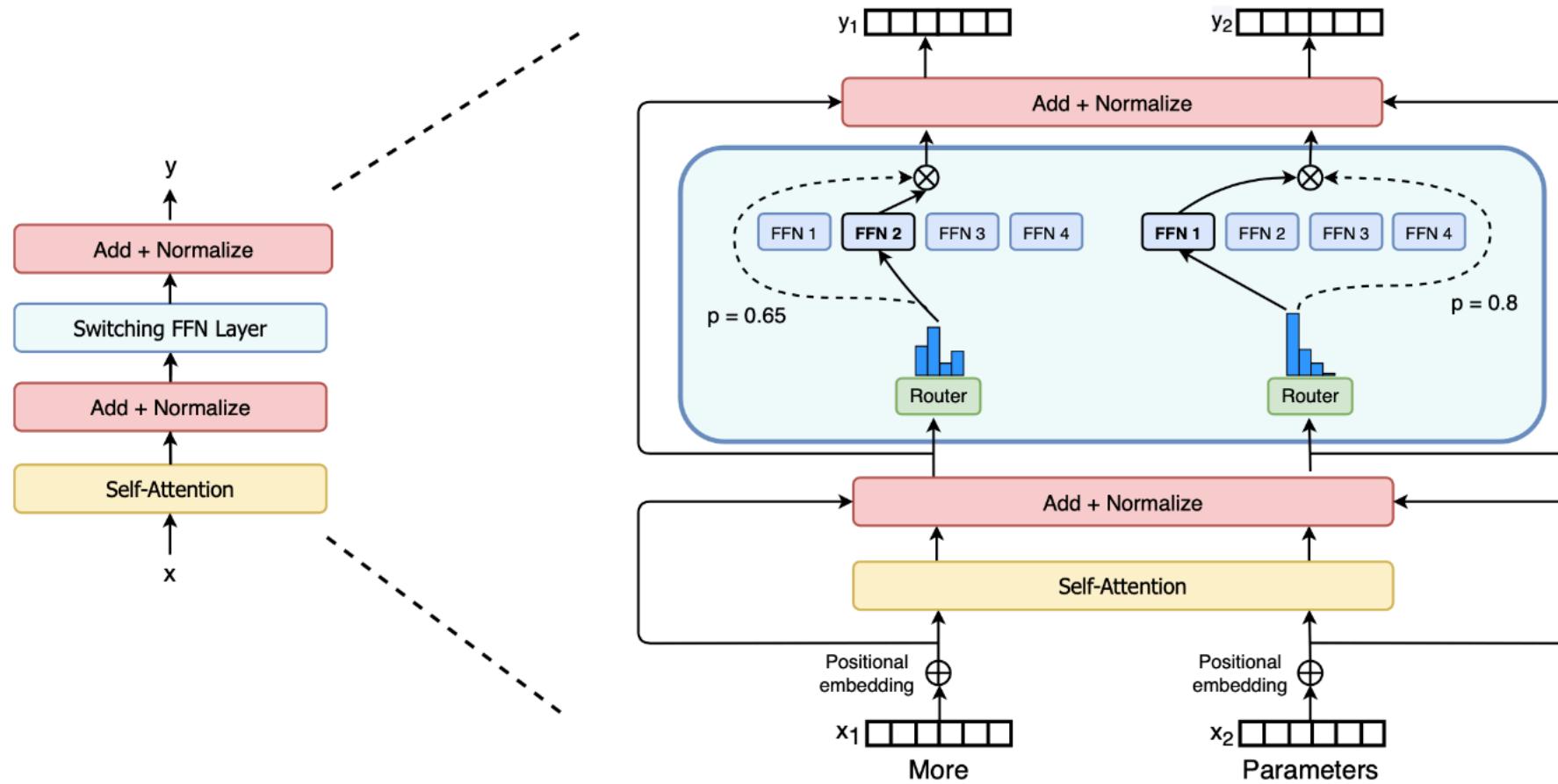
(1)  $c_E \leftarrow 0$                                 ▷ gating decisions per expert
(2)  $g_{S,E} \leftarrow softmax(wg \cdot x_S)$       ▷ gates per token per expert,  $wg$  are trainable weights
(3)  $m_E \leftarrow \frac{1}{S} \sum_{s=1}^S g_{s,E}$     ▷ mean gates per expert
(4) for  $s \leftarrow 1$  to  $S$  do
(5)    $g1, e1, g2, e2 = top\_2(g_{s,E})$           ▷ top-2 gates and expert indices
(6)    $g1 \leftarrow g1/(g1 + g2)$                   ▷ normalized  $g1$ 
(7)    $c \leftarrow c_{e1}$                           ▷ position in  $e1$  expert buffer
(8)   if  $c_{e1} < C$  then
(9)     |  $\mathcal{G}_{s,e1} \leftarrow g1$                 ▷  $e1$  expert combine weight for  $x_s$ 
(10)   end
(11)    $c_{e1} \leftarrow c + 1$                       ▷ incrementing  $e1$  expert decisions count
(12) end
(13)  $\ell_{aux} = \frac{1}{E} \sum_{e=1}^E \frac{c_e}{S} \cdot m_e$ 
(14) for  $s \leftarrow 1$  to  $S$  do
(15)    $g1, e1, g2, e2 = top\_2(g_{s,E})$           ▷ top-2 gates and expert indices
(16)    $g2 \leftarrow g2/(g1 + g2)$                   ▷ normalized  $g2$ 
(17)    $rnd \leftarrow uniform(0, 1)$                  ▷ dispatch to second-best expert with probability  $\propto 2 \cdot g2$ 
(18)    $c \leftarrow c_{e2}$                           ▷ position in  $e2$  expert buffer
(19)   if  $c < C \wedge 2 \cdot g2 > rnd$  then
(20)     |  $\mathcal{G}_{s,e2} \leftarrow g2$                 ▷  $e2$  expert combine weight for  $x_s$ 
(21)   end
(22)    $c_{e2} \leftarrow c + 1$ 
(23) end

```

Mixture-of-Experts (MoE)

- Switch Transformer(Fedus等,2021)
- Replace **dense feed forward layer** by **sparse switch FFN layer**
- 将模型规模扩大到了万亿参数量级(!!)
- 在稀疏开关前馈层中,每个输入仅路由到一个专家网络
- 用于负载均衡的辅助损失为 $\text{loss}_{aux} = w_{aux} \sum_{i=1}^n f_i p_i$ ，其中给定n个专家， f_i 是路由到第i个专家的token比例， p_i 是门控网络为第i个专家预测的路由概率。

Mixture-of-Experts (MoE)



- Switch transformer. The sparse switch FFN layer is in the blue boxes.

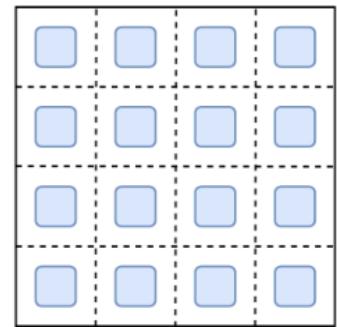
Mixture-of-Experts (MoE)

- 为了改善训练稳定性, Switch Transformer结合了以下设计:
 - 选择性精度。仅将模型的局部部分转换为FP32精度可以提高稳定性,同时避免了FP32张量的高昂通信成本。FP32精度仅在路由器函数主体内使用,结果被重新转换为FP16。
 - 更小的初始化。权重矩阵的初始化从 $\mu = 0, \sigma = \sqrt{\frac{s}{n}}$ 的截断正态分布中采样。同时建议将transformer的初始化缩放参数s从1降低到0.1。
 - 使用更高的专家**dropout**。微调通常使用小数据集。为避免过拟合,每个专家内的**dropout**率显著增加。有趣的是,他们发现增加所有层的**dropout**会导致较差的性能。在该论文中,非专家层的**dropout**率为0.1,而专家FF层的**dropout**率为0.4。

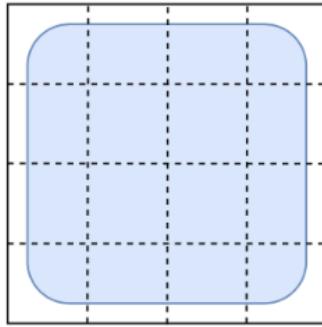
Mixture-of-Experts (MoE)

How the *model weights* are split over cores

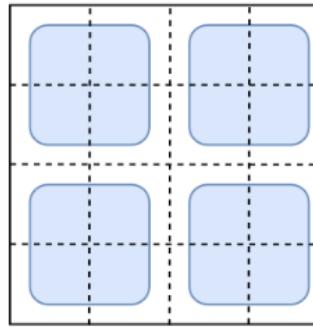
Data Parallelism



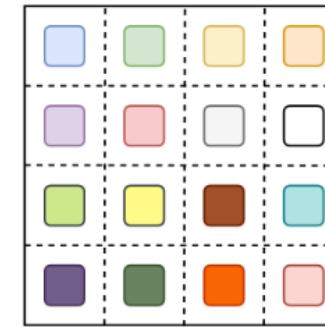
Model Parallelism



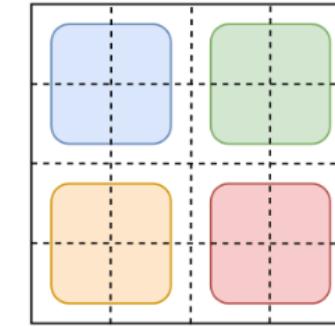
Model and Data Parallelism



Expert and Data Parallelism

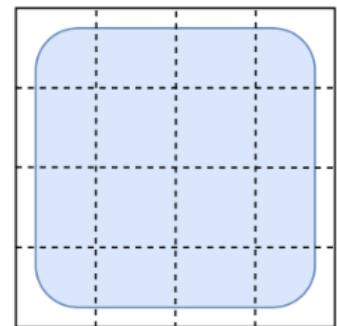


Expert, Model and Data Parallelism

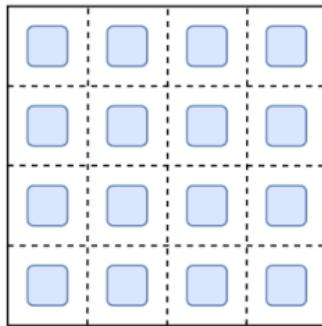


- The switch transformer paper summarized different data and model parallelism strategies for training large models with a nice illustration:

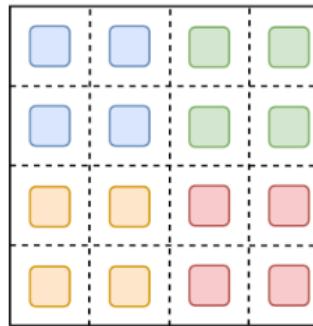
Data Parallelism



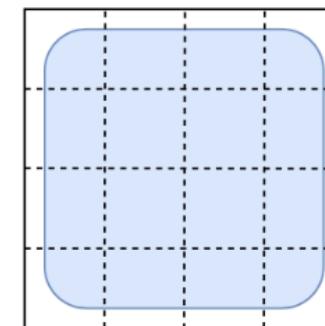
Model Parallelism



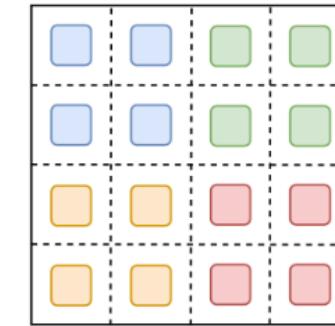
Model and Data Parallelism



Expert and Data Parallelism



Expert, Model and Data Parallelism



Mixture-of-Experts (MoE)

- GShard top-2 和 Switch Transformer top-1 的共同局限：
 - 依赖 token 选择，无法保证最优路由
 - 辅助损失仅鼓励负载均衡，不能保证最佳性能
 - 存在专家容量限制可能导致 token 被丢弃浪费
- 这些局限说明仅依靠标记自主选择路由以及辅助损失驱动的负载均衡仍有优化空间。
- 需要更全局优化的路由机制来提高性能，同时管理好专家容量。

Mixture-of-Experts (MoE)

- Export Choice (EC) (Zhou等,2022) 允许每个专家选择顶部k个标记，而不是标记选择专家。
- 每个专家自然保证了固定容量，每个标记可能被路由到多个专家。EC可以实现完美的负载平衡，并且可以提高训练收敛速度2倍。
- 给定e个专家和一个输入矩阵 $X \in \mathbb{R}^{n \times d}$, token to expert的亲和度分数通过计算得到

$$S = \text{softmax}(X \cdot W_g), \text{ where } W_g \in \mathbb{R}^{d \times e}, S \in \mathbb{R}^{n \times e}$$

Mixture-of-Experts (MoE)

- token与专家的分配用三个矩阵 $I, G \in \mathbb{R}^{e \times k}$ 和 $P \in \mathbb{R}^{e \times k \times n}$ 表示
- $I[i,j]$ 是第*i*个专家的第*j*个选择的token
- 门控矩阵 G 存储所选token的路由权重
- P 是 I 的 onehot 版本,用于为门控 FFN 层产生输入矩阵($P \cdot X \in \mathbb{R}^{e \times k \times n}$)

$$G, I = \text{top-k}(S^\top, k) \quad P = \text{one-hot}(I)$$

Mixture-of-Experts (MoE)

- 可以通过限制每个token的最大专家个数来加正则化项

$$\begin{aligned} & \max_A \langle S^\top, A \rangle + \lambda H(A) \\ \text{s.t. } & \forall i : \sum_{j'} A[i, j'] = k, \quad \forall j : \sum_{i'} A[i', j] \leq b, \quad \forall i, j : 0 \leq A[i, j] \leq 1 \end{aligned}$$

- 其中 $A[i,j]$ 标注第*i*个专家是否选择第*j*个token。求解这一约束问题非常困难。论文采用Dykstra算法，运行多次迭代计算步骤。限制专家选择数量会略微降低微调性能。
- 参数k确定为 $k=nc/e$ ，其中n是一个batch中的token总数，c是容量因子，表示每个token平均使用的专家数量。
- EC的一个大缺点是在batch大小太小时无法工作，也不适用于自回归文本生成，因为它需要知道未来token来进行top-k选择。

Thank you!