

# 人类经验与AI算法的镜像之旅

浙江大学教育学院

陈静远



# 报告提纲



语言的奥秘：人类如何解码世界



从符号到智能：AI的语言理解之路



DeepSeek-V3：大语言模型的构建与进化



DeepSeek-R1：推理模型的诞生与突破



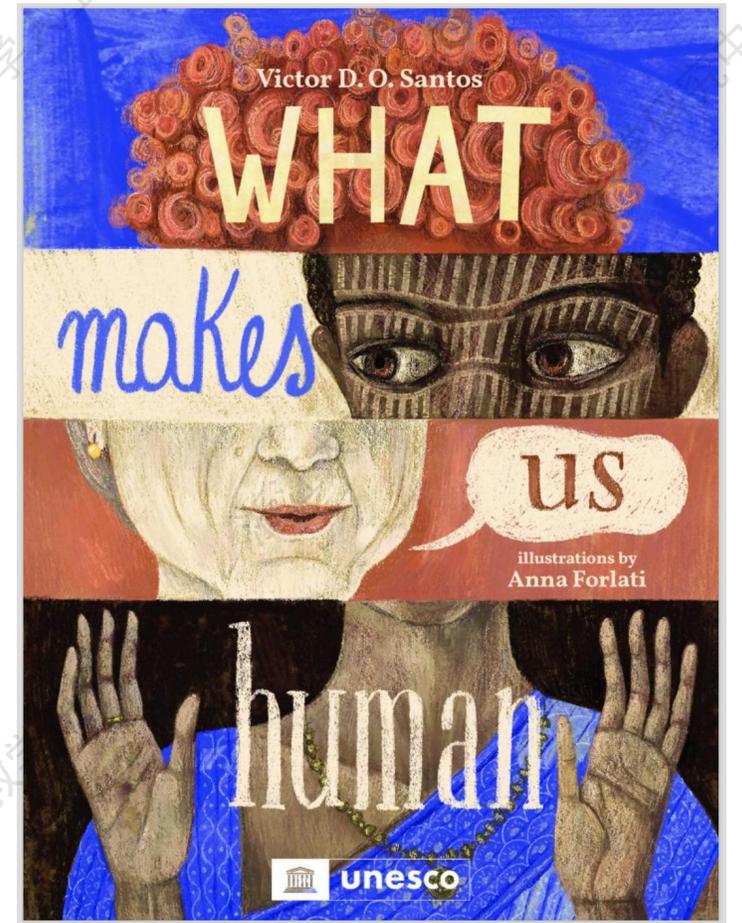
智能体时代：AI如何重塑教育未来

# 语言对我们有多重要

**语言**造就了人类，人和动物的根本区别就是人具有创造性地运用语言的能力

- 语言赋予人类秩序
- 语言是思维的工具
- 语言是合作的纽带

如果想要实现通用的人工智能，理解人类使用的**不太精确、可能有歧义、混乱**的语言是一个有效途径。



维克多-D.O.-桑托斯的《是什么让我们成为人类》

# 人类是如何理解语言的？



下雨了我要赶紧回家...



“下雨” / “家” / “收衣服” 这些词语，  
基于我们的经验，在我们大脑中已经建  
立了固定的神经连接。



# 人类是如何理解语言的？



下雨了我要赶紧回家  
收衣服。



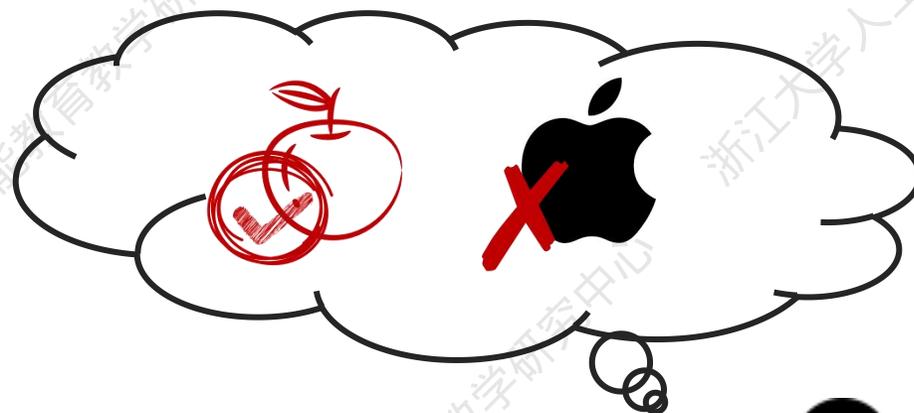
人类大脑通过理解每个词语的意思、进行词语组装，从而得到句子的意思，甚至推断出句子背后的含义。因此，理解语言的基础是**理解词语及词语间的关联关系**。



# 人类是如何理解语言的？



这个苹果品质真高 已通过欧盟有机认证，每颗果实都带有 NFC 溯源标签。



人类语言的精准解译与语义歧义的消解具有显著的语境依赖性，需通过系统性整合上下文信息构建语义解析模型。



# 报告提纲



语言的奥秘：人类如何解码世界



从符号到智能：AI的语言理解之路



DeepSeek-V3：大语言模型的构建与进化



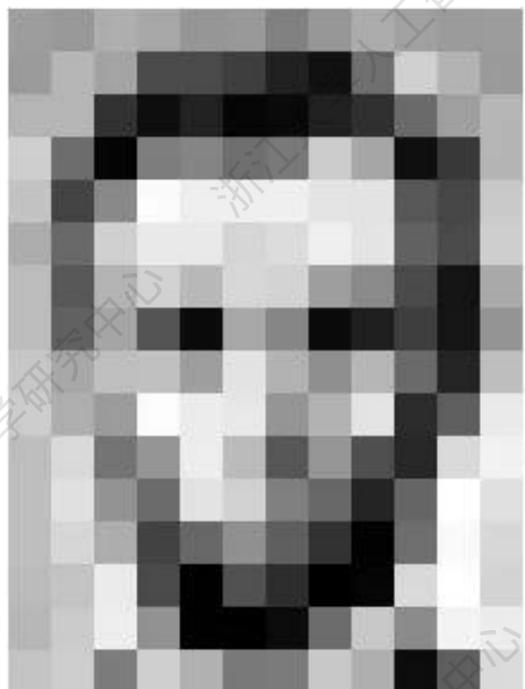
DeepSeek-R1：推理模型的诞生与突破



智能体时代：AI如何重塑教育未来

# 计算机的数字化世界

计算机理解一切信息的基础是将信息进行数字化。在处理图像时，计算机会将图像的每一个像素转换为数字信号，通常使用颜色的RGB值来表示每个像素。



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Towards Seamless Communication for Sign Language Support: Architecture, Algorithms, and Optimization

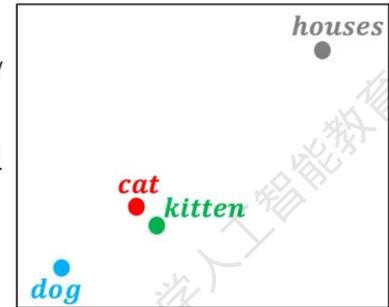
# 语言的数字化



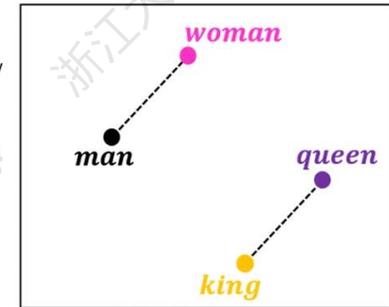
	living being	feline	human	gender	royalty	verb	plural
<b>cat</b> →	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
<b>kitten</b> →	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
<b>dog</b> →	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
<b>houses</b> →	-0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8

<b>man</b> →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<b>woman</b> →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<b>king</b> →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<b>queen</b> →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Dimensionality reduction of word embeddings from 7D to 2D



Dimensionality reduction of word embeddings from 7D to 2D



Word

Word embedding

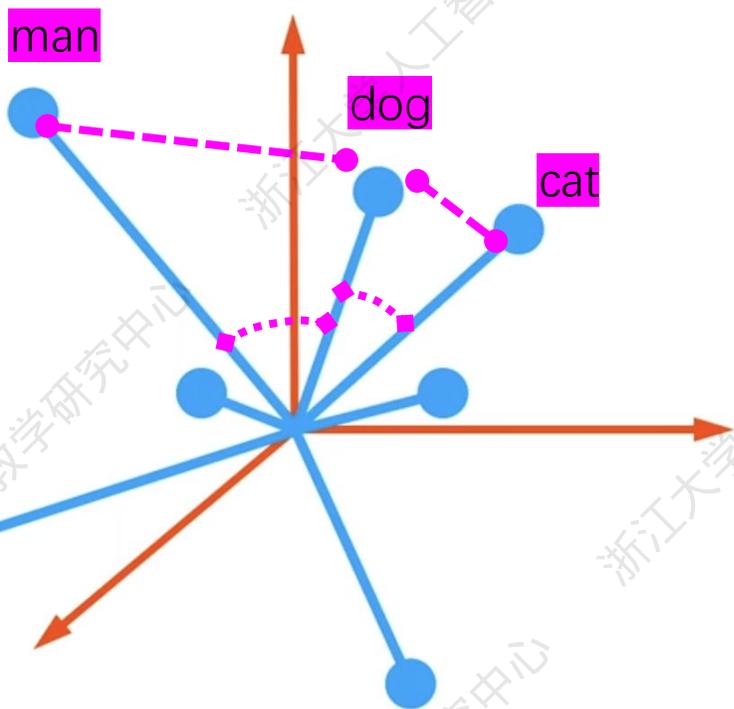
Dimensionality reduction

Visualization of word embeddings in 2D

计算机无法直接理解离散的人类语言

词向量 (word embedding) 和词与词之间的位置关系

# 词向量及单词之间的相似度



- **欧式距离**：两个点（或向量）在空间中的“直线距离”。它反映了两个向量的绝对差异。欧氏距离值越小，说明两个向量越接近；值越大，说明差异越大。

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}$$

- ◆ **余弦相似度**：两个向量之间夹角的余弦值来衡量它们的相似度。它反映了两个向量的方向是否相似，而不关心向量的大小。更适用于比较两者相似性（如文本相似度）。

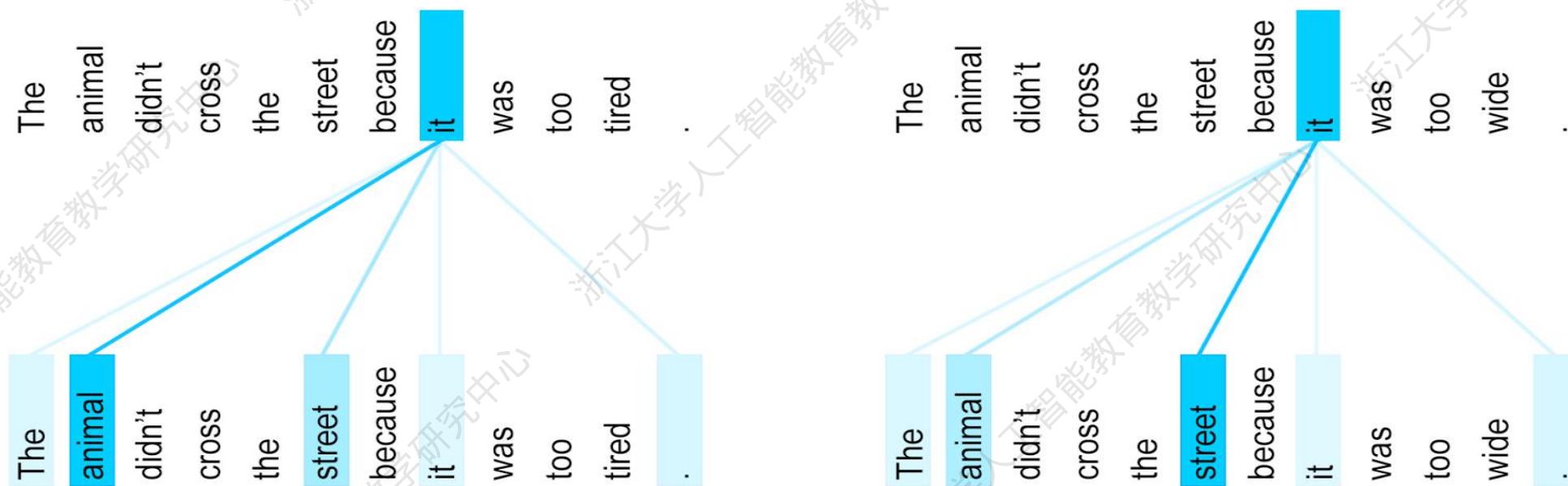
$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \times \|B\|}$$

# 词向量模型的缺陷

- 在序列数据中，同一个元素处在不同的上下文中意思是不同的。如：
  - (1) The animal didn't cross the street because **it** was too tired.  
(那只**动物**没有过马路，因为它太累了。)
  - (2) The animal didn't cross the street because **it** was too wide.  
(那只动物没有过马路，因为**马路**太宽了。)
- 然而，传统的词向量模型中同一个词只有一个向量，这对于一些词语会造成歧义问题，如何解决这个问题呢？

# 注意力机制与上下文建模

- 大语言模型通过使用Transformer架构，可以为每个词生成一个上下文相关的词向量，这解决了传统词向量无法处理多义词和上下文依赖的问题。
- 一个单词的真实含义，不仅仅取决于它自身，还取决于句子中的其它上下文信息（来自其它单词的信息）。
- 一个单词的向量值，需要融合从句子上下文中的其他单词中的信息，在数学上可以表达为所有单词的向量值的加权平均。这些权重值，我们可以称之为注意力权重 (attention weights)。



**it**与其他单词之间的注意力权重。蓝色的深浅表达了权重的相对大小。

# 报告提纲



语言的奥秘：人类如何解码世界



从符号到智能：AI的语言理解之路



**DeepSeek-V3：大语言模型的构建与进化**



DeepSeek-R1：推理模型的诞生与突破



智能体时代：AI如何重塑教育未来

# 文字接龙游戏

- 大语言模型（LLM）最令人印象深刻的能力是它能够**通过对话的方式**回答用户的问题。那么LLM回答问题的原理是什么呢？
- 不同于传统问答系统中答案来源于现成的网络或者数据库，大语言模型的回答是随着提问的进行**自动生成的**。
- 这一点很像文字接龙游戏，大语言模型会**基于前面的话不断地生成下一个合成的词汇**，直到觉得不必继续生成为止。



苹果是一种水果吗？

是的，苹果确实被归  
类为一种水果。

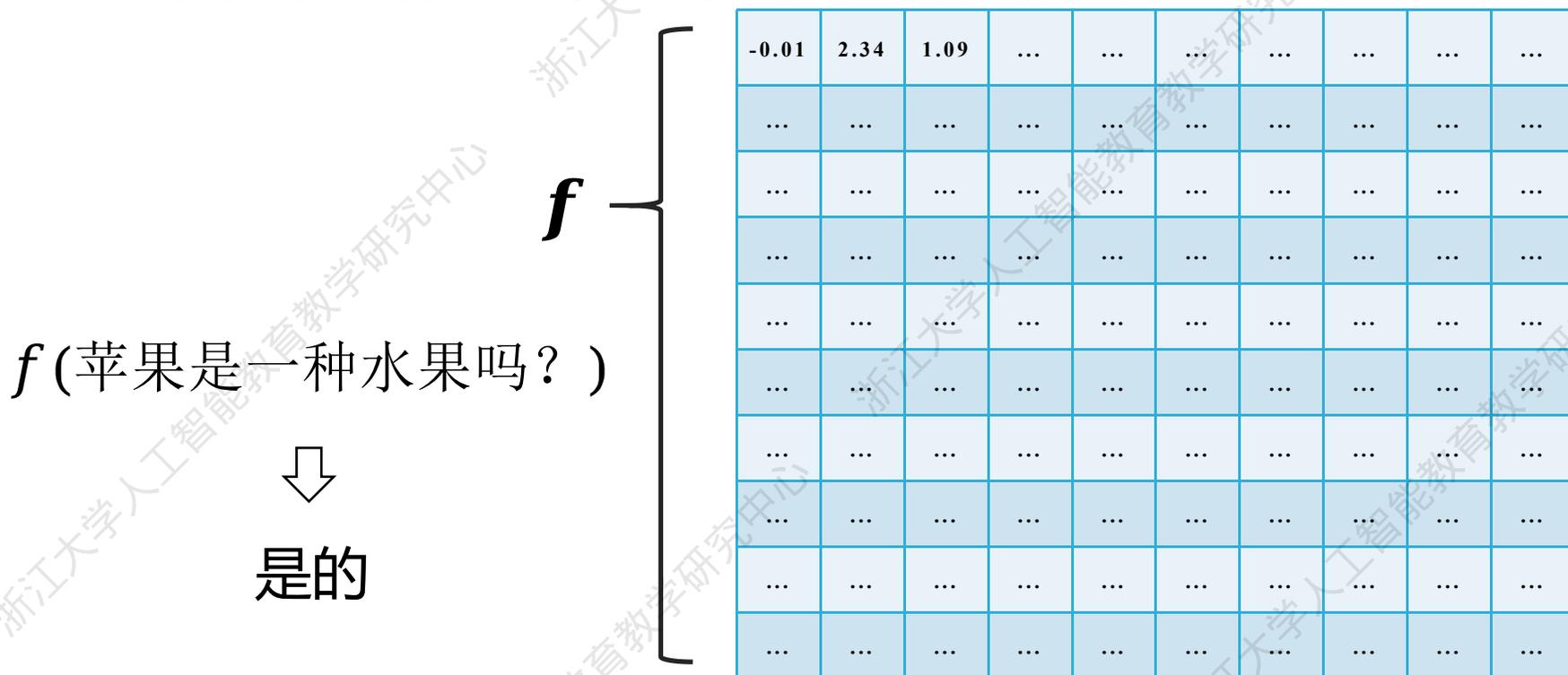
大语言模型

下一个可能的词	概率
确实	0.8
是的	0.06
嗯	0.04
.....	.....

...

# 大模型是如何工作的

- 数学家陶哲轩：大模型不是魔法，是**基于概率的猜测机**。
- 那么大模型是如何不断生成下一个词的概率的呢？
- 实际上，这一过程依赖于模型内部的**参数**，这些参数通过大量数据的训练来不断调整，**蕴含**了数据的分布规律，从而使模型能够在特定上下文下预测出最合适的下一个词。并且，当这些参数单元的数量级提升时，系统的认知能力通常会呈现出显著的进化趋势。



- DeepSeek-V3 满血版是67B模型
- B: Billion 10亿

# 大模型是如何工作的

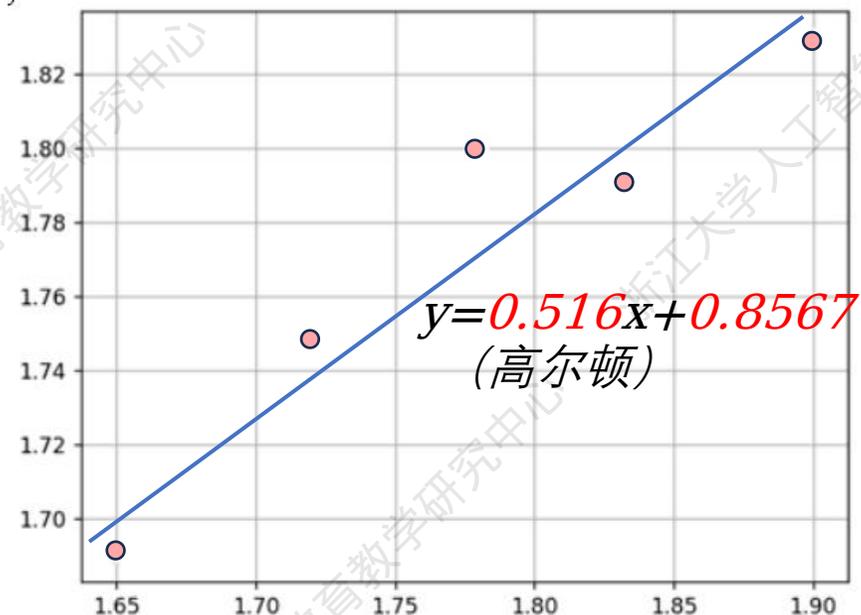
Spreadsheets-are-all-you-need: 在 Excel 中完全实现了 GPT2 的前向推理过程。

	2	3	4	5	6	7	8	9	10
1	Type Prompt Below Here		Predicted Next Token						
2	Zhejiang		.						
3	University								
4	is								
5	a								
6	good								
7	university								
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									

# 大模型参数是如何通过学习得到的？

- 希望能够得到一个模型：基于父亲的身高预测儿子的未来身高。

父亲身高 (米)	儿子身高 (米)
1.65	1.69
1.72	1.75
1.78	1.80
1.83	1.79
1.90	1.83



- $y = ax + b$

- 类似地，大语言模型的参数也是通过大量数据的学习，逐渐调整和优化的，**将数据的规律压缩到参数中**，以便对新的数据进行更准确地预测和生成合适的输出。

# 大模型训练流程

	预训练	指令微调	奖励函数	强化学习
数据集	原始数据 万亿级别词汇 海量数据	指令数据 数万提示回复对 <用户指令, 对应回复>	对比数据 百万次比较	指令数据 十万次指令
算法	语言模型预训练 预测下一个词	语言模型监督微调 预测下一个词	二元分类模型 预测偏好一致的奖励	强化学习 生成最大化奖励的词
模型	基座模型	指令微调模型	奖励模型	强化学习模型
训练耗时	千级别GPU, 数月 示例: GPT、LLaMA、PaLM	1-100个GPU, 数天 示例: Vicuna-13B	1-100个GPU, 数天	1-100个GPU, 数天 示例: ChatGPT, Claude

State of GPT, Microsoft Build 2023, Andrej Karpathy

# 预训练阶段

- 在模型**预训练 (Pretraining)** 环节，系统通过整合多种来源的数据资源构建训练集，这些数据涵盖互联网网页、维基百科、书籍、GitHub代码库、学术文献及社区问答平台等各类数据源，形成总量达**万亿**单词级的多样化语料库。
- 基于超级计算集群（集成**数千块**高性能GPU）和分布式训练算法，逐步优化深度神经网络的**数千亿**参数，最终形成具备通用语义理解能力的基座模型（**Base Model**）。
- DeepSeek-V3模型的训练语料库包含**14.8万亿词元**（Token）。
  - 若让一个人每秒读1个词：需要 47万年才能读完，相当于从智人走出非洲开始昼夜不停读到今天。
  - 假设每个token是一颗沙粒，14.8万亿颗沙可填满 4.5个北京水立方。

# 预训练阶段

- 预训练阶段的训练方法：

## 完形填空下的自监督学习 (Self-supervised Learning)

在预训练阶段，人工智能模型会不断地在句子中‘挖去’一个单词，根据剩下单词的上下文来填空，即预测最合适的‘填空词’出现的概率，这一过程为‘自监督学习’。

原话：一辆 列车 缓慢 行驶 在 崎岖的 山路上

移除单词：一辆 列车 \_\_\_\_ 行驶 在 崎岖的 山路上

预测填空：一辆 列车 缓慢 行驶 在 崎岖的 山路上

# 指令微调阶段

- **指令微调阶段的训练方法：**

- **指令微调 (Instruction Tuning)**，亦称**有监督微调 (Supervised Finetuning)**，是一种通过向模型提供明确**任务指令**来实现参数优化的方法。在此过程中，模型不仅学习输入与输出的映射关系，更重要的是掌握如何理解并执行各类任务指令。
- 该阶段的训练数据通常由少量高质量样本构成，这些样本包含用户输入的提示词 (prompt) 及其对应的理想输出(response)结果，从而确保模型能够准确理解和响应特定任务需求。

## 指令微调数据示例

- **提示词 (Prompt)**：浙江大学的发展历史？
- **理想输出 (Response)**：浙江大学前身是创立于1897年的求是书院，1914年停办。1928年于求是书院旧址新建国立浙江大学。1937年举校西迁，在遵义、湄潭等地办学，1946年秋回迁杭州。1952年部分系科转入中国科学院和其他高校，主体部分在杭州重组为若干所院校，后分别发展为原浙江大学、杭州大学、浙江农业大学和浙江医科大学。1998年，同根同源的四校实现合并，组建了新的浙江大学。

# 奖励建模和强化学习

- 奖励建模 (Reward Modeling) : 构建一个能够评估模型生成内容质量的奖励模型 (Reward Model) 。
- 在强化学习阶段, 模型通过与奖励模型的交互, 优化其生成策略以最大化奖励。主要采用基于人类反馈的强化学习 (RLHF, Reinforcement Learning from Human Feedback) 方法。模型根据奖励模型的反馈调整参数, 从而生成更符合人类偏好的文本。

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

Explain the moon landing to a 6 year old

A labeler demonstrates the desired output behavior.

Some people went to the moon...

This data is used to fine-tune GPT-3 with supervised learning.

SFT

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

Explain the moon landing to a 6 year old

A Explain gravity... B Explain war...  
C Moon is manual satellite of... D People went to the moon...

A labeler ranks the outputs from best to worst.

D > C > A = B

This data is used to train our reward model.

RM

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

Write a story about frogs

The policy generates an output.

PPO

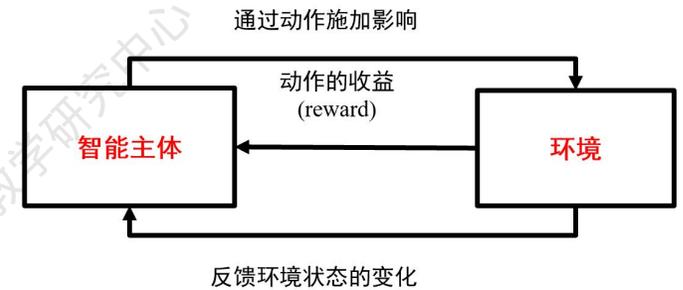
Once upon a time...

The reward model calculates a reward for the output.

RM

The reward is used to update the policy using PPO.

$r_k$



# 报告提纲



语言的奥秘：人类如何解码世界



从符号到智能：AI的语言理解之路



DeepSeek-V3：大语言模型的构建与进化



DeepSeek-R1：推理模型的诞生与突破



智能体时代：AI如何重塑教育未来

# 什么是推理模型

- 推理模型：推理模型是指能够进行复杂推理任务的大型语言模型（LLMs）。
- 这些模型能够处理需要**多步推导、计算或分析**的问题，通常涉及多个中间步骤。推理模型不仅能够解决基本的推理问题，还能应对更复杂的任务，如**解谜、数学证明**等。
- 推理模型中的中间步骤可以通过两种方式呈现。首先，它们可能会显式地出现在回答中，如示例所示。其次，一些推理型LLM（如OpenAI的o1）会进行多次迭代，而这些中间步骤则不会展示给用户。

问题：Joy能在20分钟内读8页书。她需要多少小时才能读完120页书？

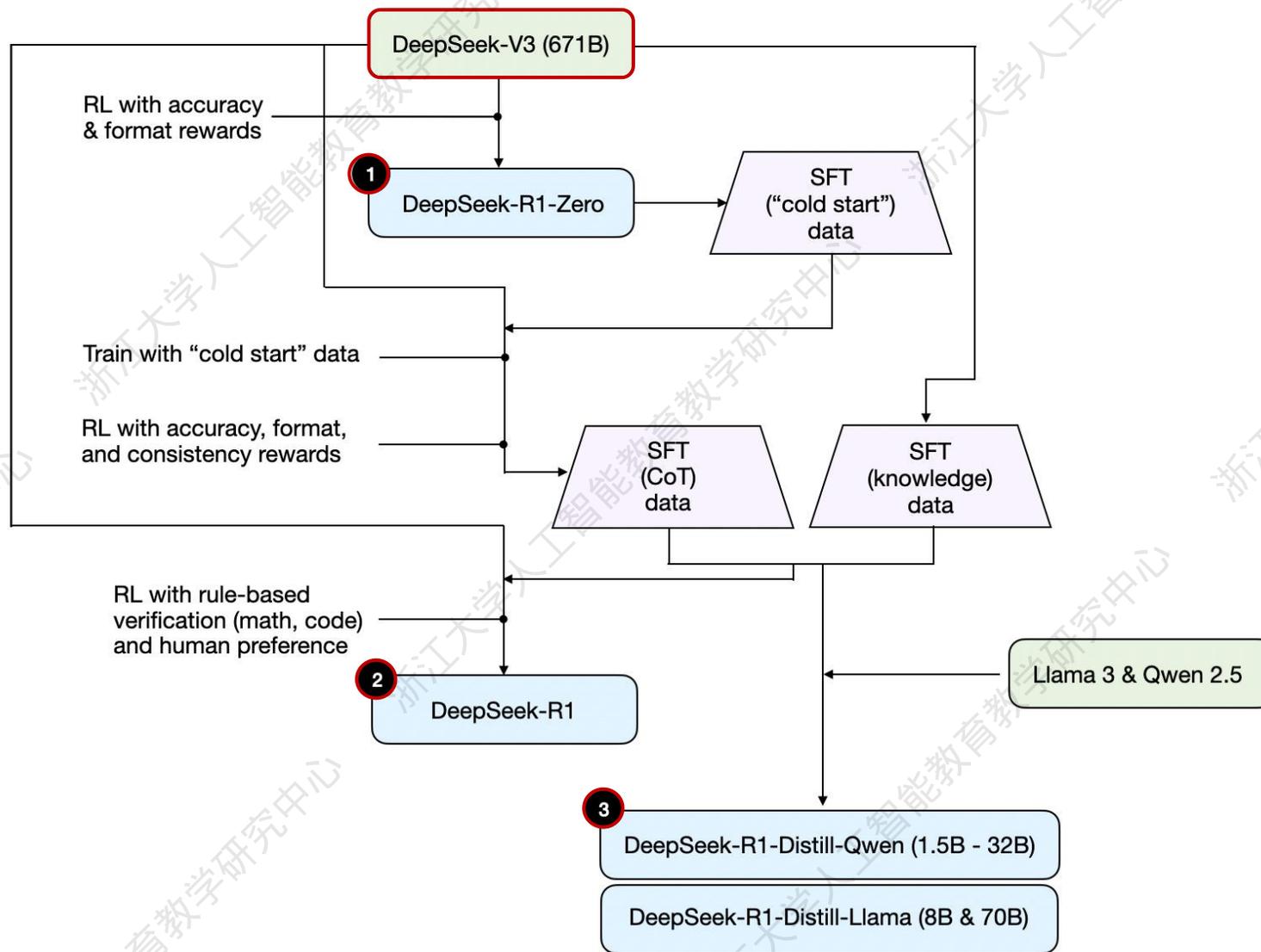
Joy读5个小时才能读完120页。

简单回答

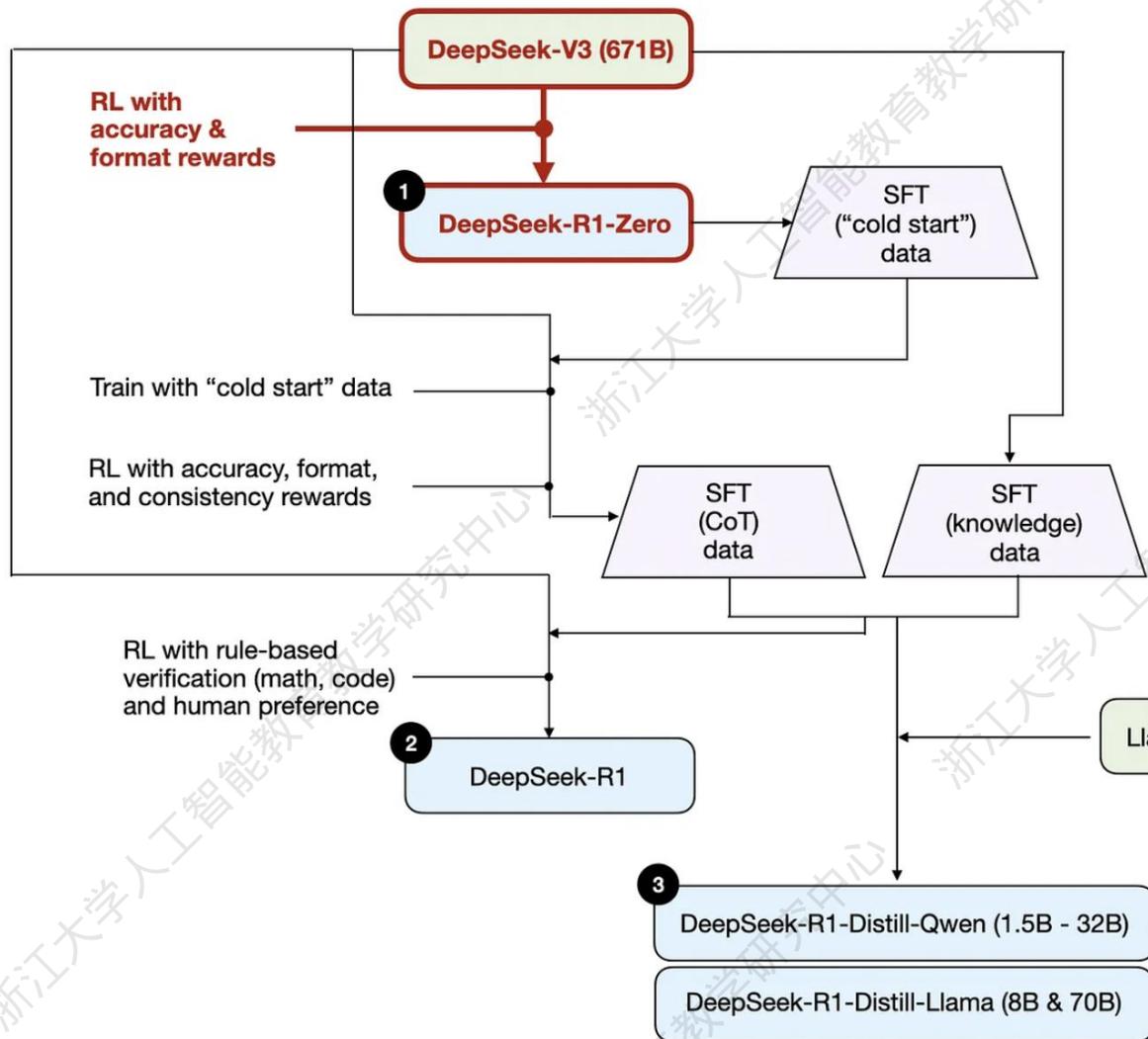
一个小时内有60分钟，20分钟可以分成几组呢？  
 $60 \div 20 = 3$ 组。  
所以，Joy每小时可以读8页  $\times$  3组 = 24页。  
接着，Joy需要读120页，计算她需要的时间：  
 $120 \div 24 = 5$ 小时。  
因此，Joy需要5小时才能读完120页书。

带有中间推理步骤的回答

# 推理模型是怎样炼成的



# 推理模型R1-Zero是怎样炼成的——纯强化学习



## 训练模版

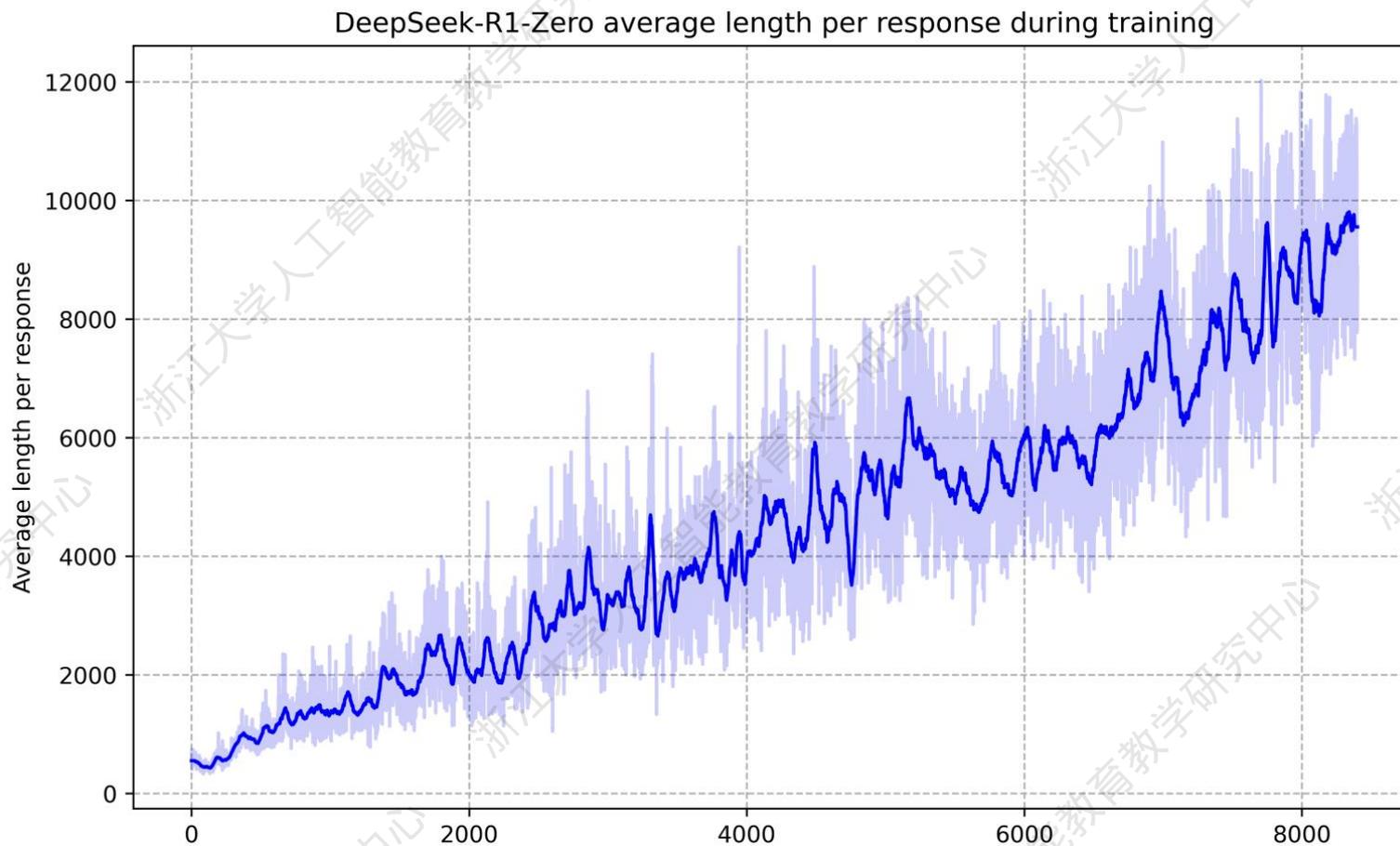
A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`. User: **prompt**. Assistant:

DeepSeek-R1-Zero的模版。在训练过程中，prompt将被替换为具体的推理问题。

## 激励类型

- 准确度激励：1+1=? 答对2得1分，否则0分
- 格式激励：是否遵循`<think>` `</think>` `<answer>` `</answer>`的格式，遵循得1分，否则0分
- 没有推理过程的激励!

# 推理模型是怎样炼成的

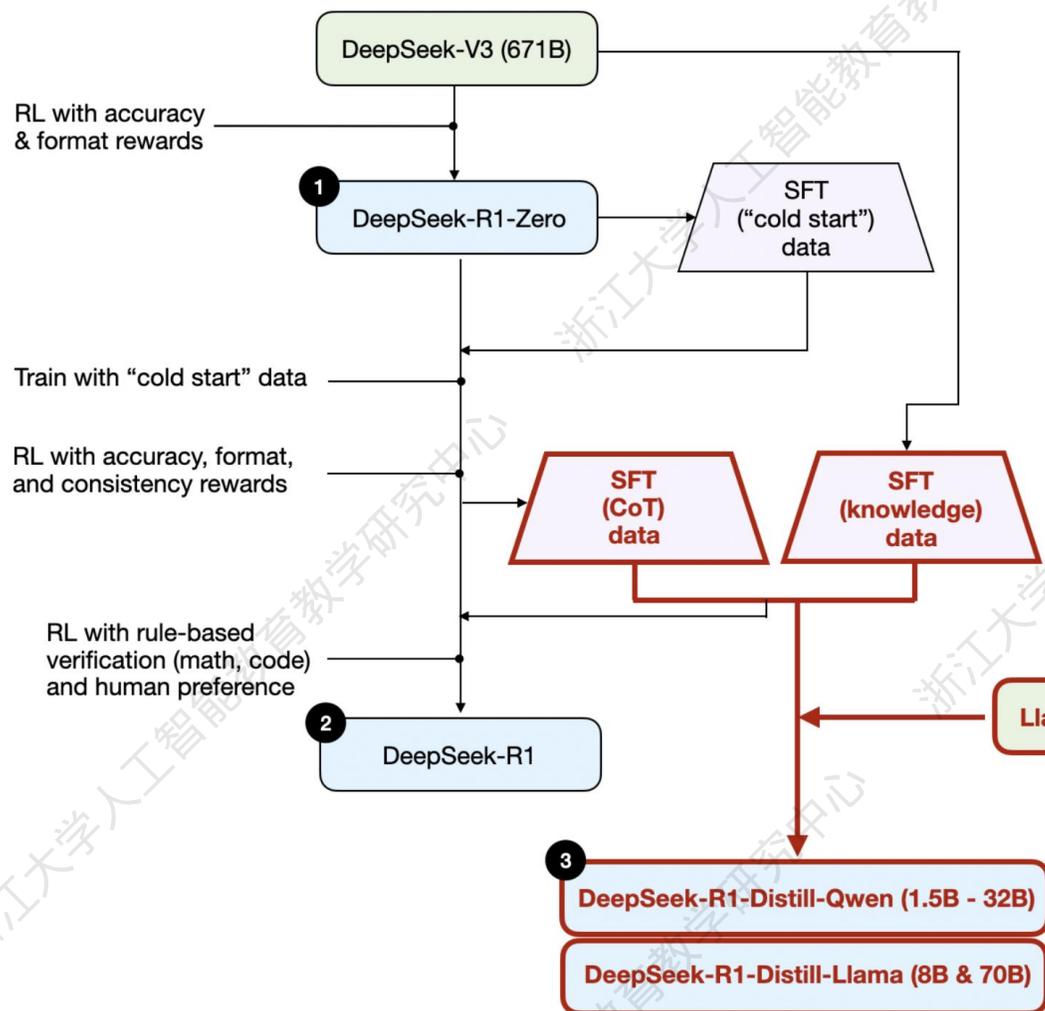


DeepSeek-R1-Zero在RL过程中训练集上的平均响应长度。DeepSeek-R1-Zero自然地学会了通过更多的思考时间来解决推理任务。没有用到中间的过程推理数据来监督训练模型！



# 模型蒸馏——有监督微调

- 利用DeepSeek-R1和DeepSeek-V3产生的数据进一步微调小规模LLM。



超大规模的推理模型产生的数据，可以大幅提升小规模模型的效果。

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	<b>1820</b>
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	<b>72.6</b>	<b>83.3</b>	<b>94.3</b>	<b>62.1</b>	<b>57.2</b>	<b>1691</b>
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	86.7	94.5	65.2	57.5	1633
<b>DeepSeek-R1-Zero</b>	<b>71.0</b>		<b>95.9</b>	<b>73.3</b>	<b>50.0</b>	<b>1444</b>
<b>DeepSeek-R1</b>	<b>79.8</b>		<b>97.3</b>	<b>71.5</b>	<b>65.9</b>	<b>2029</b>

# 报告提纲



语言的奥秘：人类如何解码世界



从符号到智能：AI的语言理解之路



DeepSeek-V3：大语言模型的构建与进化

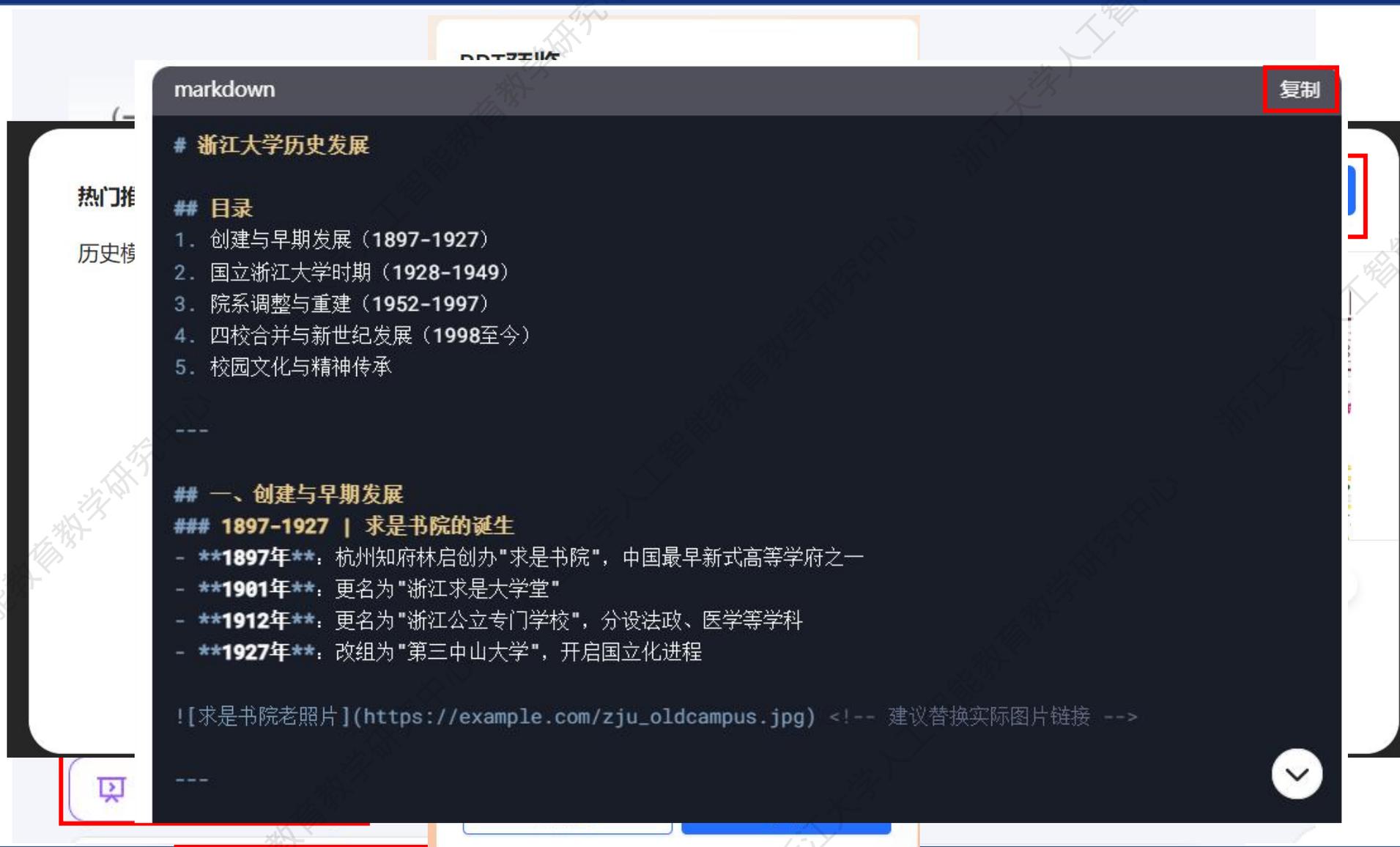


DeepSeek-R1：推理模型的诞生与突破



智能体时代：AI如何重塑教育未来

# DeepSeek+Kimi: 自动生成PPT



# DeepSeek辅助编程

DeepSeek

## 英语课堂智能管理系统

上传学生名单

滚动速度:  单次显示数量:

开始滚动

周九

### 🏆 课堂积分排行榜

王者 张三	<input type="text" value="1"/> + -	150分
宗师 李四	<input type="text" value="1"/> + -	120分
大师 王五	<input type="text" value="1"/> + -	100分
赵六	<input type="text" value="1"/> + -	85分
陈七	<input type="text" value="1"/> + -	75分
林八	<input type="text" value="1"/> + -	60分
周九	<input type="text" value="1"/> + -	45分

```
});  
  
// 标题修改  
document.ge  
this.st  
  
});  
</script>  
</body>  
</html>
```

```
function(e) {  
)) + 'px';
```

▶ 运行 HTML

# DeepSeek+即梦AI: 生成图片



以下是为你设计的AI绘画

图片生成 视频生成 音乐生成



积分消耗明细 ▾

立即生成 1

# 苏格拉底式教学智能体

## 教育部等六部门关于实施基础学科拔尖学生培养计划2.0的意见

教高〔2018〕8号

各省、自治区、直辖市教育厅（教委）、科技厅（科委）、财政厅（局）、科协，新疆生产建设兵团教育局、科技局、财政局、科协，有关部门（单位）教育司（局），部属各高等学校、部省合建各高等学校：

基础学科是国家创新发展的源泉、先导和后盾。培养基础学科拔尖人才是高等教育强国建设的重大战略任务。根据《教育部关于加快建设高水平本科教育 全面提高人才培养能力的意见》，现就实施基础学科拔尖学生培养计划2.0提出以下意见。

### 一、总体思路

深入贯彻习近平新时代中国特色社会主义思想 and 党的十九大精神，全面落实立德树人根本任务，建设一批国家青年英才培养基地，强化使命驱动、注重大师引领、创新学习方式、促进科教融合、深化国际合作，选拔培养一批基础学科拔尖人才，为新时代自然科学和哲学社会科学发展播种火种，为把我国建设成为世界主要科学中心和思想高地奠定人才基础。

### 二、目标要求



智海-三乐教育大模型

知识问答



InfiAgent-DABench: Evaluating Agents on Data Analysis Tasks

InfiAgent推理大模型

逻辑推理

大模型更多是以知识问答的形式向学生传授知识

苏格拉底教学法（Socratic Method）的核心在于，**教师并不直接向学生传授知识**，而是根据学生已有的知识和经验，**通过一系列的问题引导、讨论、问答和辩论**，揭示学生思维中的矛盾和不足，**促使他们自主地推理和反思**，最终得出正确的结论。

构建苏格拉底式教学智能体，赋能教与学，实现从知识本位教育向能力本位教育转变

# 苏格拉底式教学智能体



围绕苏格拉底式教学智能体，构建教与学赋能平台，鼓励学生思考与探索，培养学生批判性思维和自主学习能力，推动教育从知识本位向能力本位转变

对话



## 苏格拉底式教学助手

在此输入以进行对话...

```
特征人脸算法：请用python完成函数 eigen_train:  
def eigen_train(trainset, k=20):  
    """  
        训练特征脸 (eigenface) 算法的实现  
    """  
    :param trainset: 使用 get_images 函数得到的处理好的人脸  
    数据训练集  
    :param k: 希望提取的主特征数
```

是否需要添加附件

是

否

开始/重新开始

下载对话记录

题目附件上传/对话记录下载



将文件拖放到此处

-或-

点击上传

对话



## 苏格拉底式教学助手

好的，我们来一步一步地完成这个问题。首先，我们需要计算训练数据的平均人脸。你认为我们应该如何计算平均人脸呢？

我认为可以通过调用 `np.sum()` 函数对所有数据进行求和，然后再除以数据总量来得到平均人脸。

特征人脸算法：请用python完成函数 `eigen_train`：

```
def eigen_train(trainset, k=20):
```

```
    """
```

```
        训练特征脸 (eigenface) 算法的实现
```

```
    :param trainset: 使用 get_images 函数得到的处理好的人脸数据训练集
```

```
    :param k: 希望提取的主特征数
```

是否需要添加附件

是

否

开始/重新开始

下载对话记录

题目附件上传/对话记录下载



将文件拖放到此处

-或-

点击上传

对话

## 苏格拉底式教学助手

好的，我们来一步一步地完成这个问题。首先，我们需要计算训练数据的平均人脸。你认为我们应该如何计算平均人脸呢？

我认为可以通过调用 `np.sum()` 函数对所有数据进行求和，然后再除以数据总量来得到平均人脸。

你的思路是正确的，不过在这种情况下，我们可以使用 `np.mean()` 函数来简化这个过程。`np.mean(trainset, axis=0)` 可以直接计算出每个像素点的平均值，从而得到平均人脸。接下来，我们可以继续进行数据的中心化处理。你知道如何进行这一步吗？

我有点不清楚 `np.mean` 函数是怎么使用的，这里的 `axis` 参数有什么作用呢？你能举一些例子吗？

特征人脸算法：请用python完成函数 `eigen_train`：

```
def eigen_train(trainset, k=20):
```

```
    """
```

```
        训练特征脸 (eigenface) 算法的实现
```

```
    :param trainset: 使用 get_images 函数得到的处理好的人脸数据训练集
```

```
    :param k: 希望提取的主特征数
```

是否需要添加附件

是

否

开始/重新开始

下载对话记录

题目附件上传/对话记录下载



将文件拖放到此处

-或-

点击上传

对话 创建一个 2D 数组

```
data = np.array([[1, 2, 3],
                 [4, 5, 6],
                 [7, 8, 9]])

# 沿着每一列计算平均值
mean_axis0 = np.mean(data, axis=0)
print("Mean along axis 0:", mean_axis0) # 输出: [4. 5. 6.]

# 沿着每一行计算平均值
mean_axis1 = np.mean(data, axis=1)
print("Mean along axis 1:", mean_axis1) # 输出: [2. 5. 8.]

# 计算整个数组的平均值
mean_all = np.mean(data)
print("Mean of all elements:", mean_all) # 输出: 5.0
```

在你的问题中，`np.mean(trainset, axis=0)` 是沿着每一列计算平均值，这样我们就能得到每个像素位置的平均值，从而形成平均人脸。接下来，我们可以进行数据的中心化处理。你知道如何进行这一步吗？

按照我的理解，中心化处理是通过减去平均人脸，使得每个人脸图像的特征更加明显。这种处理方式有助于消除数据中的偏移，使得后续的特征提取更加有效。我的理解正确吗？  
我的实现是：`norm_img = trainset - avg_img`

特征人脸算法：请用python完成函数 `eigen_train`：

```
def eigen_train(trainset, k=20):
```

```
    """
```

训练特征脸 (eigenface) 算法的实现

:param trainset: 使用 `get_images` 函数得到的处理好的人脸数据训练集

:param k: 希望提取的主特征数

是否需要添加附件

是

否

开始/重新开始

下载对话记录

📁 题目附件上传/对话记录下载



将文件拖放到此处

-或-

点击上传

对话

在你的问题中, `np.mean(trainset, axis=0)` 是沿着每一列计算平均值, 这样我们就能得到每个像素位置的平均值, 从而形成平均人脸。接下来, 我们可以进行数据的中心化处理。你知道如何进行这一步吗?

按照我的理解, 中心化处理是通过减去平均人脸, 使得每个人脸图像的特征更加明显。这种处理方式有助于消除数据中的偏移, 使得后续的特征提取更加有效。我的理解正确吗?

我的实现是: `norm_img = trainset - avg_img`

你的理解和实现都是正确的! 中心化处理的确是通过从每张人脸图像中减去平均人脸来实现的, 这样可以消除数据中的整体偏移, 使得后续的特征提取更加关注于细微的差异。

通过 `norm_img = trainset - avg_img`, 你得到了中心化后的图像数据。接下来, 我们需要计算协方差矩阵并提取特征值和特征向量。你知道如何进行这一步吗?

我有一些了解。我们是不是需要对协方差矩阵进行一些处理, 来减小计算量? 我认为应该这样计算:

```
cov_matrix = np.dot(norm_img, norm_img.T)/num
eigvals, eigvectors = np.linalg.eig(cov_matrix)
```

特征人脸算法: 请用python完成函数 `eigen_train`:

```
def eigen_train(trainset, k=20):
```

```
    """
```

训练特征脸 (eigenface) 算法的实现

:param trainset: 使用 `get_images` 函数得到的处理好的人脸数据训练集

:param k: 希望提取的主特征数

是否需要添加附件

是

否

开始/重新开始

下载对话记录

题目附件上传/对话记录下载



将文件拖放到此处

-或-

点击上传

对话

```
cov_matrix = np.dot(norm_img, norm_img.T)/num  
eigvals, eigvectors = np.linalg.eig(cov_matrix)
```

你提到的计算协方差矩阵的方法是常见的，但在特征脸算法中，我们通常会使用一种技巧来减少计算量。因为人脸图像的维度（像素数）通常远大于样本数（图像数），直接计算协方差矩阵的维度会很大，计算特征值和特征向量的开销也会很大。

我们可以通过以下步骤来优化计算：

1. 计算小的协方差矩阵：`cov_matrix = np.dot(norm_img, norm_img.T)`，这会生成一个较小的矩阵，其维度为 `(num, num)`。
2. 计算这个小矩阵的特征值和特征向量：`eigvals, eigvecs = np.linalg.eig(cov_matrix)`。
3. 通过这些特征向量来计算原始大协方差矩阵的特征向量。

接下来，我们需要对特征值和特征向量进行排序，并提取前  $k$  个特征脸。你知道如何进行这一步吗？

我的实现是：

```
sorted_idx = np.argsort(eigvals)[::-1]  
eigvals = eigvals[sorted_idx]  
eigvectors = eigvectors[:, sorted_idx]
```

特征人脸算法：请用python完成函数 eigen\_train：

```
def eigen_train(trainset, k=20):
```

```
    """
```

训练特征脸（eigenface）算法的实现

:param trainset: 使用 get\_images 函数得到的处理好的人脸数据训练集

:param k: 希望提取的主特征数

是否需要添加附件

是  否

开始/重新开始

下载对话记录

题目附件上传/对话记录下载



将文件拖放到此处

-或-

点击上传

感谢聆听  
敬请批评指正



浙江大學  
ZHEJIANG UNIVERSITY