



机器学习

夏令营第一讲

汤远航

NLP group, NJU



NLP

从ChatGPT说起



机器学习是什么？ChatGPT和机器学习的关系是什么？

- ✓ 正在搜索: 机器学习是什么
- ✓ 正在搜索: ChatGPT和机器学习的关系是什么
- ✓ 正在为你生成答案...

您好，这是Bing。很高兴为您解答。😊

机器学习是一门人工智能的科学，该领域的主要研究对象是人工智能，特别是如何在经验学习中改善具体算法的性能。机器学习是对能通过经验自动改进的计算机算法的研究。机器学习是用数据或以往的经验，以此优化计算机程序的性能标准。¹

ChatGPT是一个由浮点数参数表示的深度学习神经网络大模型，属于深度学习的框架。它是一个生成型预训练变换模型，目的是为了使用深度学习生成人类可以理解的自然语言。² ChatGPT和机器的关系是，它是机器学习（包含深度学习）的一个应用，利用了机器学习的方法和技术来实现自然语言处理和对话生成等任务。^{3 4}

了解详细信息:

[1. zh.wikipedia.org](https://zh.wikipedia.org)

[2. openai.com](https://openai.com)

[3. zhuanlan.zhihu.com](https://zhuanlan.zhihu.com)

[+4 更多](#)

1 共 30 ●

AI还能做什么



微软推出办公“智能副驾”Microsoft 365 Copilot

2023年 3月 17日

Mr. Ranedeer: Your personalized AI Tutor!

Unlock the potential of GPT-4 with Mr. Ranedeer AI Tutor, a customizable prompt that delivers personalized learning experiences for users with diverse needs and interests.

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.



机器学习——学习?



$$1+1=2$$

$$1+2=3$$

$$1+3=4$$

.....



学习数手指头.img



$$2+3=???$$

5 !!!

总结**经验**，形成**技能**



机器学习——机器？

- 学习: 总结**经验**, 形成技能
- but ...
 - 机器的“经验”?
 - 机器的“**大脑**”?
 - 以及...怎么“**学习**”?
 - 又如何评估机器学习的**效果**?
- 机器学习
 - **数据**: 提供**经验**
 - **模型**: 装载**大脑**
 - **训练**: 进行**学习**
 - **测试**: 评估**效果**

数据



图像



音频

初次见面，我已经关注你很久了。我叫纳西妲，别看我像个孩子，我比任何一位大人都了解这个世界。所以，
我可以用我的知识，换取你路上的见闻吗？

文本

特征	数值
歪的概率	1
氪金	0
平均出金	89

特征向量: [1, 0, 89, ...]

- 不同数据形式最终都要转化为数值表示
 - 人工抽取: 统计学习
 - 学习得到: 深度学习

数据 / 任务



- 任务要求决定数据形式
 - 分类: 文本 + 标签
 - 翻译: 中文句子 + 英文句子
 - AI 孙燕姿: 孙燕姿的作品音频等
 - 任务分类
 - 有监督: 需要人类标注, e.g. ImageNet (图1, 猫), (图2, 狗)
 - 无监督: 无需标注信息, e.g. 学习
 - 自监督: 从无标注数据中获得监督信息
 - ...
 - 讨论:
 1. ChatGPT是使用什么数据训练的?
 2. 又是用什么任务训练的?
-

模型



- 模型就是一个函数
 - 输入: 文本, 图片, 音频, 特征 ...
 - 输出: 类别, 下一个词, 字幕 ...
 - 映射: 树, 线性分类器, 神经网络 ...
- 如何确定这个函数
 - 选择一个映射形式, e.g. 神经网络
 - 选择一个优化方法, e.g. 梯度下降
 - 拟合参数
- ChatGPT使用了什么模型?

机器学习 \approx 构建一个映射函数

- 语音识别 $f(\text{audio}) = \text{"你好"}$
- 图像识别 $f(\text{image of 9}) = \text{"9"}$
- 围棋 $f(\text{Go board}) = \text{"6-5"}$
- 机器翻译 $f(\text{"你好!"}) = \text{"Hello!"}$

训练



- 训练一般指拟合模型参数或结构
- 我们希望训练后模型效果比较好
 - 训练时, 如何评估模型的效果?
- 目标函数 / 误差函数 / 损失函数
 - 评估模型训练时在数据集上表现的好坏
 - 需要根据任务特性来设计
 - 希望损失越小, 模型在数据集上表现越好

均方误差 (回归):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

交叉熵 (分类):

$$L = - \sum_{i=1}^C y_i \log(p_i)$$

训练



- 降低损失, 形式化为最优化问题
- 为不同模型和任务设计不同损失函数和优化方法

- 比如, 线性回归

- 均方误差
- 梯度下降

回归形式: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$

损失函数: $\min_{\beta_0, \beta_1, \dots, \beta_p} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \dots - \beta_p x_{ip})^2$

- 梯度下降

- 参数的梯度(导数)的反方向更新参数: $\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$
- 需要模型的计算过程可微
- 对多种损失函数有效

深度学习



你好，这是必应。我可以帮你了解深度学习相比于机器学习的优点。😊

根据我的搜索结果¹²³，深度学习相比于机器学习的优点有以下几点：

- **学习能力强**：深度学习可以从大量的数据中自动提取特征和规律，而不需要人工设计和选择特征。深度学习可以处理复杂的非线性关系，适用于图像、语音、文本等高维数据。
- **覆盖范围广，适应性好**：深度学习可以利用多层神经网络来构建各种各样的模型，理论上可以拟合任意函数。深度学习可以应用于不同的领域和任务，例如计算机视觉、自然语言处理、推荐系统、强化学习等。
- **数据驱动，上限高**：深度学习高度依赖数据，数据量越大，其表现就越好。在许多任务中，深度学习已经超过了人类的水平，例如图像分类、人脸识别、机器翻译等。深度学习还可以通过调整参数和结构来进一步提高性能。
- **可移植性好**：深度学习有很多成熟的框架和工具，例如TensorFlow、PyTorch等，可以兼容不同的平台和设备。深度学习还可以利用迁移学习的技巧，将预训练的模型应用于新的数据集或任务，减少训练时间和成本。

了解详细信息：[1. zhuanlan.zhihu.com](#) [2. zhihu.com](#) [3. blog.csdn.net](#) [+2 更多](#)

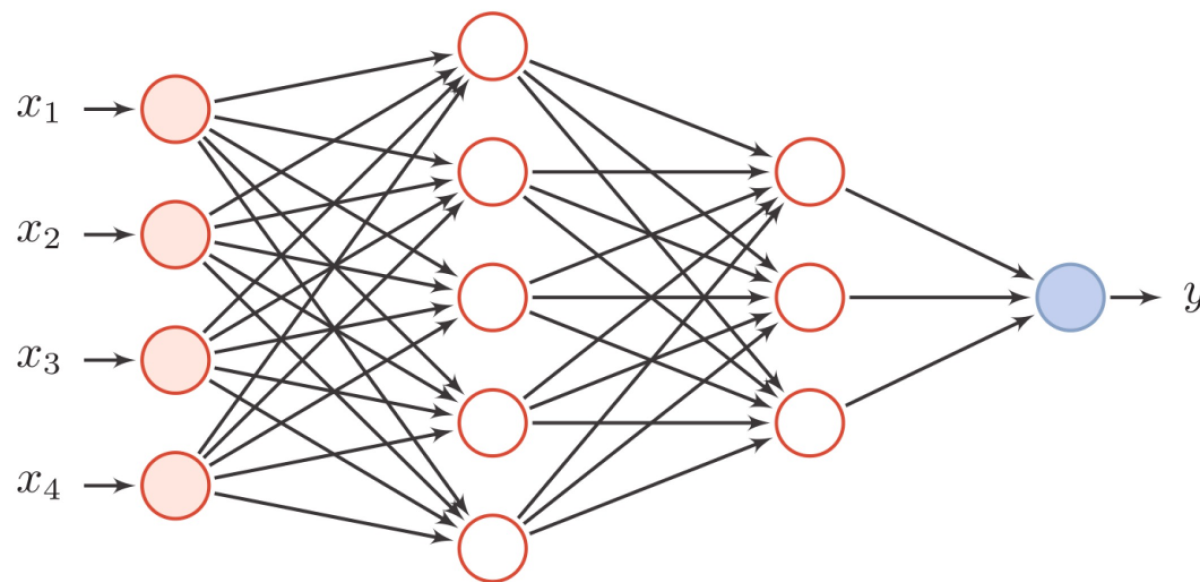
1 共 30 ●

1. new bing 给出的答复
2. 其中不乏过于绝对的表述

人工神经网络



- 不同视角
 - 仿生学: 仿照人脑结构设计的计算形式
 - 结构: 不同计算节点相互关联
 - 数学: $y = f^1 \left(f^2 \left(f^3 \left(f^4 (\dots) \right) \right) \right)$
- 和机器学习的关系
 - 人工神经网络是一种**模型**
 - 没那么玄乎, 就是一类模型而已
- 结构
 - 节点, 激活函数
 - 基本单位: 层
 - 堆叠很多层



多层感知机 (MLP)

- 单个节点的计算

- 线性组合:

$$z = w_1x_1 + \dots + w_dx_d$$

- 非线性激活: $a = \sigma(z)$

- 网络层的计算:

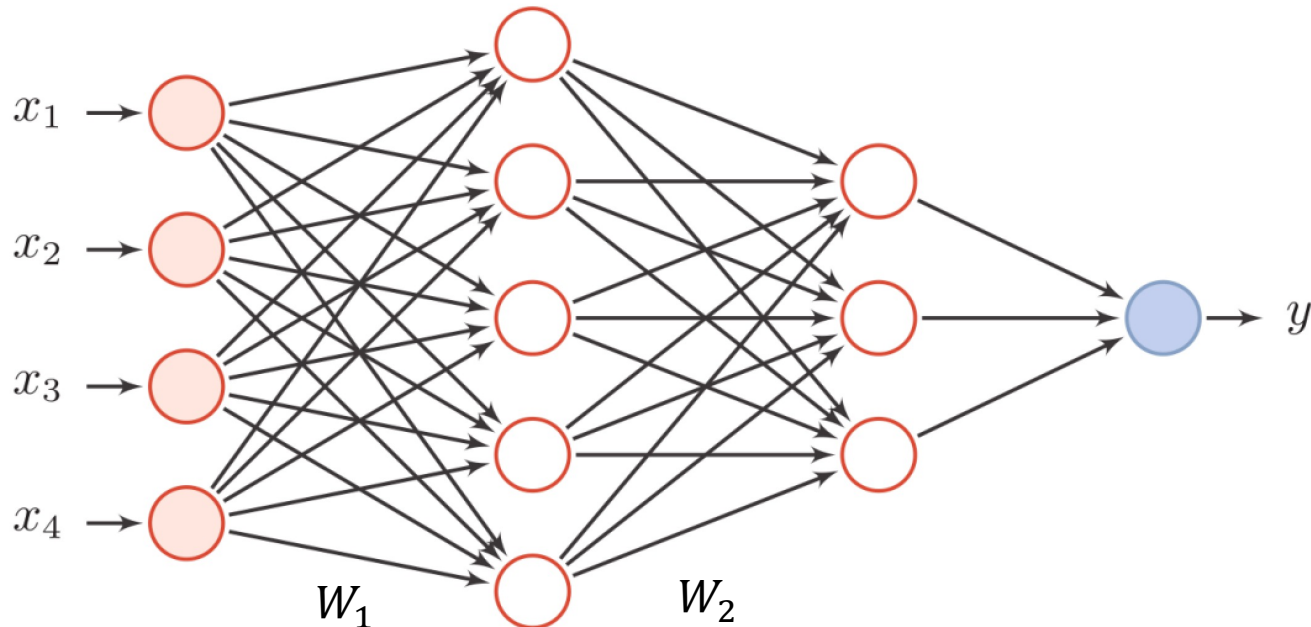
$$z_1 = W_1X + b_1$$

$$a_1 = \sigma(z_1)$$

$$z_2 = W_2a_1 + b_2$$

$$a_2 = \sigma(z_2)$$

...



- 整个模型的计算

$$L_i = \sigma(W_i a_{i-1} + b_i)$$

$$MLP(x) = L_n(L_{n-1}(\dots(L_1(x)) \dots))$$

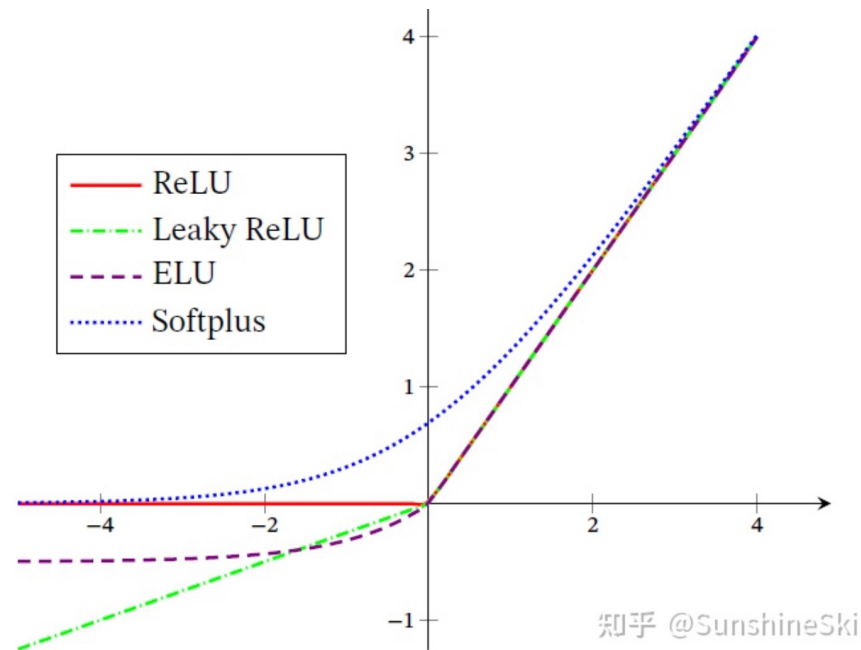
激活函数

- 计算分两种: 线性 / 非线性
- 线性函数的复合还是线性函数
 - 线性代数的特点
- 通过激活函数引入非线性
 - 拟合更复杂的映射关系
- 常见的激活函数

- $Sigmoid(x) = 1 / (1 + \exp(-x))$

- $tanh(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$

- $ReLU(x) = \max(0, x)$





特征抽取

- 深度学习一般指利用深度神经网络进行建模的机器学习方法
- 为什么要深?
 - 进行更深入的特征抽取
 - 模型浅层抽取简单特征
 - 浅层特征在网络深层组合成更复杂的特征
- 有研究指出 NLP 预训练模型BERT
 - 较浅层学习到词组特征
 - 中间层学习到句法特征
 - 较深层学习到语义特征

表示学习



- 深度学习的特征抽取特性使之更适用于图片, 文本, 语音等数据
 - 人工抽取数值特征较为困难
 - 特征通过神经网络学出来
 - 深度学习是表示学习的范例
 - 表示学习
 - 学习将输入转化为数值表示 e.g. 将文本转化为向量
 - 学得的数值表示应该反映输入的特性
 - e.g. 比较相似的输入的向量表示距离比较近
 - 文本的表示是如何学得的?
-



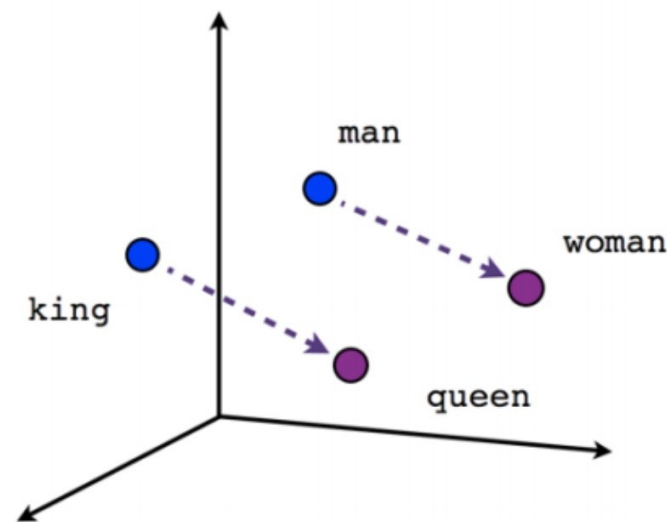
词嵌入 (embedding)

- 文本是离散符号, 而计算机更擅长进行数值计算
- Fill the gap
 - 把单词转化成数值表示
 - 每个词分配一个向量
- 人工分配词向量感觉很困难
 - 词表太大
 - 怎么分配才合理?
- 那就.....学一个?
 - 那如何设计损失呢?
 - “好”的词向量表示应该满足什么性质?

词向量



- 词的分布假设
 - 词的语义由其上下文决定
 - A word is characterized by the company it keeps -- By Firth, 1957.
- Word2vec的假设
 - 良好的词向量应该能预测句子中周围的词
- Word2vec 训练一个MLP, 从而
 - 用周围的词预测中间的词 (CBOW), 或者
 - 用中间的词预测周围的词 (Skip-gram)
- Word2vec 训练得到的词向量反映了词的语义



循环神经网络 (RNN)



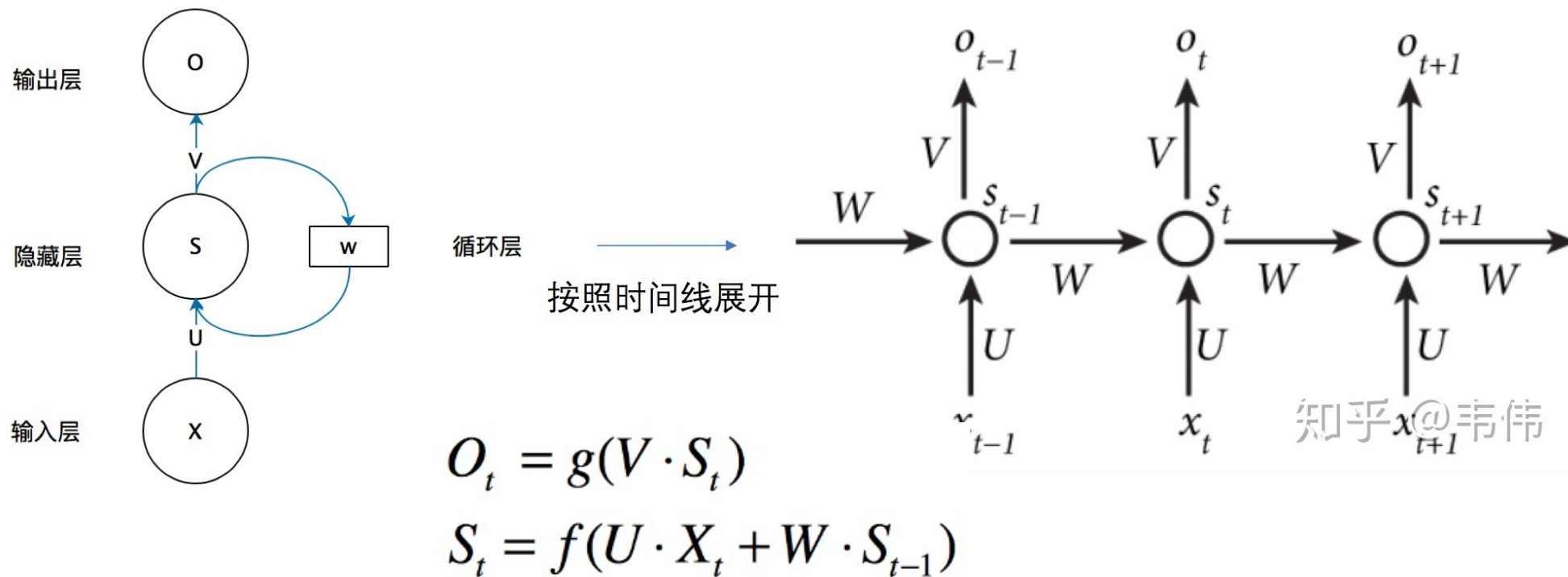
循环神经网络 (RNN) 相比于多层感知机 (MLP) 更适合处理文本数据, 因为:

1. RNN可以处理时序信息和上下文关系, 而文本通常是时序数据, 单词或字符的顺序对理解文本很重要。
2. RNN能够处理可变长度的输入, 适应不同长度的句子或段落, 而MLP需要固定大小的输入。
3. RNN可以更好地捕捉文本中的长期依赖关系和上下文依赖性。

简而言之, RNN在处理文本时更具优势, 因为它们能够更好地处理时序信息、变长输入和上下文依赖性。

1. 在Transformer架构出现之前, RNN 是 NLP 最常用的神经网络架构
2. LSTM 是最有代表性的 RNN 架构之一, 但是讲起来太烦了
3. 我们用一个简单的RNN和MLP比较一下

RNN: 例子



- 从头到尾逐个处理
- x_t : 第 t 个词
- S_t 存储 $[0, t]$ 个 token 的综合语义, 也就是语境
- 当前词的计算涉及前文的语境

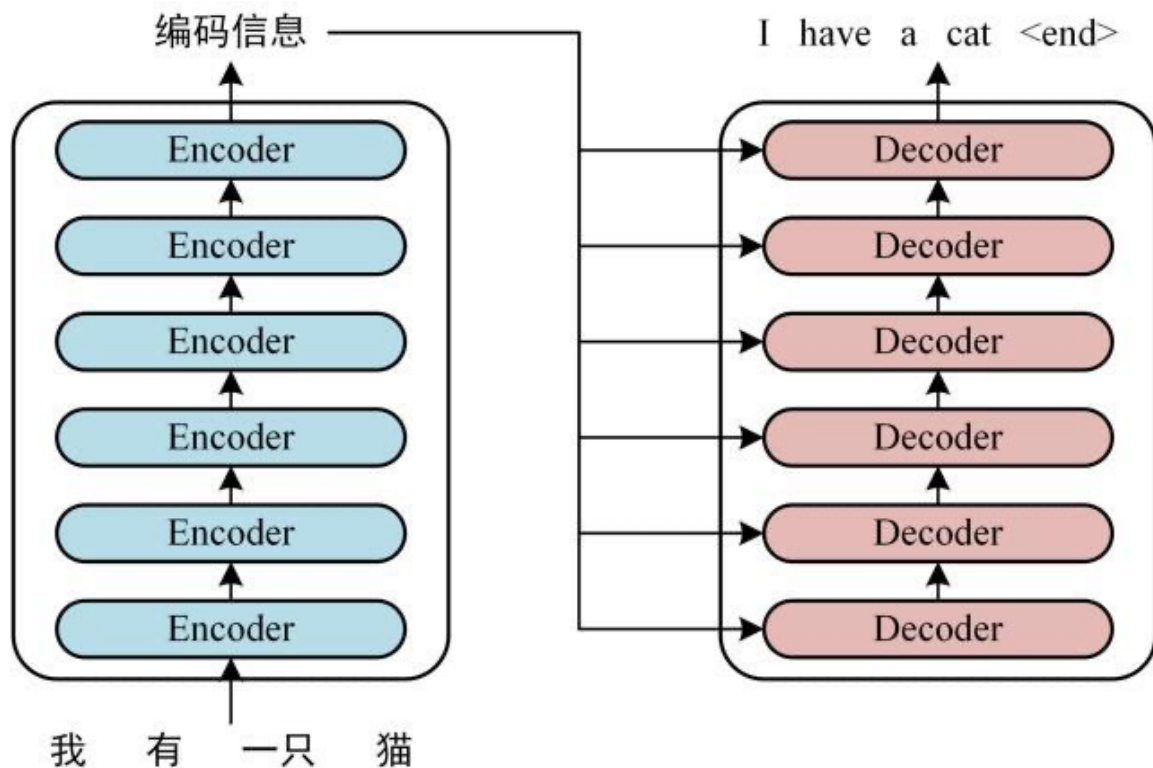
Transformer



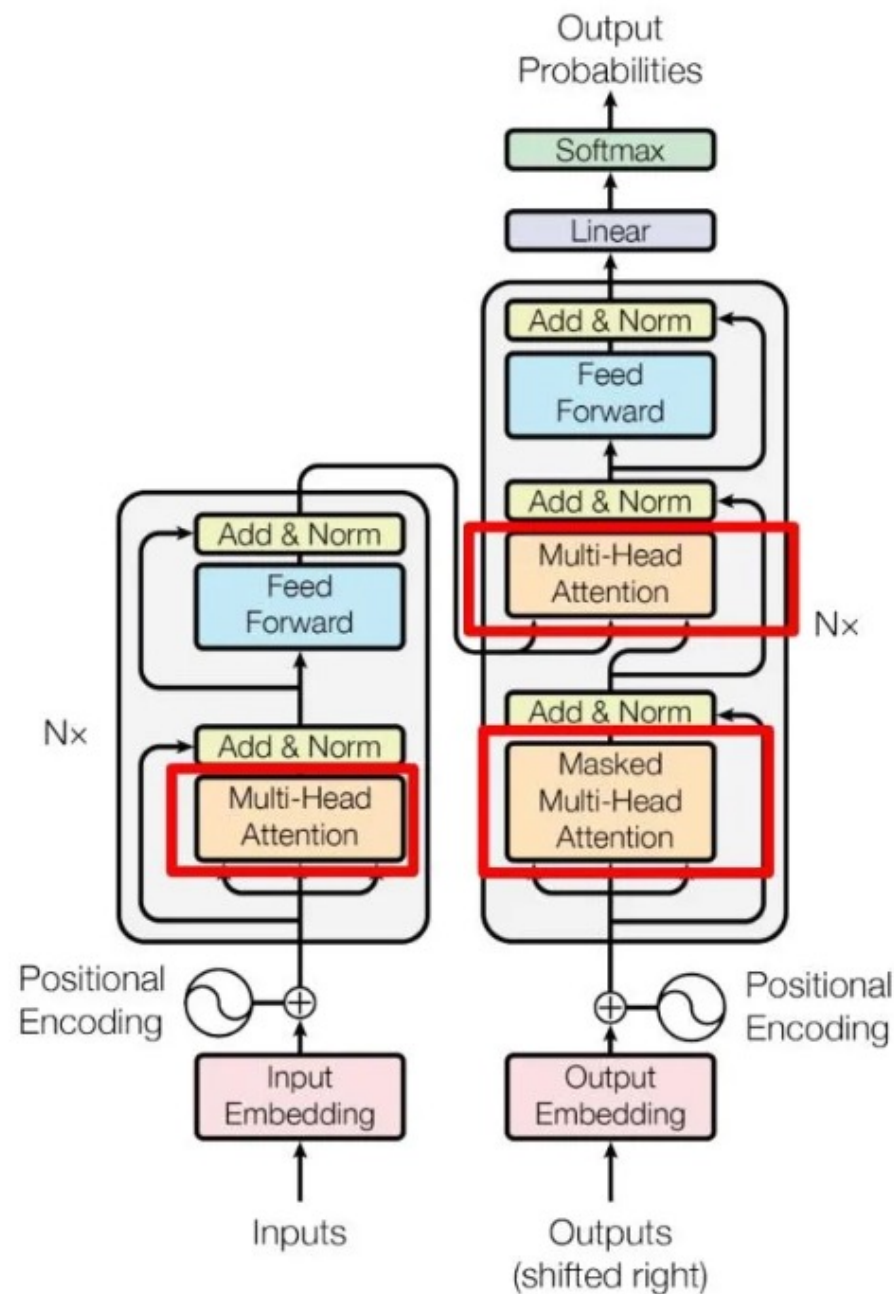
Transformer相比于LSTM的优势主要包括：

1. 并行计算：Transformer可以同时计算输入序列中的所有位置，而LSTM是逐步计算的。这使得Transformer在处理长序列时更快。
2. 长期依赖建模：Transformer使用自注意力机制，能够更好地捕捉输入序列中不同位置的长期依赖关系。
3. 可解释性：Transformer的自注意力机制使得注意力权重可视化，提供了更好的模型可解释性。

Transformer



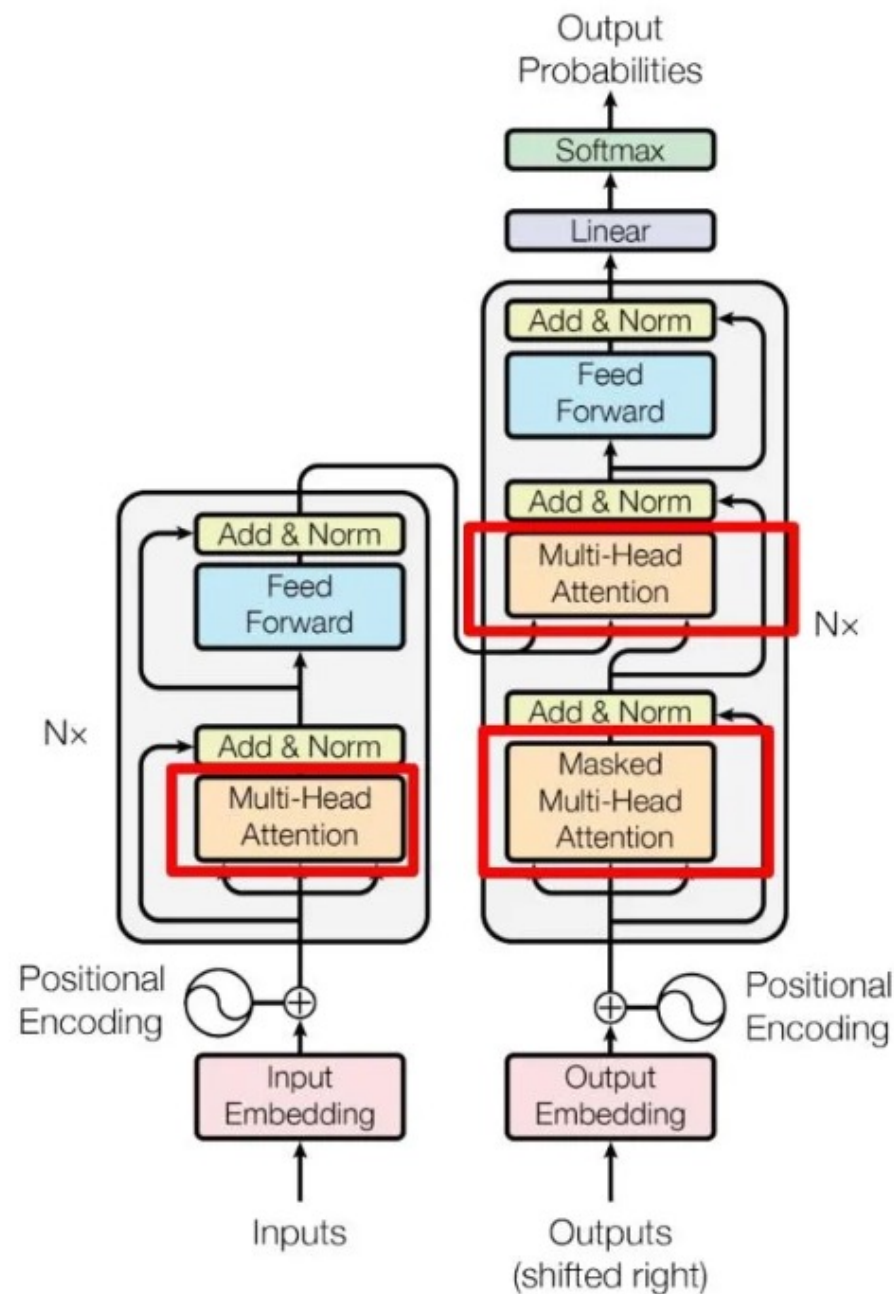
- Encoder-Decoder架构
 - Encoder负责学习表示
 - Decoder将表示解码为目标输出



Transformer



- Multi-Head Attention机制
 - RNN中, 单词无法直接和其他单词交互
 - Attention机制中, 每个词直接和其他词交互
- Add & Norm
 - Add: 残差连接
 - Norm: layer normalization
- Feed Forward
 - MLP
- Transformer 是 ChatGPT 等大模型的基础架构
 - 但是Transformer训练很不稳定, 容易训崩
 - 咋办呢?



预训练



预训练技术的出现是因为：

1. 数据稀缺：预训练模型可以利用大规模未标记数据来学习有用的特征表示。
2. 迁移学习：预训练模型可以将在一个领域上学到的知识迁移到其他相关领域，提升模型性能。
3. 初始化参数：预训练技术提供了更好的初始参数设置，使得模型更容易训练和收敛。
4. 提高效率：通过利用他人已经训练好的模型，可以避免从零开始训练，节省时间和计算资源。

综上所述，预训练技术的出现是为了解决数据稀缺、迁移学习和模型训练效率等问题。

- 目标: 学习人类语言的通用表示
- 方法: 在大规模的语料上为选定的模型架构学习一个初始化参数
- 听起来有点像word2vec?

预训练

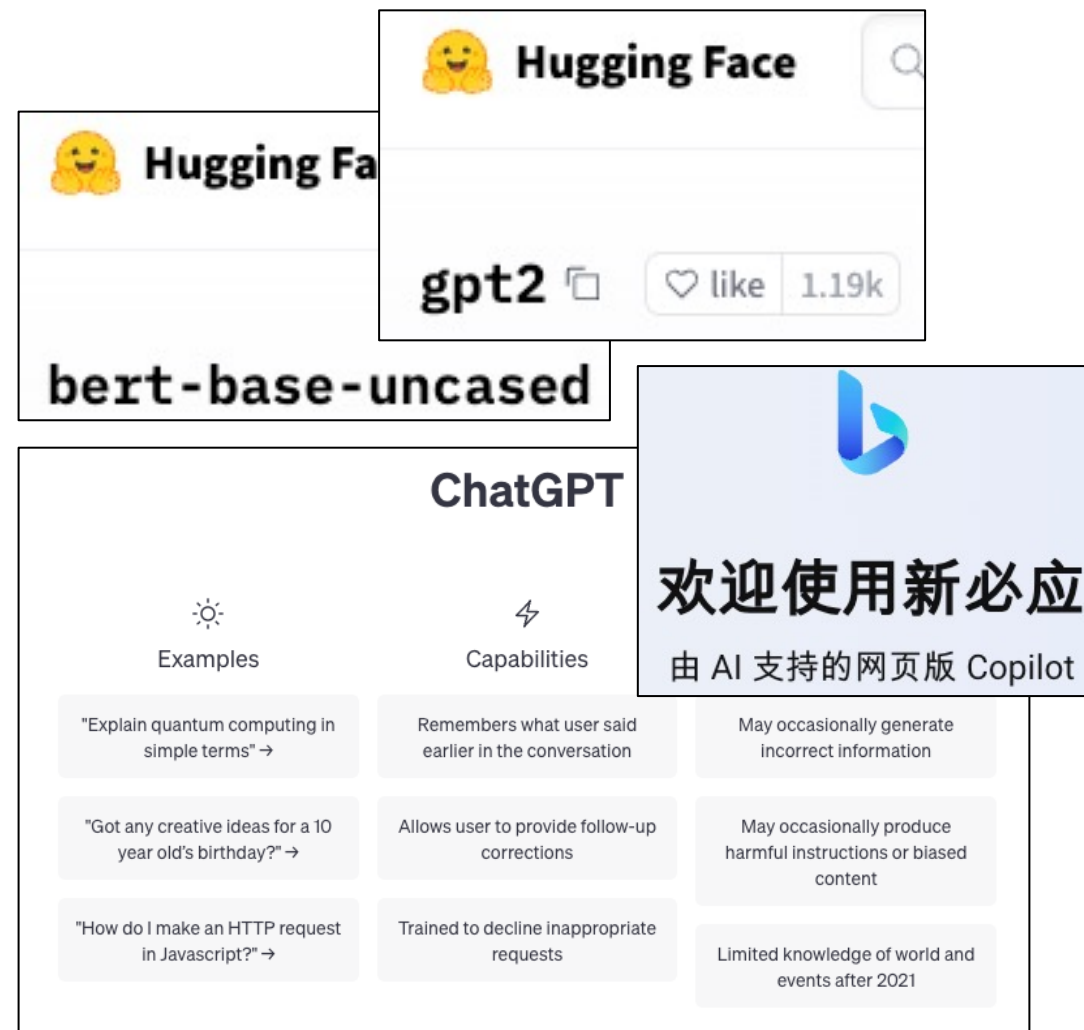


- 某个**模型**在大规模的**语料**上为某个**任务**进行训练
 - 模型: Transformer架构的各种变体
 - 语料: 互联网(wiki, 论坛, github代码), 书籍, 对话 ...
 - 任务: **预测下一个词**, 完形填空, 预测下一个句子 ...
- 产出: 模型的初始化参数
- 基本趋势
 - 模型更大: ChatGPT约175B个参数 (B: billion)
 - 数据量更大, 多样性更强
 - **大模型**出现了**通用智能**的迹象
- 大模型一般指 **10B** 参数量以上的模型
 - 无需为特定任务训练而具有多任务能力
 - 进一步微调后能遵从人类指示, 尊重人类价值观, 调用工具...

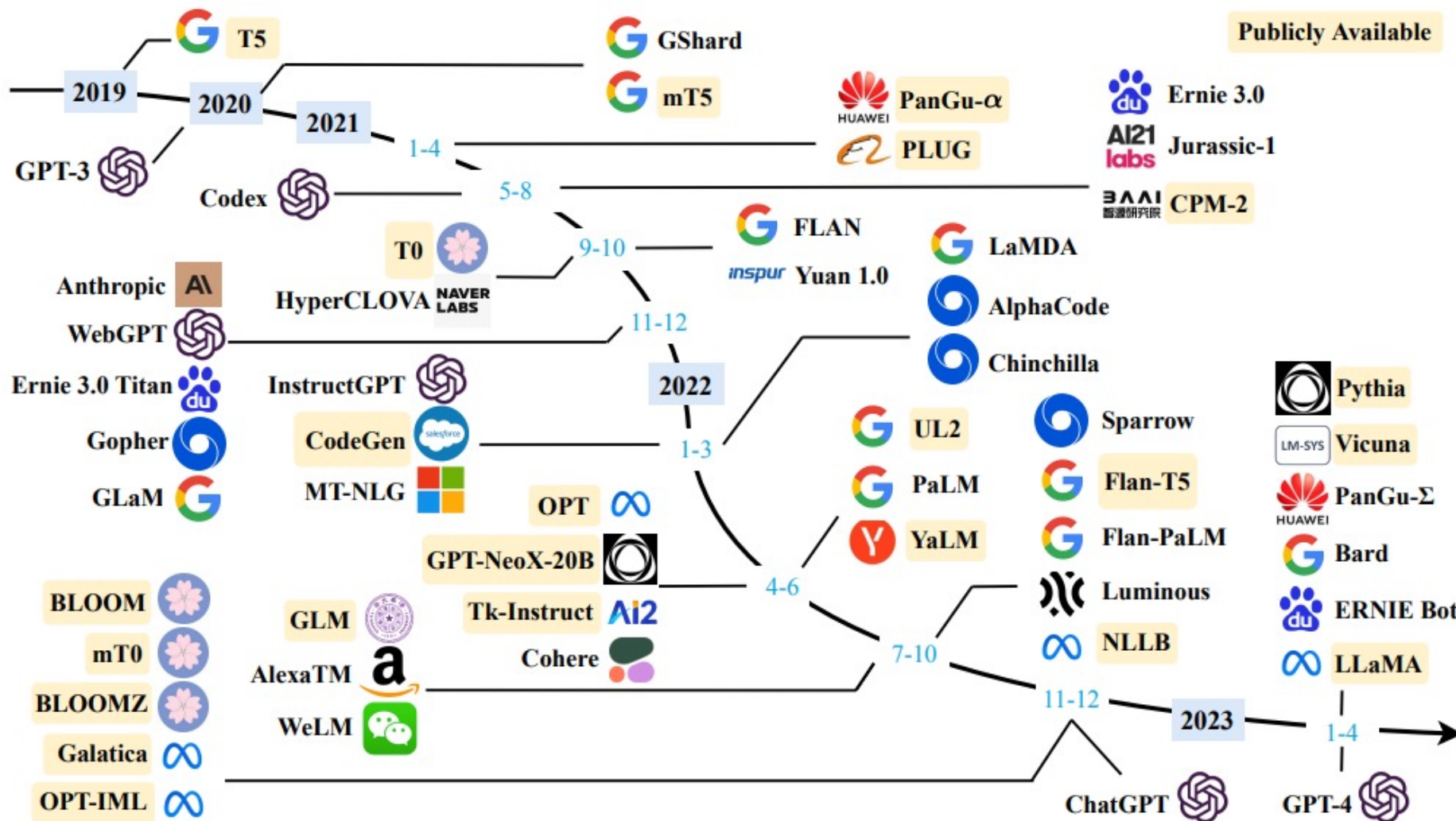
预训练



- 文本分类: BERT系列
 - 开源
 - BERT-base, RoBERTa, ...
- 文本生成: GPT2系列
 - 开源
 - GPT2-small, GPT2-large, ...
- 大模型: GPT3+
 - 未开源, 提供网页和 api 接口
 - GPT3, ChatGPT, GPT-4
- 大模型 + 插件
 - New Bing: 网页检索
 - GPT4 plugin: GPT4调用各种工具



大模型



炼丹术



- 训练深度神经网络十分tricky
 - Epoch, learning rate, 参数初始化, 归一化, 优化器, ...
 - 过拟合, 数值溢出, 存储不足, 分布式, ...
 - 如同炼丹
- 要是从零搞起, 我们要做些什么
 - 数据收集, 数据清理, 网络搭建, 模型微分
 - 从入门到放弃
- 好在我们不必从零做起
 - PyTorch: 神经网络搭建, 训练和预测
 - Huggingface transformers: 预训练模型



Transformers

推荐材料



- [Stanford CS 224N | Natural Language Processing with Deep Learning](#)
 - [CS230 Deep Learning \(stanford.edu\)](#)
 - [Deep Learning by deeplearning.ai | Coursera](#)
 - [ML 2021 Spring \(ntu.edu.tw\)](#)
 - [神经网络与深度学习 \(nndl.github.io\)](#)
 - [Welcome to PyTorch Tutorials](#)
 - [Introduction - Hugging Face NLP Course](#)
 - [新必应 - 了解详细信息 \(bing.com\)](#)
 - [Introducing ChatGPT \(openai.com\)](#)
 - [《动手学深度学习》](#)
-

Homework



- 两个 Python 小工程
 - 使用 PyTorch 搭建 MLP 实现 MNIST 手写数字识别
 - 借助 transformers 包载入 BERT 实现IMDB情感分类
- 着重练习
 - 使用深度学习框架
 - 调超参数 (炼丹)
 - 通过Google, New Bing, ChatGPT 以及各其他网络资源解决困难