



# T12

## ChatGPT原理使用及Prompt基础 ——使用Replit搭建你的AI助手



 **bilibili** @AFAN的费曼生活

 **知识星球** @AFAN的费曼生活

 **GitHub** @BILI-AFAN

 **gitee** @BILI-AFAN

# 本教程依赖项

## 依赖情况

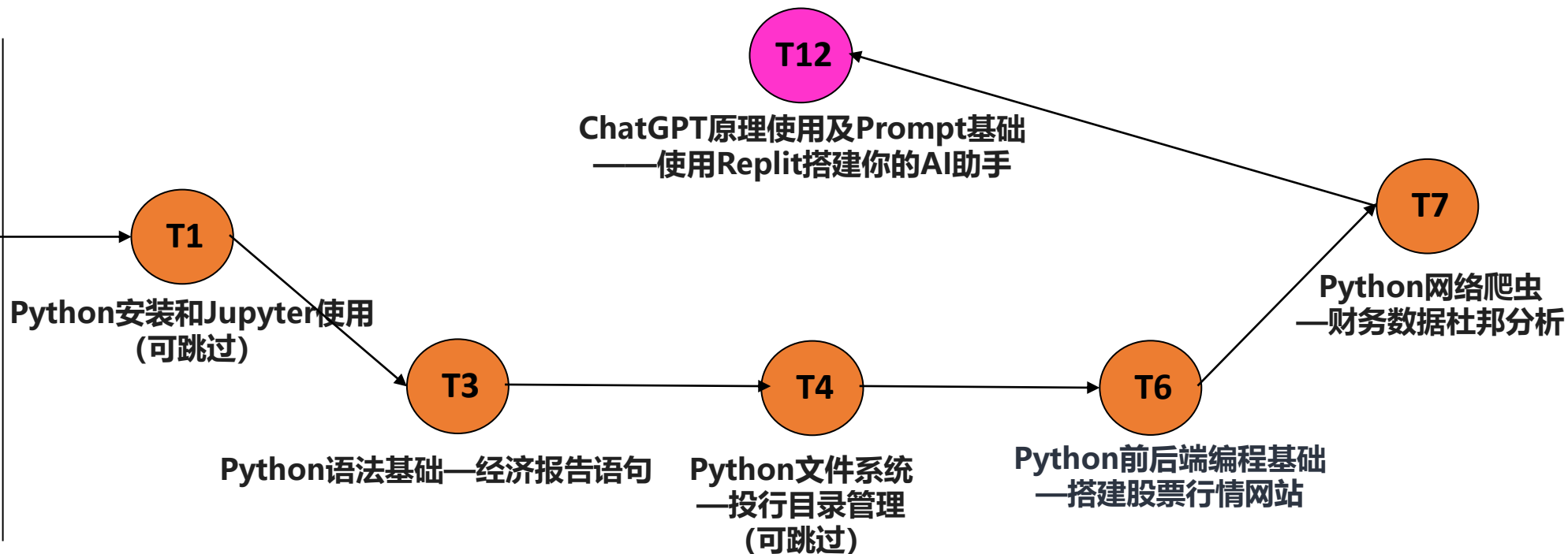
### 全局环境变量

会使用计算机操作

了解金融基本概念

具备基本逻辑思维

了解点金融的高中生



# 和AFAN交流互动



资料下载、答疑、深入探讨

免费下载



[github.com/bili-afan/bili-afan](https://github.com/bili-afan/bili-afan)



[gitee.com/bili-afan/bili-afan](https://gitee.com/bili-afan/bili-afan)

视频中的资料代码  
全部免费公开下载

免费答疑



知识星球 @AFAN的费曼生活

视频内容免费答疑  
个人研究不定期分享

付费干货



知识星球 @AFAN的金融科技

专业分享 3次/周  
主题干货直播 1次/月  
专业话题提问 不限次/1年  
付费用户微信群交流/永久



# YOU CAN GET



ChatGPT基础原理



+



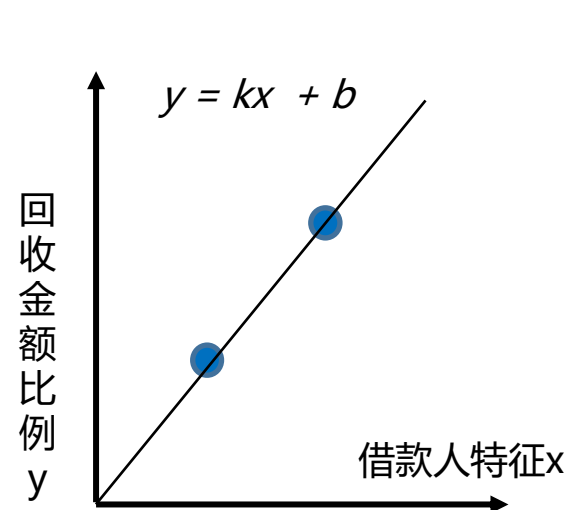
DIY免费应用



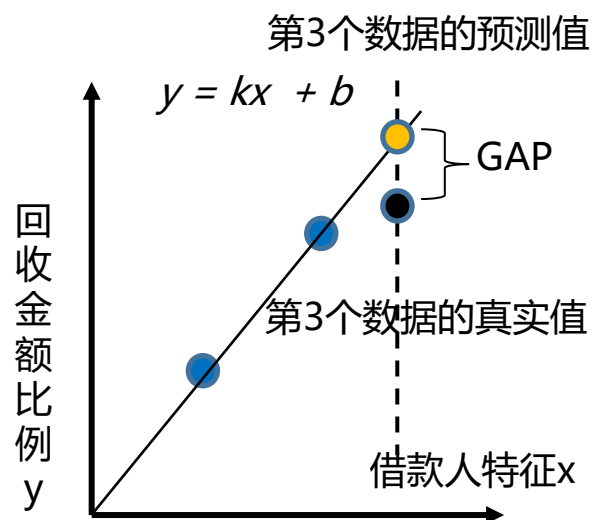
大模型Prompt原则

# 1 ChatGPT基本原理

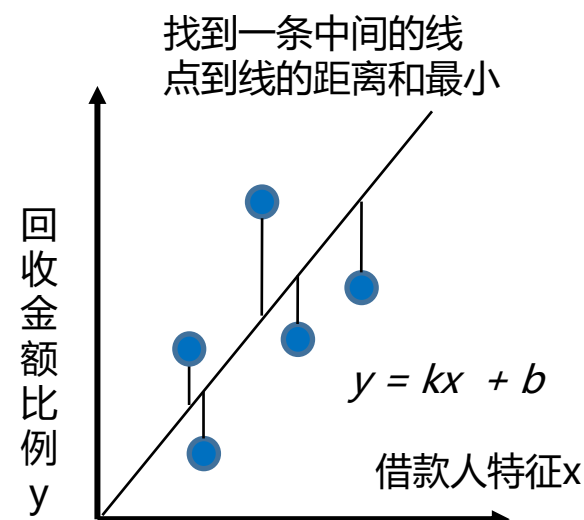
## 1.1 什么是机器学习——线性模型



构建线性模型  
基于2个数据点



评估模型的好坏  
预测和真实对比

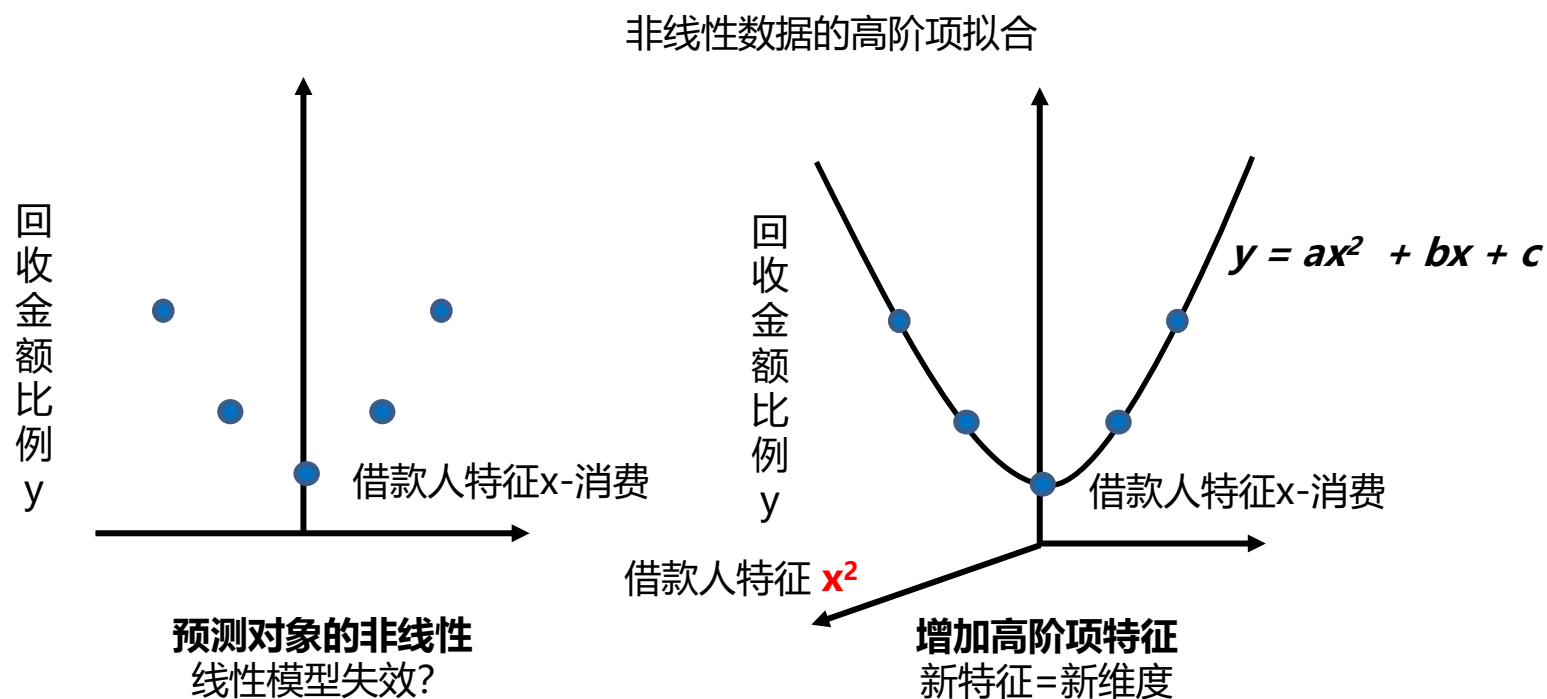


线性回归模型  
有较多数据点时

人像区域

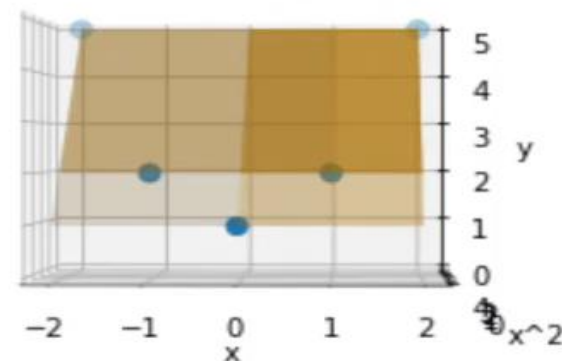
# 1 ChatGPT基本原理

## 1.1 什么是机器学习——非线性模型



人像区域

找到一个中间的平面  
点到平面的距离和最小



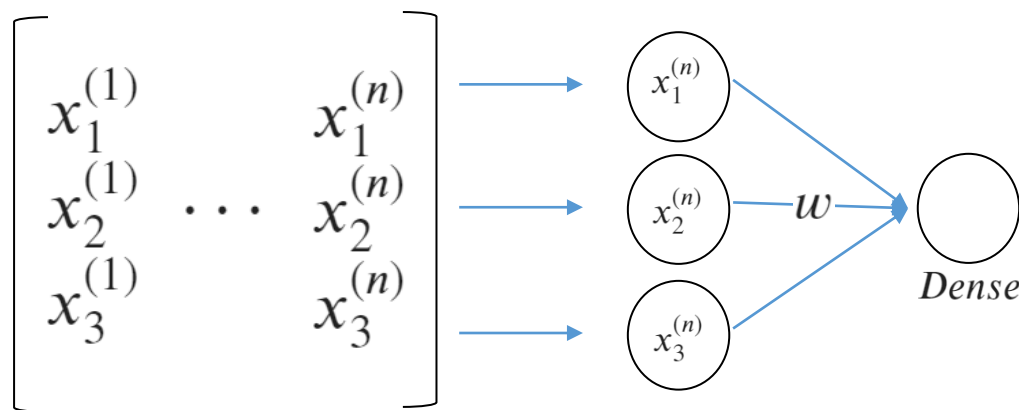
高维的最优化  
超平面的线性回归

# 1 ChatGPT基本原理

人像区域

## 1.2 什么是深度学习——单层神经网络

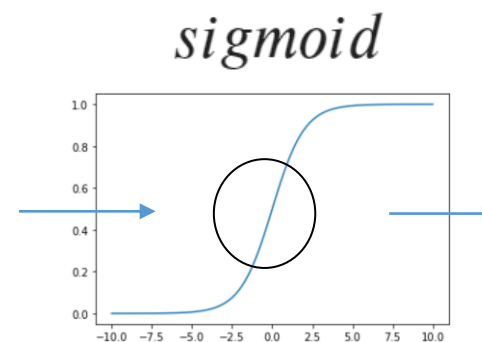
单层神经网络 = 线性回归 + 非线性变换



数据样本  
3个特征 \* n个数据

输入层  
等于特征个数

计算结果聚合  
到此=线性回归



非线性变换  
增加高维特征



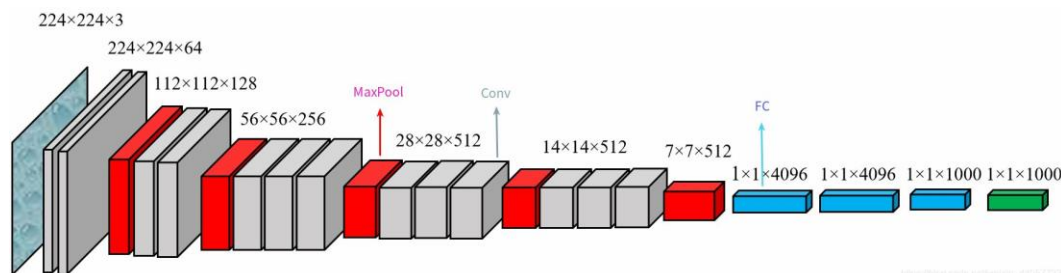
Hinton  
 $Y$

输出层  
预测结果输出

# 1 ChatGPT基本原理

## 1.2 什么是深度学习——深度神经网络

神经网络模型的优势在于其可以叠加计算复杂度  
中间层越多，网络越深，模型的 学习潜力 越强



VGG-16模型专门用于图像识别，其诞生于2014年  
其网络深度达到**16层**，模型参数量达到**1.39亿**个  
获得当年的分类竞赛的第2名，定位竞赛的第1名

## 人像区域

海量的数据是任何模型学习的基础材料。

“如果你把孩子的眼睛 都看作是生物照相机，那他们每200毫秒就拍一张照，这是眼球转动一次的平均时间。所以到3岁大的时候，一个孩子已经看过了 上亿张的真实世界照片。这种“训练照片”的数量是非常大的。”



李飞飞

计算机视觉领域的ImageNet项目诞生于2009年，该项目最终形成了**1500万张照片**的数据库，涵盖了**22000种物品**。使用者会通过各种方式**生成远大于其量级的数据集**，基于这些海量数据开展的竞赛推动了计算机视觉的技术和产业发展。

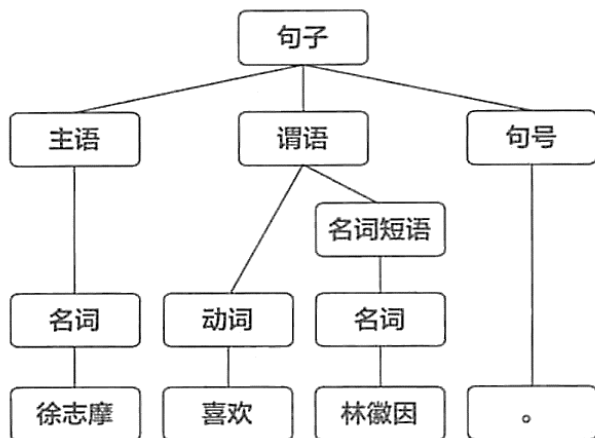
IM  GENET



# 1 ChatGPT基本原理

## 1.3 什么是自然语言处理——从规则到统计

### 基于规则的困境



### 无法穷举的句法规则

分析：“徐志摩喜欢林徽因。”

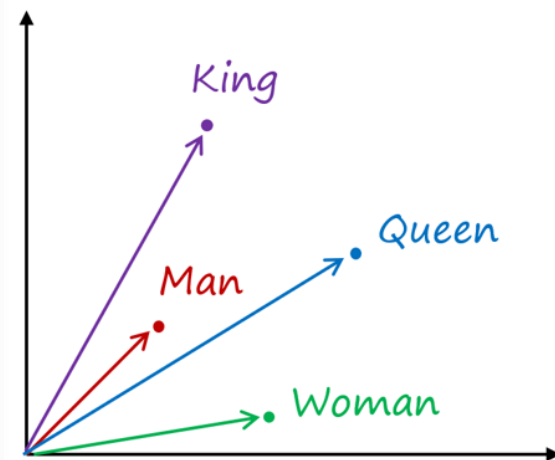
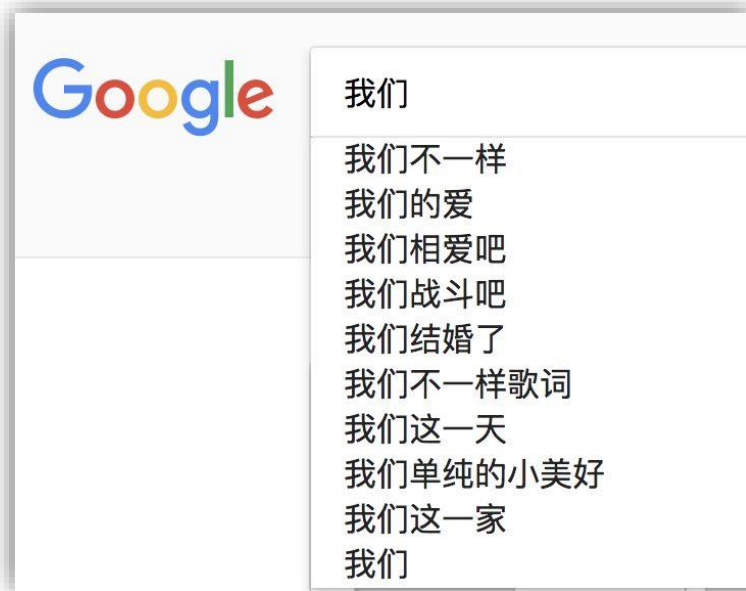
The **pen** is in the box. 钢笔  
The box is in the **pen**. 围栏

### 复杂多义的语义处理

词的多义性严重依赖于上下文，及外部世界的知识

## 人像区域

### 基于统计的优势

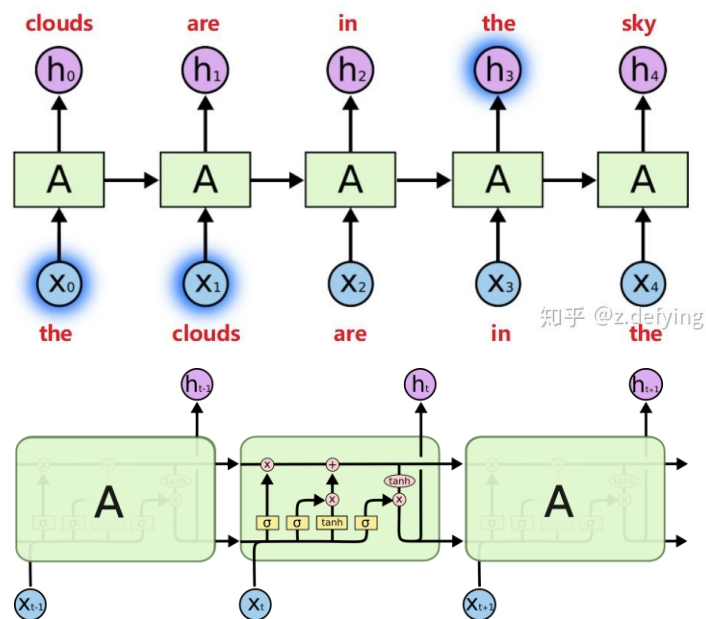


### 数据驱动的上下文概率生成和词义理解

# 1 ChatGPT基本原理

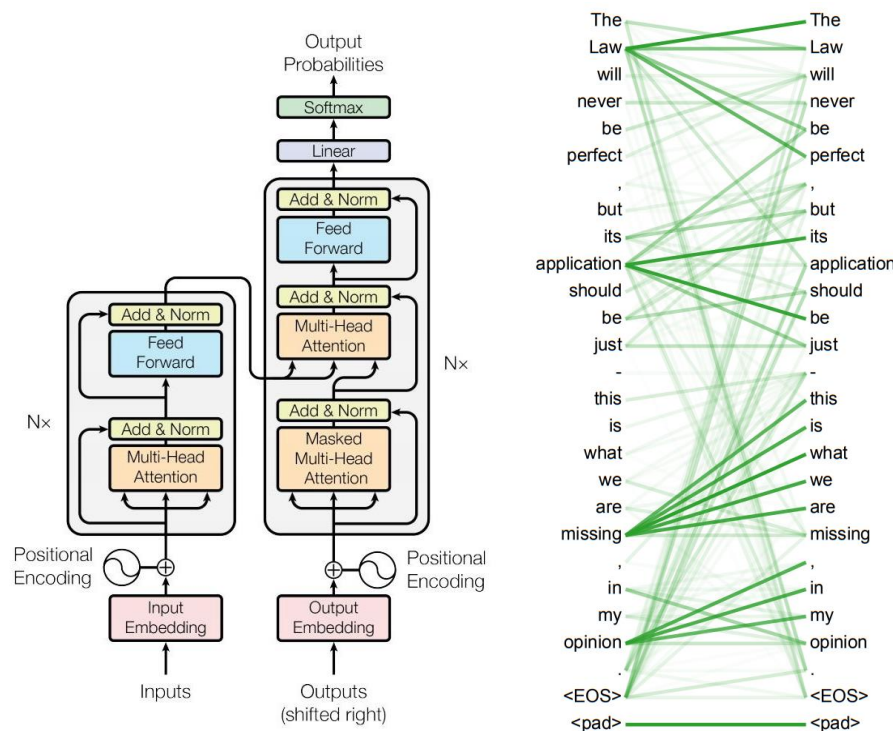
## 1.3 什么是自然语言处理——模型结构的演进

今天天气太美好了，尽管路上发生了一件不愉快的事，...，我马上调整好状态，开开心心地准备迎接美好的一天。



RNN: 循环神经网络

LSTM: 长短记忆网络



Transformer

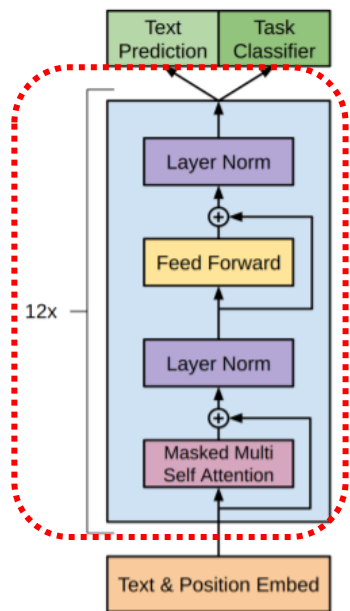
人像区域

# 1 ChatGPT基本原理

人像区域

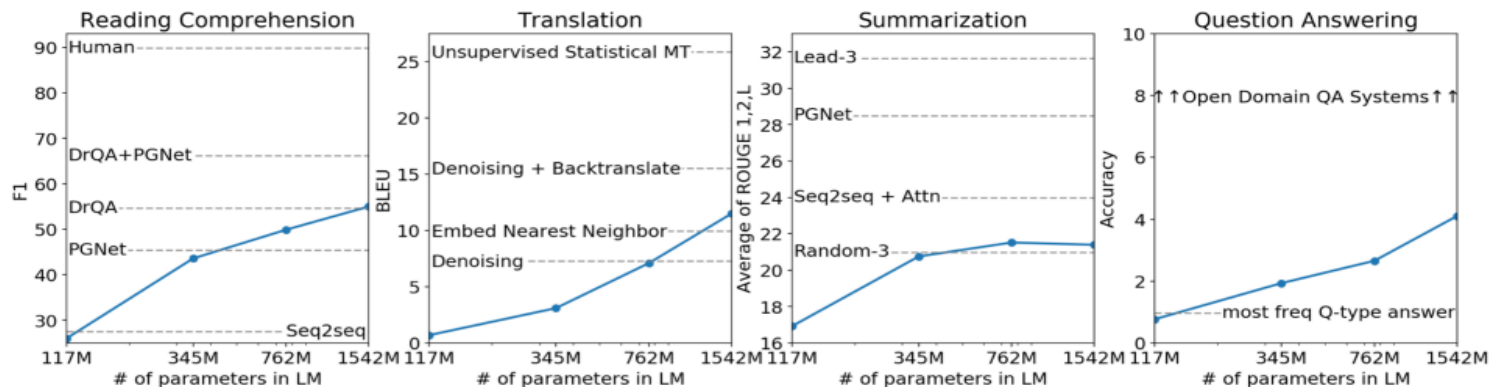
## 1.4 GPT和大模型——大模型参数的狂飙

### GPT (Generative Pre-training Transformer)



采用了12层的transformer结构  
拥有更大的参数数量和模型深度

模型	发布时间	参数量	预训练数据量
word2vec	2013年9月	约9千万	约4G
GPT	2018年6月	1.17亿	约5GB
GPT-2	2019年2月	15亿	40GB
GPT-3	2020年5月	1,750亿	45TB



随着参数量的上升,大部分任务的性能都会一直有显著提升

# 1 ChatGPT基本原理

## 1.4 GPT和大模型——GPT训练过程

Step 1

Collect demonstration data and train a supervised policy.

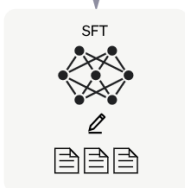
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



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



数据准备和训练

Step 2

Collect comparison data and train a reward model.

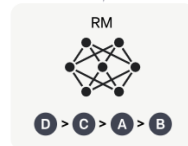
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



human feedback RL

Step 3

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

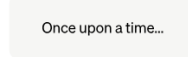
A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



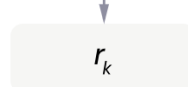
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



强化学习不断迭代

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



GPT = Generative Pre-trained Transformer



核心步骤：文字接龙+老师引导

# 1 ChatGPT基本原理

## 1.4 GPT和大模型——大模型发展的两个路线



VS



专才-Finetuning

VS

通才-Prompt



专才+适配器 => 不同场景

通才+专项训练 => 特定场景

对立和统一，互相促进发展  
但得承认：通才是之间的桥梁

人像区域

## 2 Replit平台及API

新手友好的编程平台

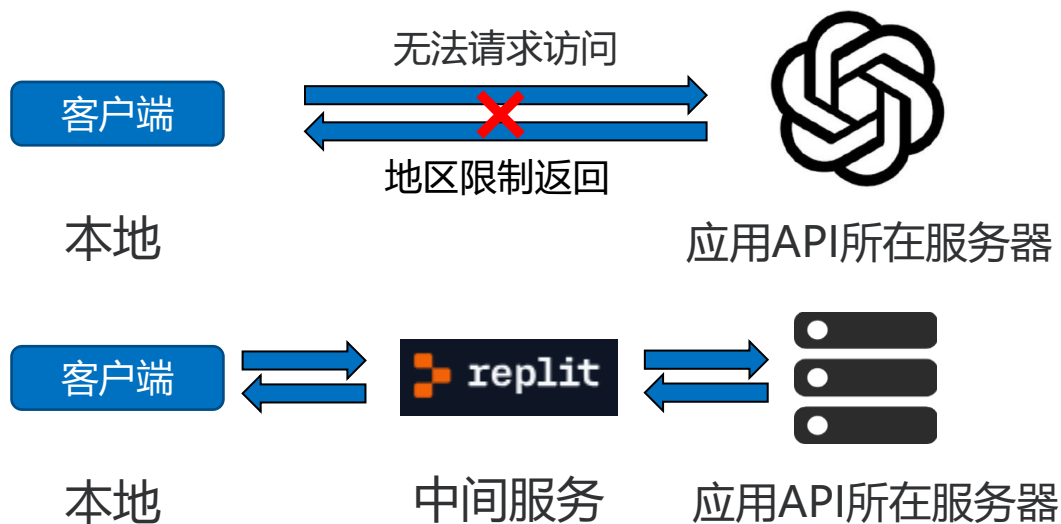


Replit是美国一家编码平台，提供了简单但功能强大的在线IDE、编辑器、编译器、解释器和REPL，支持PHP、Python等50多种编程语言进行编码、编译、运行和托管。使用Replit免费的协作式浏览器，直接在浏览器中编写代码，从浏览器快速启动，共享和开发任何编程语言项目的理想工具。

Replit前身为Repl.it，名字来自首字母缩写REPL，代表“阅读-评估-打印循环”，是由Amjad Masad、Faris Masad和Haya Odeh于2016年共同创建的初创公司和在线IDE，总部位于美国旧金山。

使用编程平台的目的是

人像区域





## 2 Replit平台及API

## 人像区域

### ChatGPT的API调用

知识星球: <https://t.zsxq.com/11INkb6Ld>

#### 默认对话模式

```
def get_completion(prompt, model="gpt-3.5-turbo"):  
    messages = [{"role": "user",  
                  "content": prompt}]  
    response = openai.ChatCompletion.create(  
        model=model,  
        messages=messages,  
        temperature=0)
```

Chat model

assistant



user

因此，用户消息是输入，而助手消息是输出

#### 增加角色概念

```
messages = [  
    {"role": "system",  
      "content": "You are an assistant... "},  
    {"role": "user",  
      "content": "tell me a joke "},  
    {"role": "assistant",  
      "content": "Why did the chicken... "},  
    ...  
]
```

system

Sets behavior of assistant



assistant

Chat model



user

而用户不会意识到系统消息

#### 输出自由度

Temperature

my favorite food is



Temperature = 0

my favorite food is pizza  
my favorite food is pizza  
my favorite food is pizza

for tasks that require reliability, predictability

Temperature = 0.3

my favorite food is pizza  
my favorite food is sushi  
my favorite food is pizza

For tasks that require variety

Temperature = 0.7

my favorite food is tacos  
my favorite food is sushi  
my favorite food is pizza

可能需要使用更高的温度

## 3 Prompt基础

### 原则1：指定分隔符

为了让LLM更清晰的知道哪些是**我们指定的内容**可以使用对应的分隔符区分，如：  
`` ` 或 --- --- 或 < > 等等

### 原则2：格式化输出

如要求json和html结构输出，方便**标准化数据传输**

### 原则3：大问题指定步骤

可以要求其分步骤输出，**给模型思考的时间**

## 人像区域

Coursera: <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>

### 原则4：输出分析过程的引用素材

大模型不知道自己的知识边界，让其输出它分析参考的素材资料，防止出现**模型幻觉**



林黛玉倒拔垂杨柳是《红楼梦》中的一个典故。在小说中，林黛玉因为受到情伤和家庭压力而生病，一度情绪低落。后来，她到了荣府的花园里，看到一棵垂杨柳，感慨万千，用力摇晃树枝，并表示自己希望能够像垂杨柳一样，在悲伤中保持坚韧和生命力。

"林黛玉倒拔垂杨柳"这个典故成为了形容女子坚强、不屈不挠的代名词，也成为了传统文化中的一个重要典故。





 Bilibili @ AFAN的费曼生活



**THANKS FOR WATCHING !**