

# 人工智能之深度学习

2023-10

# 内容

- 人工智能简史
- 深度学习简介
- 深度学习的应用
- 深度学习的优缺点
- DNN深度神经网络
- CNN卷积神经网络
- RNN循环神经网络
- GAN生成式对抗网络
- PLM预训练语言模型
- LLM大语言模型
- 学习资源

# 人工智能简史

# 1949 EDVAC

世界上第一台可存储程序并运行的计算机。

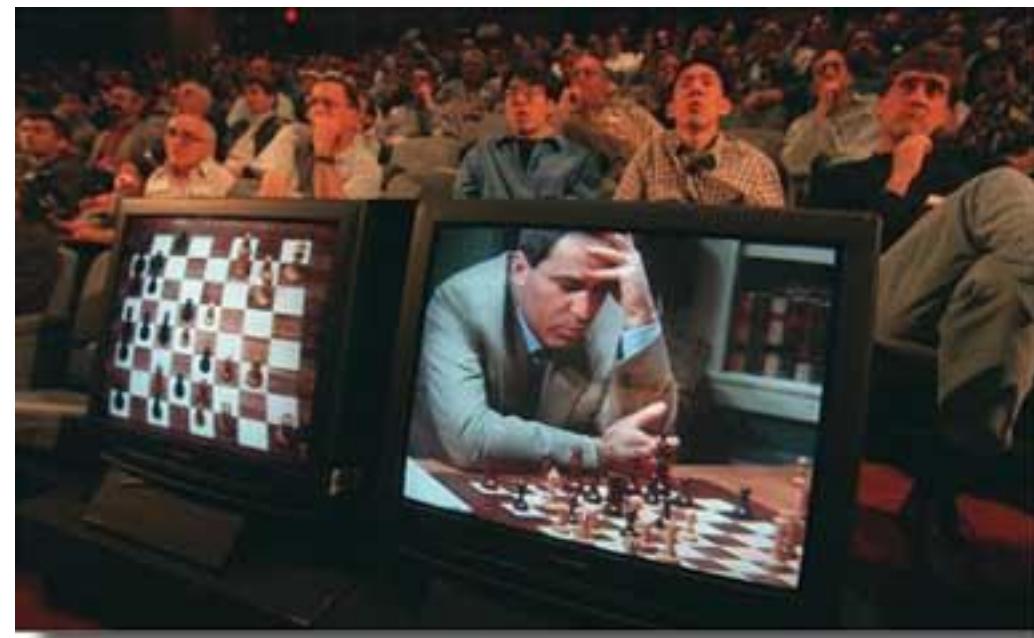
与它的前任ENIAC不同EDVAC采用二进制，是一台冯·诺伊曼结构的计算机。



# 20年前。。。DEEP BLUE



1997年， IBM深蓝击败国际象棋冠军卡斯帕罗夫。



# 2016年 3月 AlphaGo



AlphaGo 4:1 获胜

AlphaGo 综合了神经网络与  
博弈算法



AlphaGo  
Vs  
Lee Sedol  
李世石

2017年5月23–27日 AlphaGo Master



AlphaGo 3–0 获胜

Google在中国乌镇  
组织的对弈

 AlphaGo  
Vs  
Ke Jie  
中國棋王柯潔

# 人工智能简史

## 1943–1956 人工智能诞生

- 图灵测试 (1950)
- 游戏AI
- 约翰·麦卡锡提出“人工智能”一词，在1956年达特茅斯会议上被本领域接受

## 1956–1974 黄金年代

- 搜索式推理 (启发式搜索、棋路树、A\*算法)
- 基于符号学的自然语言处理 (词法分析、句法分析、语义网、聊天机器人ELIZA)

## 1974–1980 第一次低谷

- AI被认为只是“玩具”
- **感知器** (神经网络的一种形式) 的研究遇到瓶颈
- 基于规则和推理的专家系统，被批评“人类在解决问题时并没有使用逻辑运算”

## 1980–1987 繁荣

- 专家系统得到赏识 (CMU – XCON)
- 知识革命 (使用专家系统存储专业知识，试图建立巨型知识库)
- **链接主义重生** (Hopfield网络、**反向传播算法**)
- 第五代计算机工程

## 1987–1993 第二次低谷

- 第五代工程并未实现
- XCON维护费用居高不下

## 1993–2011 再次起飞

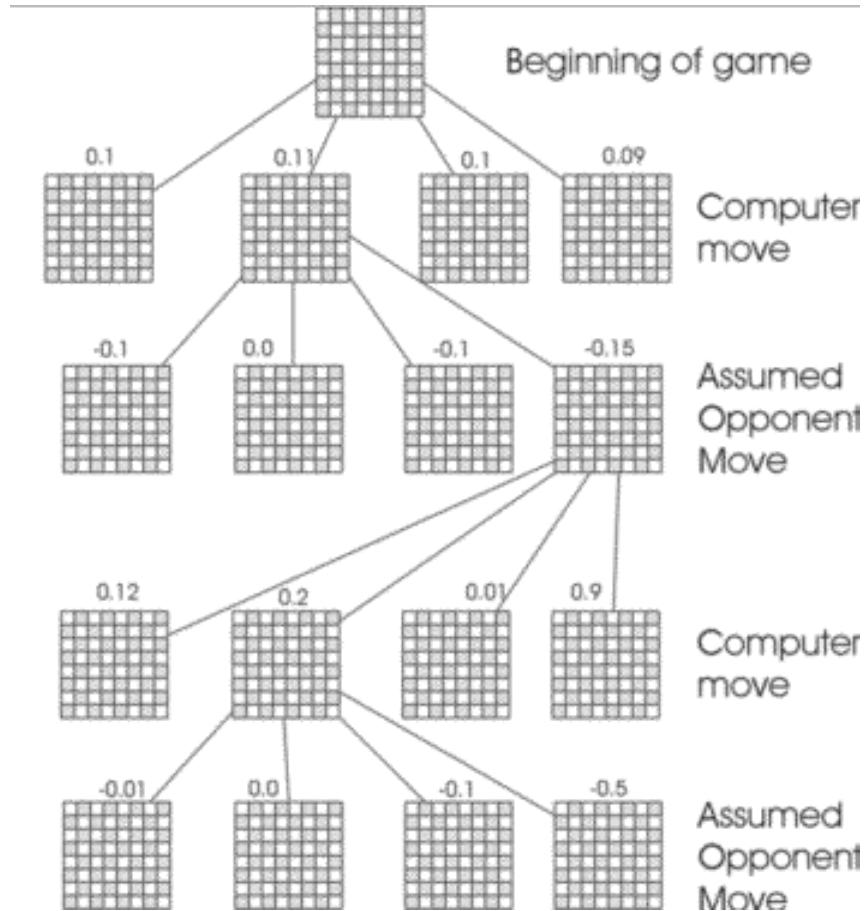
- IBM深蓝战胜国际象棋冠军卡斯帕罗夫
- AI研究开始借助复杂的数学工具，贝叶斯网络、隐马尔可夫模型、信息论、随机模型、优化理论、**神经网络的数学描述**...
- 在幕后使用AI的领域：数据挖掘、工业机器人、物流、语音识别、银行业软件、辅助医疗诊断、搜索引擎...

2011至今：深度学习、大数据

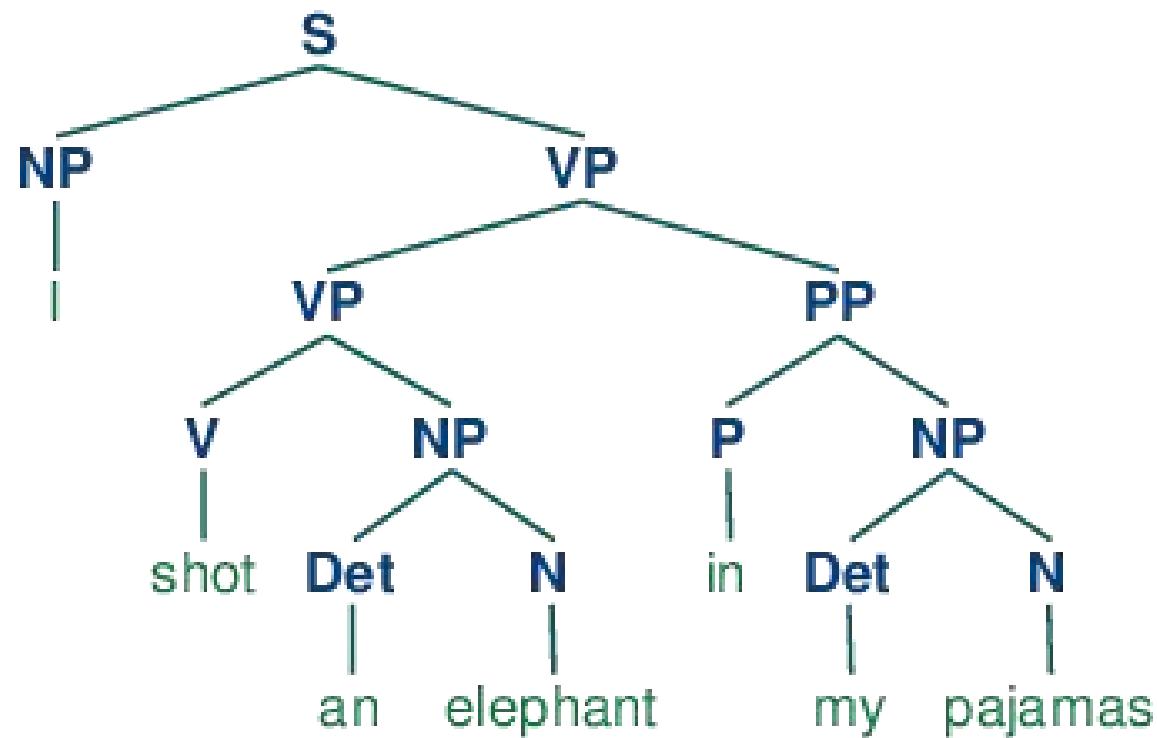
# 棋路树

每下一步，需要在很多的棋局空间中进行搜索。

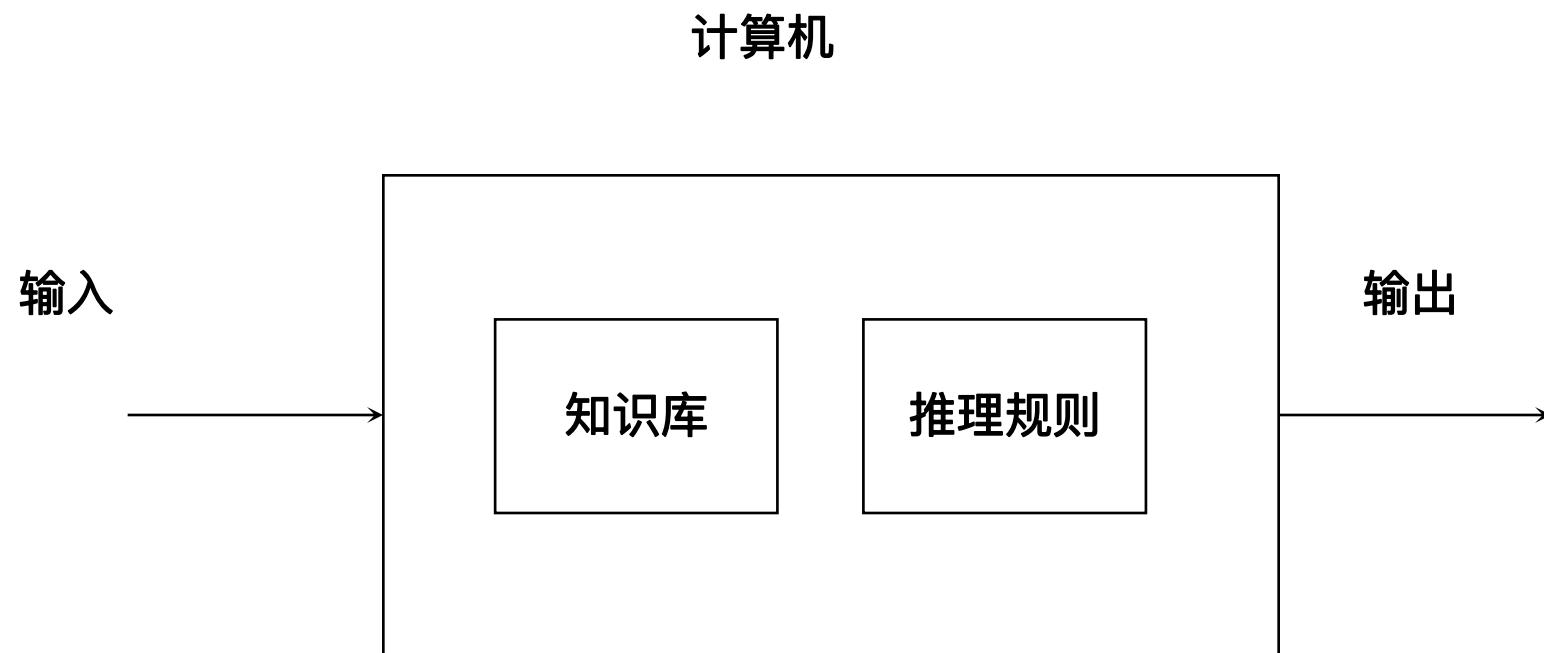
想要具备大师级国际象棋水平，  
至少需要向前探索8步，大约要  
搜索 $35^8$  或  $2 \times 10^{12}$ 个棋局。



## 自然语言处理：语法分析



# 基于知识库的专家系统



## 机器人 – Aibo

1999年 Sony发布的宠物狗

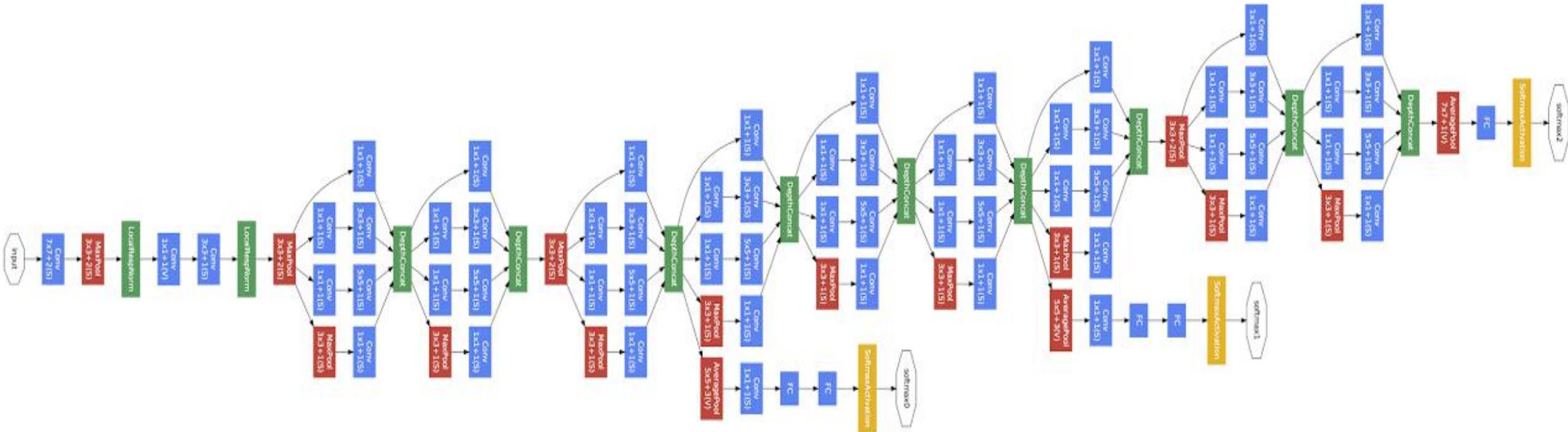


# 你家也许就有机器人

2001开始出现的扫地机器人

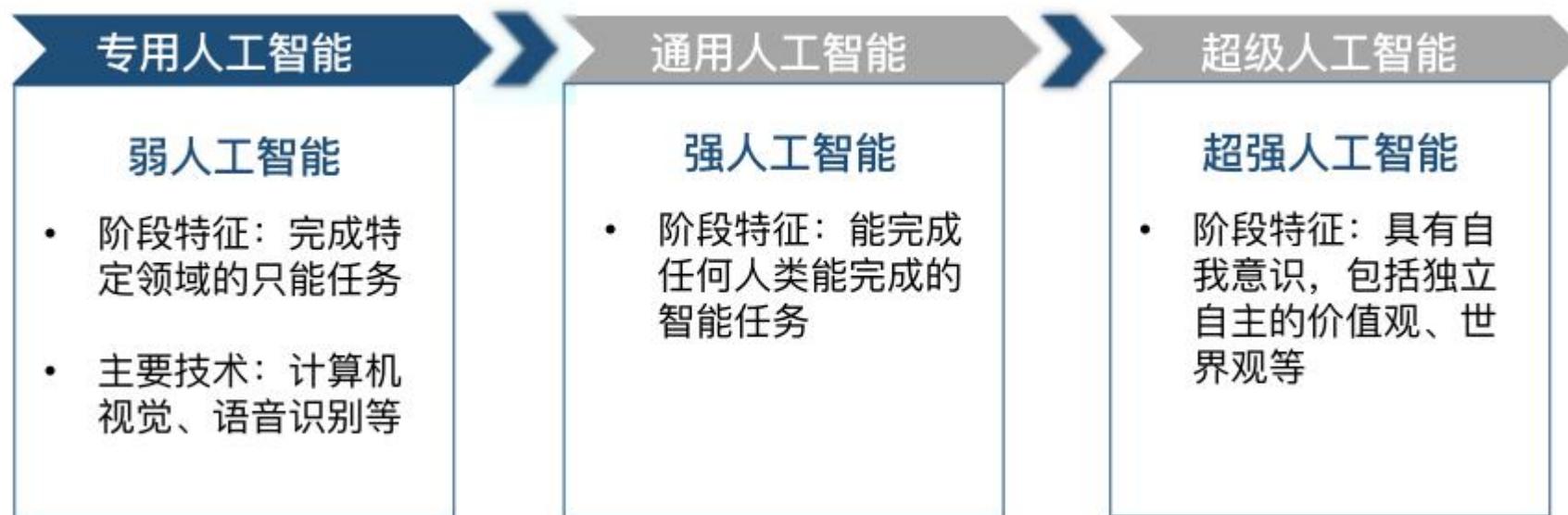


# GoogLeNet



卷积层  
池化层  
**Softmax**  
其他

# 目前还是初级阶段



# 深度学习简介

# 人工智能、机器学习、深度学习之间的关系

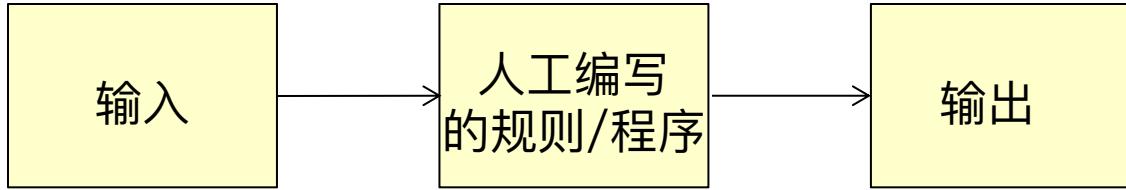
人工智能

机器学习

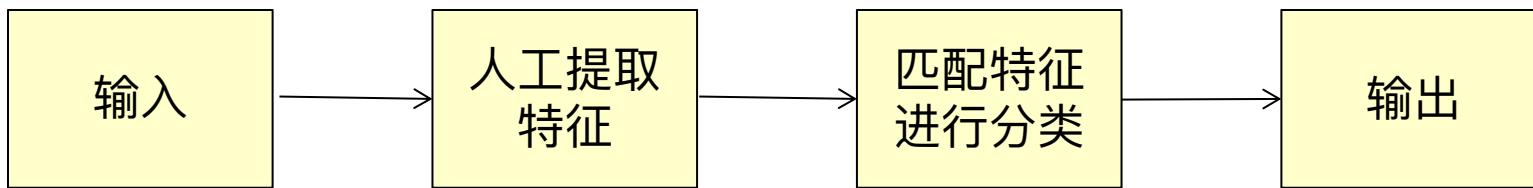
深度学习

# 不同人工智能方法的问题处理过程

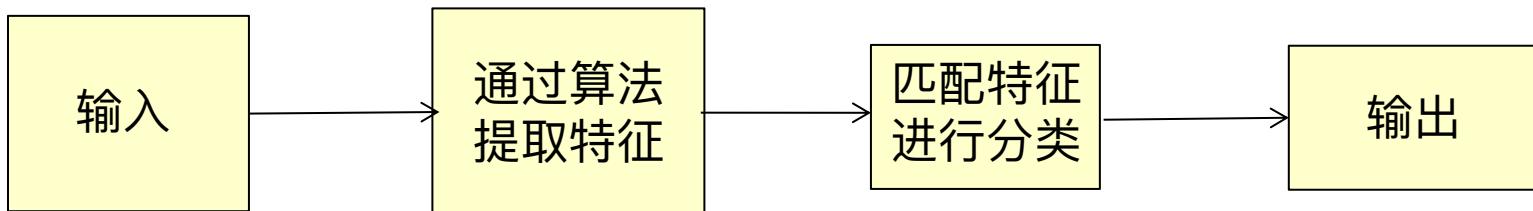
基于规则的系统



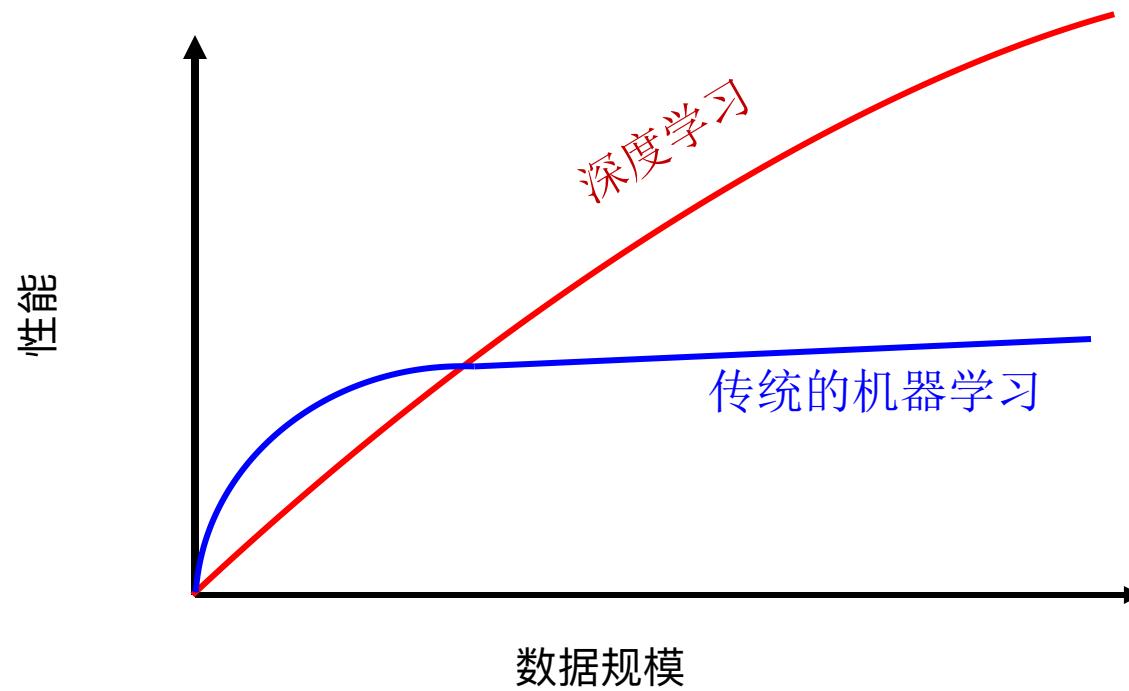
传统的机器学习



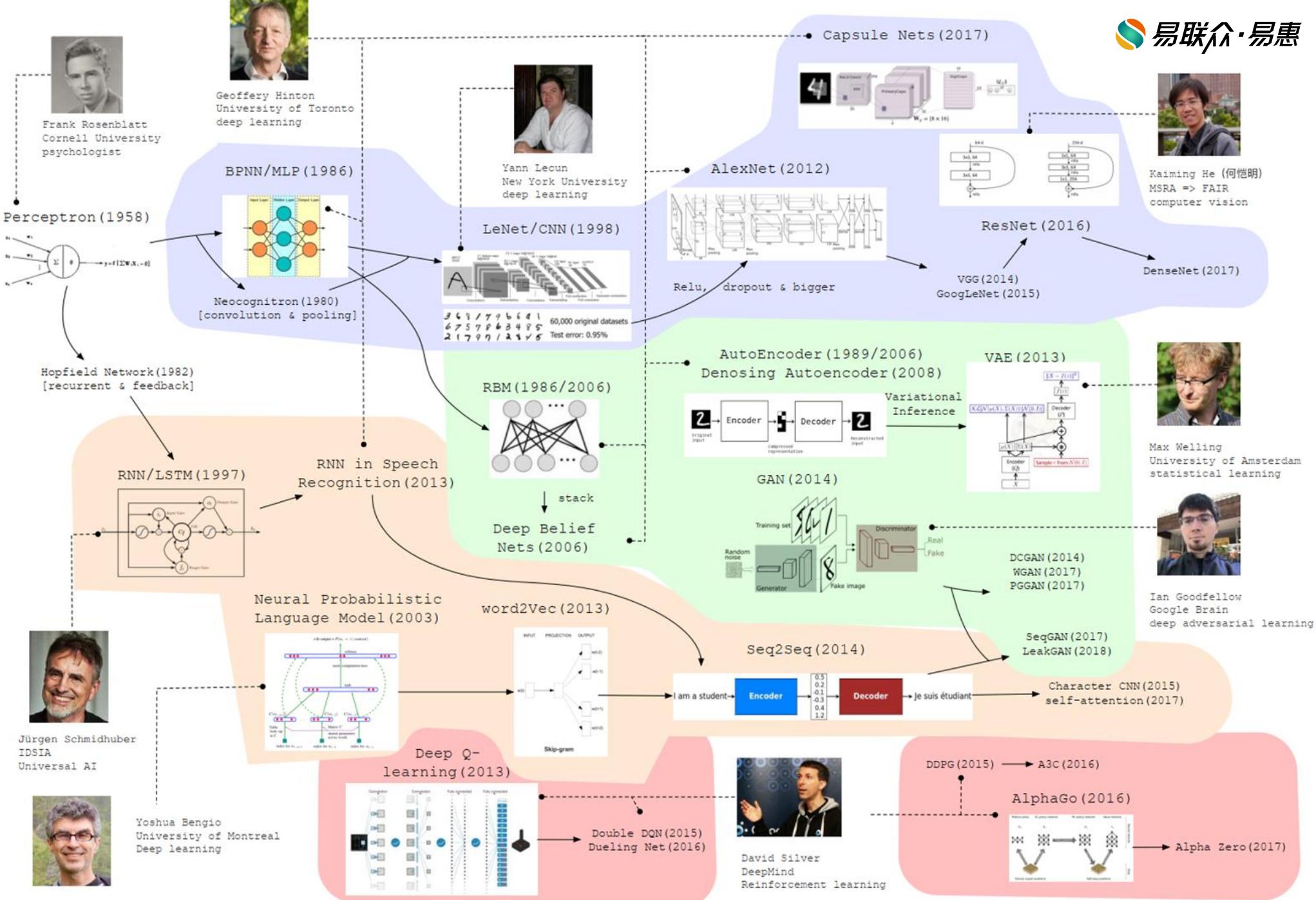
深度学习



# 样本数量 和 性能的关系

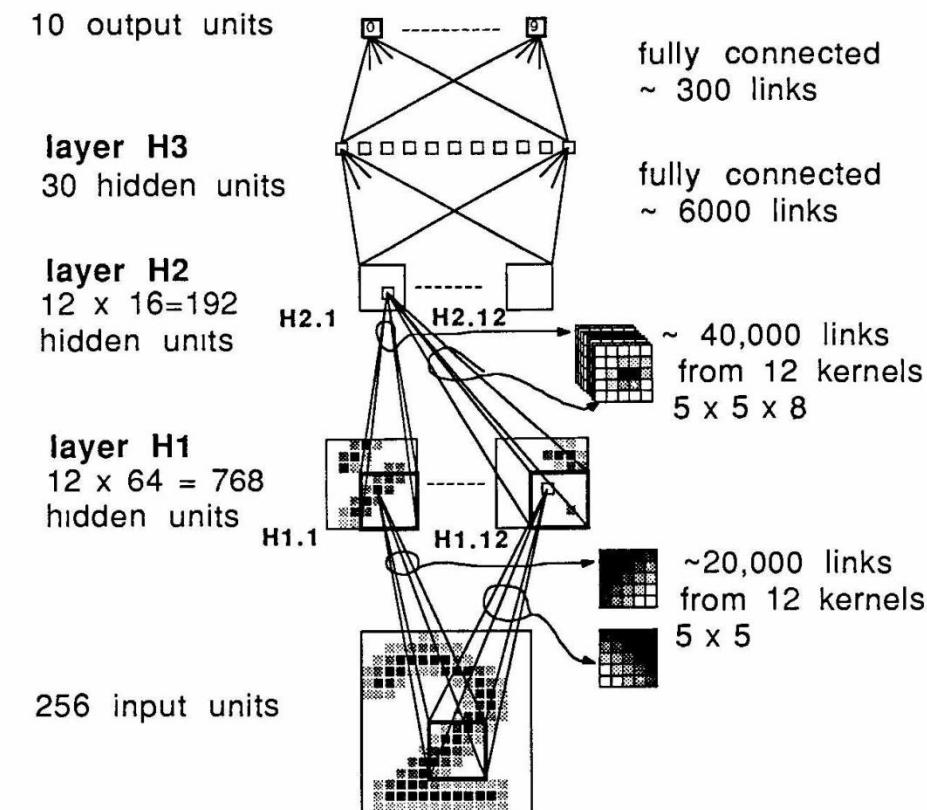


# 深度学习发展简史

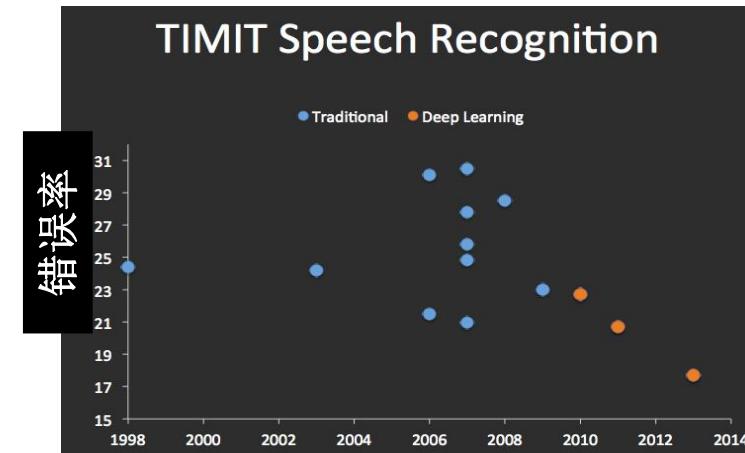
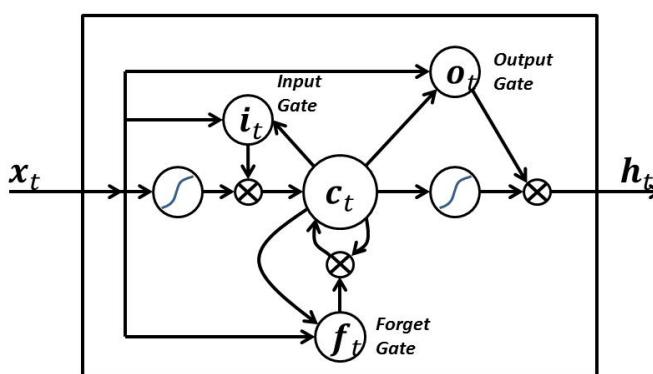


## LeNet 1989: 在美国邮政系统实现自动识别手写体邮政编码

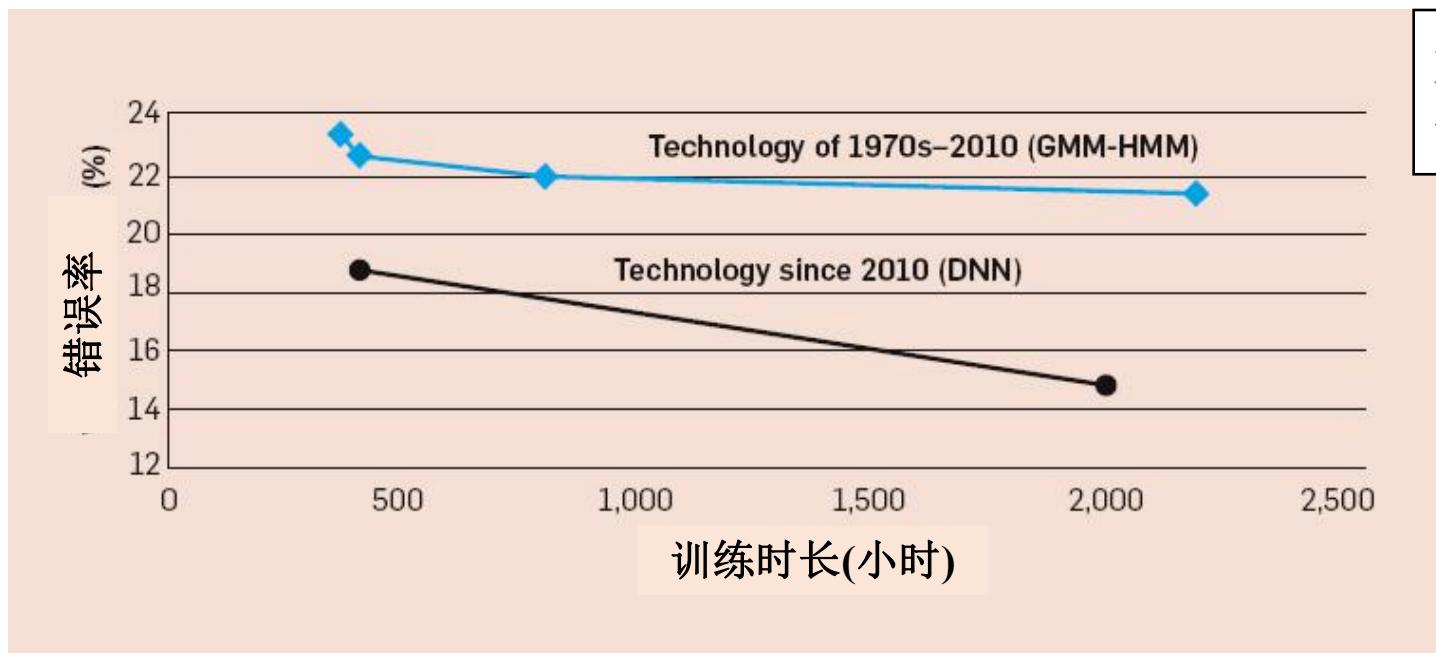
80322-4129 80306  
40004 14310  
37878 05153  
35502 75216  
35460 44209



## RNN 循环神经网络: LSTMs (1997):

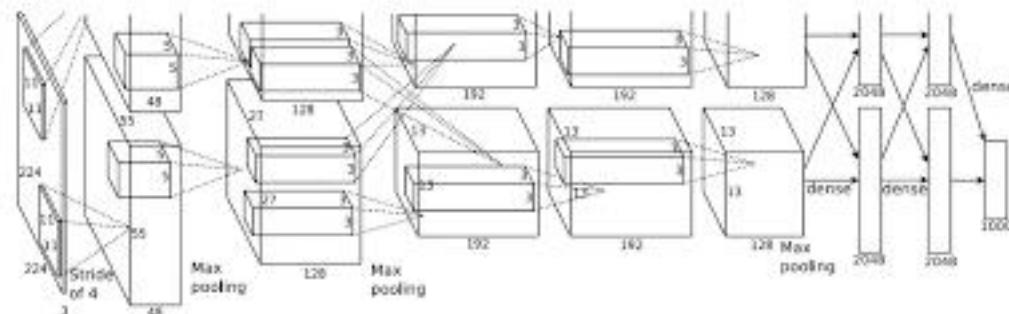
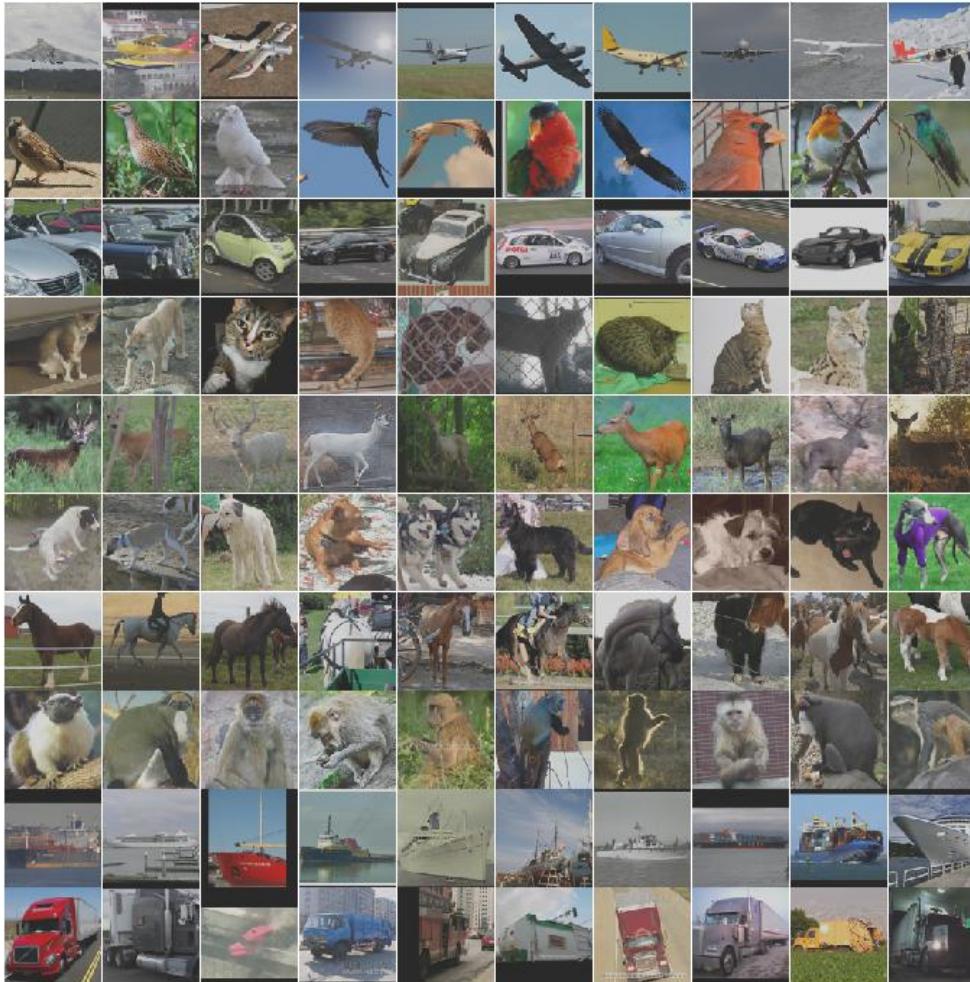


蓝色: 传统方法  
红色: 深度学习



蓝色: 混合高斯+隐马尔可夫模型  
黑色: 深度学习网络

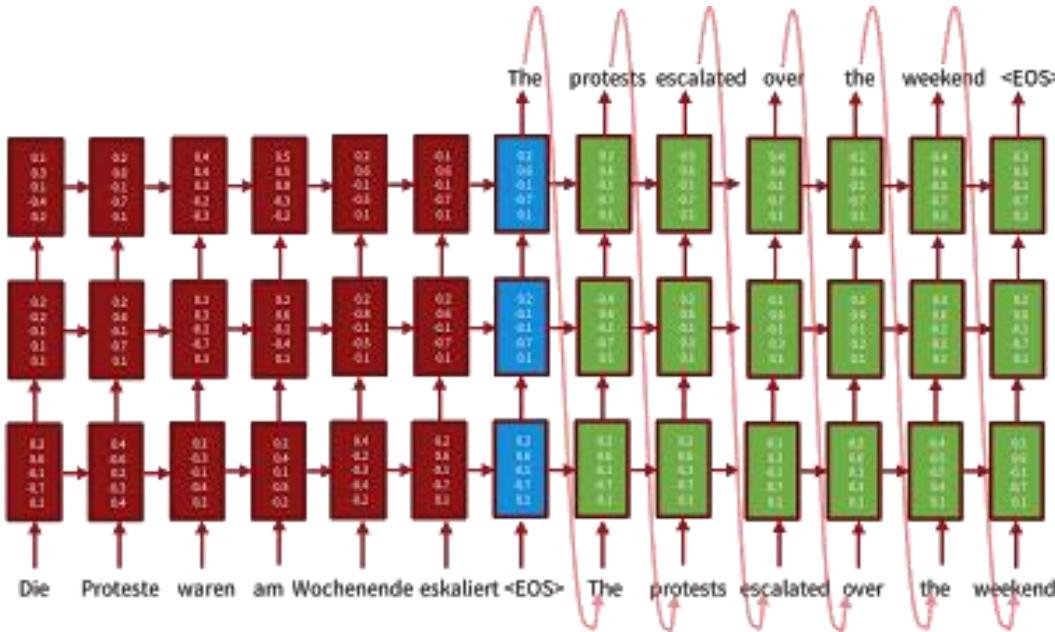
CNN 卷积网络: AlexNet (2012): 使用ImageNet的200GB图片进行训练, 获得当年图片分类比赛冠军



ImageNet目标分类竞赛的历年趋势

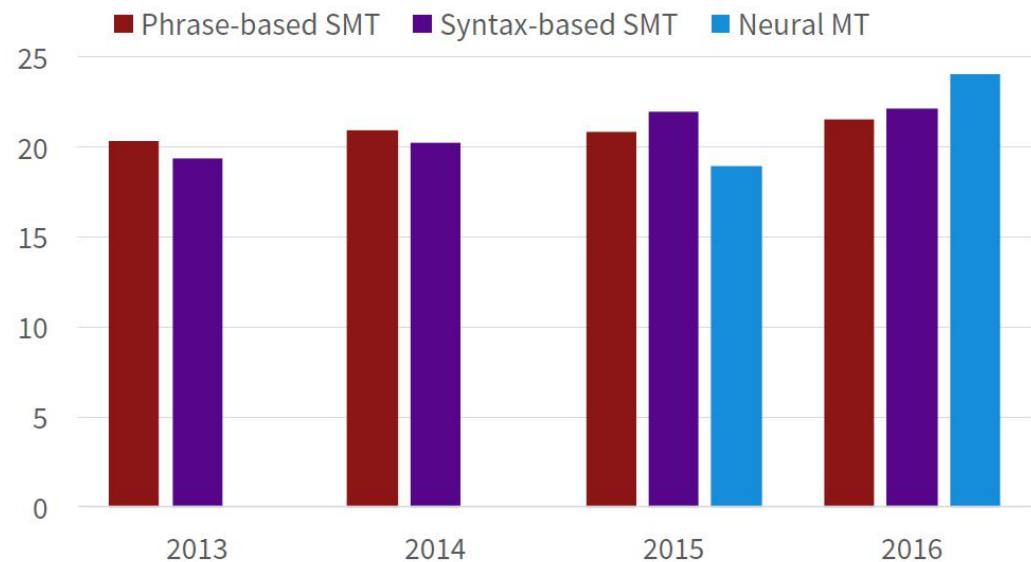


## 带attention机制的Seq2seq模型 (Source Luong, Cho, Manning ACL Tutorial 2016)



### 机器翻译的进展

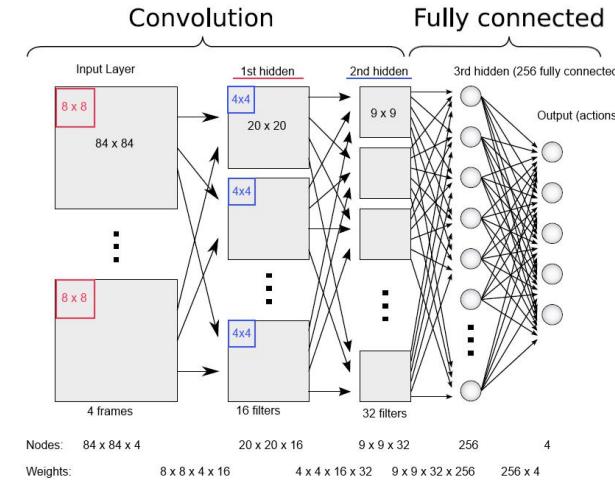
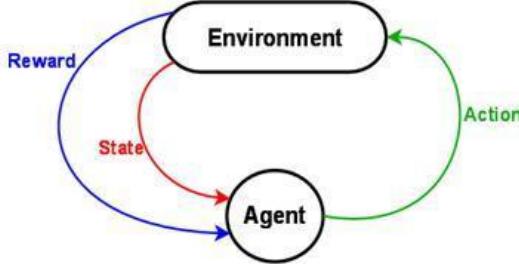
[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]



From [Sennrich 2016, [http://www.meta-net.eu/events/meta-forum-2016/slides/09\\_sennrich.pdf](http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf)]

红色：基于词组的统计机器翻译  
紫色：基于句法的统计机器翻译  
蓝色：基于神经网络的机器翻译

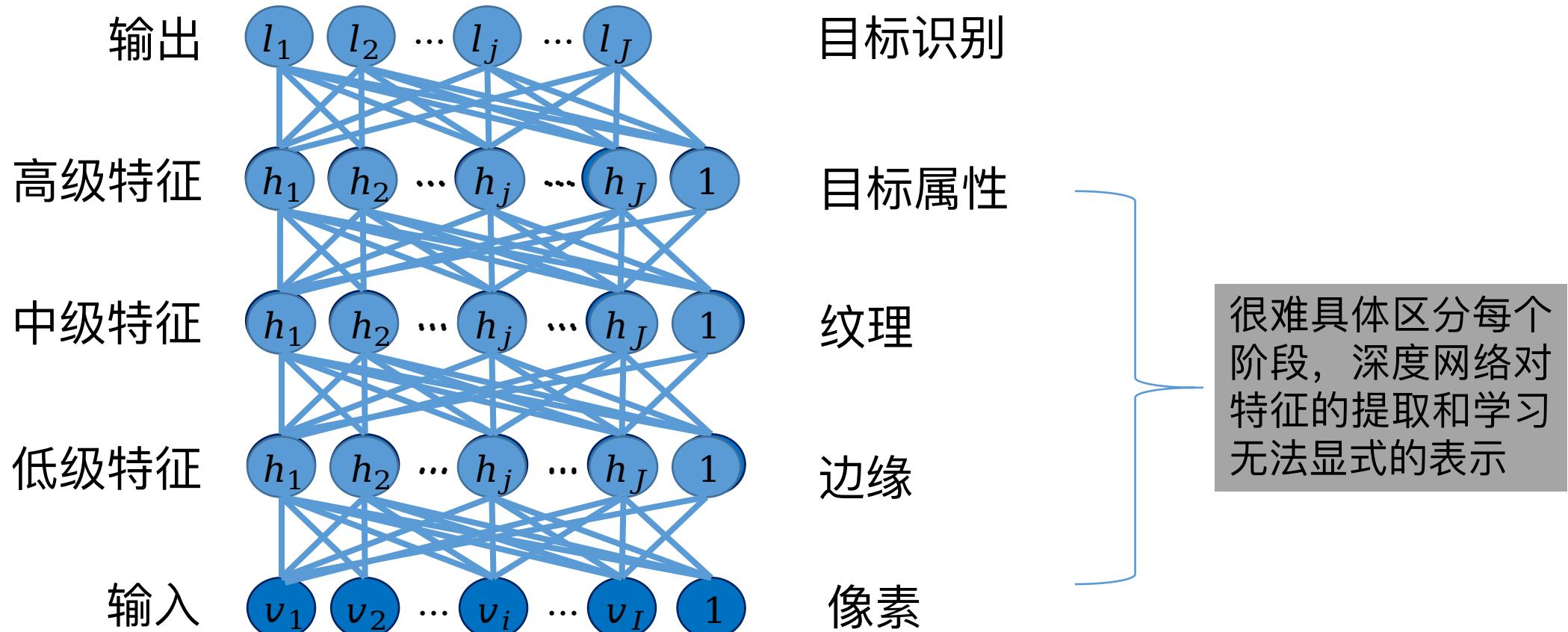
2013年, Deep Mind的街机游戏程序在6项游戏比赛中打败人类, 2014年被Google收购。



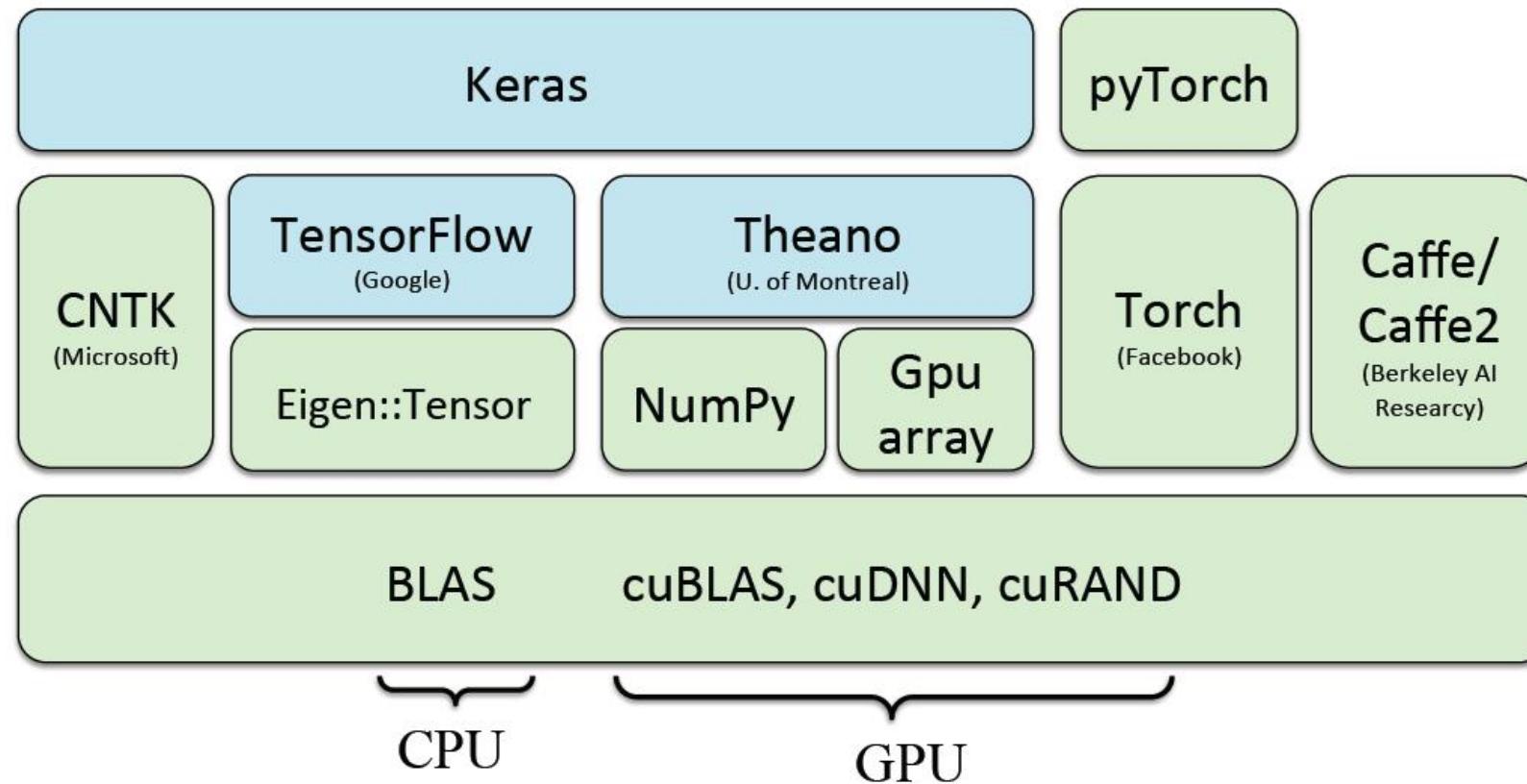
2016年, Deep Mind的阿尔法狗战胜围棋前世界冠军 Lee Sedol 李世石



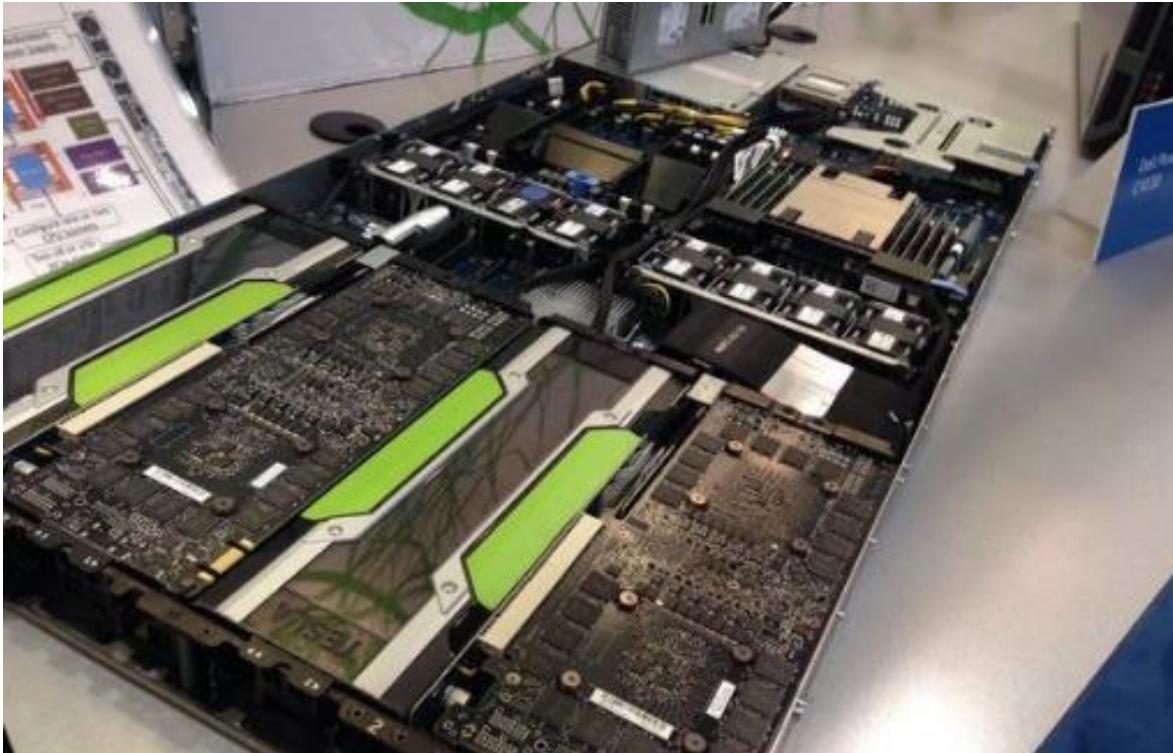
# DNN深度网络学习复杂特征的示例



# 深度学习的软件工具



## GPU - 俗称“显卡”



GPU显卡非常适合神经网络的数学运算，比基于CPU的训练快几个数量级

## TPU - 张量 (tensor) 处理器

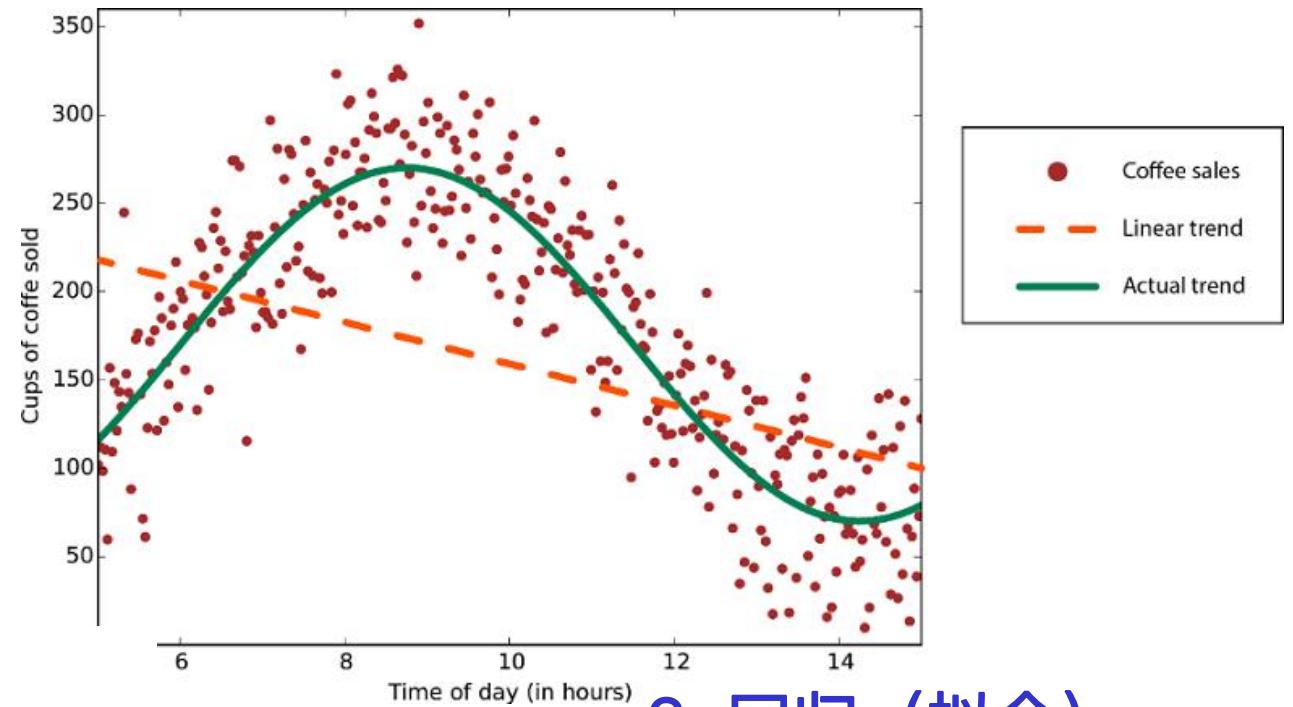
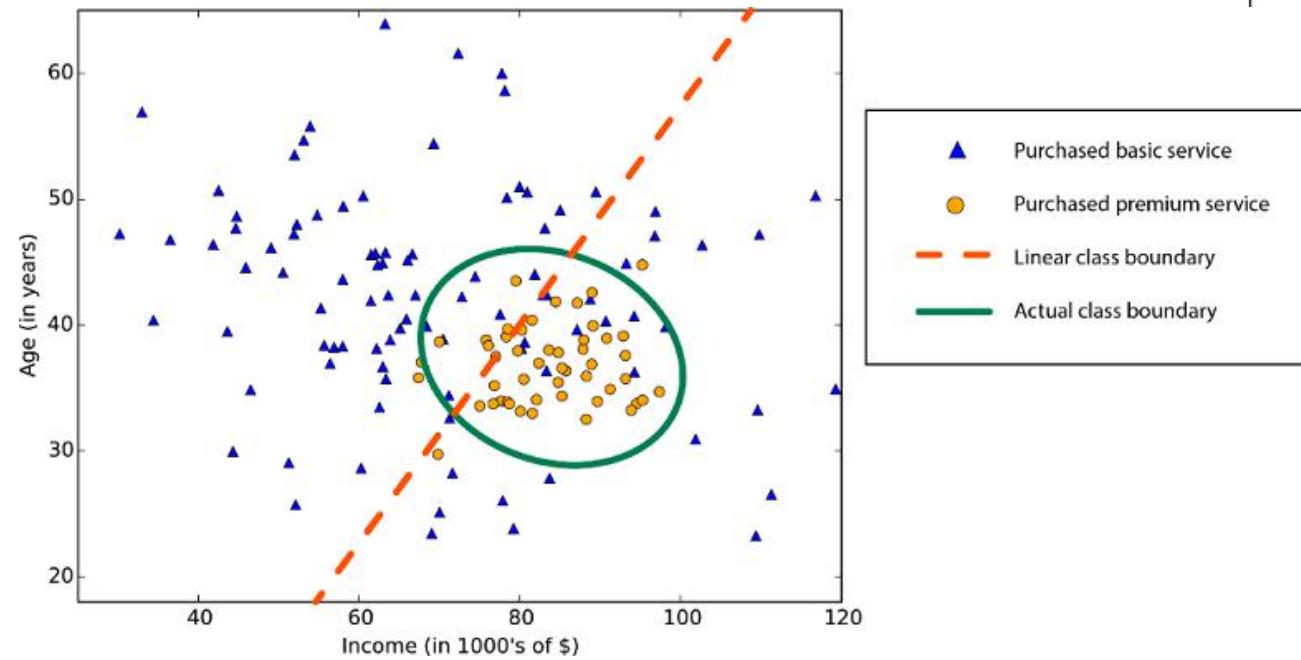
- 第三代TPU于2018年5月发布。第三代TPU的性能是第二代TPU的两倍，可实现420TFLOPs浮点运算



# 深度学习的应用

# 深度学习的基本功能

## 1. 分类



## 2. 回归 (拟合)

# 深度学习的应用方向

- Trend Prediction 趋势预测
- Recognition 识别
- New Knowledge 知识获取
- Making Sense 情感分析

# 深度学习的具体应用

- 信息获取（搜索引擎）
- 模式识别
- 受众定位
- 情绪分析（基于文本）
- 个性化
- 自动化
- 自然语言处理
- 公共媒体信息挖掘
- 产品差异化
- 语言翻译
- 语音识别
- 产生手写笔迹
- 人脸识别
- 自动驾驶
- 艺术创作
- 模仿著名画家
- 音乐创作
- 图片生成

# 深度学习的优缺点

## 深度学习的优点

- **学习能力强**

从结果来看，深度学习的表现非常好，他的学习能力非常强。



- **覆盖范围广，适应性好**

深度学习的神经网络层数很多，宽度很广，理论上可以映射到任意函数，所以能解决很复杂的问题。

- **数据驱动，上限高**

深度学习高度依赖数据，数据量越大，他的表现就越好。在图像识别、面部识别、NLP 等部分任务甚至已经超过了人类的表现。同时还可以通过调参进一步提高他的上限。

- **可移植性好**

由于深度学习的优异表现，有很多框架可以使用，例如 TensorFlow、Pytorch。这些框架可以兼容很多平台。

# 深度学习的缺点



- “黑箱”问题

你不知道神经网络如何以及为何会得出一定的输出（可解释性不足）

- 开发时间长

虽然像Keras这样的库让神经网络的开发变得简单，但有时你需要更多地控制算法的细节。特别是当你试图解决之前没人做过的难题时。大部分人只能使用现成的模型。

- 数据量需求大

神经网络通常需要更多的数据，至少需要数千甚至数百万个标记样本。在某些领域，标记数据的获取是个难题。

- 计算成本高

普通的CPU已经无法满足，主流的算力都是使用GPU和TPU，成本很高。

- 没有“人性”，容易存在偏见

由于深度学习依赖数据，并且可解释性不高。在训练数据不平衡的情况下会出现性别歧视、种族歧视等问题。

# DNN 深度神经网络

深度学习类似一个水流系统



## 深度学习 – 识别“田”字

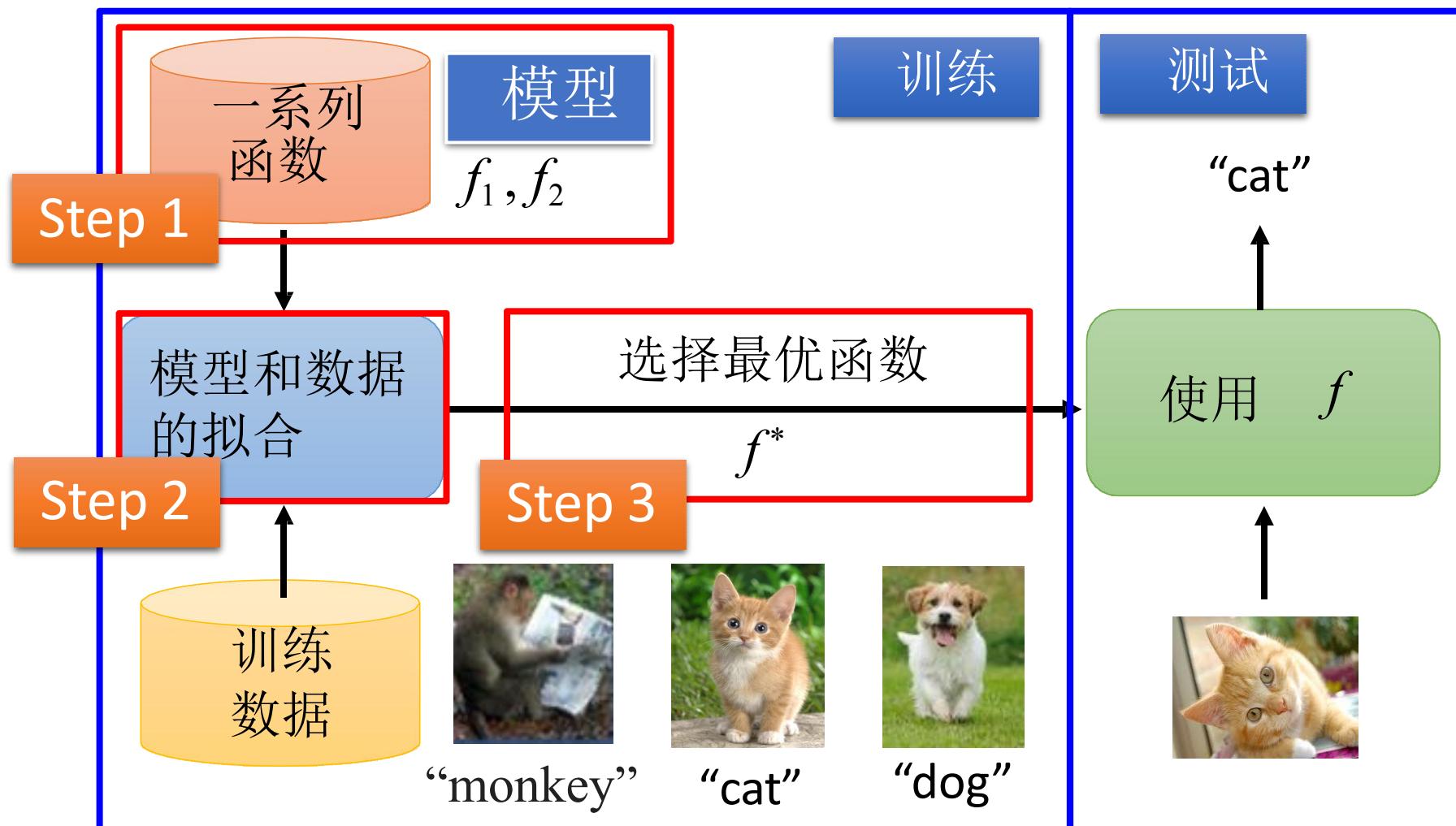


## 深度学习 – 识别“申”字



# 深度学习框架

图像识别:



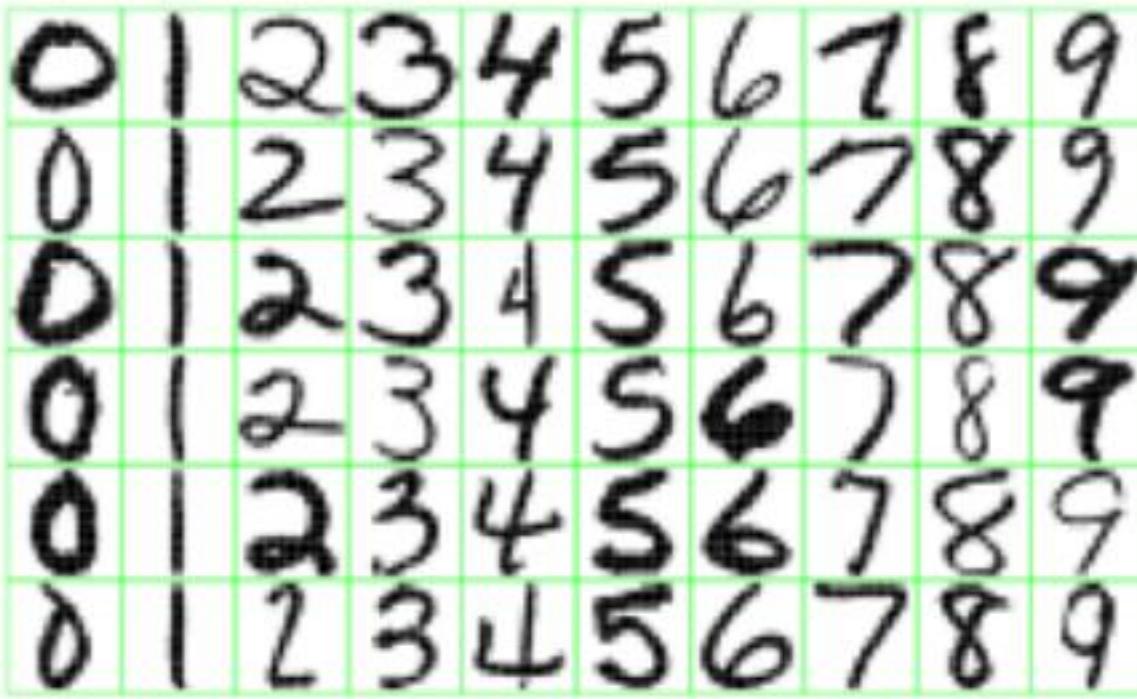


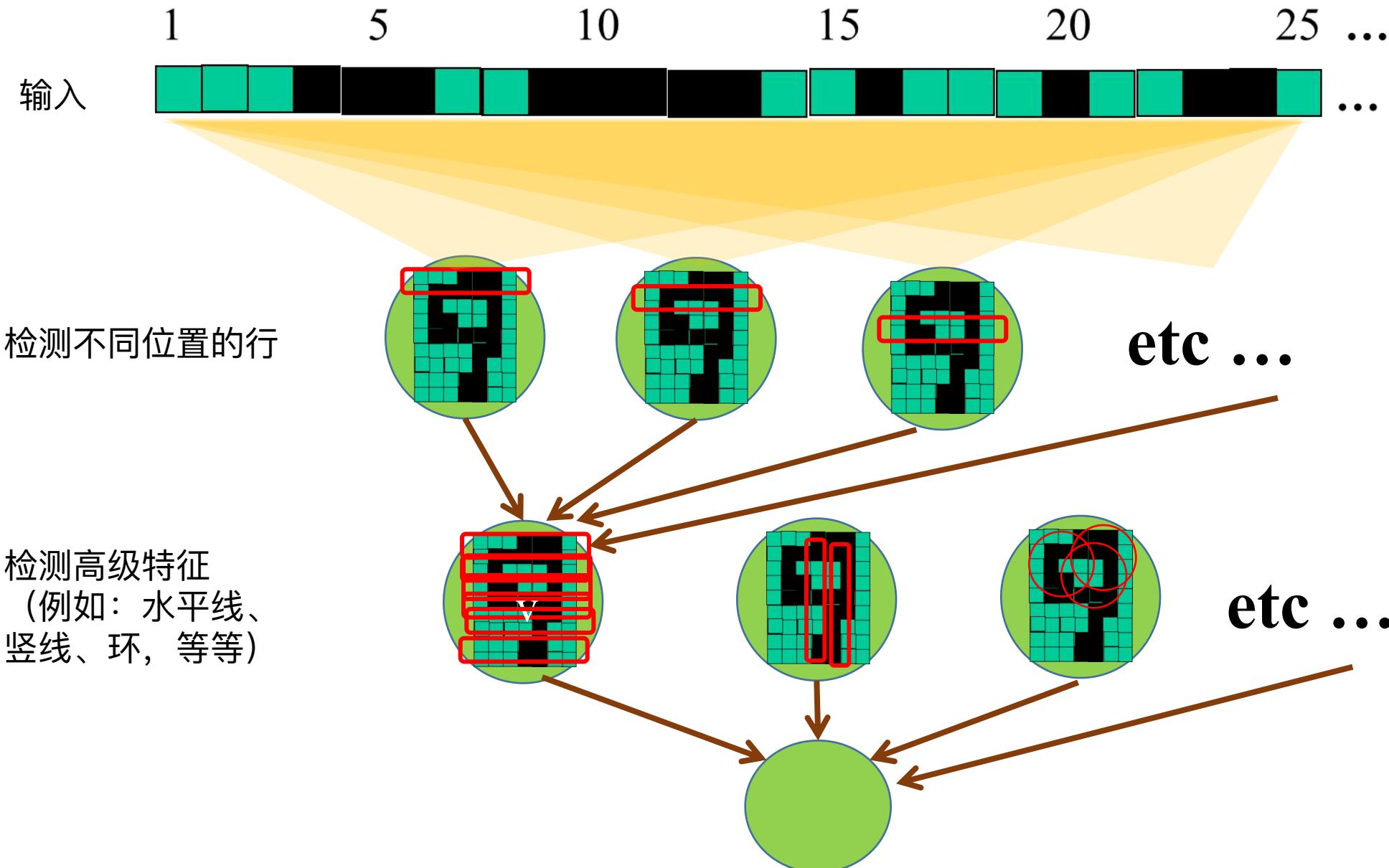
Figure 1.2: Examples of handwritten digits from U.S. postal envelopes.

你想让深度网络学到哪些特征？

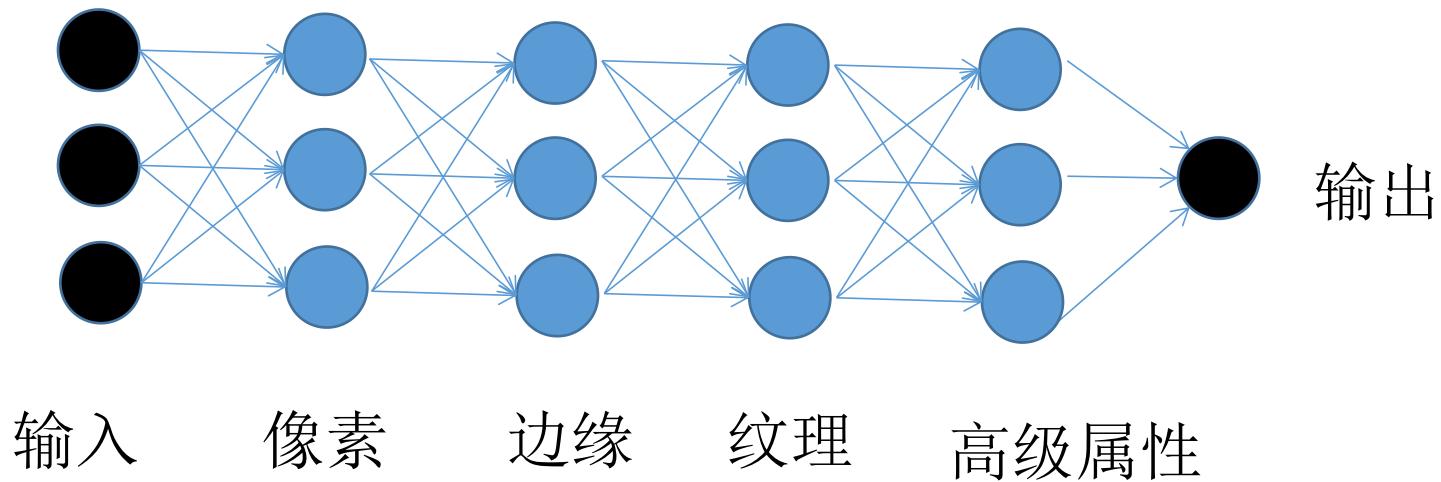


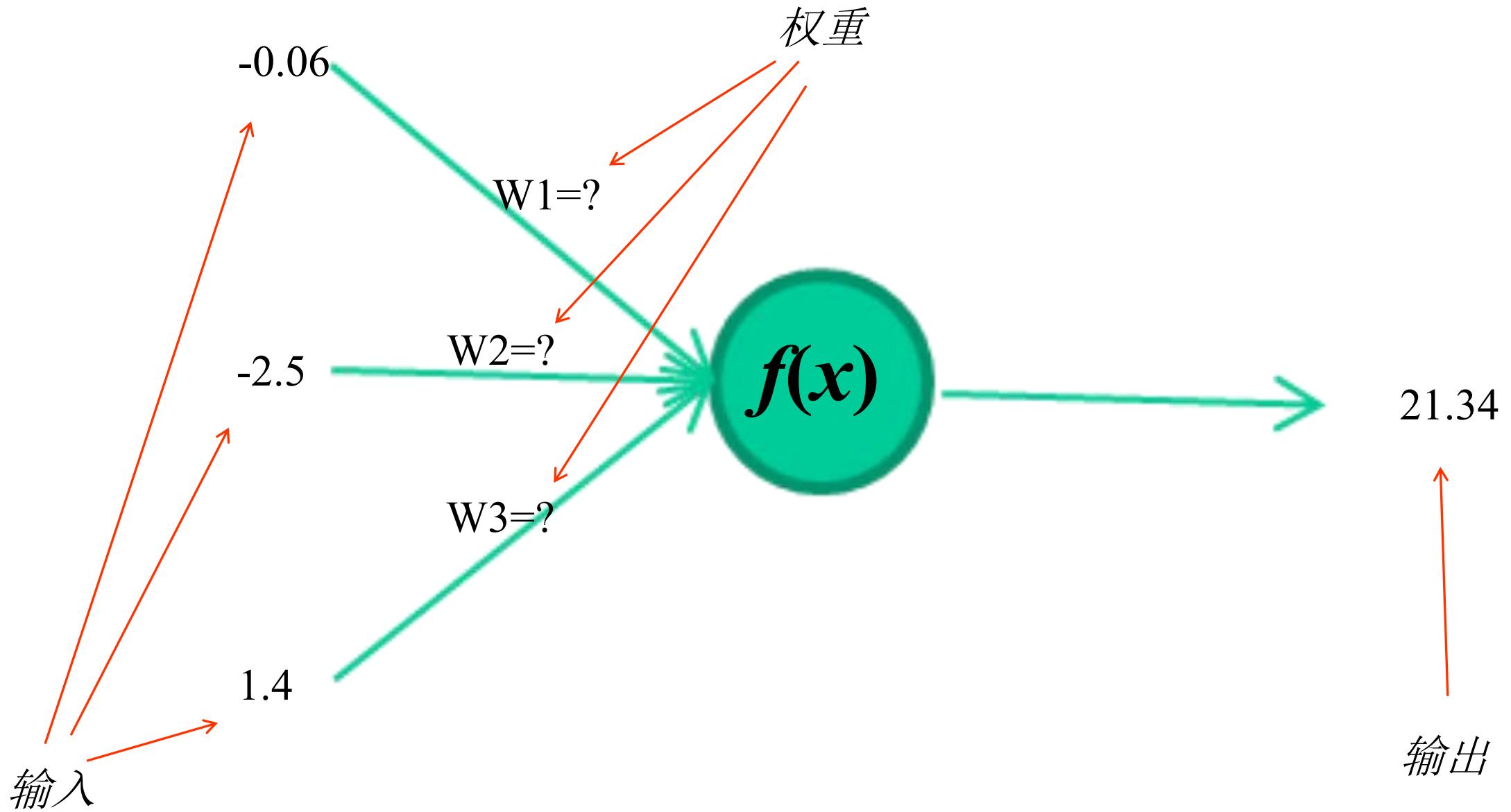
2: Examples of handwritten digits from U.S.  
postal envelopes.

# 有多个层时，后面的层可以用来学习高级特征



因此，多层网络可以实现复杂的检测和识别 --> DNN深度网络

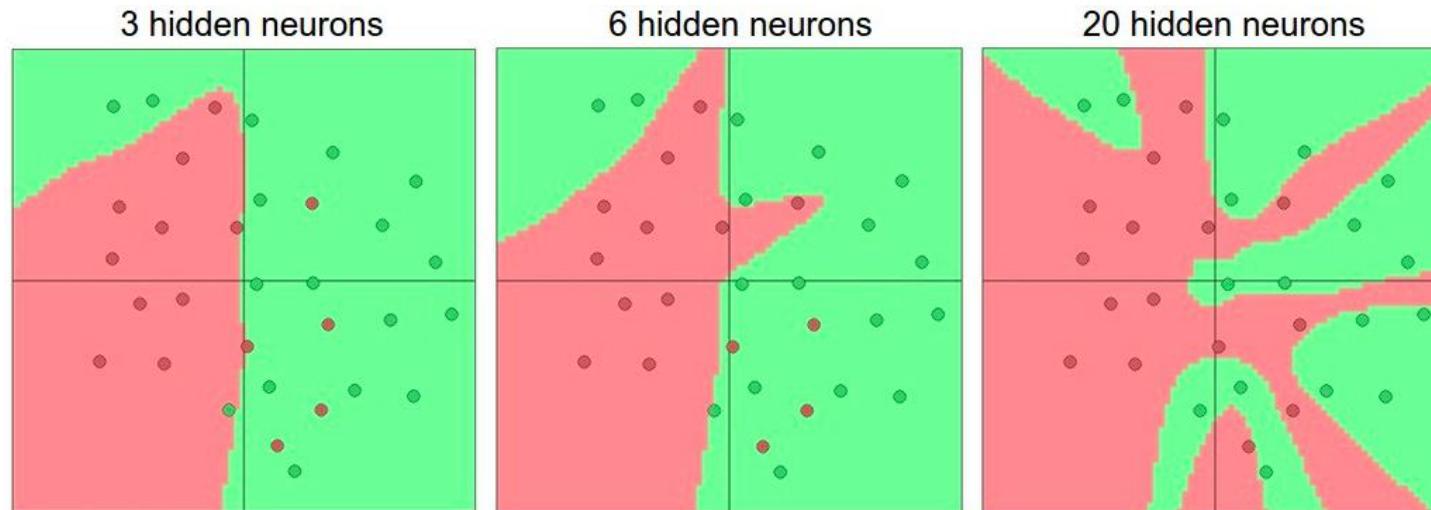




$$f(x) = -0.06 \times W1 + 2.5 \times W2 + 1.4 \times W3 = 21.34$$

非线性函数可以用来学习复杂的数据特征

$$W_1 W_2 x = Wx$$



[http://cs231n.github.io/assets/nn1/layer\\_sizes.jpeg](http://cs231n.github.io/assets/nn1/layer_sizes.jpeg)

更多层和神经节点数量可以拟合更复杂的函数

## 通俗的理解

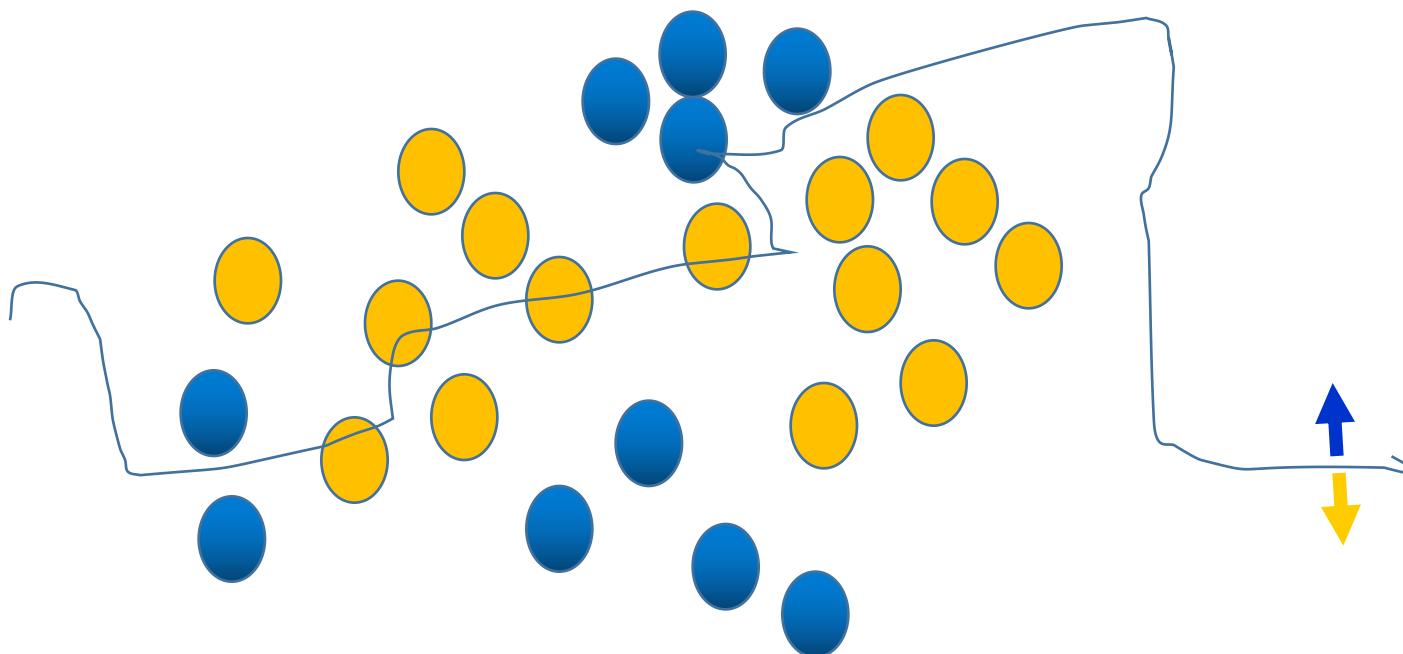
- 深度网络权重的学习算法是很傻瓜的！
- 通过成千上万的权重微调来学习，每次调整都试图使网络权重在下一次计算时表现更好，但在许多情况下可能会稍差一些。
- 如果运气足够好（大多时候是这样），最终可以学到有效进行分类的特征，可以应用到实际的分类操作中。

## 更严谨的理解

- 如果节点函数 $f(x)$ 是非线性的，从理论上讲，具有1个隐藏层的网络足以完美地学习任何分类问题。并且存在一组权重，可以从输入中产生与目标一致的输出。
- 问题是我们如何找到这组权重。

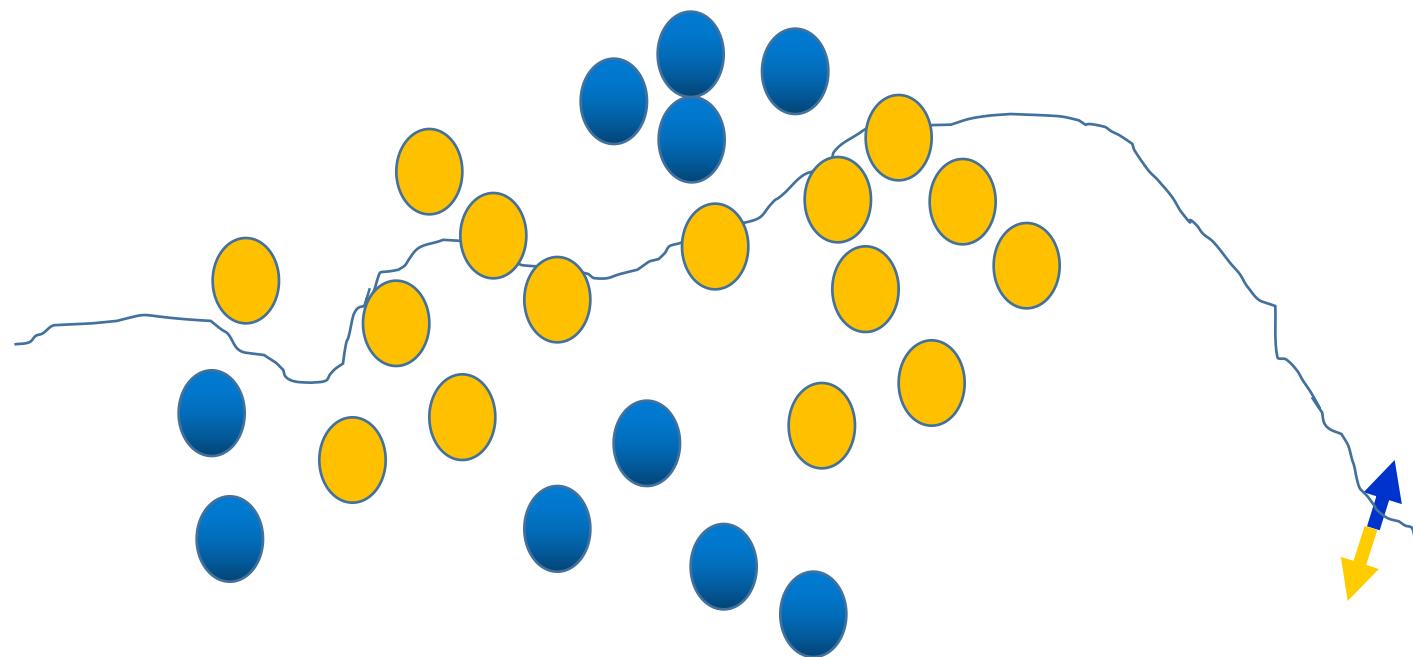
# 可视化的理解 – 寻找分类边界 (函数 $f(x)$ 的图像)

随机初始化时 – 边界是随机的



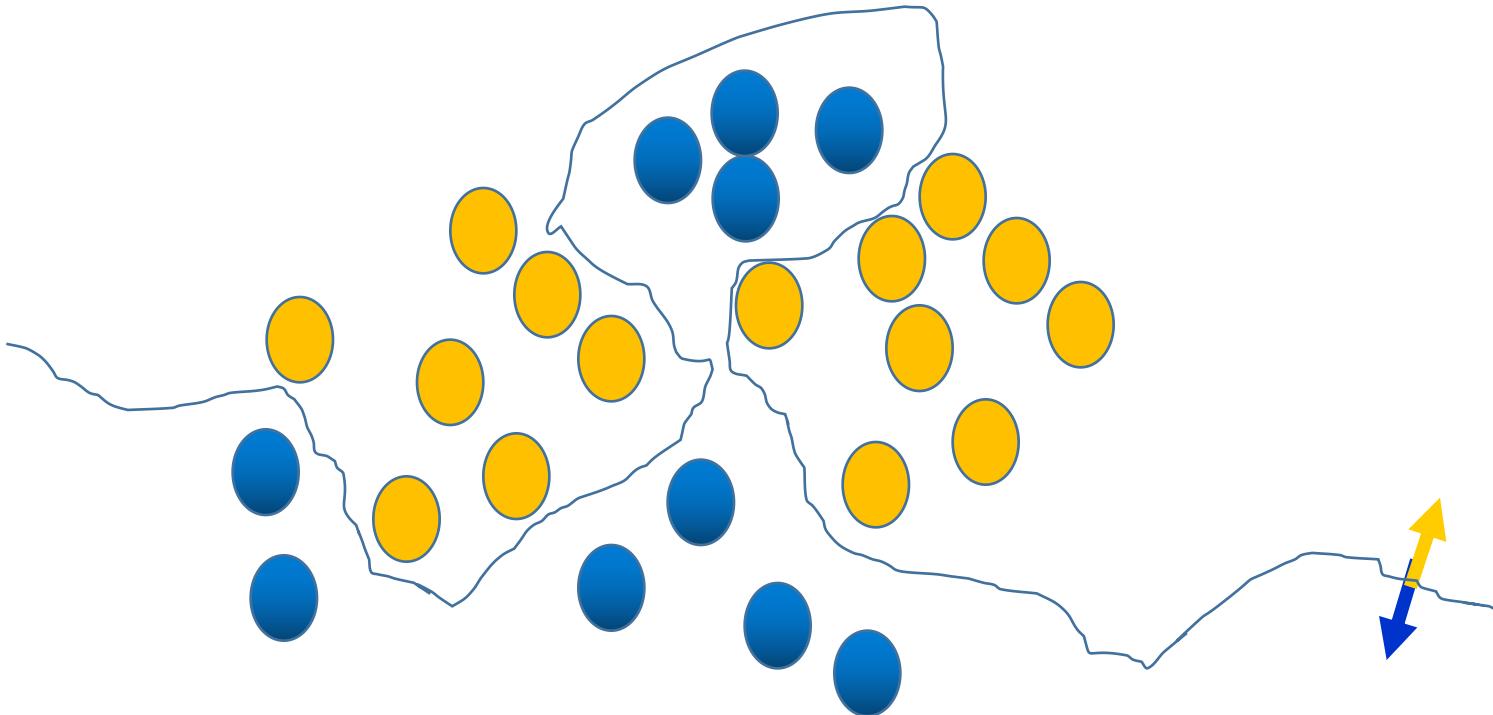
# 可视化的理解 – 寻找分类边界 (函数 $f(x)$ 的图像)

重复多次训练并调整权重后

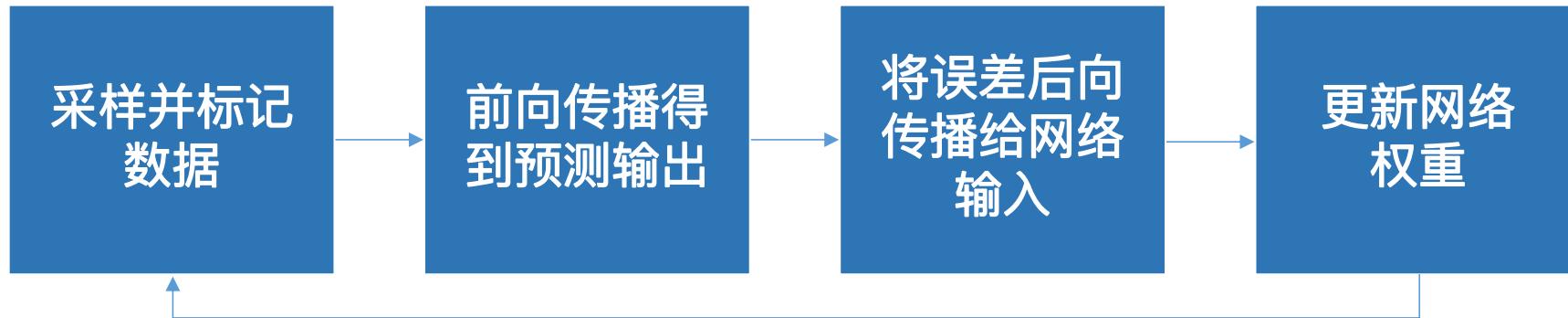


# 可视化的理解 – 寻找分类边界（函数 $f(x)$ 的图像）

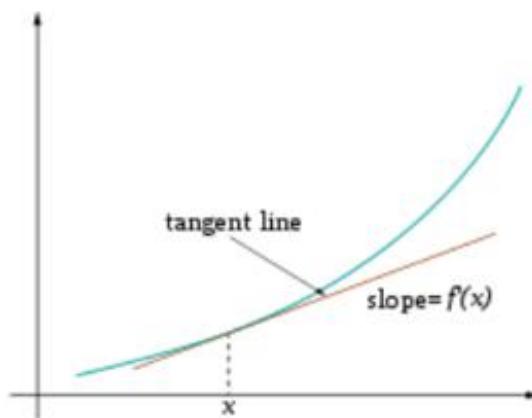
最终 ....



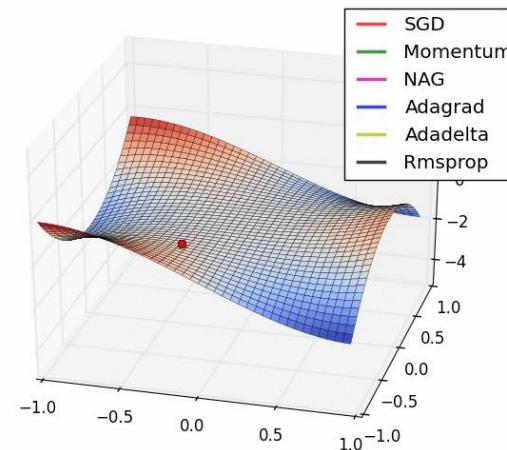
# 训练



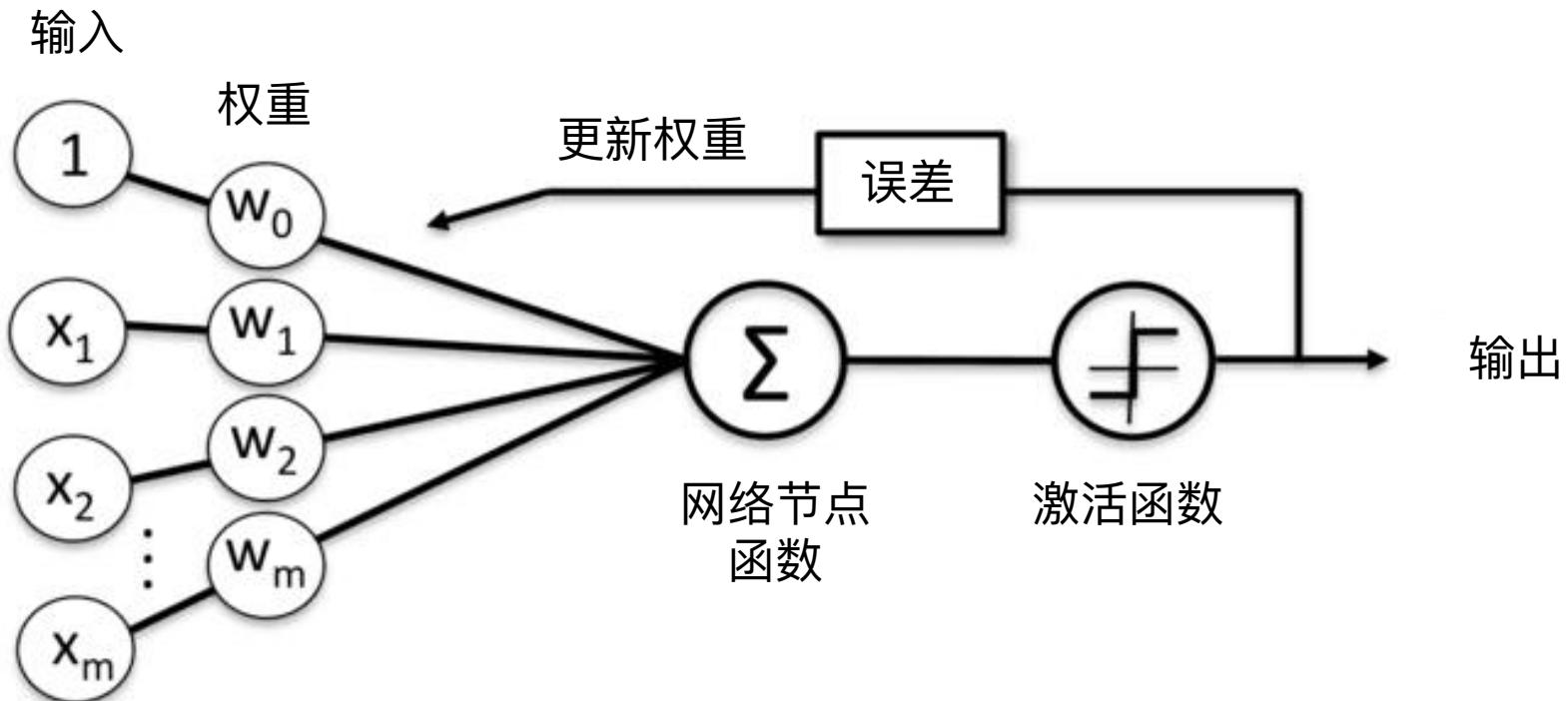
优化（最小或最大）**目标/损失函数** $J(\theta)$   
生成测量预测值与目标值之间差异的**误差信号**



使用**误差信号**调整**网络权重**并获得更准确的预测；  
减去**梯度**的一小部分使移动到损失函数的**(局部) 最小值**



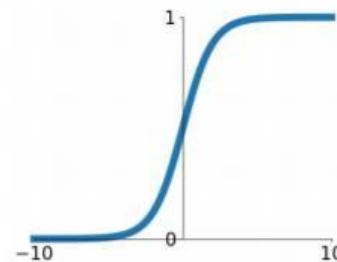
# 训练



# 激活函数

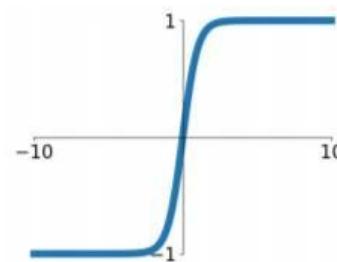
**Sigmoid**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



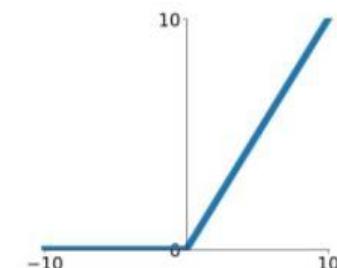
**tanh**

$$\tanh(x)$$



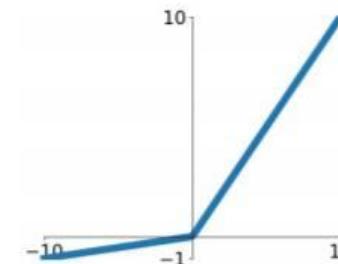
**ReLU**

$$\max(0, x)$$



**Leaky ReLU**

$$\max(0.1x, x)$$

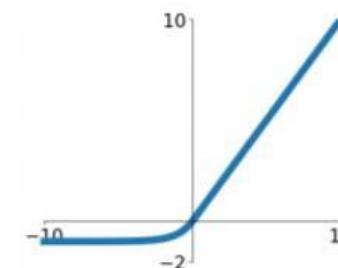


**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

**ELU**

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



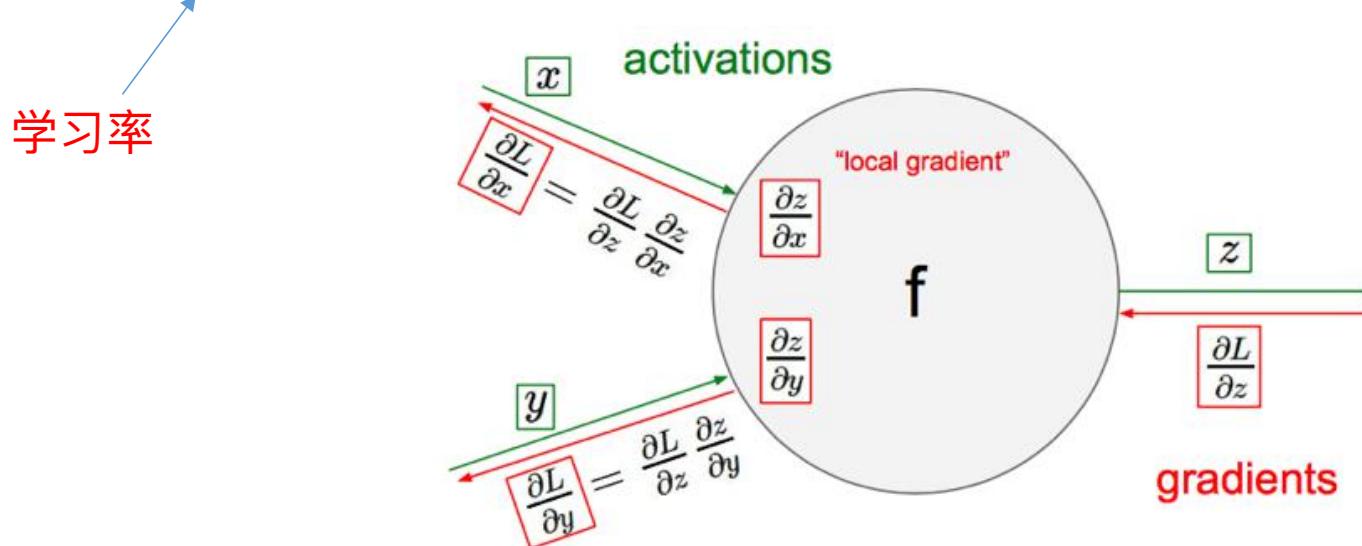
# 梯度下降

目标函数/损失函数  $J(\theta)$

反向传播

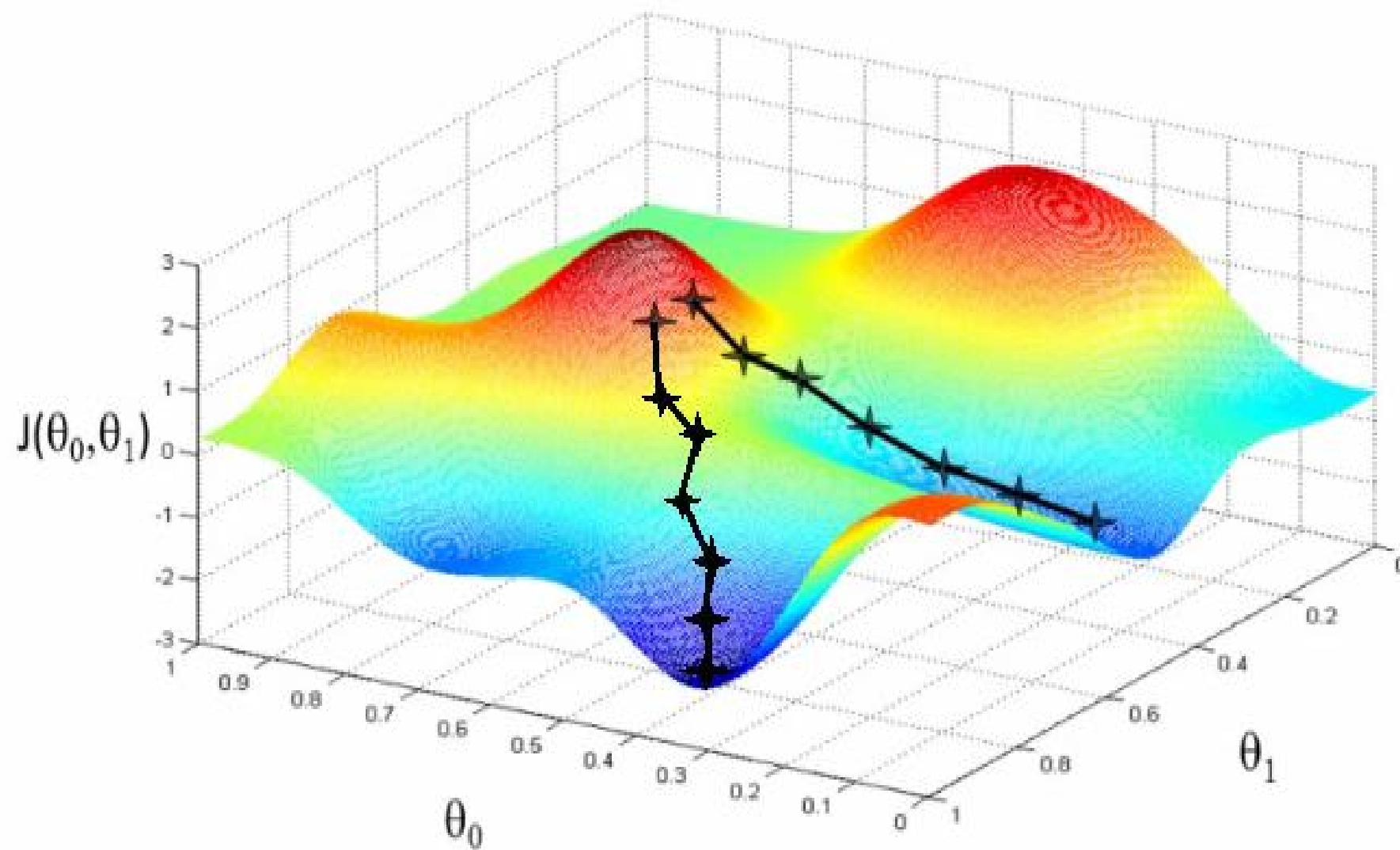
$$\theta_j^{new} = \theta_j^{old} - \alpha \frac{d}{d\theta_j^{old}} J(\theta) \quad \text{每次更新 } \theta$$

$$\theta^{new} = \theta^{old} - \alpha \nabla_{\theta} J(\theta) \quad \text{将所有参数向量化}$$

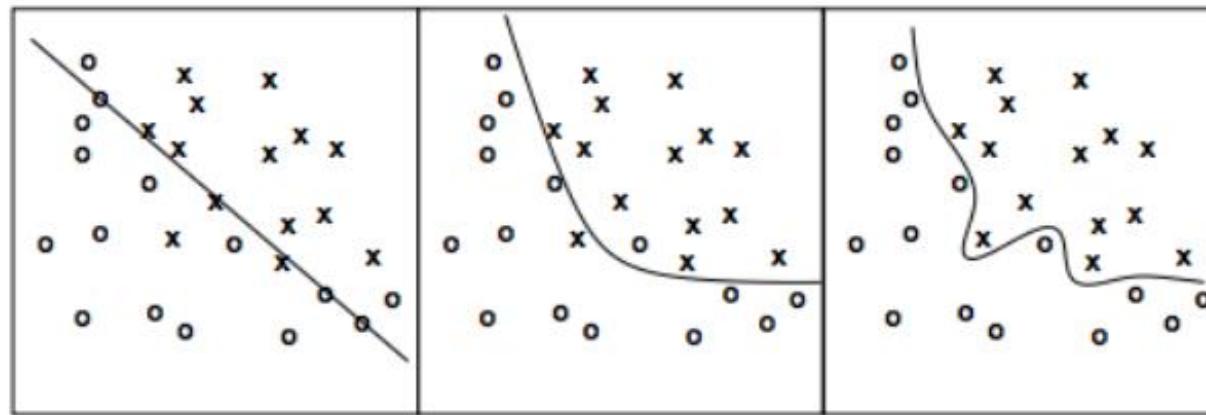


对每个节点递归应用规则链式规则 (求复合函数导数时一种法则)

# 梯度下降



# 过拟合

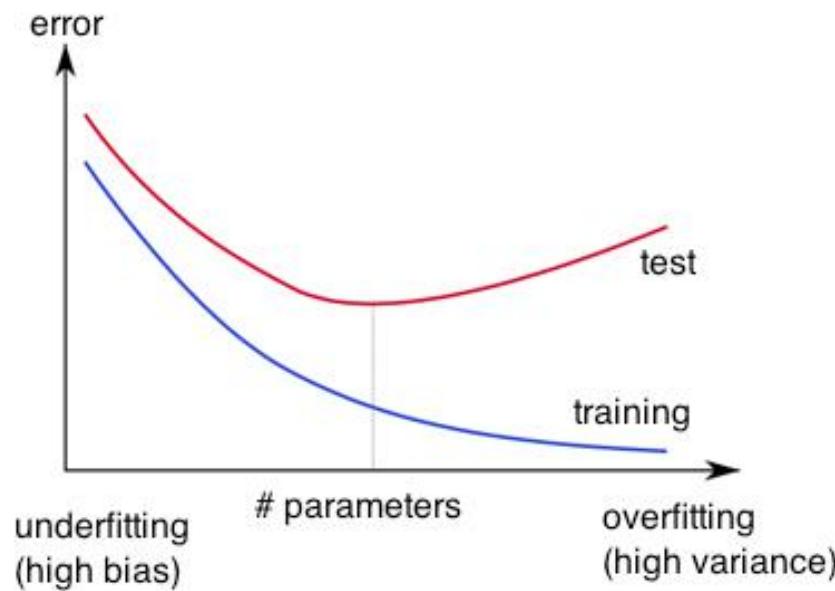


拟合不足

拟合较好

过拟合

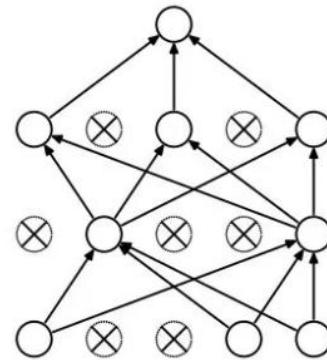
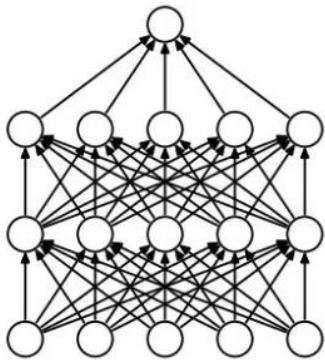
<http://wiki.bethanycrane.com/overfitting-of-data>



学习到的特征非常拟合训练数据，  
甚至是特异值（**噪声**），但无法  
对新样本（测试数据）进行**泛化**。

[https://www.neuraldesigner.com/images/learning/selection\\_error.svg](https://www.neuraldesigner.com/images/learning/selection_error.svg)

# 防止过拟合



## Dropout

- 在训练时随机去掉一些节点
- 每个节点保持固定的概率 $p$ , 独立与其他节点
- 超参数 $p$ 可以被手工调整

Srivastava, Nitish, et al. "["Dropout: a simple way to prevent neural networks from overfitting."](#)" Journal of machine learning research (2014)

## L2 = weight decay 权重衰减

- Regularization term that penalizes big weights, added to the objective
- Weight decay value determines how dominant regularization is during gradient computation
- Big weight decay coefficient → big penalty for big weights

$$J_{reg}(\theta) = J(\theta) + \lambda \sum_k \theta_k^2$$

## Early-stopping 提前停止

- 使用验证集的错误率觉得何时停止训练
- 在n次训练后, 如果网络性能未得到提升, 则停止训练。n为忍耐度

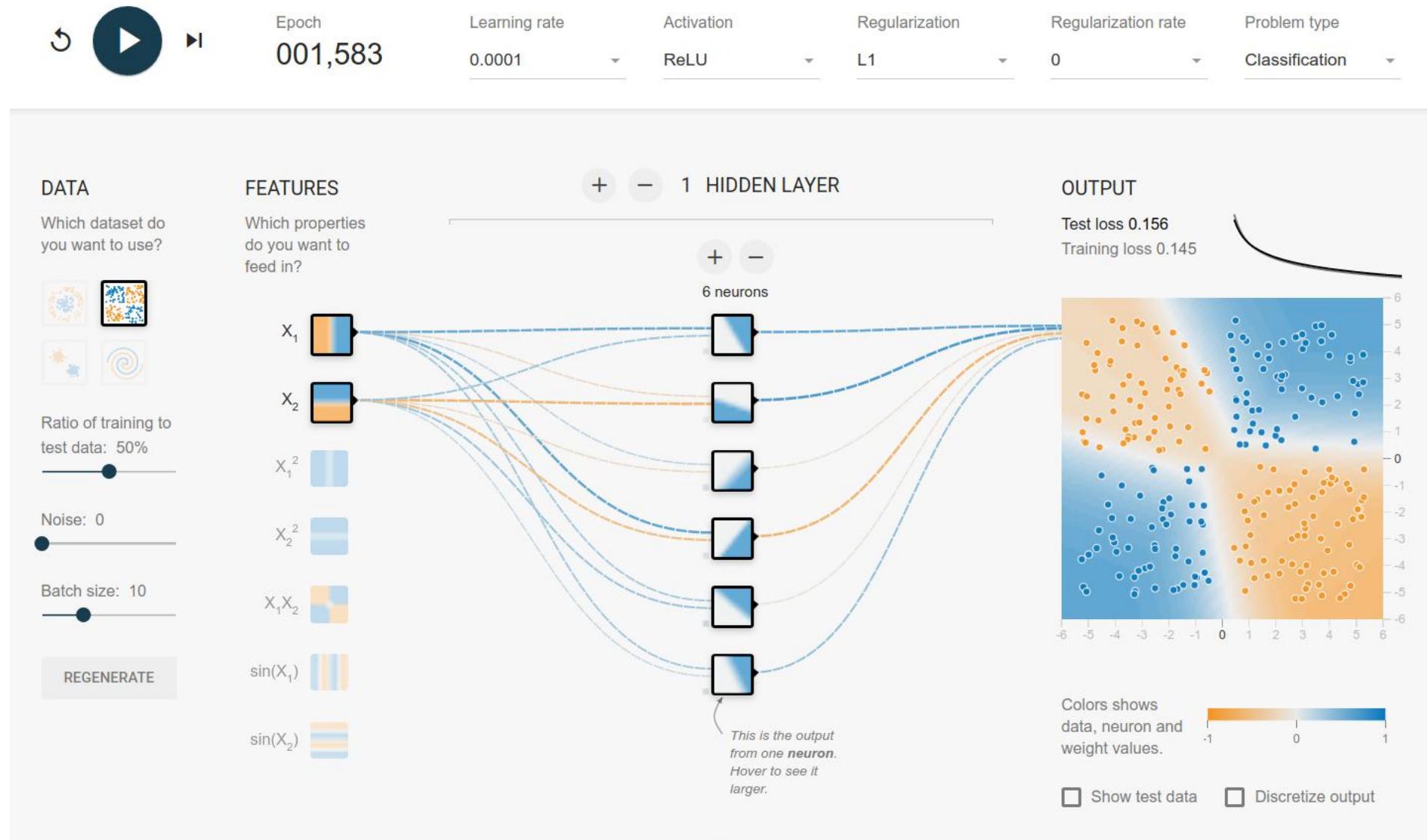
# 超参数

参数（权重）是深度神经网络最终要学习的目标。

超参数是我们控制模型结构、功能、效率等的“调节旋钮”。

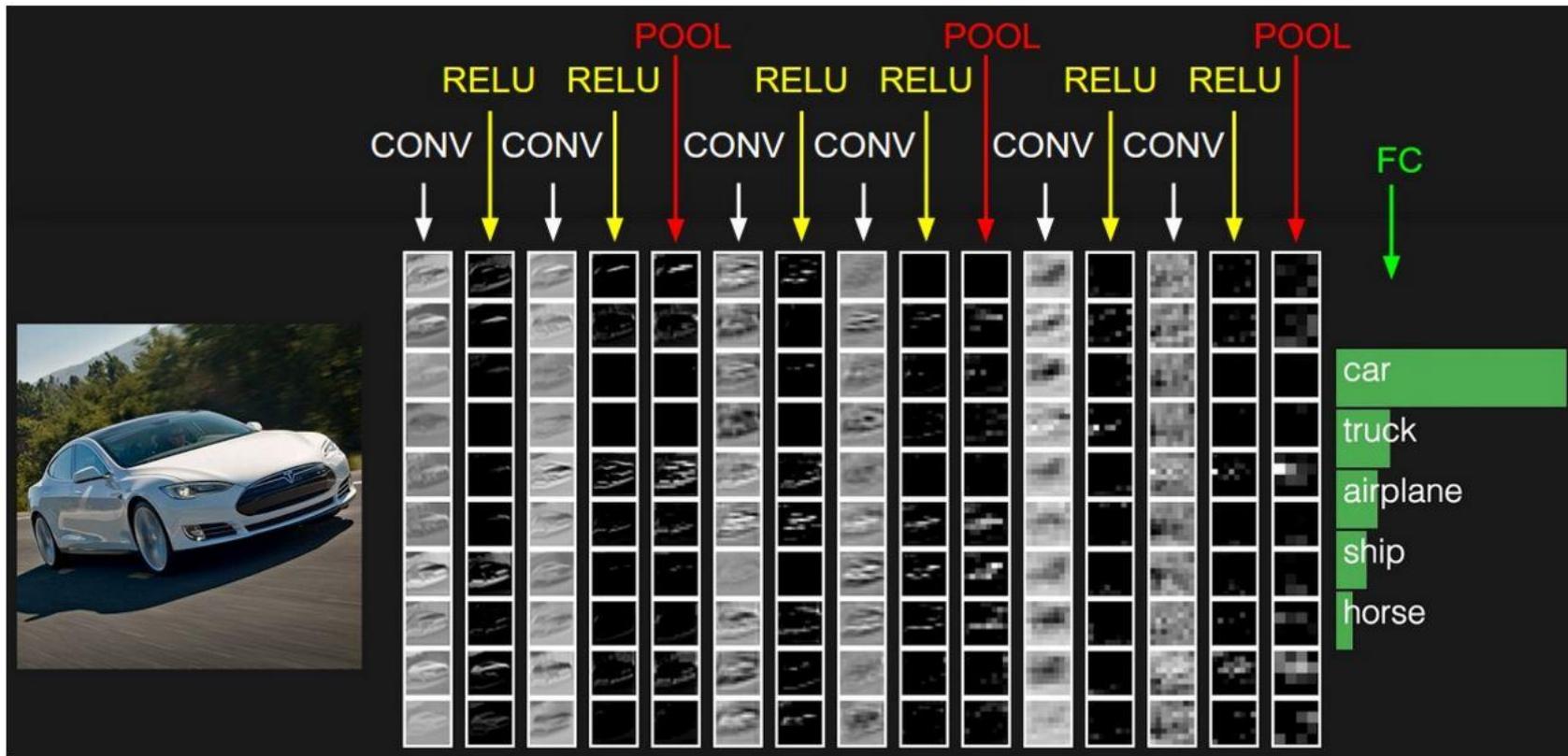
- learning rate (学习率)
- epochs (迭代次数)
- num of hidden layers (隐藏层数目)
- num of hidden layer units (隐藏层的神经元/节点数)
- activation function (激活函数)
- batch-size (用mini-batch SGD的时候每个批量的大小)
- optimizer (优化器, 如SGD、RMSProp、Adam)
- .....

# 可视化的演示：<http://playground.tensorflow.org>



# CNN – 卷积神经网络

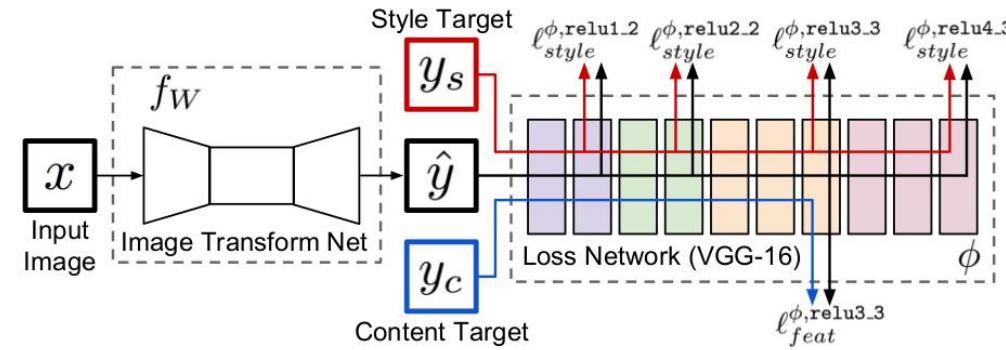
# CNN的例子：图像分类（识别）



# 目标识别的例子

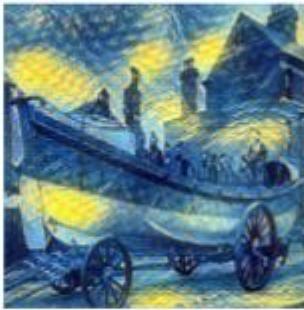


# 图片风格迁移



**Style**

*The Starry Night*,  
Vincent van Gogh,  
1889

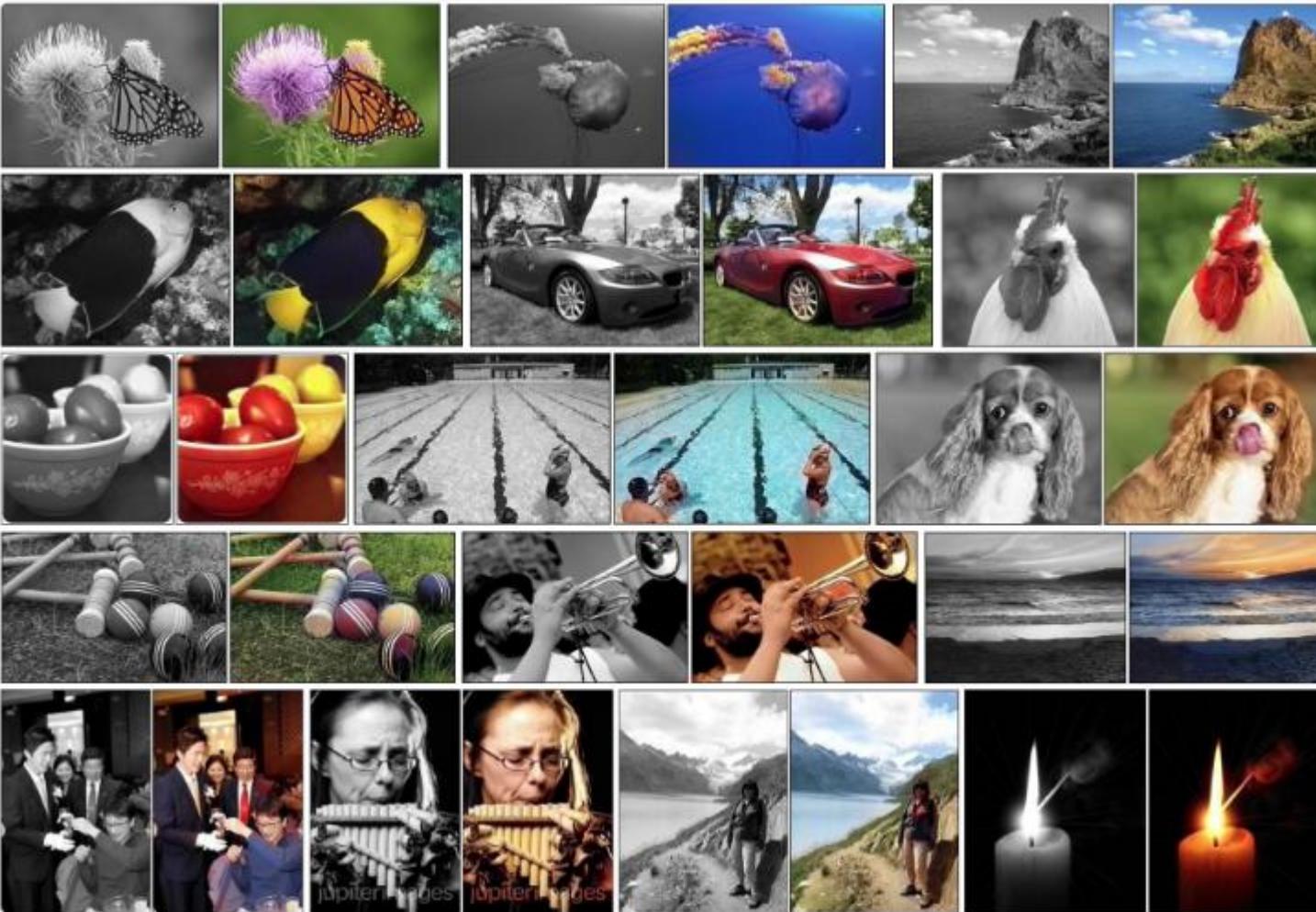


**Style**

*The Muse*,  
Pablo Picasso,  
1935



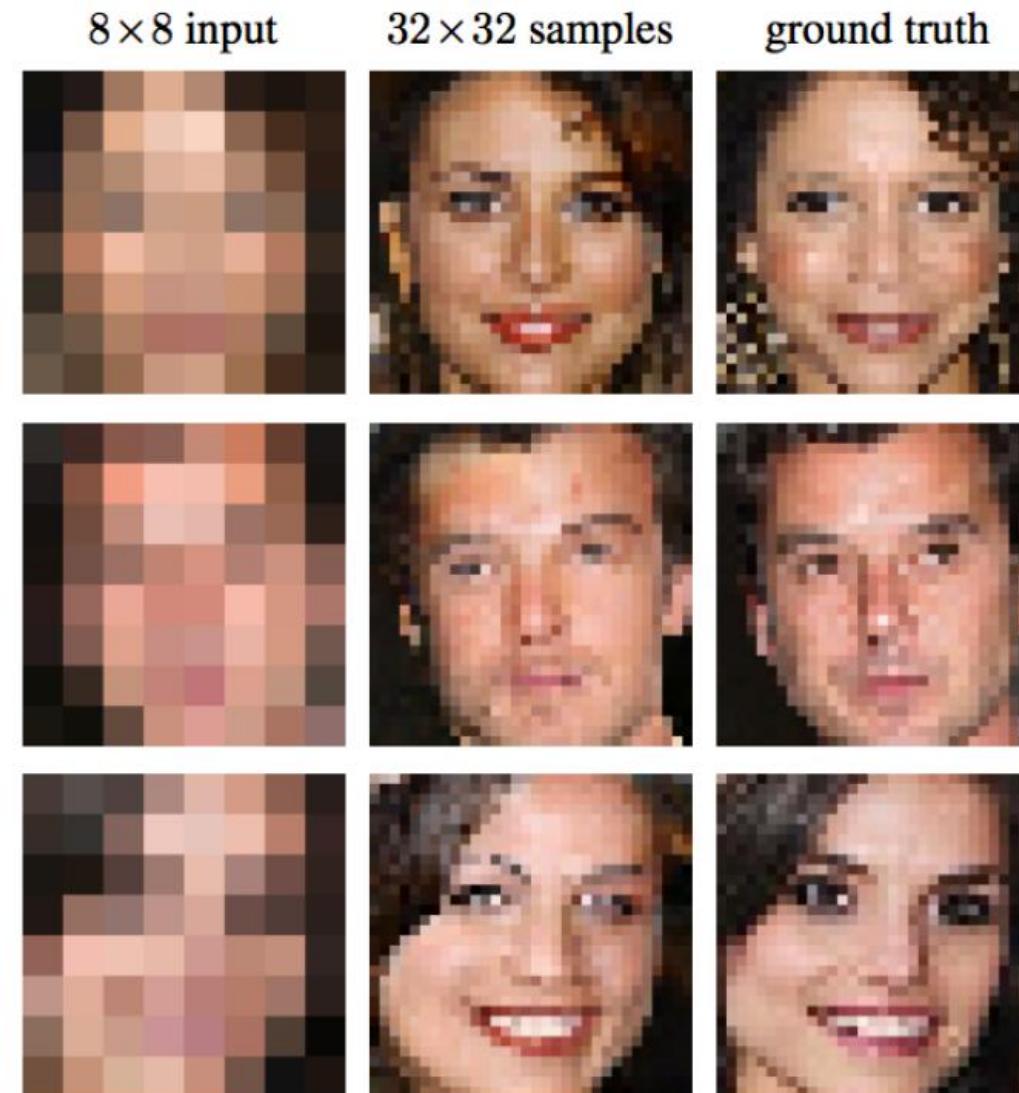
# 为黑白照片自动上颜色



# 对照片进行优化

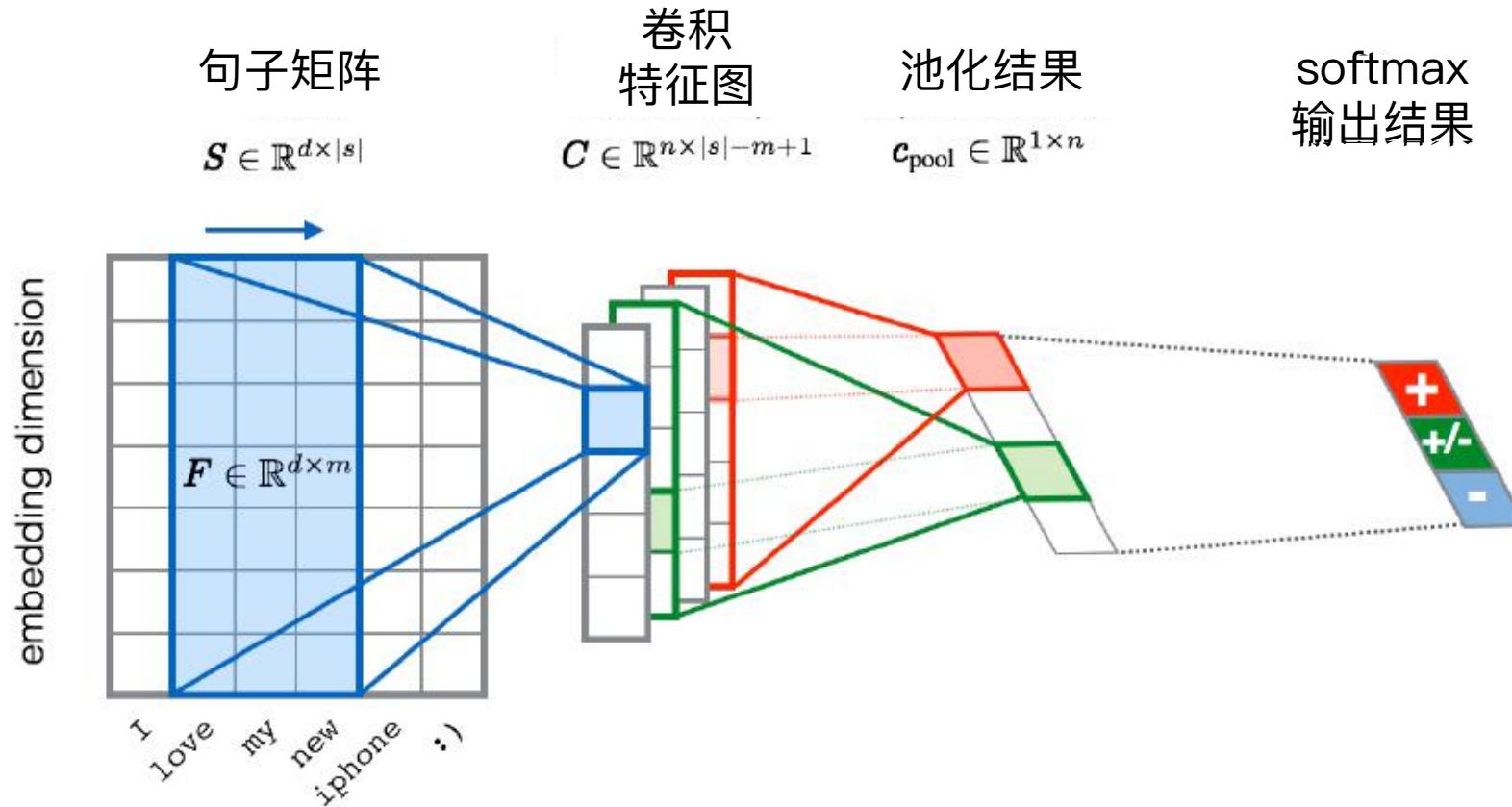


# 去除马赛克



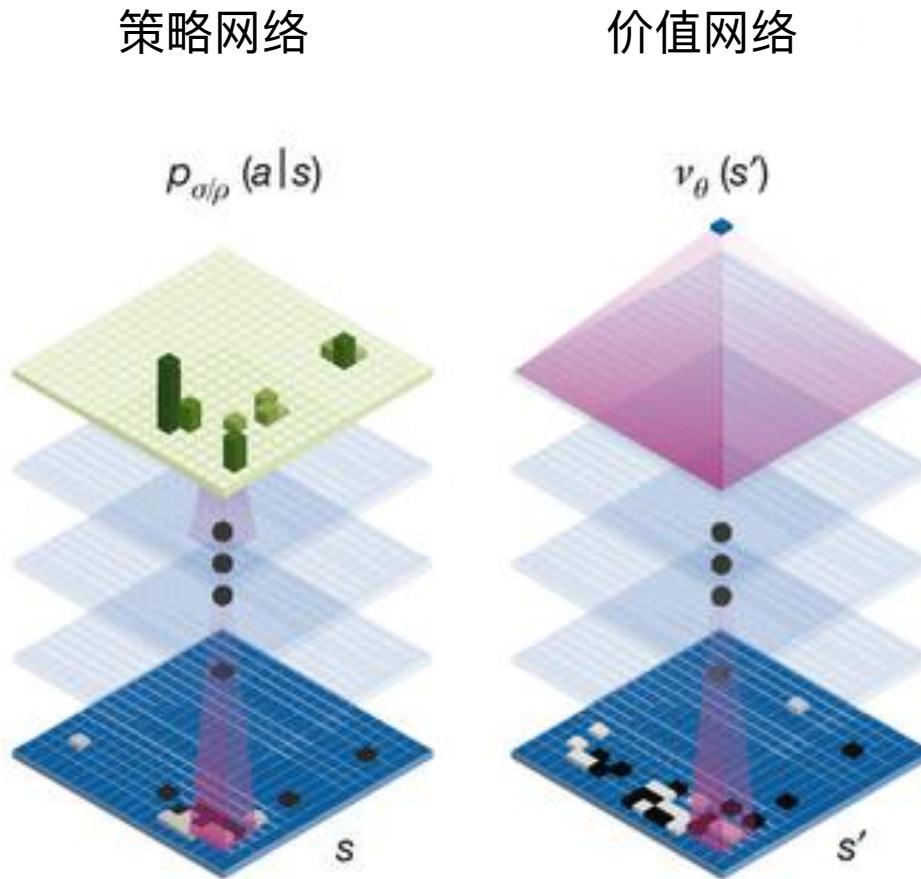
8x8 pixel photos were inputted into a Deep Learning network which tried to guess what the original face looked like. As you can see it was fairly close (the correct answer is under "ground truth").

# CNN网络用于文本情感分类



Severyn, Aliaksei, and Alessandro Moschitti. "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification." *SemEval@ NAACL-HLT*. 2015.

# AlphaGO的CNN网络



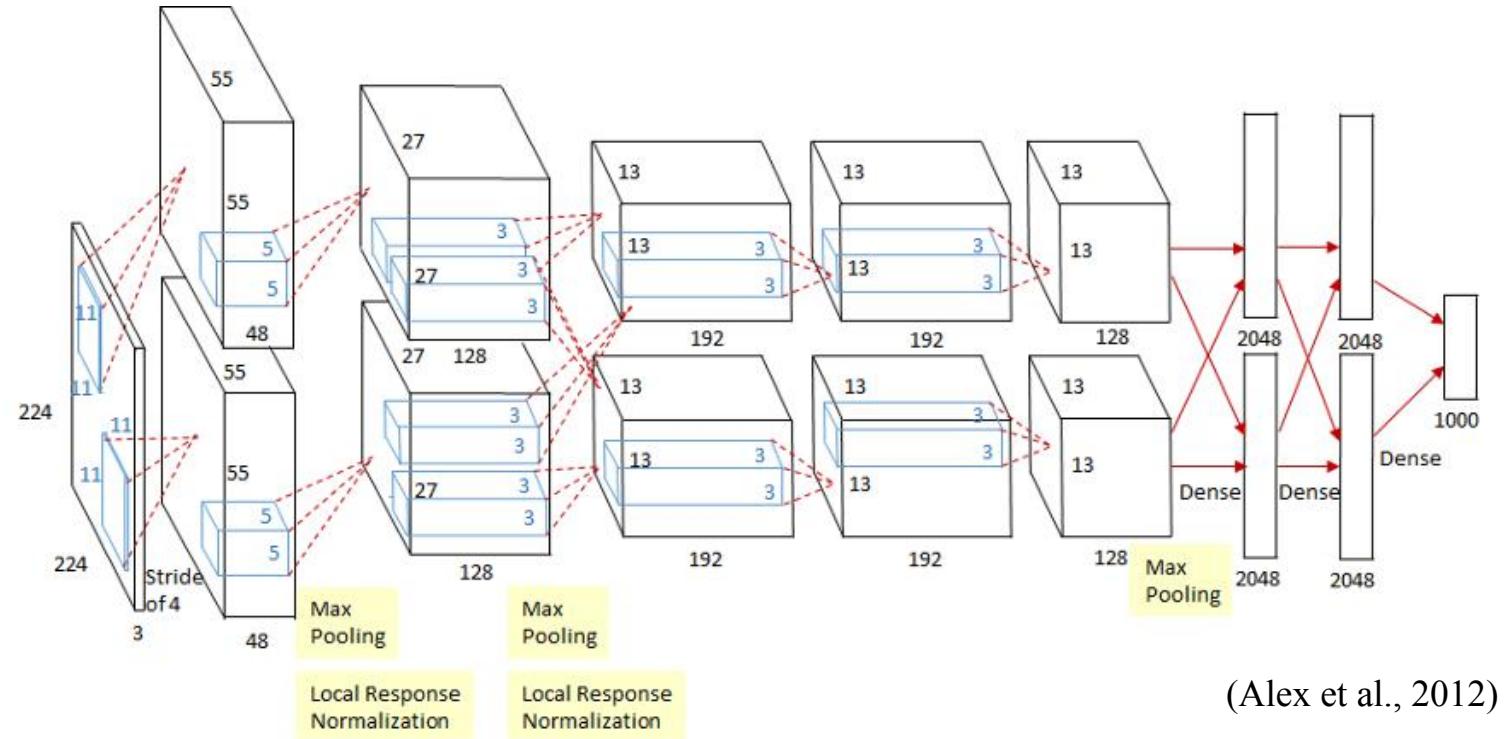
(Silver et al, 2016)

策略网络:

- 输入:  $19 \times 19$ , 48 通道
- 层1:  $5 \times 5$  卷积核, 192 过滤器
- 层2到12:  $3 \times 3$  卷积核, 192 过滤器
- 层13:  $1 \times 1$  卷积核, 1 过滤器

价值网络结构与策略网络结构相同

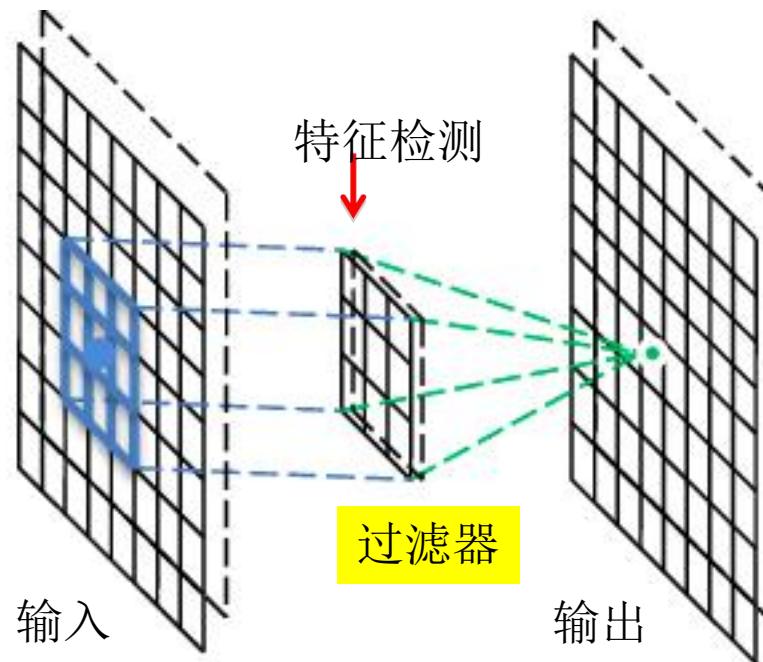
# AlexNet: ImageNet 2012 冠军



- 每个卷积层对应多个特征
- 不同卷积层抽取不同等级的特征
- 较高层将较低层中的特征矩阵作为输入

# 卷积层

CNN是由一些卷积层组成的神经网络（以及其他一些层），  
每个卷积层有一些用于进行卷积运算的过滤器（用于特征检测）。



# 卷积运算

- Filter



$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



输入图像



卷积后的图像

# 卷积运算

步进=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

点积  
结果



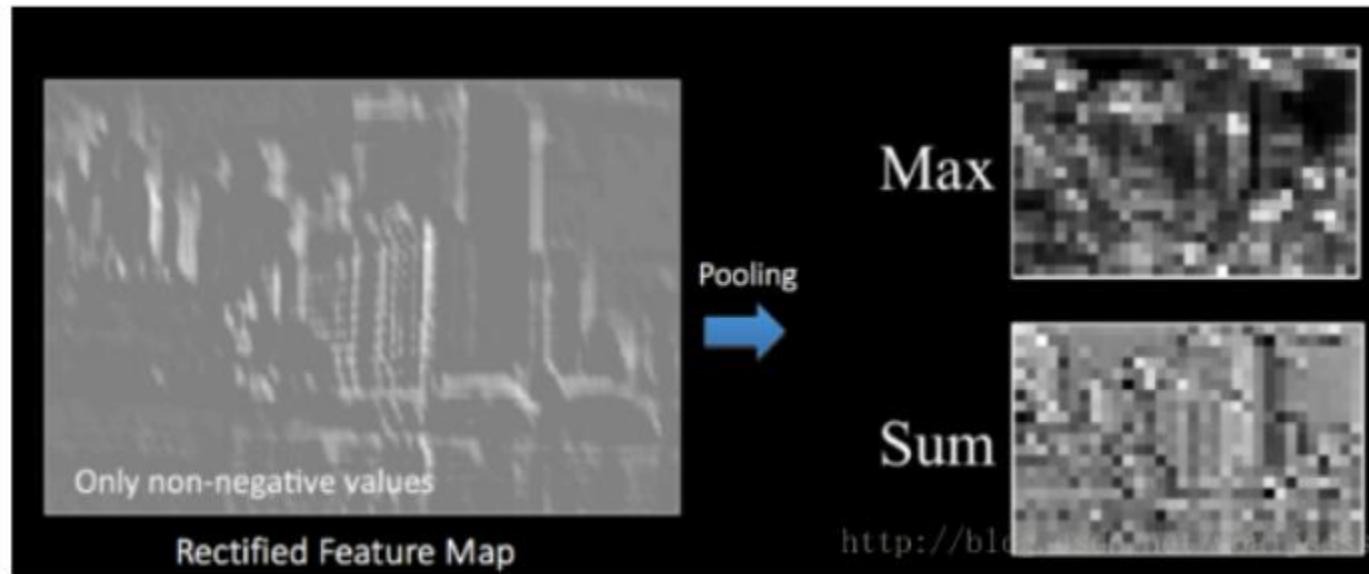
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

6 x 6 图像

# 为什么要池化

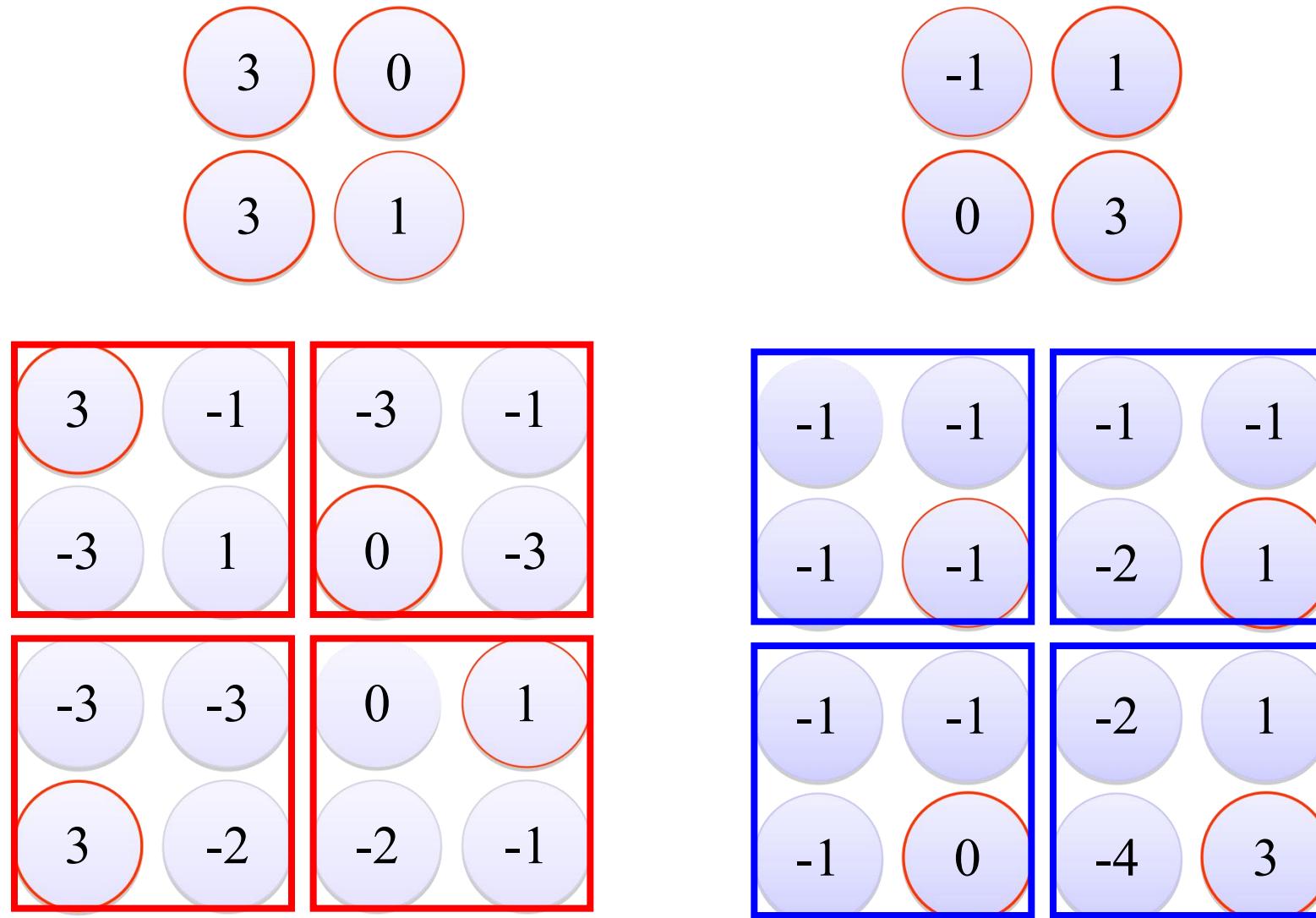
- 对像素抽样并不影响图片特征



可以将图片缩小，而不改变特征

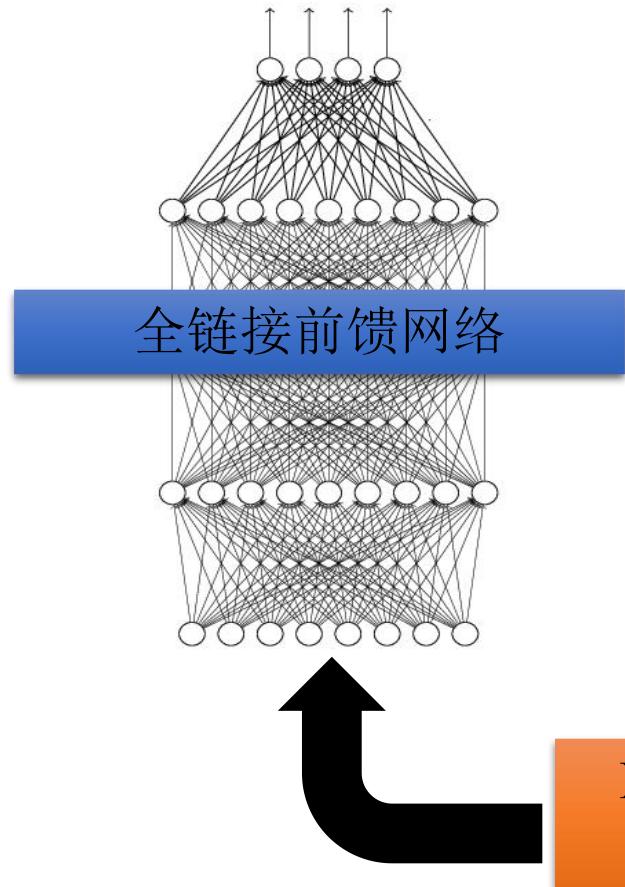
→ 更小的图像只需要更少的参数，可以突出特征图

# Max Pooling 池化



# 完整的 CNN

猫？ 狗？



可以重复n多次

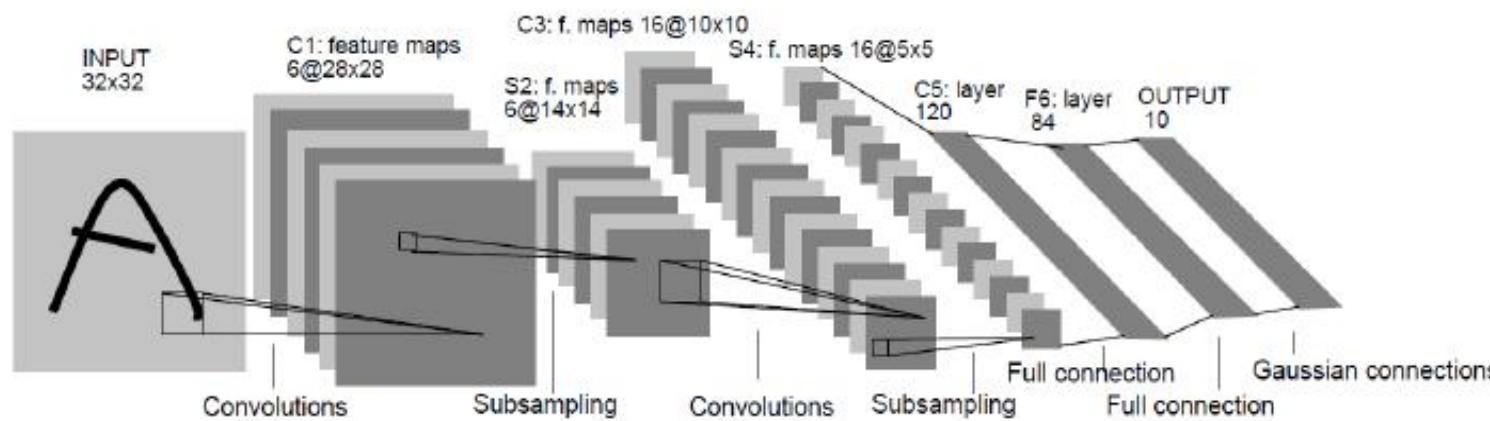
Flattened  
展平

# 模型示例 - LeNet

## Convolutional Networks Used Case

Local receptive field + weight sharing

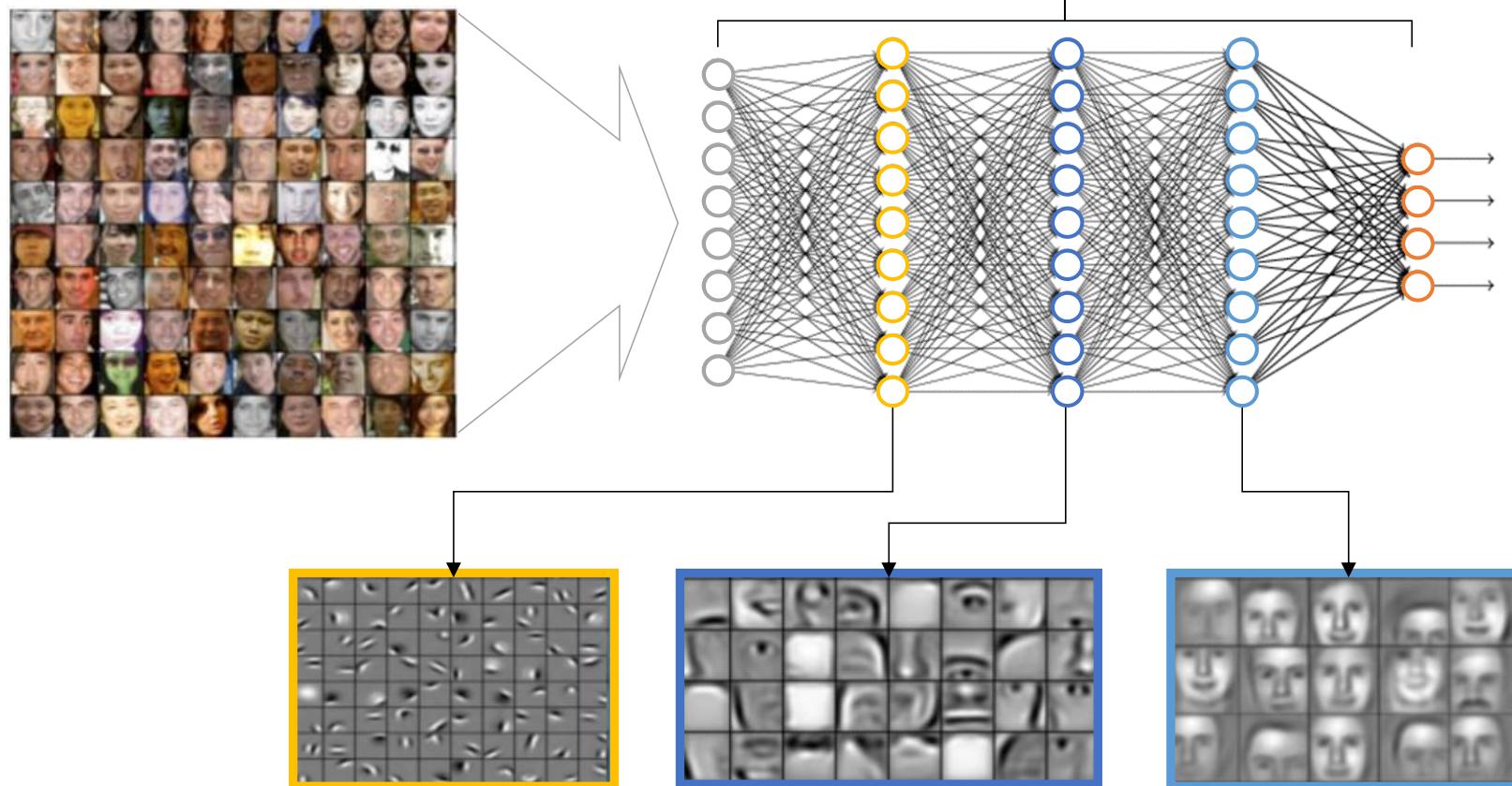
Yann LeCun et al, 1998



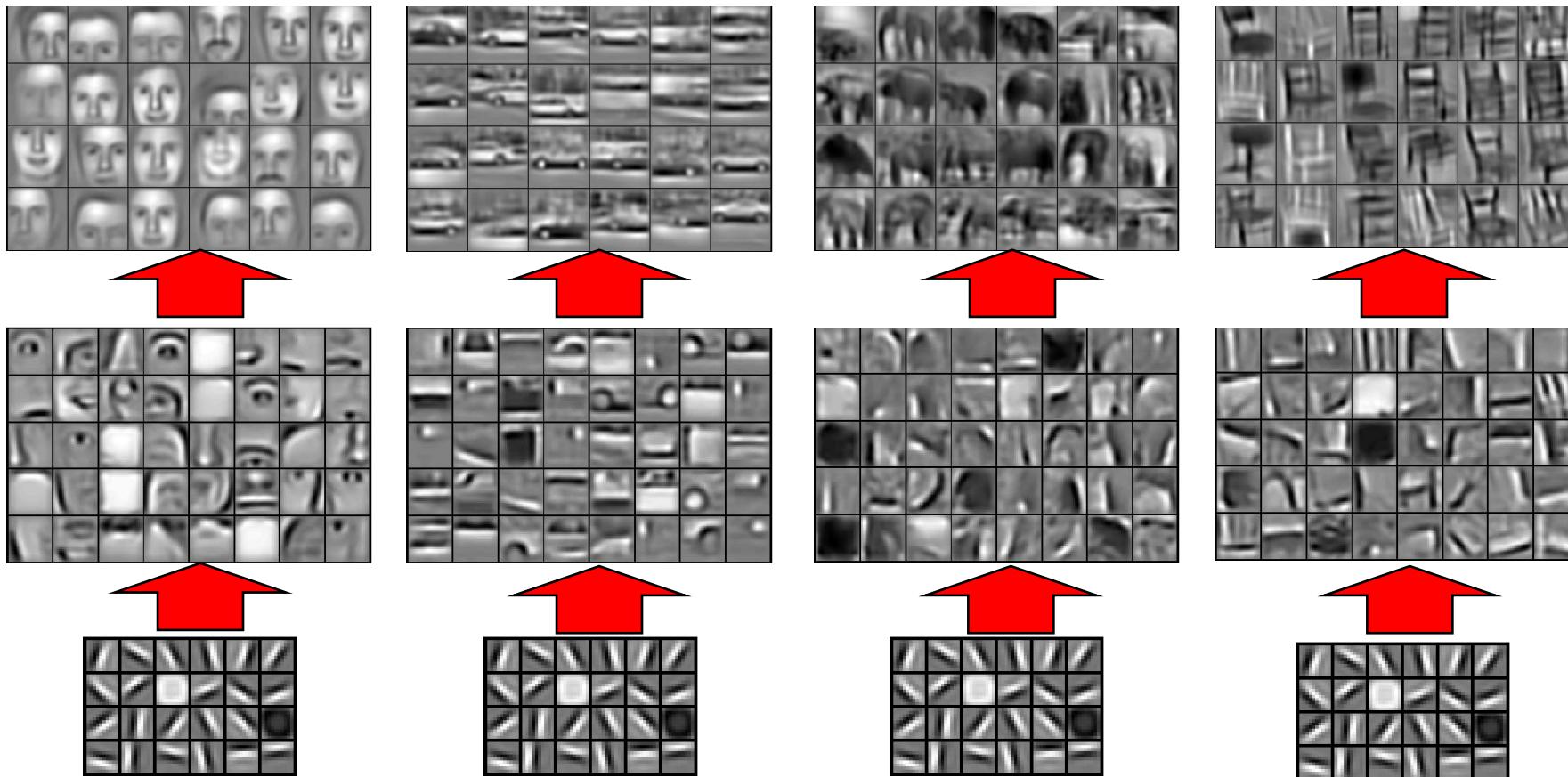
- ▶ MNIST: 0.7% error rate

# 模型示例

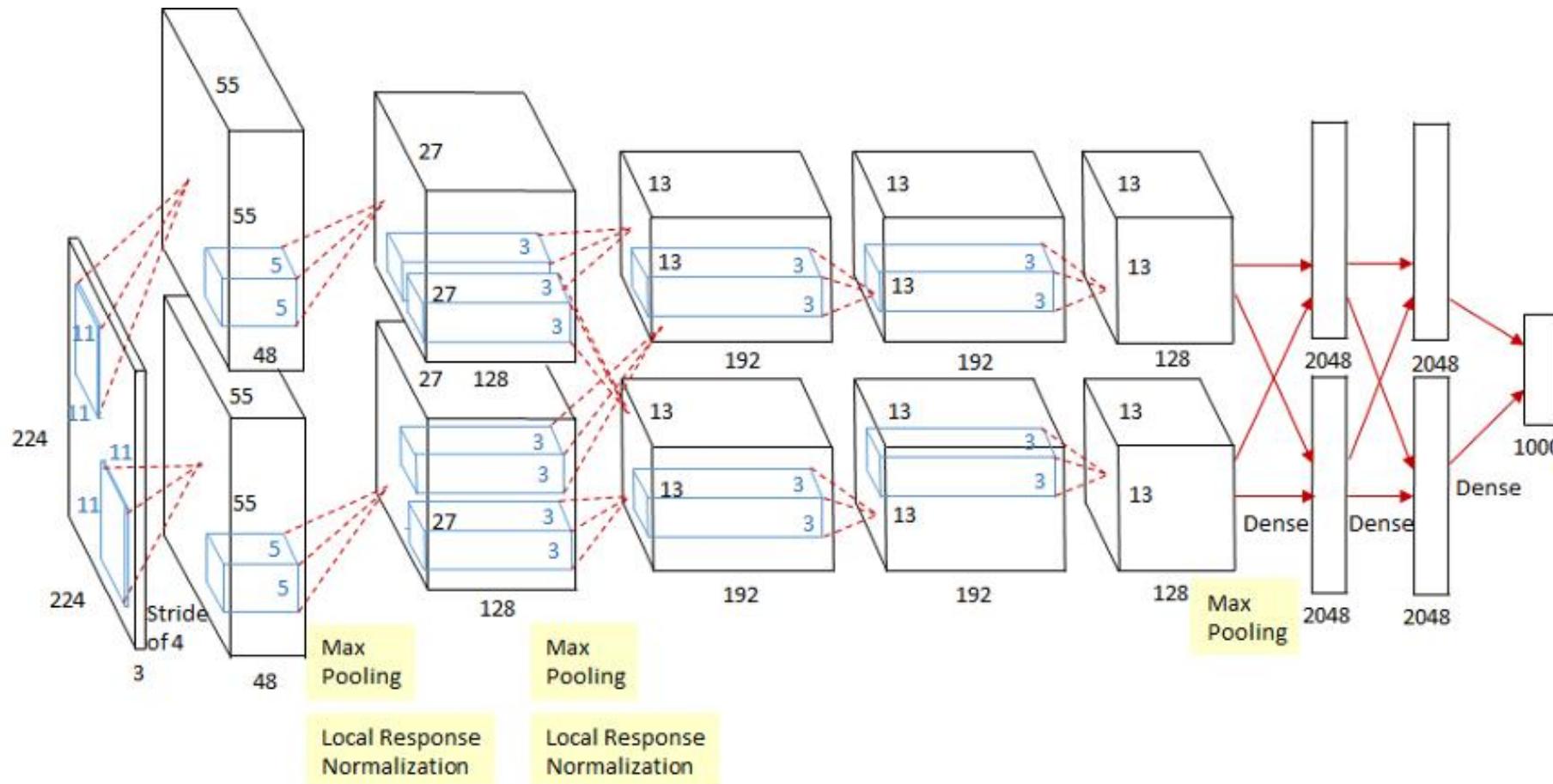
## 卷积神经网络



# 对目标局部特征的学习

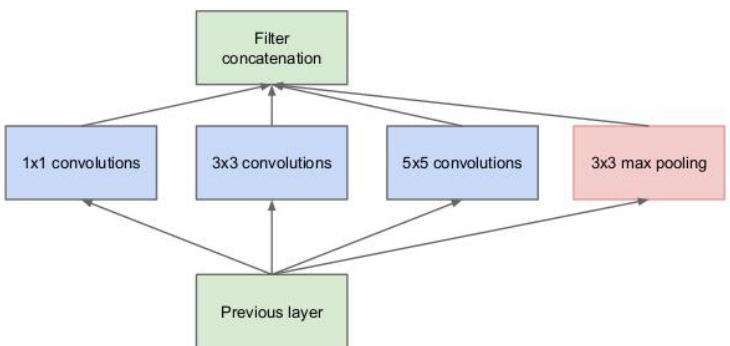
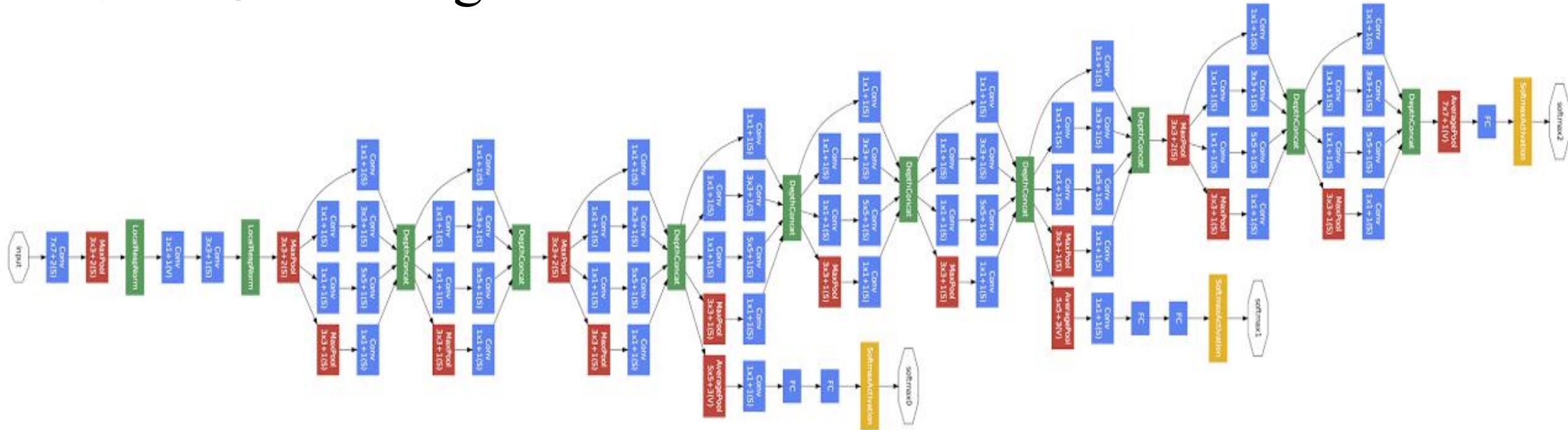


# 模型示例 – AlexNet

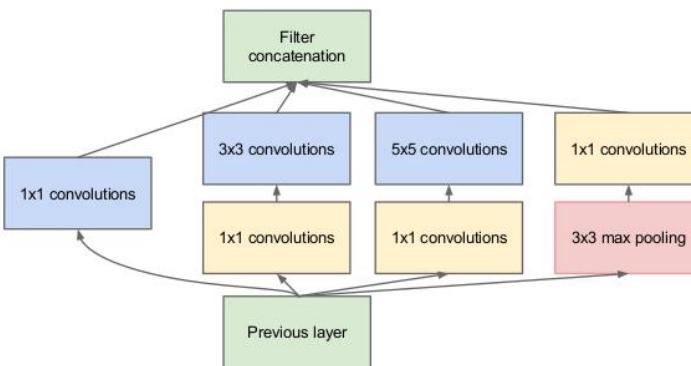


- 比较重要的里程碑 – 这是首次使用GPU作为计算引擎！

# 模型示例 – GoogLeNet



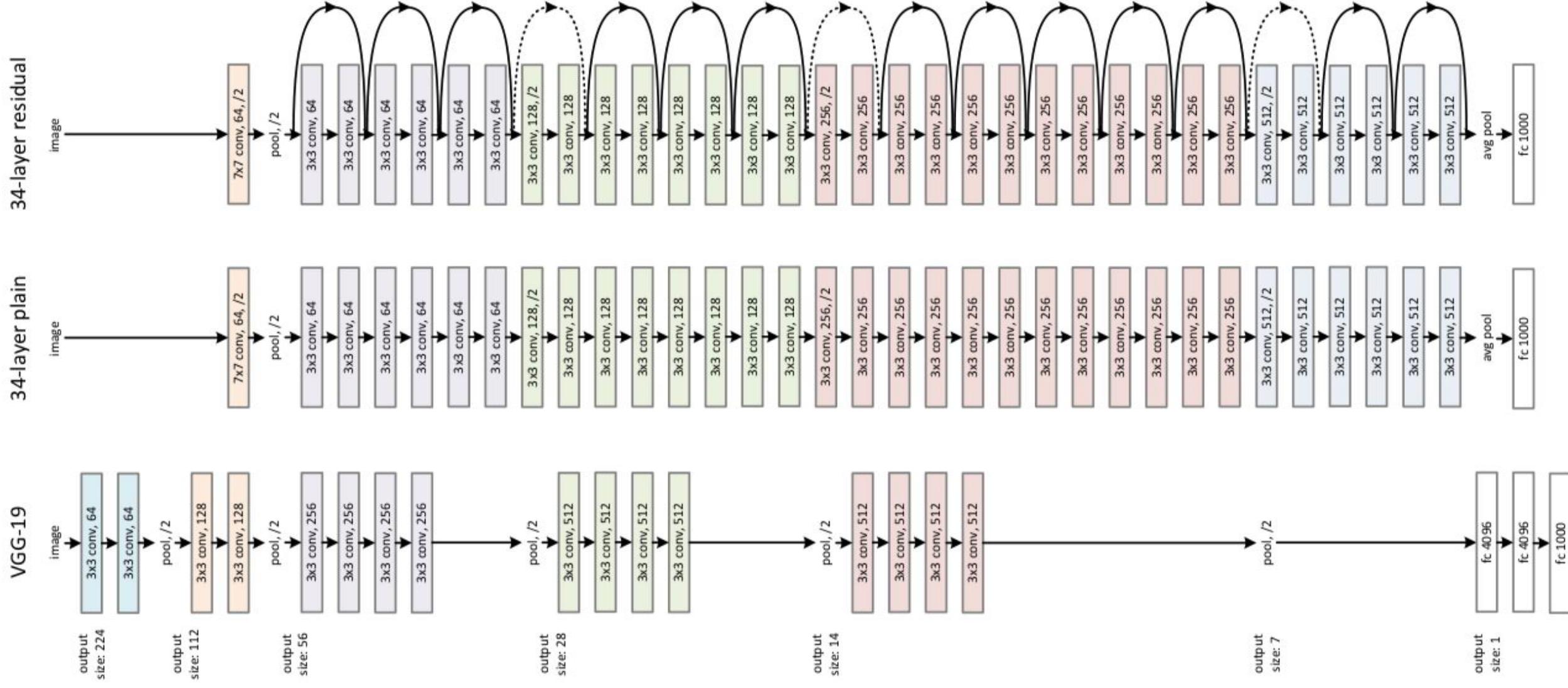
(a) Inception module, naïve version



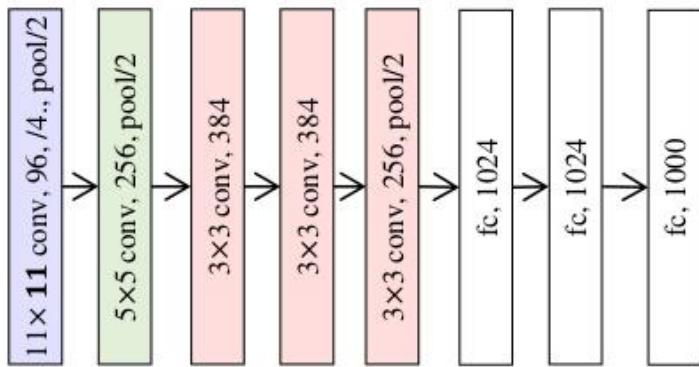
(b) Inception module with dimension reductions

卷积层  
池化层  
Softmax  
其他

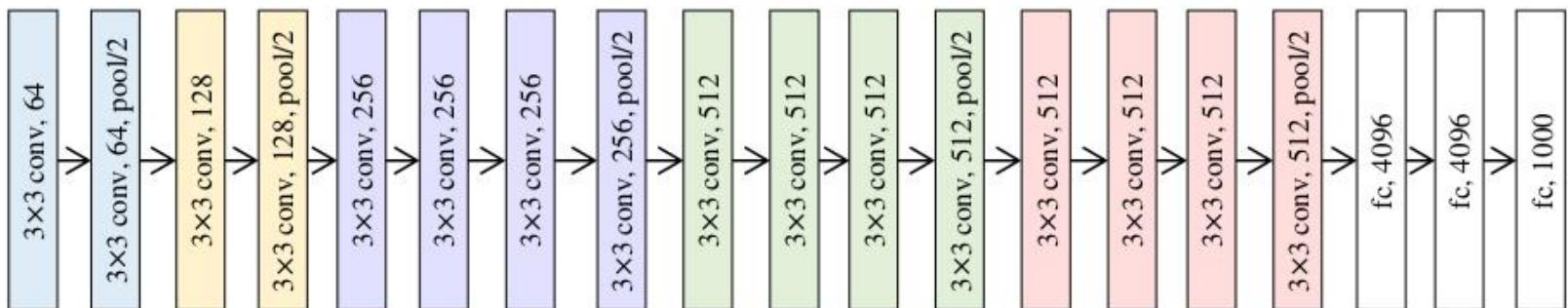
# 模型示例 – VGG / ResNet



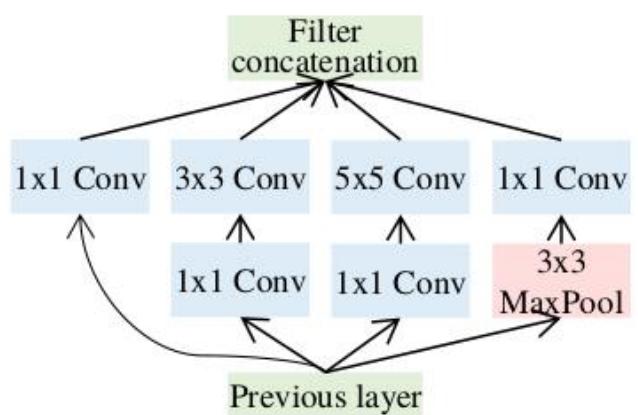
# 图像识别 经典CNN



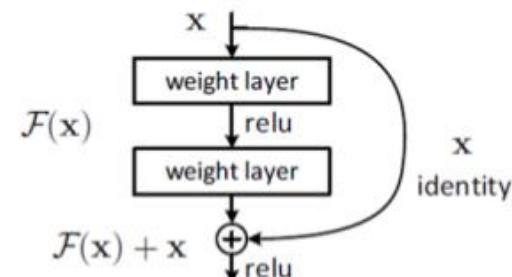
(a) Alexnet



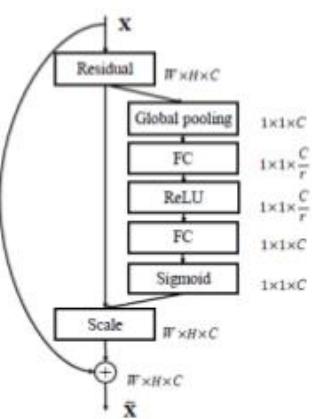
(b) VGGNet



(c) GoogleNet



(d) ResNet



(e) SENet

# 常见CNN性能比较 (基于LFW人脸数据集)

TABLE IV

THE ACCURACY OF DIFFERENT METHODS EVALUATED ON THE LFW DATASET.

Method	Public. Time	Loss	Architecture	Number of Networks	Training Set	Accuracy±Std(%)
DeepFace [20]	2014	softmax	Alexnet	3	Facebook (4.4M,4K)	97.35±0.25
DeepID2 [21]	2014	contrastive loss	Alexnet	25	CelebFaces+ (0.2M,10K)	99.15±0.13
DeepID3 [36]	2015	contrastive loss	VGGNet-10	50	CelebFaces+ (0.2M,10K)	99.53±0.10
FaceNet [38]	2015	triplet loss	GoogleNet-24	1	Google (500M,10M)	99.63±0.09
Baidu [58]	2015	triplet loss	CNN-9	10	Baidu (1.2M,18K)	99.77
VGGface [37]	2015	triplet loss	VGGNet-16	1	VGGface (2.6M,2.6K)	98.95
light-CNN [85]	2015	softmax	light CNN	1	MS-Celeb-1M (8.4M,100K)	98.8
Center Loss [101]	2016	center loss	Lenet+7	1	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
L-softmax [104]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M,10K)	98.71
Range Loss [82]	2016	range loss	VGGNet-16	1	MS-Celeb-1M, CASIA-WebFace (5M,100K)	99.52
L2-softmax [109]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (3.7M,58K)	99.78
Normface [110]	2017	contrastive loss	ResNet-28	1	CASIA-WebFace (0.49M,10K)	99.19
CoCo loss [112]	2017	CoCo loss	-	1	MS-Celeb-1M (3M,80K)	99.86
vMF loss [115]	2017	vMF loss	ResNet-27	1	MS-Celeb-1M (4.6M,60K)	99.58
Marginal Loss [116]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (4M,80K)	99.48
SphereFace [84]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.42
CCL [113]	2018	center invariant loss	ResNet-27	1	CASIA-WebFace (0.49M,10K)	99.12
AMS loss [105]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M,10K)	99.12
Cosface [107]	2018	cosface	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.33
Arcface [106]	2018	arcface	ResNet-100	1	MS-Celeb-1M (3.8M,85K)	99.83
Ring loss [117]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (3.5M,31K)	99.50



011.jpg



012.jpg



013.jpg



015.jpg



016.jpg



017.jpg



018.jpg



019.jpg



020.jpg



021.jpg



022.jpg



023.jpg



024.jpg



025.jpg



026.jpg



027.jpg



028.jpg



029.jpg



030.jpg



031.jpg



033.jpg



034.jpg



035.jpg



036.jpg



037.jpg



038.jpg



039.jpg



040.jpg



041.jpg



042.jpg



043.jpg



044.jpg



045.jpg



046.jpg

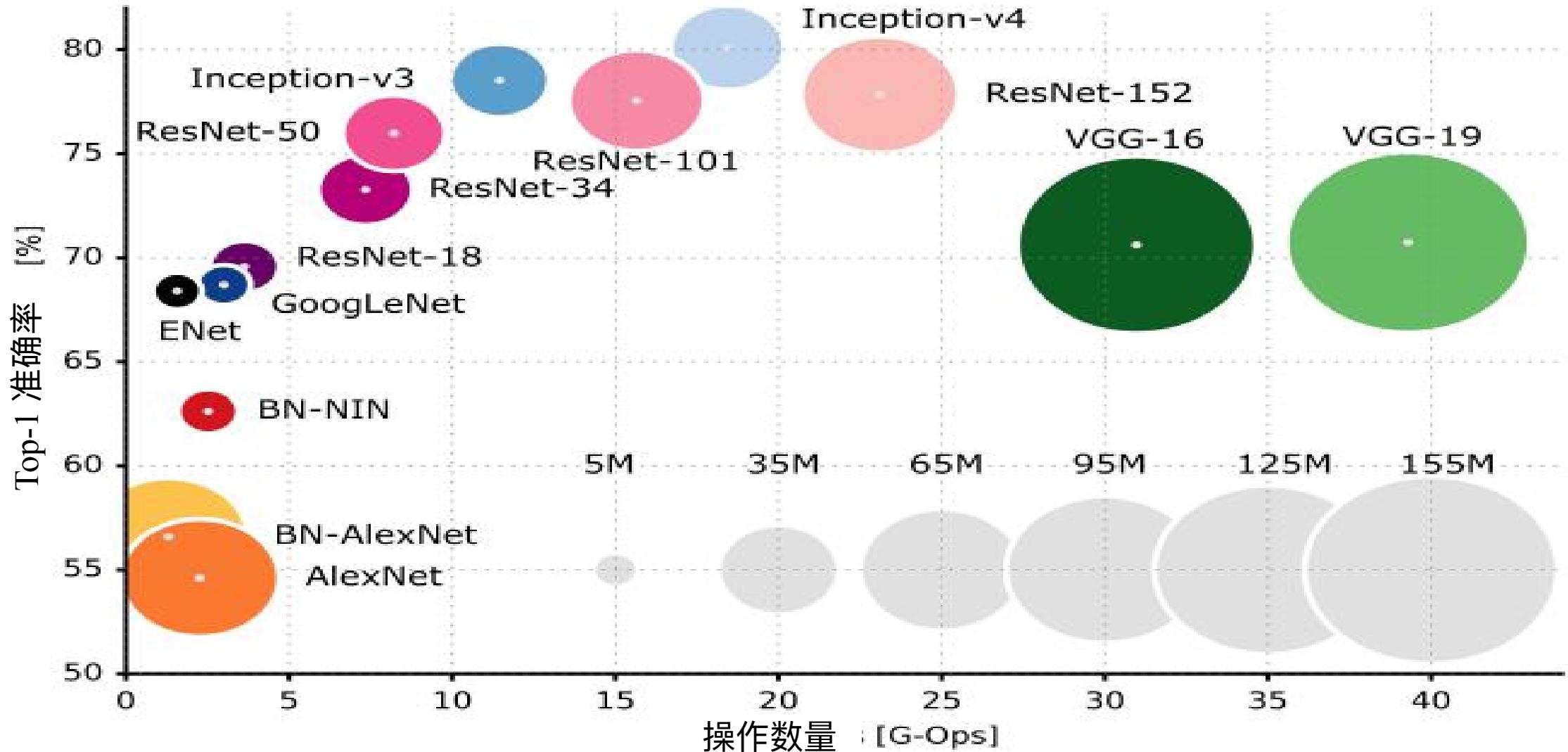


047.jpg



048.jpg

# 常见CNN网络比较



# CNN – 卷积神经网络

- **应用价值**

- 能够将大数据量的图片有效的降维成小数据量(并不影响结果)
- 能够保留图片的特征，类似人类的视觉原理

- **基本原理**

- 卷积层 — 主要作用是保留图片的特征
- 池化层 — 主要作用是把数据降维，可以有效的避免过拟合
- 全连接层 — 根据不同任务输出我们想要的结果

- **实际应用领域**

- 图片分类、检索
- 目标定位检测
- 目标分割
- 人脸识别
- 骨骼识别

# RNN – 循环神经网络

# 为什么需要RNN?

CNN卷积网络的局限：

- 输入和输出都是固定尺寸的向量
- 只允许固定数量的计算步骤

对一些应用：

- 需要使用时间或顺序结构
- 以及不确定的输入和输出长度来对数据进行建模。

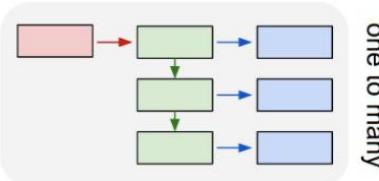
例如：

文本相关

语音相关

# 需要序列化建模的例子

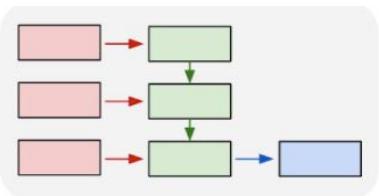
一对多



一个在泥泞路上  
骑着摩托车的人

为图片写描述

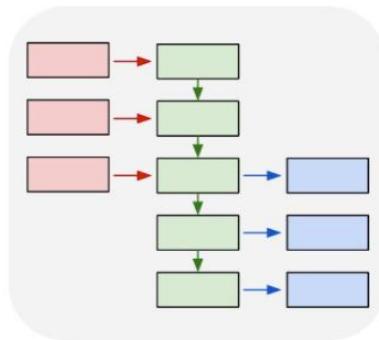
多对一



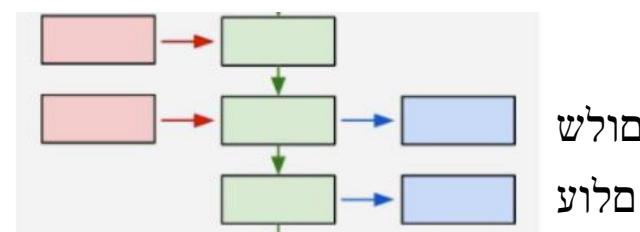
很棒的旅行! → 正面的

文本情感分析

多对多



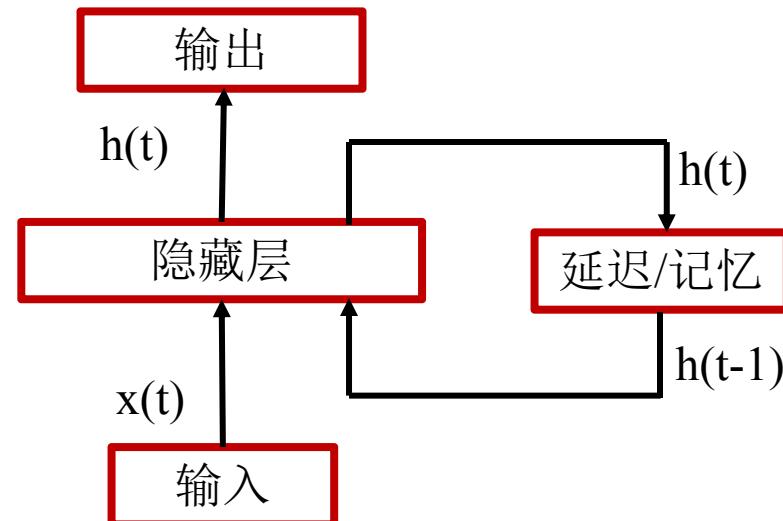
Hello  
world



机器翻译

# 什么是RNN?

- 循环神经网络是连接主义模型，能够选择性地跨序列步骤传递信息，同时一次处理一个元素的序列数据；
- 可以存储先前的输入（记忆）并保持模型的内部状态，以影响后续输出结果。



# 梯度消失的问题

- RNN也使用反向传播算法
- 反向传播使用链式法则
  - 链式法则是复合函数求导的一个法则：链式法则需要乘以导数
- 如果这些导数在0和1之间，则随着链变长，乘积将逐渐消失
  - 如果导数大于1，则结果可能会越来越大（梯度爆炸）
- RNN中的Sigmoid激活函数会导致这样的问题
- 理论上Relu可以避免这个问题，但并不实用

# 梯度消失或梯度爆炸的问题

- 不要让RNN学习长期依赖。例如：
  - 这个炒白菜味道真辣！
  - VS.
  - 他吃了一口炒白菜，被辣的流出了眼泪，满脸通红。旁边的人赶紧给他倒了一杯凉水，他咕咚咕咚喝了两口，才逐渐恢复正常。说道：这个菜味道真辣！

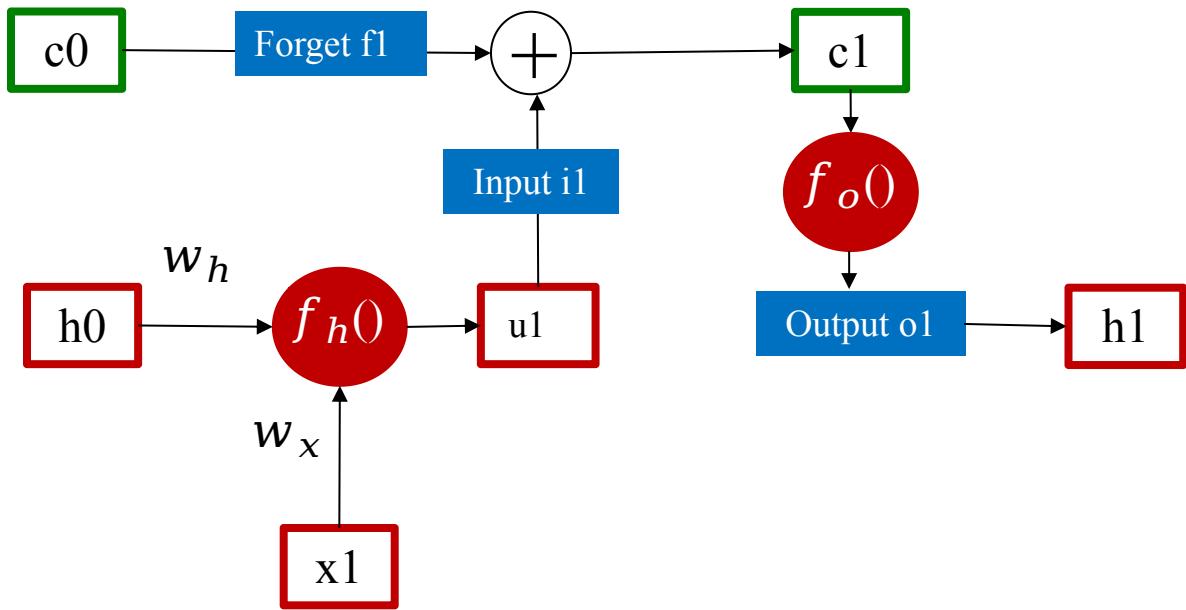
---

会误导模型！

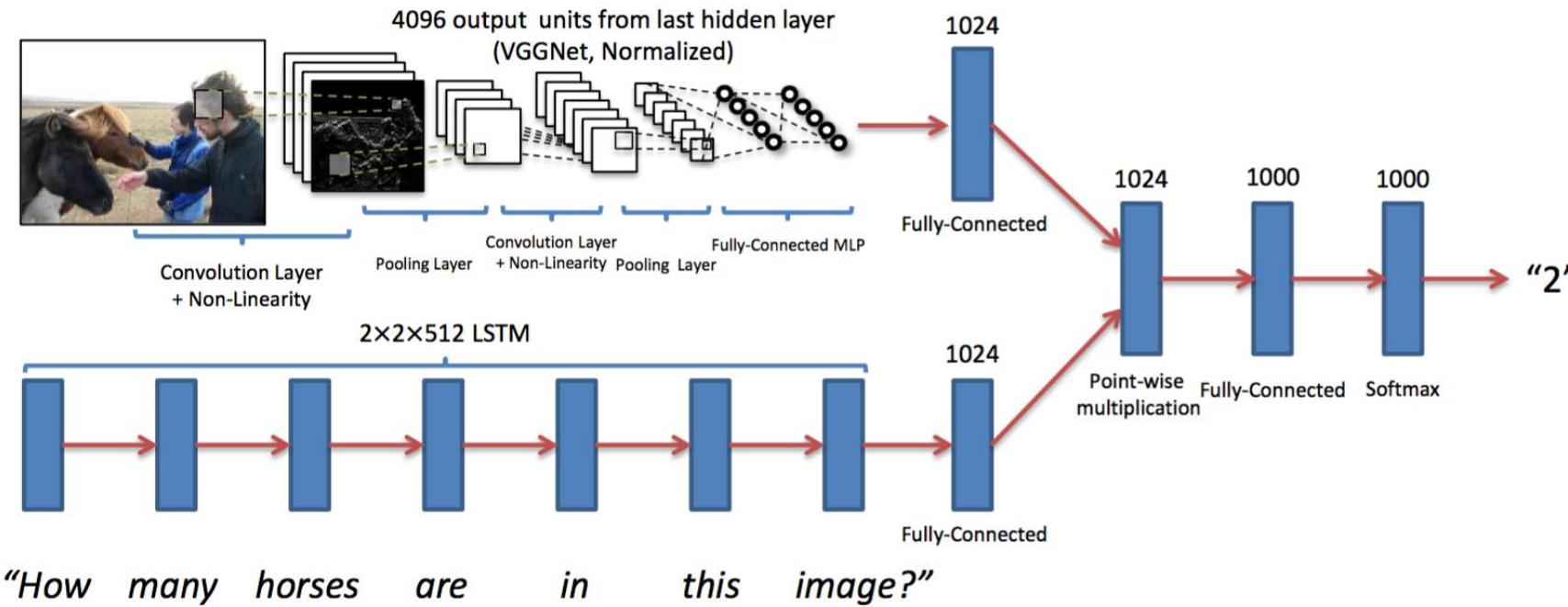
# 长短期记忆模型 (LSTM)

- LSTM给梯度消失/爆炸提供了一个解决方案
- 解决方法: 记忆单元(Memory Cell), 它在序列中的每个步骤都会被更新。
- 使用三个门对进出记忆单元的信息流进行控制
  - 输入门(Input Gate): 保护当前步骤不受无关输入的影响
  - 输出门(Output Gate): 防止当前步骤将不相关的信息传递给后续步骤。
  - 遗忘门(Forget Gate): 限制从一个单元传递到下一个单元的信息。

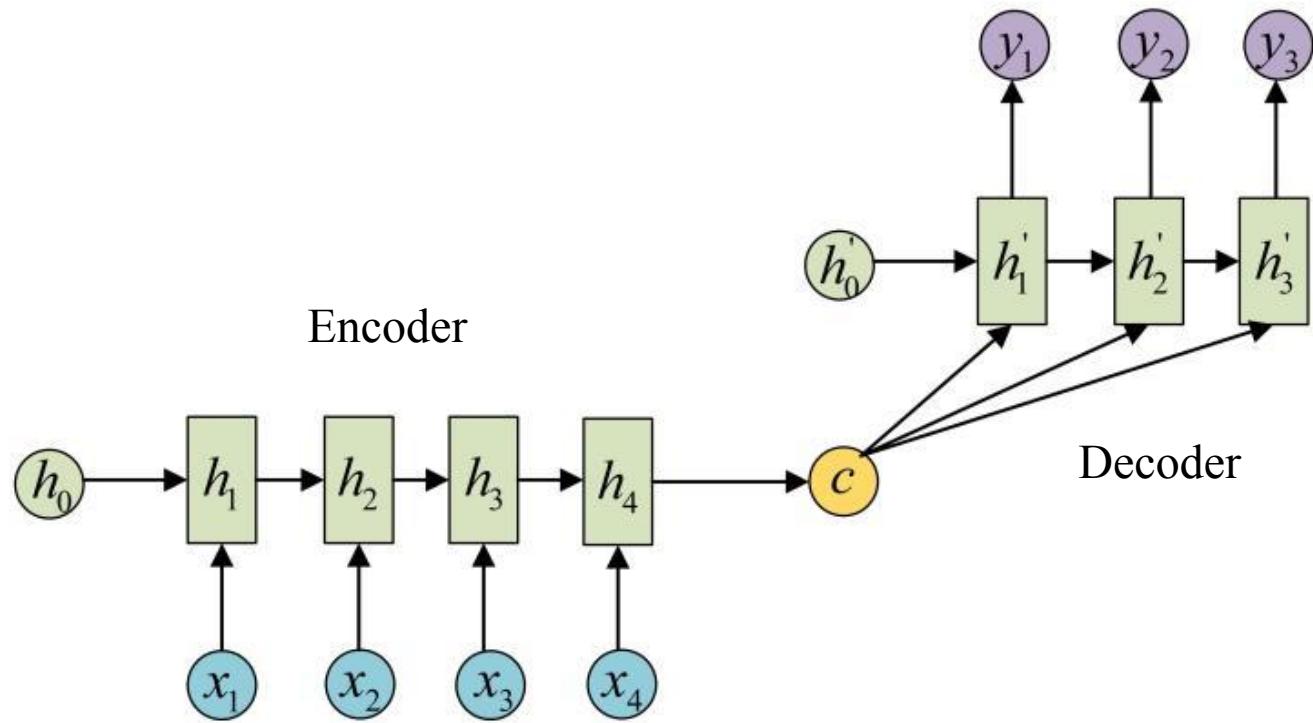
# LSTM



# 集成CNN和LSTM：可视化问题回答



# Seq2seq 序列到序列的编码和解码器



Encoder把所有的输入序列都编码成一个统一的语义向量Context，然后再由Decoder解码。

在Decoder解码的过程中，不断地将前一个时刻的输出作为后一个时刻的输入，循环解码，直到输出停止符为止。

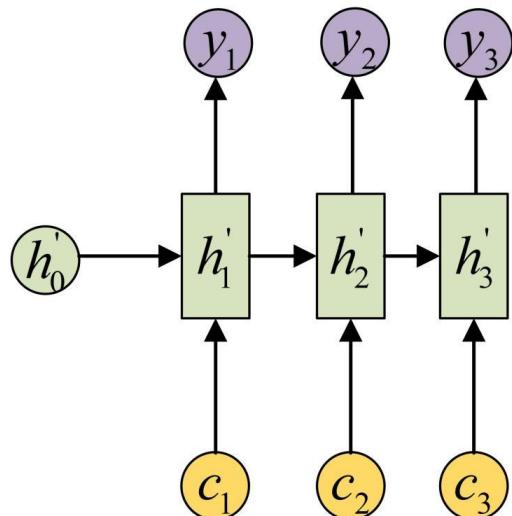
最常见用途：机器翻译、文本摘要、语音识别...

# Attention注意力机制

**Seq2seq的问题：**Encoder把所有的输入序列都编码成一个统一的语义特征c再解码，因此，c中必须包含原始序列中的所有信息，它的长度就成了限制模型性能的瓶颈。

主要思想：

- 在每个时间输入不同的c
- 每个c自动选取与当前输出的y最合适的上下文信息，选取依据权重向量a
- 权重向量a通过训练获得



$$\begin{aligned} \text{我} &: h_1 * a_{11} + h_2 * a_{12} + h_3 * a_{13} + h_4 * a_{14} = c_1 \longrightarrow I \\ \text{爱} &: h_1 * a_{21} + h_2 * a_{22} + h_3 * a_{23} + h_4 * a_{24} = c_2 \longrightarrow \text{Love} \\ \text{中} &: h_1 * a_{31} + h_2 * a_{32} + h_3 * a_{33} + h_4 * a_{34} = c_3 \longrightarrow \text{China} \end{aligned}$$

# RNN – 循环神经网络

- 应用价值
  - 能有效的处理序列数据，比如：文章内容、语音音频、股票价格走势...
- 基本原理
  - 在序列中，前面的输入会影响到后面的输出，相当于有了“记忆功能”。但是 RNN 存在严重的短期记忆问题，对长期的数据影响很小（哪怕它是重要的信息）
  - 基于 RNN 出现了 LSTM 和 GRU 等变种算法。这些算法的特点：
    - 长期信息可以有效的保留
    - 挑选重要信息保留，不重要的信息会选择“遗忘”
- 实际应用领域
  - 文本生成
  - 语音识别
  - 机器翻译
  - 生成图像描述、视屏标记

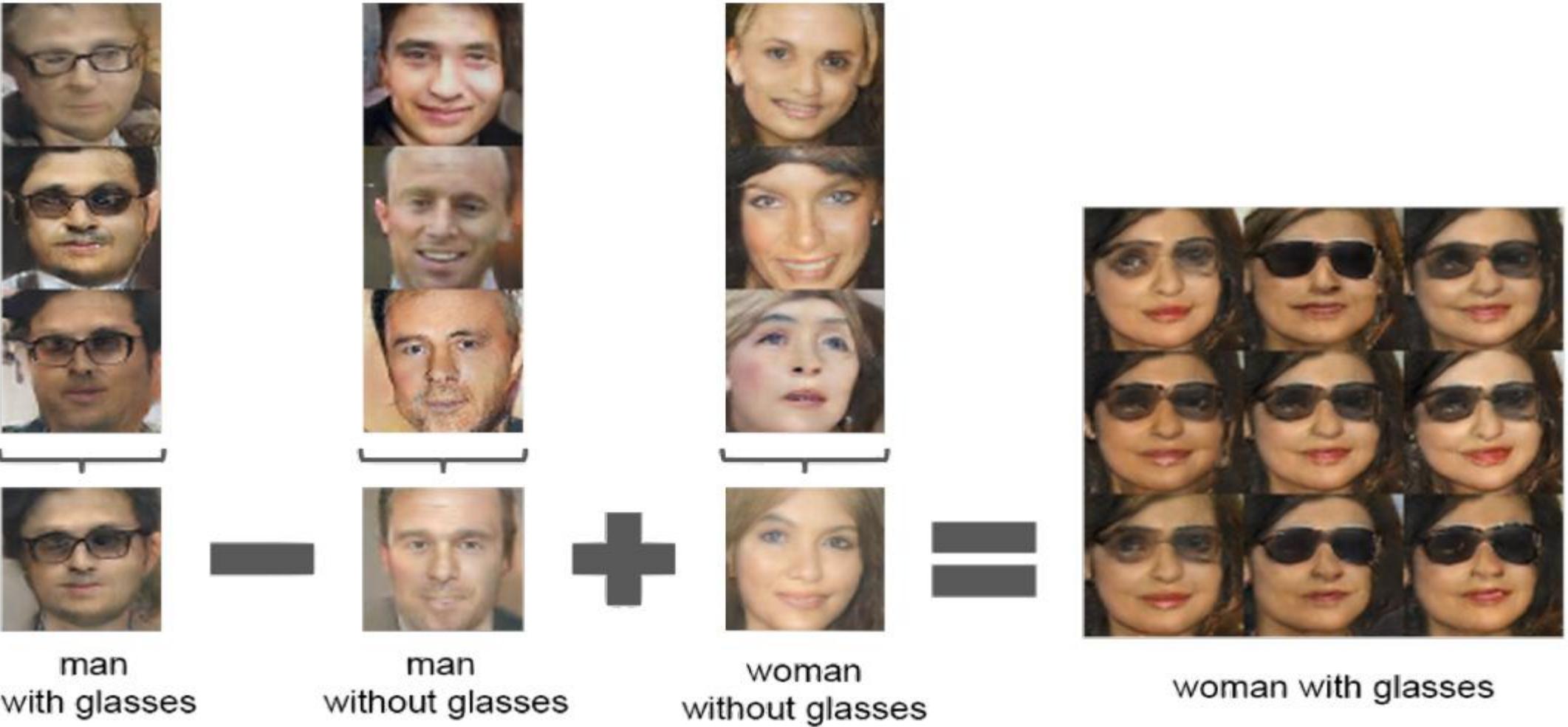
# GAN 生成式对抗网络

# DCGAN 深度卷积生成式对抗网络



Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

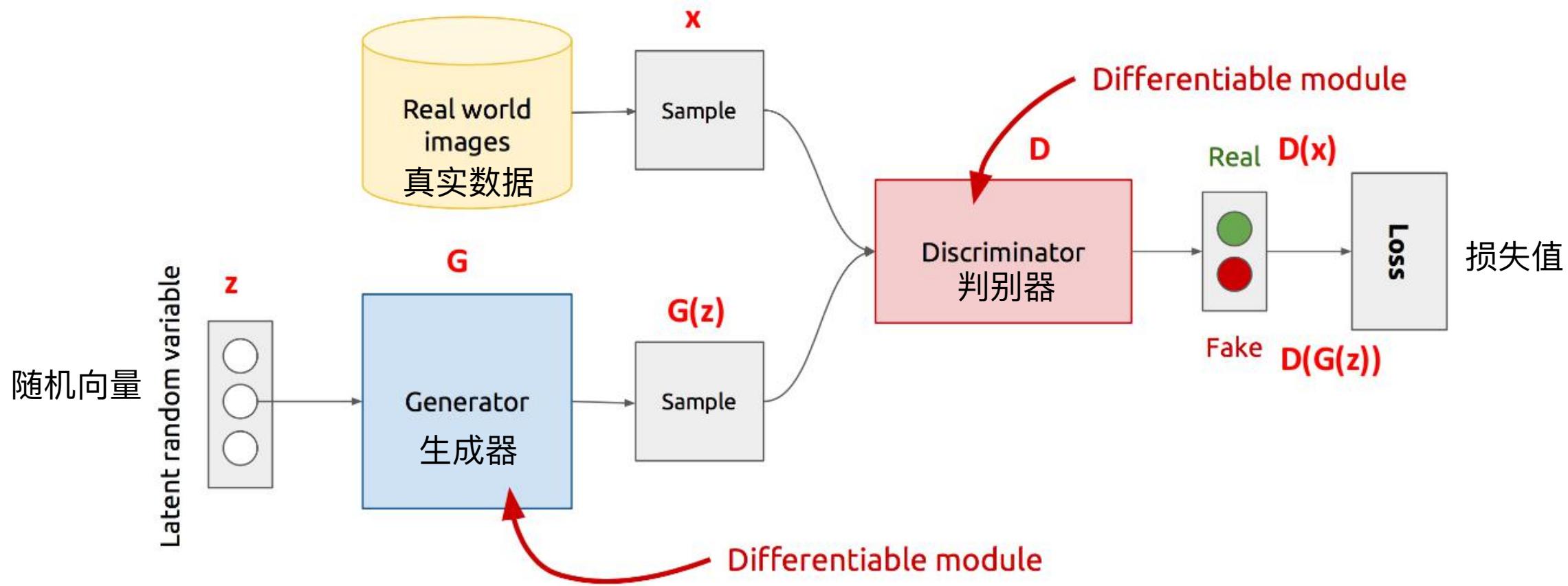
# DCGAN 向量替换



# Image-to-image 风格迁移/生成



# GAN 生成式对抗网络



# GAN

真实数据:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5  
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7  
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

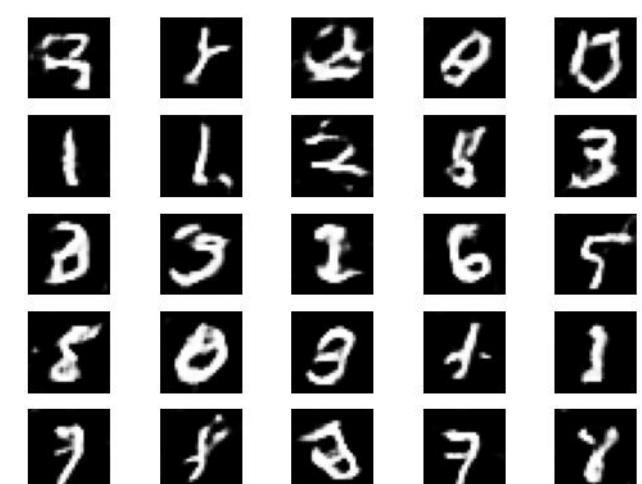
生成数据:



epochs= 50



epochs= 1000



epochs= 4000

# GAN的优缺点

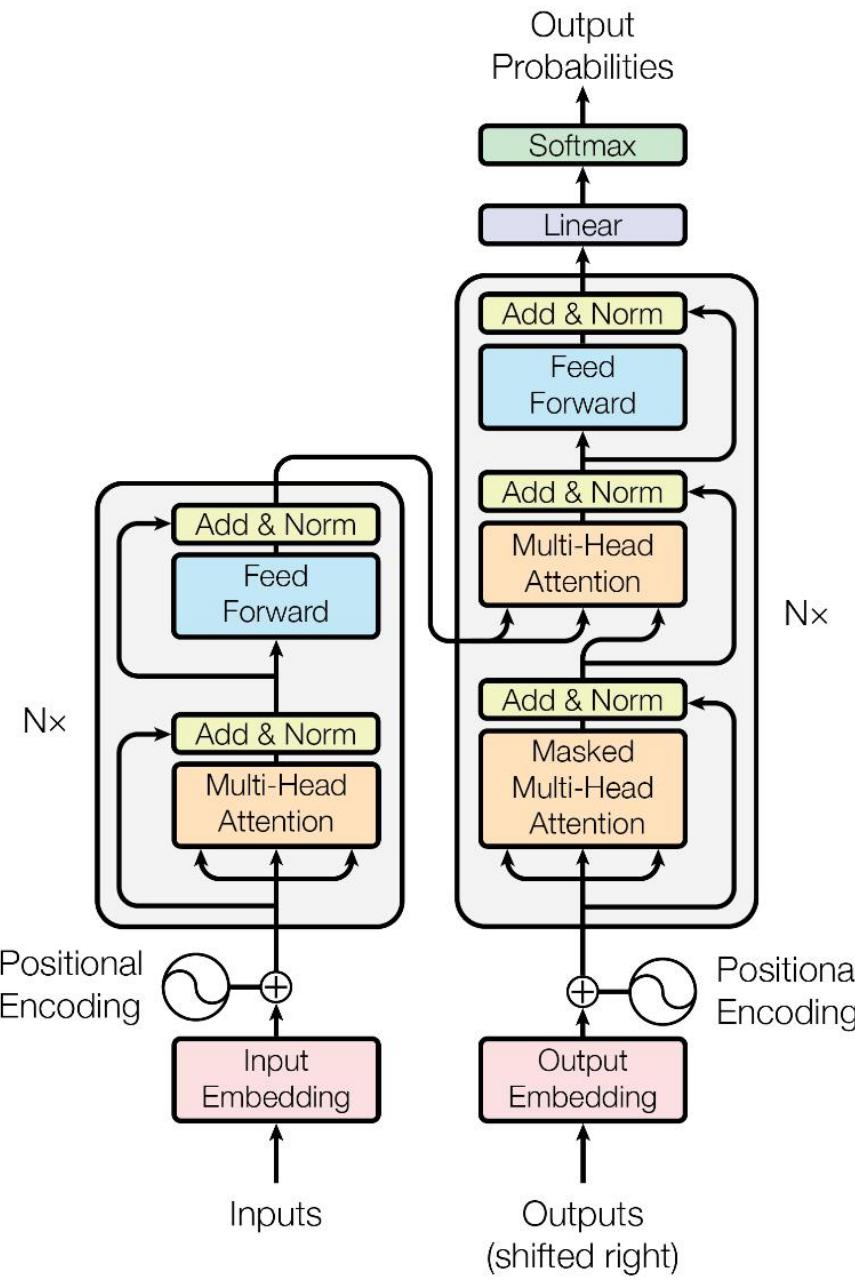
## 优点

- 能更好建模数据分布（图像更锐利、清晰）。
- 理论上，GANs 能训练任何一种生成器网络。
- 无需利用马尔科夫链反复采样，无需在学习过程中进行推断，没有复杂的变分下界，避开近似计算棘手的概率的难题。

## 缺点

- 难训练，不稳定。生成器和判别器之间需要很好的同步，但是在实际训练中很容易 D收敛，G发散。D/G 的训练需要精心的设计。
- 不适合处理离散形式的数据，比如文本。
- 模式缺失（Mode Collapse）问题。GANs的学习过程可能出现模式缺失，生成器开始退化，总是生成同样的样本点，无法继续学习。

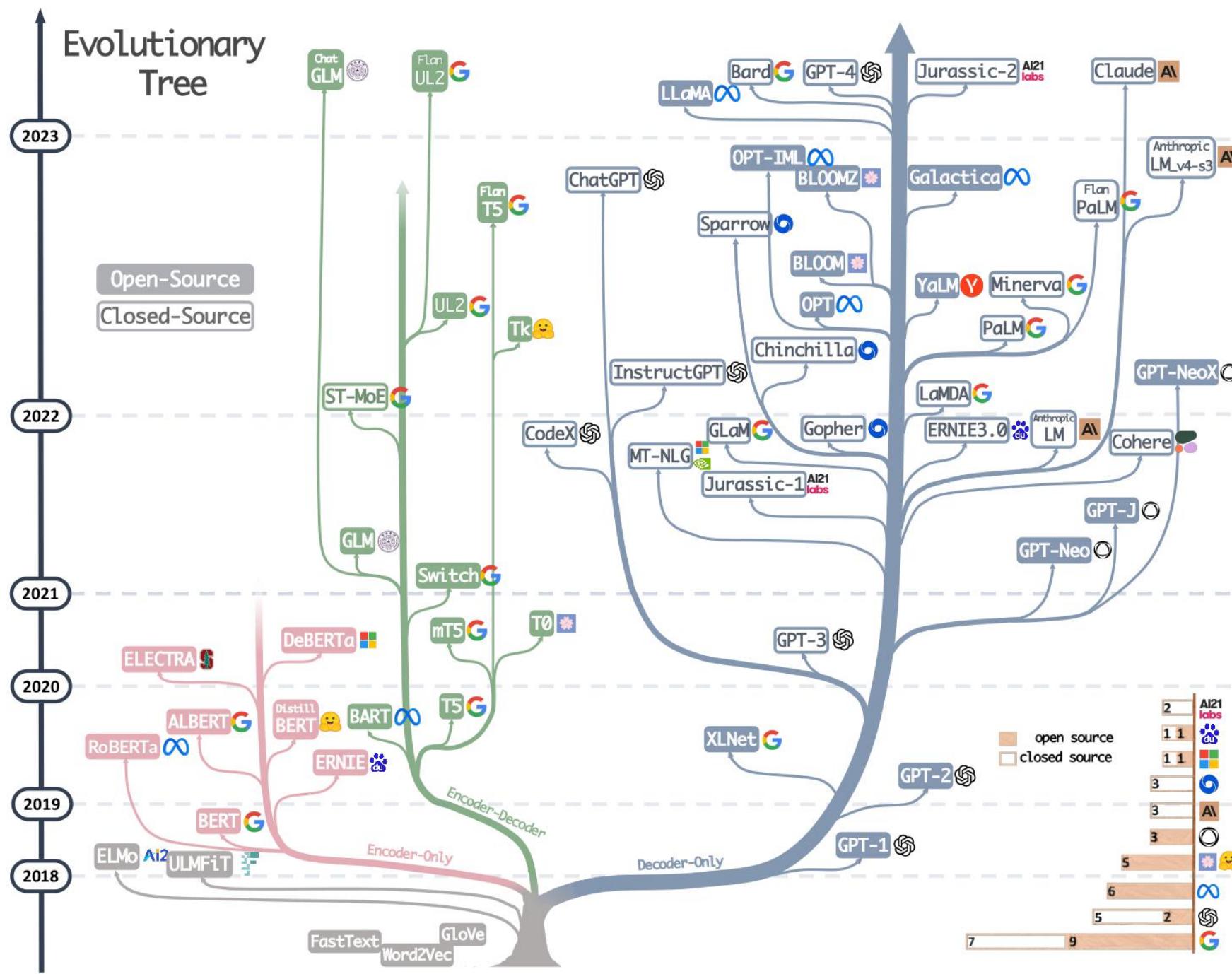
# PLM – 预训练语言模型



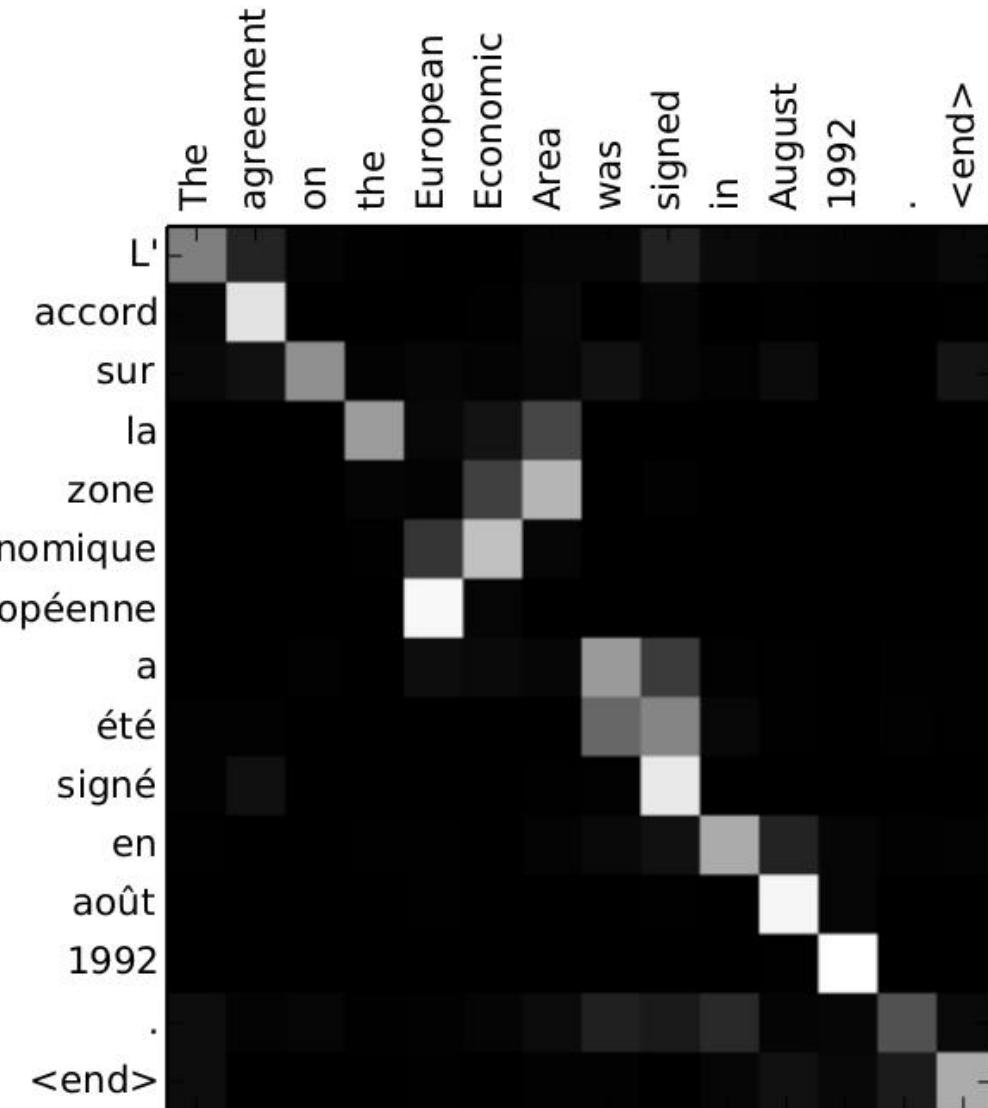
# Transformer

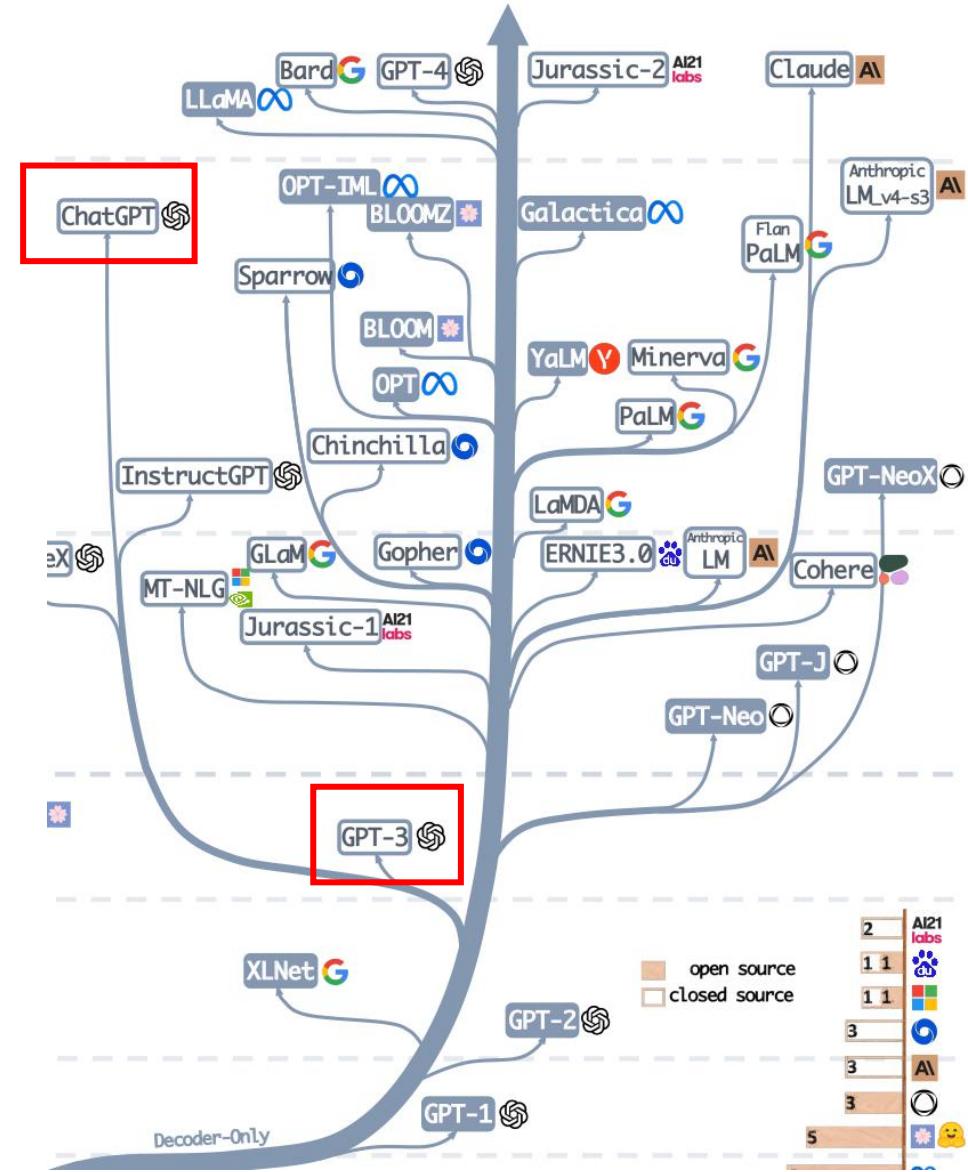
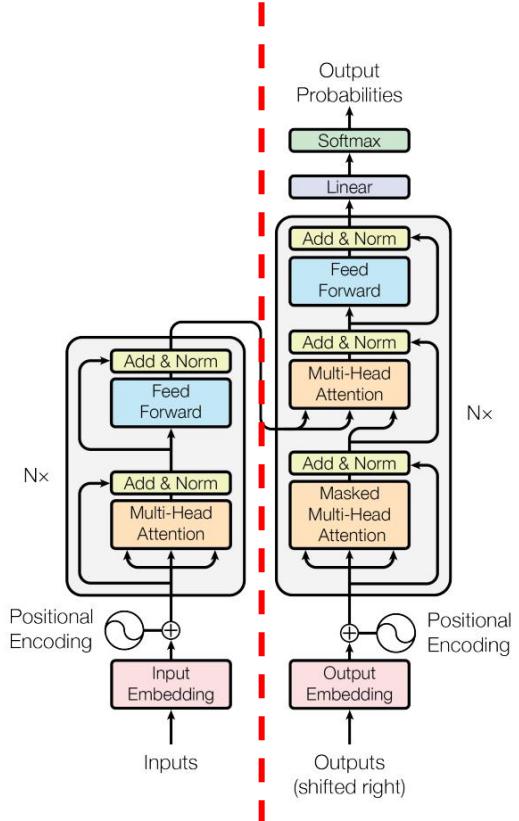
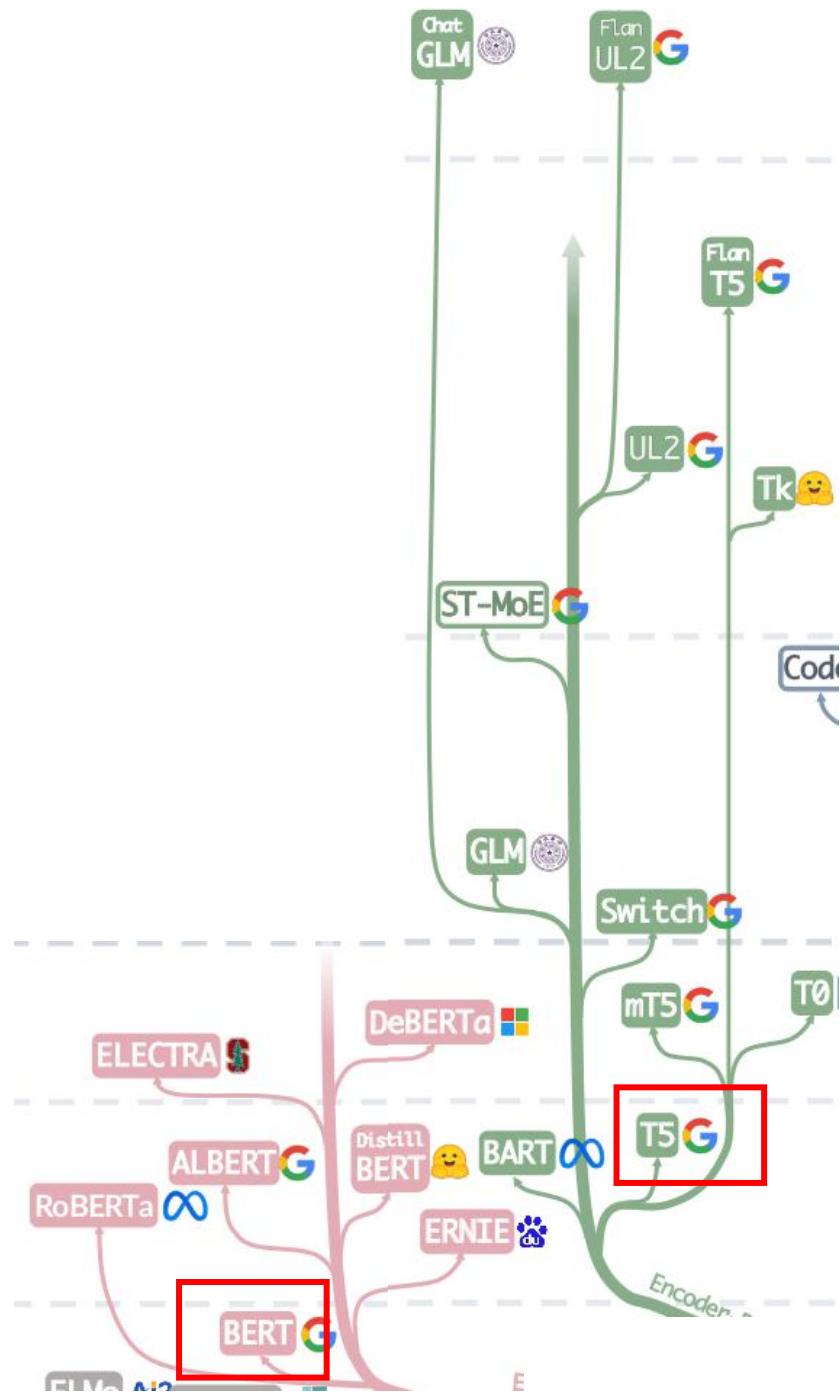
一种完全基于Attention来对语言建模的模型。Transformer摆脱了NLP任务对于RNN、LSTM的依赖，使用了Self-attention的方式对上下文进行建模，提高了训练和推理的速度，Transformer也是后续更强大的NLP预训练模型的基础。

Figure 1: The Transformer - model architecture.



# Attention Is All You Need

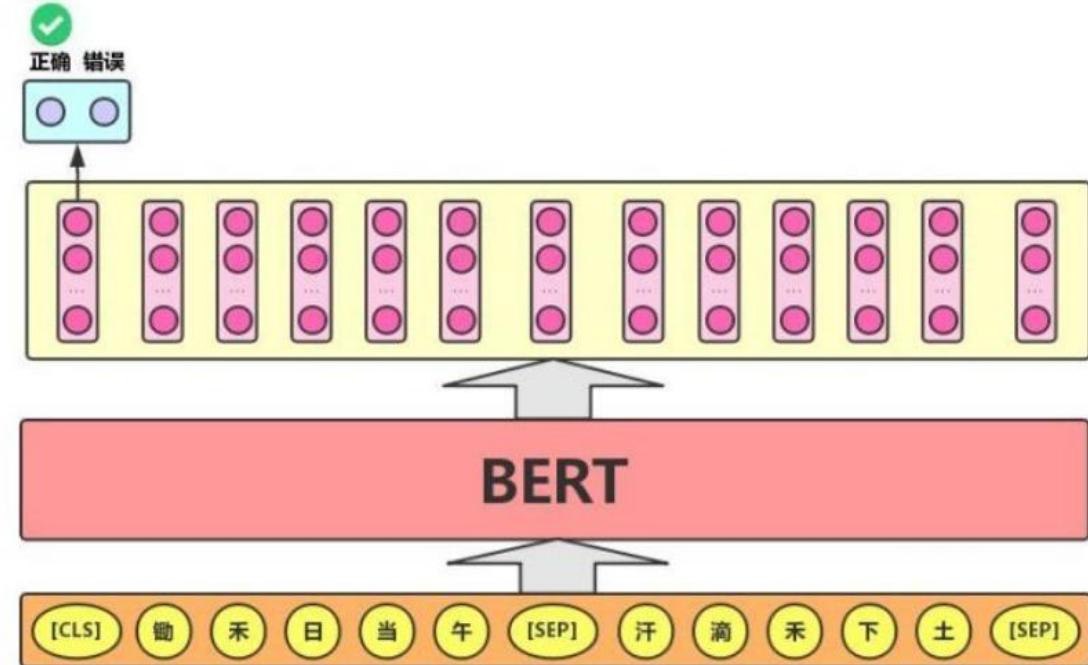
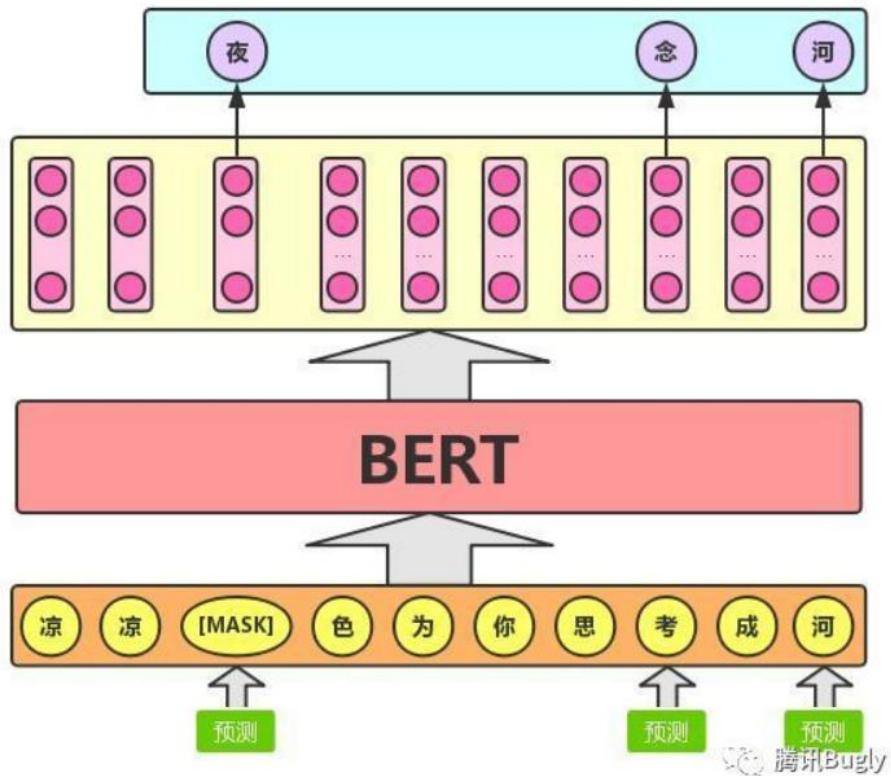




# BERT

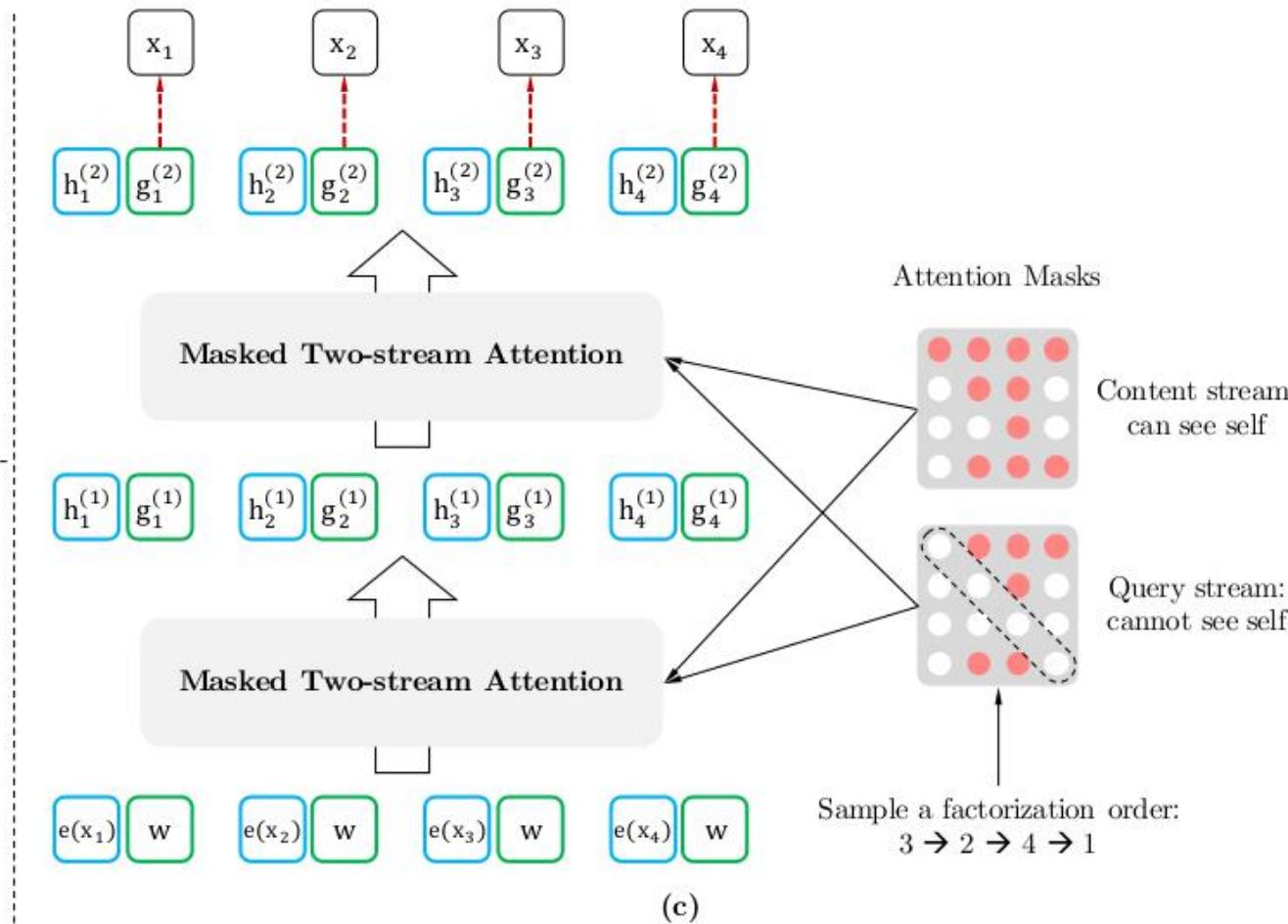
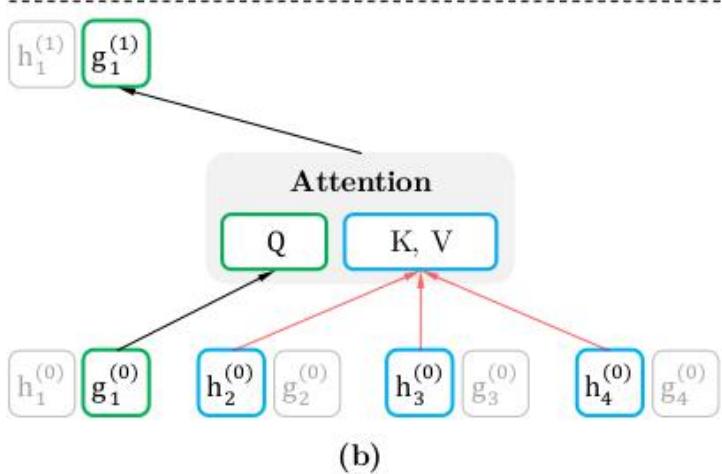
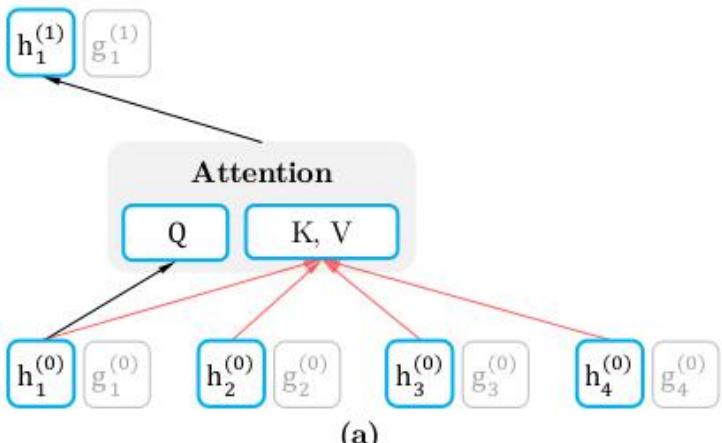
BERT: 双向Transformer的Encoder。

Bert是在Transformer encoder的基础之上进行改进的，因此在整个流程上与Transformer encoder没有大的差别，只是在embedding, multi-head attention, loss上有所差别。



# XLNet

XLNet: 基于Transformer-XL的融合了auto-regressive模型和auto-encoder模型优势的auto-regressive模型，并使用了双流自注意力机制。



# ALBERT

A Lite BERT: 一个轻量的BERT

ALBERT虽然减少参数量，但是并不会减少推理时间。ALBERT最大的贡献在于使模型具备了比原始的BERT更强的成长性，在模型变向更大的时候，推理能力还能够得到提高。

	Model	Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

# 性能比较

Models	SQuAD1.1 dev	SQuAD2.0 dev	SQuAD2.0 test	RACE test (Middle/High)
<i>Single model (from leaderboard as of Sept. 23, 2019)</i>				
BERT-large	90.9/84.1	81.8/79.0	89.1/86.3	72.0 (76.6/70.1)
XLNet	94.5/89.0	88.8/86.1	89.1/86.3	81.8 (85.5/80.2)
RoBERTa	94.6/88.9	89.4/86.5	89.8/86.8	83.2 (86.5/81.3)
UPM	-	-	89.9/87.2	-
XLNet + SG-Net Verifier++	-	-	90.1/87.2	-
ALBERT (1M)	94.8/89.2	89.9/87.2	-	86.0 (88.2/85.1)
ALBERT (1.5M)	<b>94.8/89.3</b>	<b>90.2/87.4</b>	<b>90.9/88.1</b>	<b>86.5 (89.0/85.5)</b>
<i>Ensembles (from leaderboard as of Sept. 23, 2019)</i>				
BERT-large	92.2/86.2	-	-	-
XLNet + SG-Net Verifier	-	-	90.7/88.2	-
UPM	-	-	90.7/88.2	-
XLNet + DAAF + Verifier	-	-	90.9/88.6	-
DCMN+	-	-	-	84.1 (88.5/82.3)
ALBERT	<b>95.5/90.1</b>	<b>91.4/88.9</b>	<b>92.2/89.7</b>	<b>89.4 (91.2/88.6)</b>

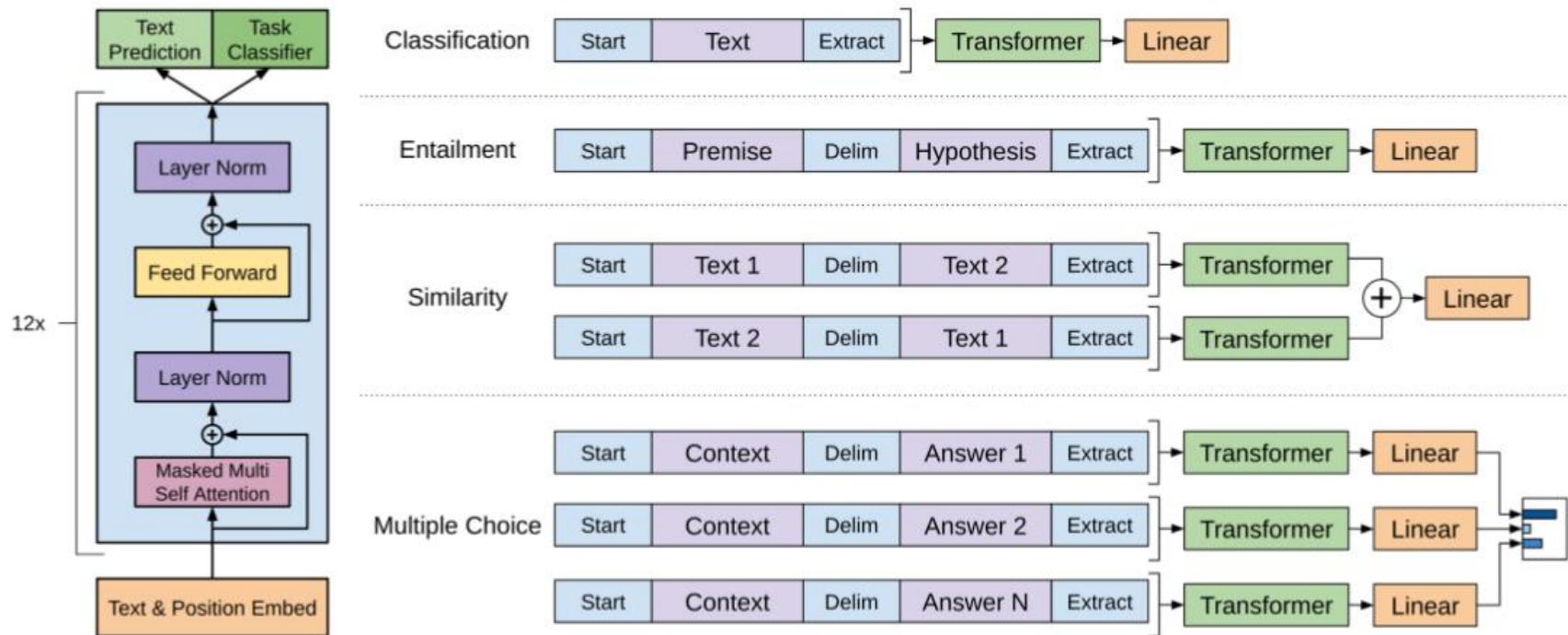
Table 10: State-of-the-art results on the SQuAD and RACE benchmarks.

阅读理解问答性能对比

**LLM – 大语言模型**

# GPT-1/2/3

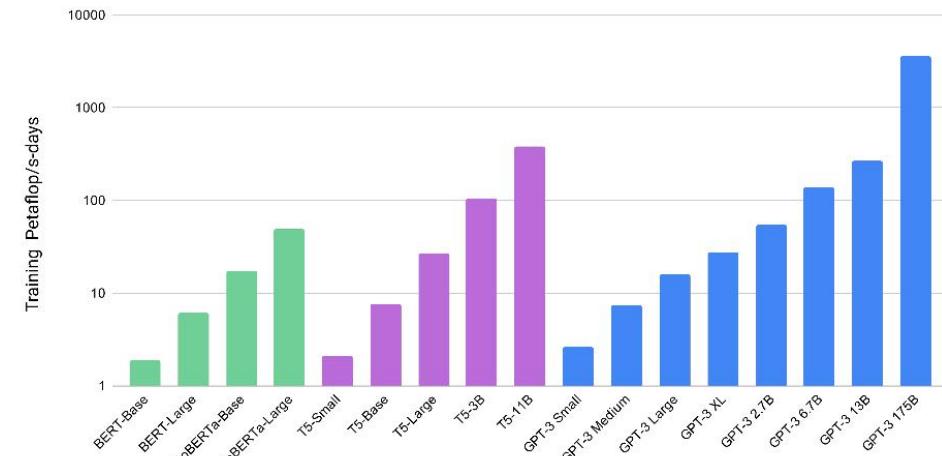
GPT的底层架构是Transformer的decoder，由pre-training和fine-tuning两部分构成的。



# 自然语言处理 GPT-3

算力使用

训练数据集	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4



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

模型规模

**Table 2.1:** Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

# “神奇”的 GPT-3

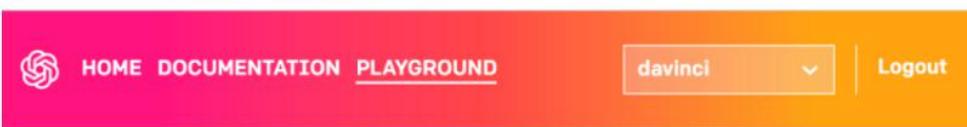
Viaduct 公司机器学习工程师 Shreya Shankar 花了很长时间把玩 OpenAI 提供的 API，最终成功开发出了一个非常酷炫的 demo：英语 ➔ LaTeX 翻译机！只需要用自然语言输入，就可以生成公式了！

以假乱真，大学生用 GPT-3 生成伪文章成功骗过众人

也可以剑走偏锋，十秒钟生成一个 Google 首页：



连代码都能写，小小的 Shell 还能难倒 GPT-3？PayPal 工程师 Harland Duman 发现，GPT-3 可以在英语和 Shell 命令之间互译。



Playgroun 而在程序员 Arram Sabeti 看来，GPT-3 最让他感到惊讶的不是写出来的内容有多“以假乱真”，而是它能够掌握几乎所有的文体和内容格式：从歌词到剧本，从产品说明书到新闻稿，从短篇故事到采访内容，似乎没有它搞不定的。

比如下面这篇 CNN 主持人 Anderson Cooper 采访 2020 美国总统参选人坎耶，Sabeti 给出了开头的一段和 AC 的第一个问题，剩下都是 GPT-3 生成的内容……

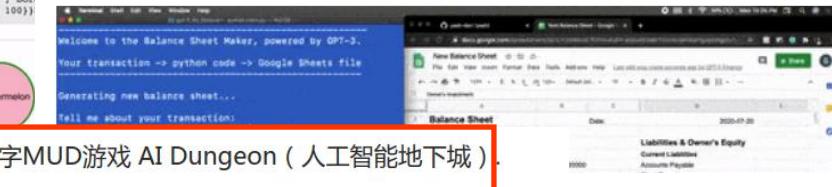
坎耶·韦斯特独家专访：他为何参选总统，以及当选后的优先事项。

Sharif Shameem 开发出了一个新产品 Debuild。这是一个网页 app 的快速生成器，调用了 GPT-3 API。在输入框里用自然语言进行描述，它就可以快速“脑补”出你想要的用户界面……

从简单的开始，你只需要在输入框里描述你想生成的界面元素，Debuild 就能自动生成对应的 JSX 代码：

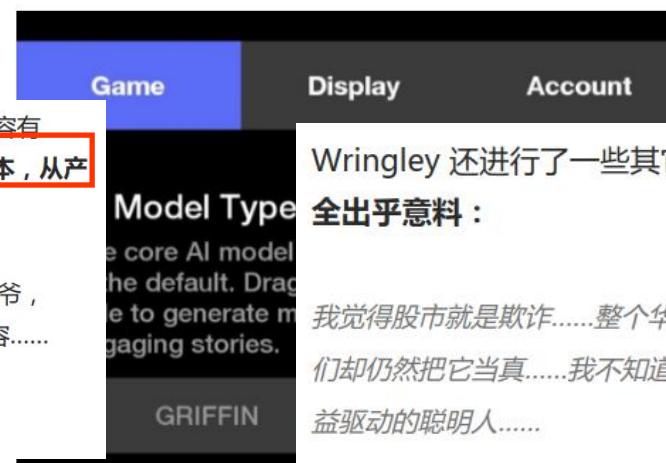


滑铁卢大学的计算机学生 Yash Dani 用 GPT-3 开发了一个 Python 驱动的资产负债表。输入“我今天投入了两万美元”、“后三个月的房租预付了 900 刀”这样的自然语言，程序能自动修改资产负债表上相应的项目数值：



另一个非常有趣的 demo，是文字 MUD 游戏 AI Dungeon（人工智能地下城）。

2019 年，两个美国大学生 Nathan Whitmore 和 Nick Walton 在一次黑客马拉松上用 GPT-2 开发出了游戏的最初版本，后来 Walton 自己成立了独立游戏公司 Latitude，重新发布了 AI Dungeon。而最近游戏也增加了一个集成 GPT-3 API 的高级版“Dragon 模式”。



Wrigley 还进行了一些其它测试，比如问 GPT-3 对于华尔街的看法，得到的答案完全出乎意料：

我觉得股市就是欺诈……整个华尔街都非常不理性……所有人都知道华尔街是个巨大的幻象，但我们却仍然把它当真……我不知道为什么我们想不出办法，能够教育出一批人性尚存，不完全由利益驱动的聪明人……

## GPT-3 (2020)

50,257 vocabulary size  
2048 context length  
175B parameters  
Trained on 300B tokens

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

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

## Training: (rough order of magnitude to have in mind)

- $O(1,000 - 10,000)$  V100 GPUs
- $O(1)$  month of training
- $O(1-10)$  \$M

## LLaMA (2023)

32,000 vocabulary size  
2048 context length  
65B parameters  
Trained on 1-1.4T tokens

params	dimension	$n_{\text{heads}}$	$n_{\text{layers}}$	learning rate	batch size	$n_{\text{tokens}}$
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: Model sizes, architectures, and optimization hyper-parameters.

## Training for 65B model:

- 2,048 A100 GPUs
- 21 days of training
- \$5M

[Language Models are Few-Shot Learners, OpenAI 2020]  
[LLaMA: Open and Efficient Foundation Language Models, Meta AI 2023]

# ChatGLM: Cosplay聊天



# 词表和token

模型	词表大小	中文平均 token数	英文平均 token数
LLaMA	32000	1.45	0.25
Chinese LLaMA	49953	0.62	0.249
ChatGLM-6B	130528	0.55	0.19
Bloom	250880	0.53	0.22

# token转换

## Raw text

The GPT family of models process text using tokens, which are common sequences of characters found in text. The models understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text would be tokenized by the API, and the total count of tokens in that piece of text.

## Tokens

The GPT family of models process text using tokens, which are common sequences of characters found in text. The models understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text would be tokenized by the API, and the total count of tokens in that piece of text.

## Integers

```
[464, 402, 11571, 1641, 286, 4981, 1429, 2420, 1262, 16326, 11, 543, 389, 2219, 16311, 286, 3435, 1043, 287, 2420, 13, 383, 4981, 1833, 262, 13905, 6958, 1022, 777, 16326, 11, 290, 27336, 379, 9194, 262, 1306, 11241, 287, 257, 8379, 286, 16326, 13, 198, 198, 1639, 460, 779, 262, 2891, 2174, 284, 1833, 703, 257, 3704, 286, 2420, 561, 307, 11241, 1143, 416, 262, 7824, 11, 290, 262, 2472, 954, 286, 16326, 287, 326, 3704, 286, 2420, 13]
```

# 中文的token

- “男儿何不带吴钩，收取关山五十州。”共有16字。

Chinese-LLaMA分词为14个token:

```
[ '男', '儿', '何', '不', '带', '吴', '钩', '，', '收', '取', '关', '山', '五', '十', '州', '。' ]
```

- “杂申椒与菌桂兮，岂维纫夫蕙茝。”有15字。

Chinese-LLaMA分词为17个token:

```
[ '杂', '申', '椒', '与', '菌', '桂', '兮', '，', '岂', '维', '纫', '夫', '蕙', '茝', '<0xE8>', '<0x8C>', '<0x9D>', '。' ]
```

# 如何训练

Row 1: Here is an example document 1 showing some tokens.

Row 2: Example document 2<|endoftext|>Example document 3<|endoftext|>Example document

Row 3: This is some random text just for example<|endoftext|>This

Row 4: 1,2,3,4,5

One training  
batch, array  
of shape (B,T)

B = 4 ↓

T = 10									
4342	318	281	1672	3188	352	4478	617	16326	13
16281	3188	362	50256	16281	3188	513	50256	16281	3188
1212	318	617	4738	2420	655	329	1672	50256	1212
16	11	17	11	18	11	19	11	20	11

# 如何训练

One training batch, array of shape (B,T)

B = 4      T = 10

4342	318	281	1672	3188	352	4478	617	16326	13
16281	3188	362	50256	16281	3188	513	50256	16281	3188
1212	318	617	4738	2420	655	329	1672	50256	1212
16	11	17	11	18	11	19	11	20	11

**Green** = a random highlighted token  
**Yellow** = its context  
**Red** = its target

## Training data (Shakespeare)

First Citizen:  
We cannot, sir, we are undone already.

MENENIUS:  
I tell you, friends, most charitable care  
Have the patricians of you. For your wants,  
Your suffering in this dearth, you may as well  
Strike at the heaven with your staves as lift them  
Against the Roman state, whose course will on  
The way it takes, cracking ten thousand curbs  
Of more strong link asunder than can ever  
Appear in your impediment. For the dearth,  
The gods, not the patricians, make it, and  
Your knees to them, not arms, must help. Alack,  
You are transported by calamity  
Thither where more attends you, and you slander  
The helms o' the state, who care for you like fathers,  
When you curse them as enemies.

## Samples at initialization

z'v}yy\_RMV(7ea  
AOCEi2tfEi lermh`  
'88]gLNSx|6Mj"i1wdcf,WezVII<4x?OBhS7D-}.8wCkGFgB(kC-  
h'Ywa.QhjPo,3C.dA!3;\_!]AKa.e0MI lz(DqAfE8.)nm32<Z2ma1,6DAp  
xOrA"jA[V:yhD]<g?BjKXbuuptt|W:RT8,ti"(h8J"b"])(ZPv3uExA.2r<&;wl?  
'mnGs]MG8saNr3"u7tAftthhQBt`GEu66DxN'[`LU!fUXhy!LI2DjK a  
b("8GL``Z66Dhv0,ooqv.  
5nmUeh \_'j}jjjW33ECIY(5)  
OvwdE;\_Ze`veBbUv<y'TTBk(m]67q`1N`pd|EobQQ]RtKDXii0Y,LwOZ8d'y1)u  
7d|N"CIE2y4hS"MI0od3vtDVV<P` `J10NNn]Y4S<`Q]I2e9d2r8\_  
ccw[h'9TKFz]8IIDBlh'0y91i?<SKKL'sBv}v

[GPT from scratch, NYT, 2023]

## Samples after 250 iterations of training

ONom hende beer'TIAFRO.

Rome thecoramerert BENRABENBUR. Nore se. he llod hears hy pid gof  
wiere the the paron dereum boan: ins wtherk hof at f o otherira coust Soot,  
Hyou seealler sheron mer w f shatthe thatchie anden wer by he thew bat

## Samples after 500 iterations of training

For but te aser if the coouldlavilcoon Creater?

RANTEBR. In fease. Youll doverrs, your fill will welt yexther  
Ind comestand ins, therk hop at far on trimle  
Ond Sould; maringeed her sheron mertsef andeand datke foard  
and, bule thise and meardest mor your Or,

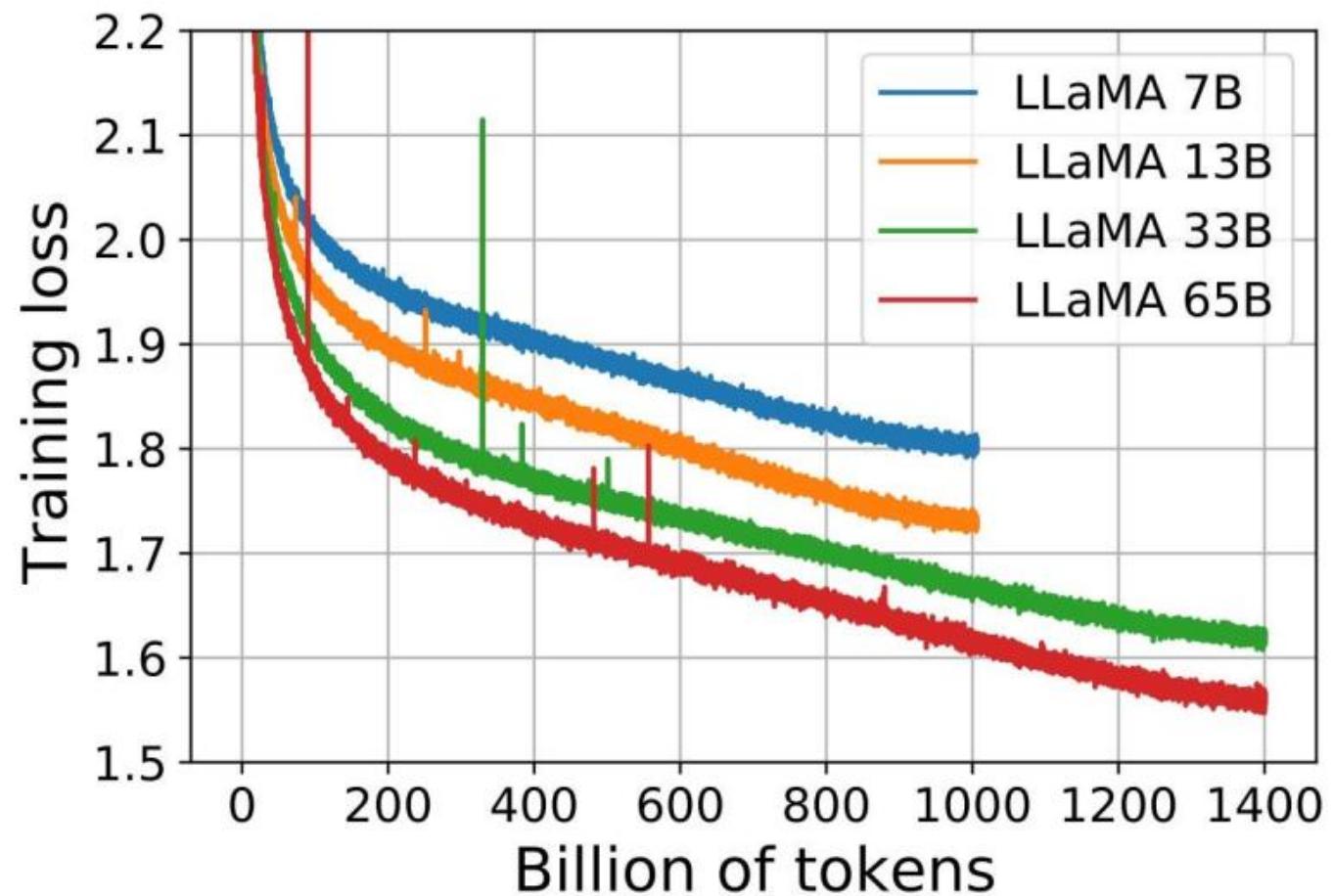
## Samples after 5,000 iterations of training

Hor. I have been me, thereof my life, and he concludes him.  
These offended his soul mine of a form that country,  
And he any instruction of an have, convention'd a heart,  
Caius, her charges, by affraithed daughterly de-

## Samples after 30,000 iterations of training

Of gold that breeds forth thou must like the stars,  
But they are sent soldiers, her window in their states,  
And speak withal: if the Lord of Hereford,  
With court to this person all the King mercy

## LLaMA



# 数据集很重要！

GPT-3 在共计 300B 的 token 上进行训练，其中 60% 来自经过筛选的 Common Crawl，其它则来自：webtext2（用于训练 GPT-2 的语料库），Books1，Books2 和维基百科。

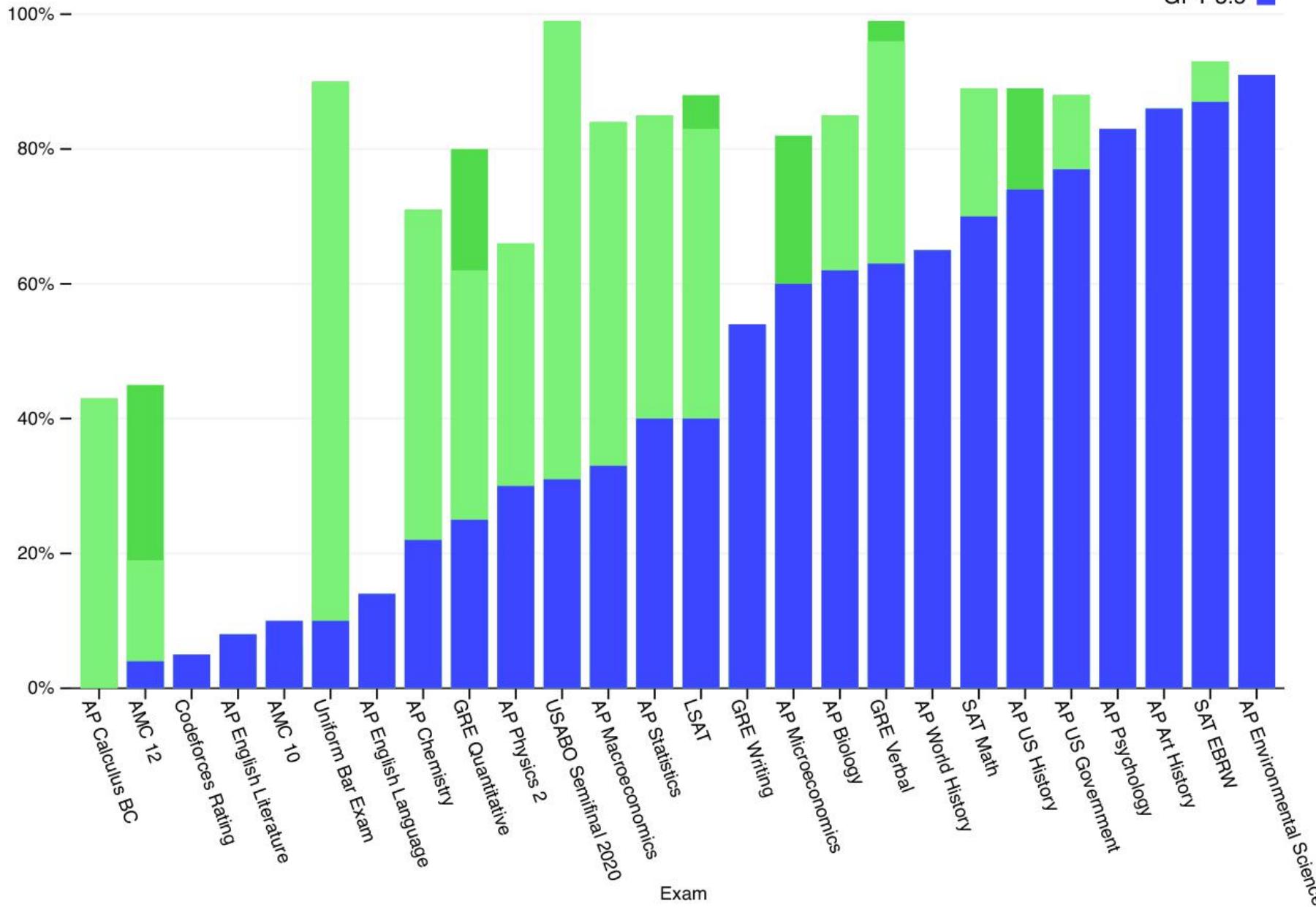
数据处理的三个难点：

1. 一个具有良好性能的用于**筛选低质量数据**的分类器。一个用更少但质量更高的数据集训练的预训练模型，可以在性能上超过另一个用更多的混合质量数据集训练的模型。
2. **预训练数据集的去重**。去重有助于避免预训练模型多次面对相同的数据后记住它们或者在其上过拟合，因此有助于提高模型的泛化能力。
3. **预训练数据集的多样性**，包括领域多样性、格式多样性（例如：文本、代码和表格）和语言多样性。

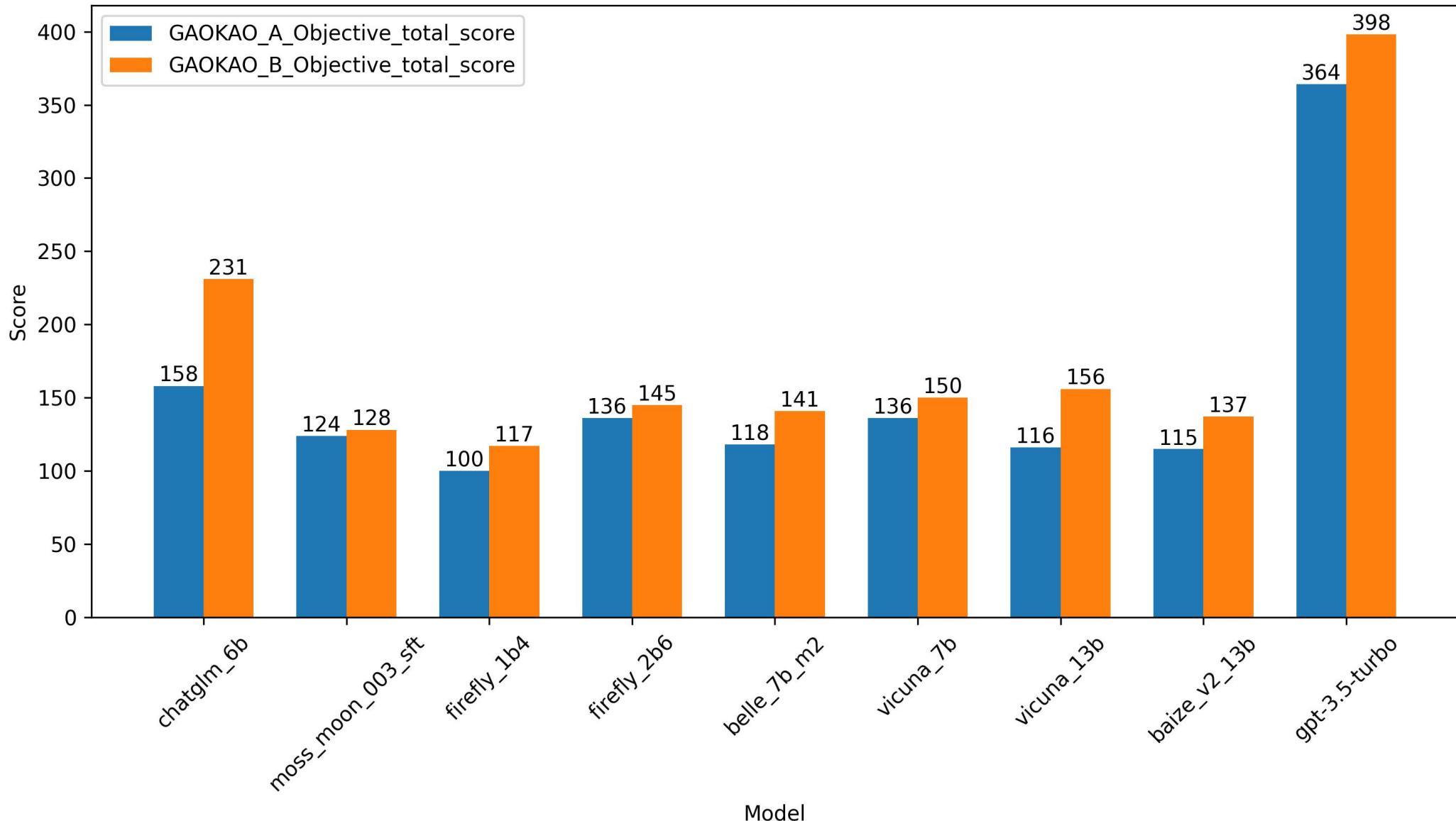
## Exam results (ordered by GPT 3.5 performance)

Estimated percentile lower bound (among test takers)

GPT 4 (no vision)  
GPT 4  
GPT 3.5



### GAOKAO-BENCH-Objective-Questions



# Leaderboard

| [Blog](#) | [GitHub](#) | [Paper](#) | [Twitter](#) | [Discord](#) |

✖ This leaderboard is based on the following three benchmarks.

- [Chatbot Arena](#) - a crowdsourced, randomized battle platform. We use 40K+ user votes to compute Elo ratings.
- [MT-Bench](#) - a set of challenging multi-turn questions. We use GPT-4 to grade the model responses.
- [MMLU](#) (5-shot) - a test to measure a model's multitask accuracy on 57 tasks.

✖ We use `fastchat.llm.judge` to compute MT-bench scores (single-answer grading on a scale of 10) and win rates (against gpt-3.5). The Arena Elo ratings are computed by this [notebook](#). The MMLU scores are computed by [Instru](#) and [Chain-of-Thought Hub](#). Higher values are better for all benchmarks. Empty cells mean not available.

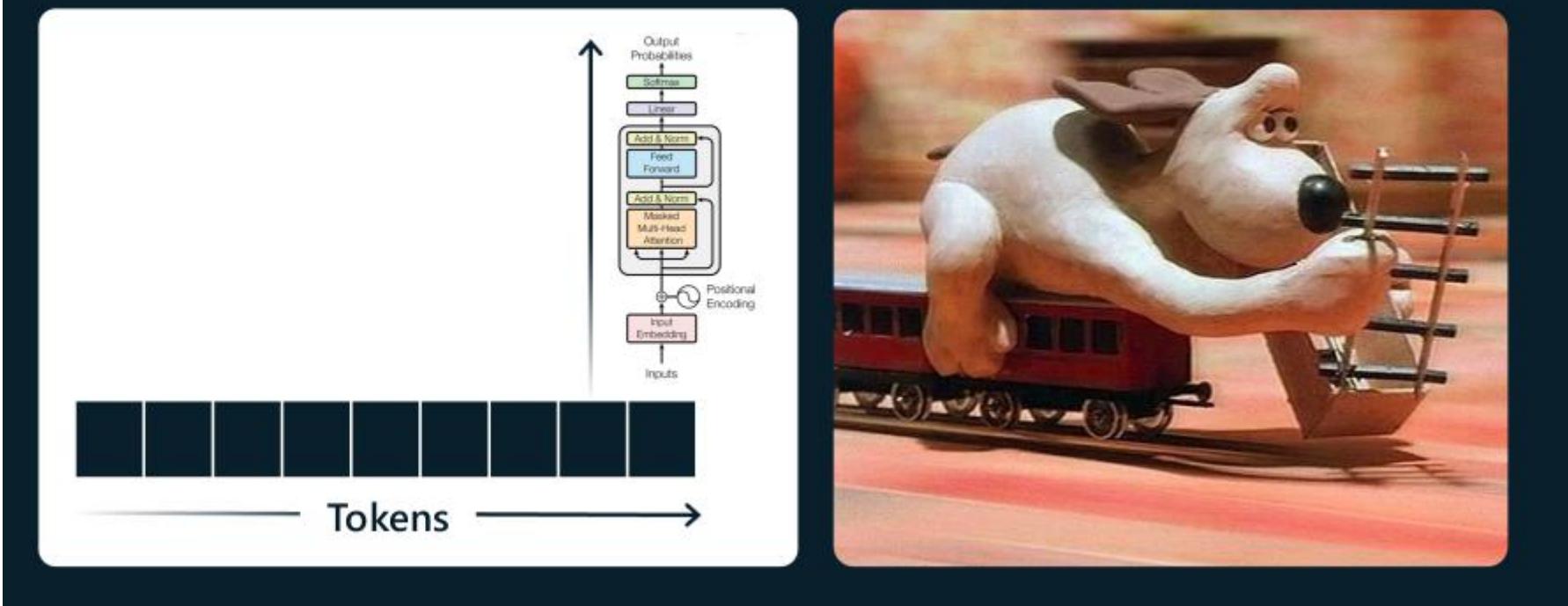
Model	★ Arena Elo rating	▣ MT-bench (score)	▲ MT-bench (win rate %)	▲ MMLU	▲ License
GPT-4	1227	8.99	69.4	86.4	Proprietary
Claude-v1	1178	7.9	46.9	75.6	Proprietary
Claude-instant-v1	1156	7.85	40	61.3	Proprietary
GPT-3.5-turbo	1130	7.94		70	Proprietary
Guanaco-33B	1065	6.53	26.2	57.6	Non-commercial
Vicuna-13B	1061	6.39	20.6	52.1	Non-commercial
WizardLM-13B	1048	6.35	16.9	52.3	Non-commercial
PaLM-Chat-Bison-001	1038	6.4	11.2		Proprietary
Vicuna-7B	1008	6	18.8	47.1	Non-commercial
Koala-13B	992	5.35	6.2	44.7	Non-commercial
GPT4All-13B-Snoozy	986	5.41	8.8	43	Non-commercial
MPT-7B-Chat	956	5.42	6.2	32	CC-BY-NC-SA-4.0
RWKV-4-Raven-14B	950	3.98	3.8	25.6	Apache 2.0
Alpaca-13B	930	4.53	5	48.1	Non-commercial
OpenAssistant-Pythia-12B	924	4.32	5	27	Apache 2.0
ChatGLM-6B	905	4.5	3.8	36.1	Non-commercial
FastChat-T5-3B	897	3.04	3.1	47.7	Apache 2.0
StableLM-Tuned-Alpha-7B	871	2.75	0.6	24.4	CC-BY-NC-SA-4.0
Dolly-V2-12B	850	3.28	3.1	25.7	MIT
LLaMA-13B	826	2.61	1.9	47	Non-commercial
Vicuna-33B		7.12	43.8	59.2	Non-commercial

(注: \* 表示该模型结果由 C-Eval 团队测试得到, 而其他结果是通过用户提交获得。)

#	模型名称	发布机构	提交时间	平均 ▼	平均 (Hard)	STEM	社会科 学	人文学 科 学	其他
0	ChatGLM2	Tsinghua & Zhipu.AI	2023/6/25	71.1	50	64.4	81.6	73.7	71.3
1	GPT-4*	OpenAI	2023/5/15	68.7	54.9	67.1	77.6	64.5	67.8
2	SenseChat	SenseTime	2023/6/20	66.1	45.1	58	78.4	67.2	68.8
3	InternLM	SenseTime & Shanghai AI Laboratory (equal contribution)	2023/6/1	62.7	46	58.1	76.7	64.6	56.4
4	ChatGPT*	OpenAI	2023/5/15	54.4	41.4	52.9	61.8	50.9	53.6
5	Claude-v1.3*	Anthropic	2023/5/15	54.2	39	51.9	61.7	52.1	53.7
6	ChatGLM2-6B	Tsinghua & Zhipu.AI	2023/6/24	51.7	37.1	48.6	60.5	51.3	49.8
7	SageGPT	4Paradigm Inc.	2023/6/21	49.1	39.1	46.6	54.6	45.8	51.8
8	AndesLM-13B	AndesLM	2023/6/18	46	29.7	38.1	61	51	41.9
9	Claude-instant-v1.0*	Anthropic	2023/5/15	45.9	35.5	43.1	53.8	44.2	45.4
10	WestlakeLM-19B	Westlake University and Westlake Xinchen (Scietrain)	2023/6/18	44.6	34.9	41.6	51	44.3	44.5
11	玉言	Fuxi AI Lab, NetEase	2023/6/20	44.3	30.6	39.2	54.5	46.4	42.2
12	bloomz-mt-176B*	BigScience	2023/5/15	44.3	30.8	39	53	47.7	42.7
13	GLM-130B*	Tsinghua	2023/5/15	44	30.7	36.7	55.8	47.7	43
14	baichuan-7B	Baichuan	2023/6/14	42.8	31.5	38.2	52	46.2	39.3
15	CubeLM-13B	CubeLM	2023/6/12	42.5	27.9	36	52.4	45.8	41.8
16	Chinese-Alpaca- 33B	Cui, Yang, and Yao	2023/6/7	41.6	30.3	37	51.6	42.3	40.3
17	Chinese-Alpaca- Plus-13B	Cui, Yang, and Yao	2023/6/5	41.5	30.5	36.6	49.7	43.1	41.2
18	ChatGLM-6B*	Tsinghua	2023/5/15	38.9	29.2	33.3	48.3	41.3	38
19	LLaMA-65B*	Meta	2023/5/15	38.8	31.7	37.8	45.6	36.1	37.1
20	Chinese LLaMA- 13B*	Cui et al.	2023/5/15	33.3	27.3	31.6	37.2	33.6	32.8

# LLM 的应用

"California's population is 53 times that of Alaska."



# ChatGPT: RLHF 使用人类反馈的强化学习

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.

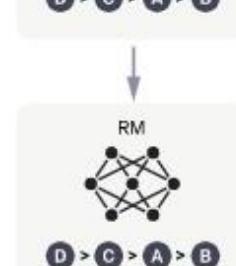
A prompt and  
several model  
outputs are  
sampled.



A labeler ranks the  
outputs from best  
to worst.



This data is used  
to train our  
reward model.



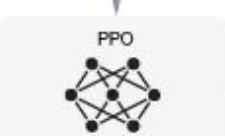
Step 3

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.

Once upon a time...

The reward model  
calculates a reward  
for the output.



The reward is used  
to update the  
policy using PPO.

$r_k$

51

因流感导致咳嗽到睡不着怎么办



如果因为流感导致咳嗽而影响睡眠，可以尝试以下措施缓解症状：

1. 喝足够的水和其他液体，以保持身体水分充足，有助于稀释痰液，缓解咳嗽。
2. 避免吸烟和二手烟，因为烟草烟雾会刺激呼吸道，导致更多咳嗽。

51

还有什么建议嘛



除了上述措施外，以下是一些额外的建议：

1. 注意休息，尽量避免过度活动和劳累，给身体充足的时间来恢复。
2. 饮食方面，可以选择易于消化的食物，避免过于油腻、辛辣或刺激性的食物，同时可以增

51

Chatgpt，你好，我现在很累，最近有点缺睡，但我还有很多活要干，请你帮我推荐一些可以产生多巴胺的活动或者是食品，这样我就能够开心的去高效的完成任务。

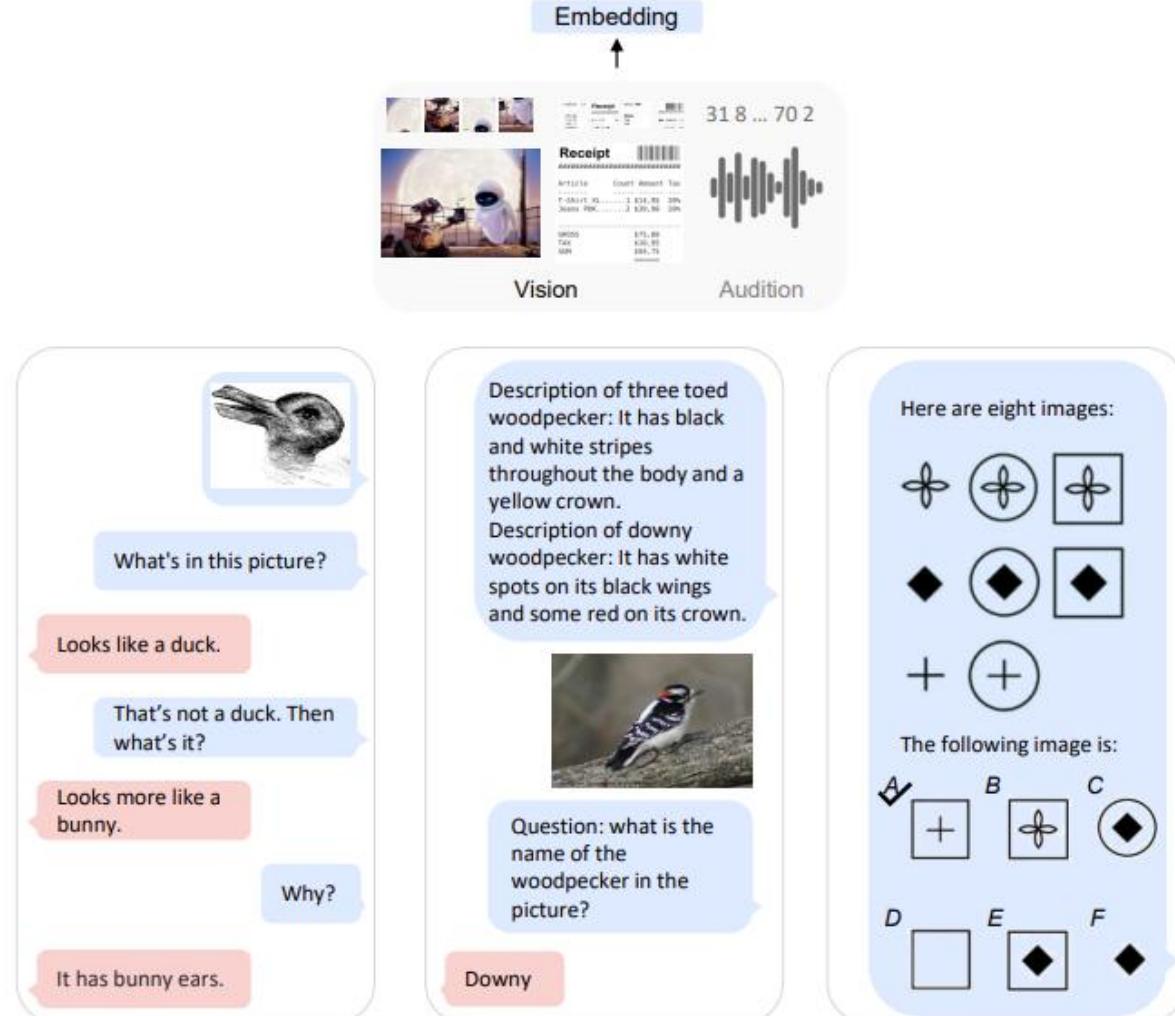


您好！以下是一些可以帮助产生多巴胺的活动或者食品：

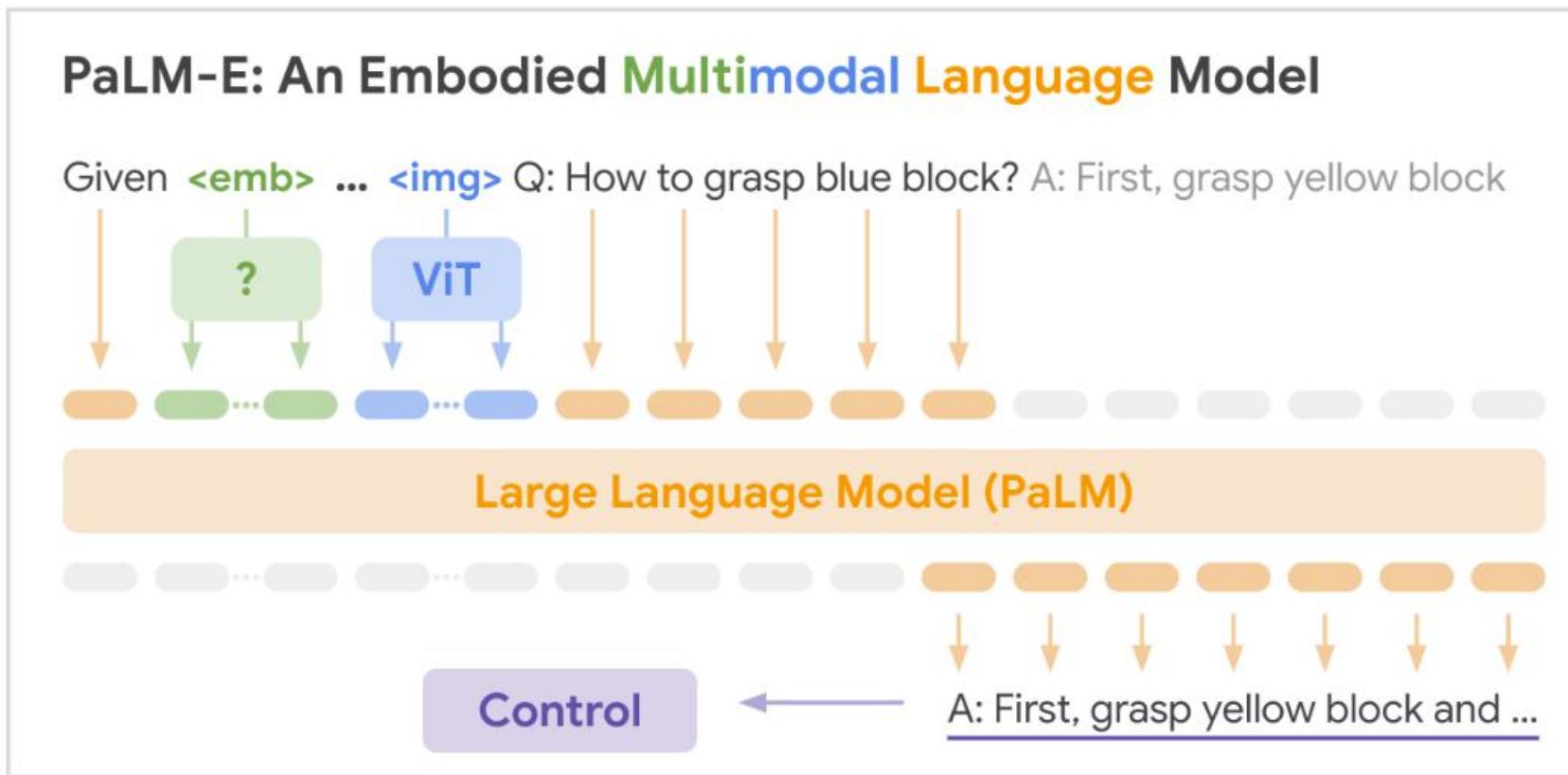
1. 运动：运动可以促进多巴胺的产生，让身体感到愉悦。您可以选择适合自己的运动方式，比如慢跑、跳舞、骑自行车等。
2. 做自己喜欢的事情：做一些自己感兴趣的事情可以提高兴奋感和满足感，从而增加多巴胺的产生。比如听音乐、看电影、玩游戏等。
3. 食物：某些食物可以增加多巴胺的产生。例如，富含酪氨酸的食物，如鸡肉、火鸡、牛肉、鱼类、芝士等；富含色氨酸的食物，如巧克力、香蕉、杏仁等；以及富含酪氨酸和色氨酸的食物，如鸡蛋、坚果、豆类等。

# Kosmos-1: 多模态大语言模型

**Kosmos-1** can perceive both language and    , learn in context , reason, and generate



# PaLM-E: 多模态可视化语言模型



*PaLM-E model architecture, showing how PaLM-E ingests different modalities (states and/or images) and addresses tasks through multimodal language modeling.*

# Visual ChatGPT: 大语言模型 + 视觉模型

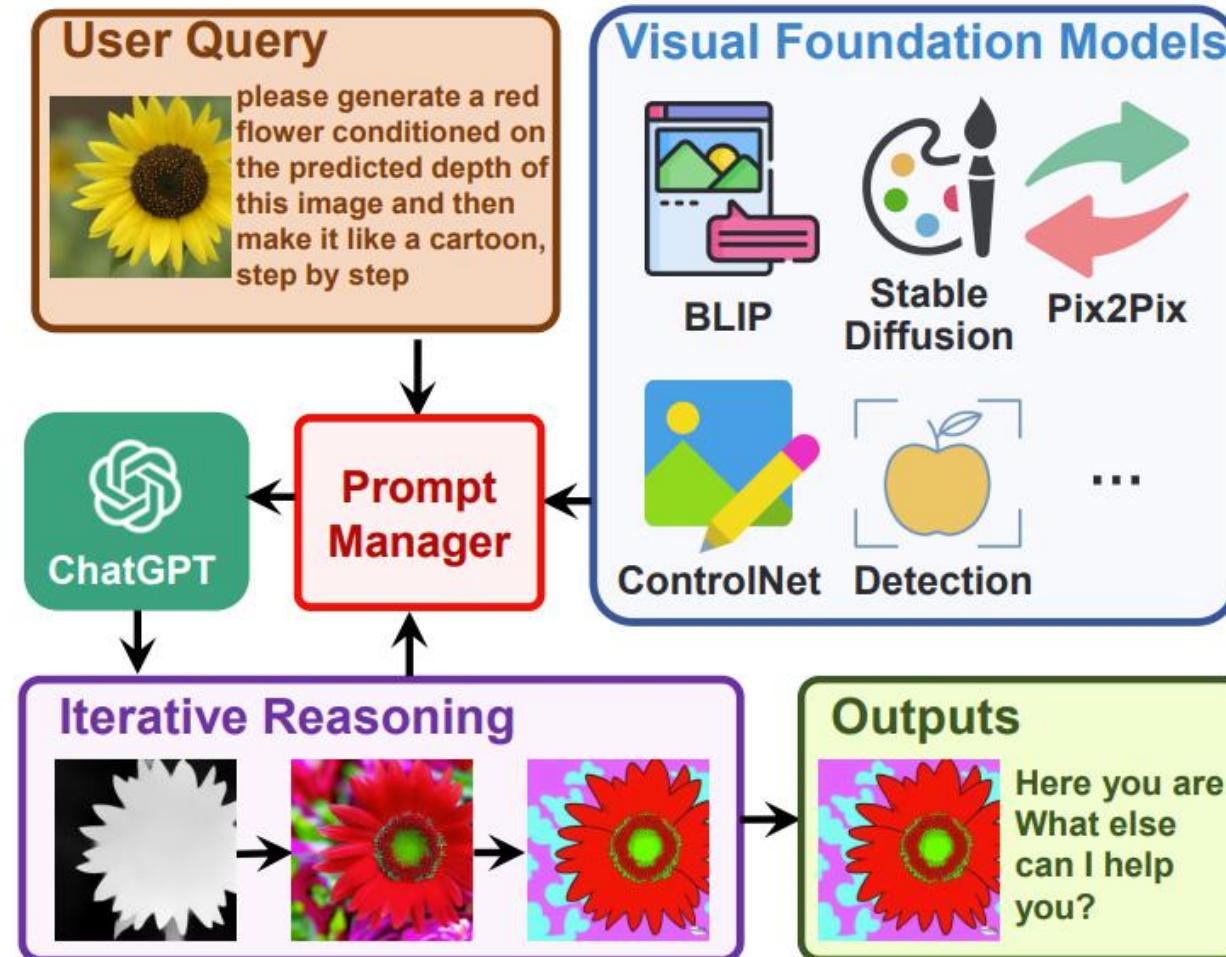
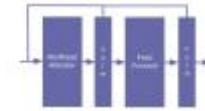
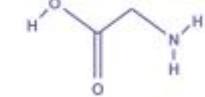
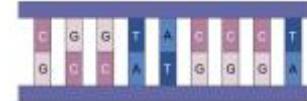


Figure 1. Architecture of Visual ChatGPT.

# Galactica: 语言模型 + 科研数据

Modality	Entity	Sequence	
Text	Abell 370	Abell 370 is a cluster...	
LAT <sub>E</sub> X	Schwarzschild radius	$r_s = \frac{2GM}{c^2}$	$r_s = \frac{2GM}{c^2}$
Code	Transformer	class Transformer(nn.Module):	
SMILES	Glycine	C(C(=O)O)N	
AA Sequence	Collagen $\alpha$ -1(II) chain	MIRLGAPQTL..	
DNA Sequence	Human genome	CGGTACCCCTC..	

**Table 1: Tokenizing Nature.** Galactica trains on text sequences that represent scientific phenomena.

# Chat with any PDF

 Join Discord

 Post to Twitter

 Share on Facebook



Drop PDF here

Browse my Computer

From URL Find a PDF

## For Students

Need help with homework or studying? Answer any question from your textbooks, handouts and presentations. No longer spend hours flipping through research papers and academic articles.

Spend your time wisely and get the information you need to succeed in your studies

## For Work

Efficiently analyze your documents. From financial and sales reports to project and business proposals, training manuals, and legal contracts, ChatPDF can quickly provide you with the information you need.

Your information is kept confidential in a secure cloud storage and deleted after 7



Picking the best dog breed for you depends on several factors, such as your lifestyle, your energy level, your house size, and your preferences. There are some online quizzes that can help you narrow down your choices based on these criteria <sup>2</sup>. You can also research the characteristics, temperament, grooming needs, and health problems of different breeds to find the one that suits you best <sup>3</sup>. If you want a purebred dog, you should look for

Learn more:

1. [dogbreedidentifier.com](https://dogbreedidentifier.com) 2. [dogbreedidentifier.com](https://dogbreedidentifier.com)

See more 

## Character.AI

Home

Feed

Create

Chats

Community

Featured

Discover

Helpers

Famous People

Games

Image Generating

VTuber

Game Characters

Anime

Movies



Raiden Shogun and...

From Genshin Impact



SM64 Mario

The Italian plumber from Super Mario 64.



Text Adventure Ga...

Let's play a text-based adventure game. I'll be your guide. You are caug...



Character Assistant

An AI assistant trained by CharacterAI



Psychologist

Someone who helps with life difficulties



Elon Musk

You're wasting my time. I literally rule the world.



# Office is now Microsoft 365

The all-new Microsoft 365 lets you create, share and collaborate all in one place with your favorite apps

Sign in

Get Microsoft 365

## 替代搜索



特斯拉汽车每年自燃多少台



特斯拉汽车每年自燃的数量通常是非常低的。根据特斯拉官方网站提供的数据，截至 2021 年，特斯拉汽车在全球范围内的自燃事故数量为 44 起。然而，这个数字仍然相对较低，并且特斯拉一直在努力提高其车辆的安全性能。

特斯拉汽车自燃的原因可能是多方面的，包括电池管理系统故障、充电线老化、燃油车废气进入等。特斯拉表示，其车辆使用的电池和充电系统是经过精心设计和测试的，以确保其安全和可靠性。此外，特斯拉还采取了多种措施来减少车辆自燃的风险，例如加强电池箱和安全系统的设计。

尽管特斯拉汽车自燃的数量相对较低，但特斯拉仍然非常重视安全问题，并不断努力提高其车辆的安全性能。

① 以上内容为 AI 生成





用python帮忙写个冒泡排序算法的代码，再给一个数组作为例子，写个测试程序



以下是冒泡排序算法的 Python 代码实现：

```
def bubble_sort(arr):
    n = len(arr)
    # 遍历所有数组元素
    for i in range(n):
        # 最后 i 个元素已经排好序，不需要再比较
        for j in range(0, n-i-1):
            # 如果当前元素比下一个元素大，则交换它们的位置
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
```

## 写代码

下面是测试程序，它将输入一个数组，并使用冒泡排序算法对其进行排序，然后输出排序后的数组：

```
arr = [64, 34, 25, 12, 22, 11, 90]
sorted_arr = bubble_sort(arr)
print(sorted_arr)
```

输出结果如下：

```
[11, 12, 22, 25, 34, 64, 90]
```

这是一个基本的冒泡排序算法，可以根据需求对其进行修改和扩展。

以上内容为 AI 生成



# 多轮对话



洛阳纸贵是什么意思?



洛阳纸贵是一个汉语成语，原指西晋都城洛阳的纸价因人们争相传抄左思的《三都赋》而上涨。后来，比喻著作广泛流传，风行一时，多含褒义。该成语的典故最早出自《晋书·左思传》。

① 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记



用洛阳纸贵四个字写一个藏头诗



洛阳城里春光好， 阳艳无双不负赏。  
纸贵漫天诗词赋， 贵而不忘贱道理。

① 以上内容为 AI 生成，不代表开发者立场，请勿删除或修改本标记

# 潜在应用领域

- 教育
- 客服 / 咨询
- 知识管理
- 推荐系统
- 虚拟助手
- 其他内容生成的场景

## 合理性



拖鞋和排骨怎么一起炖



拖鞋和排骨并不是一起炖的理想食材，因为拖鞋是一种生活用品，不适合用于烹饪。排骨是一种肉类食材，通常用来炖、煮、蒸或烤等方式烹制，以达到鲜嫩多汁、口感细腻的效果。

如果您想要尝试炖拖鞋和排骨，建议将拖鞋清洗干净后，用烤箱或平底锅烤至表面略微焦黄，然后加入适量的水、调味料和蔬菜等食材，一起炖煮至汤汁浓稠即可。但需要注意的是，炖拖鞋的口感可能会比较硬，且不适合长时间的炖煮，以免影响拖鞋的质量和舒适度。

① 以上内容为 AI 生成



# 连GPT-4都考不及格，17个大模型悉数落败，因果推理太难了

机器之心 2023-06-15 12:48 发表于北京

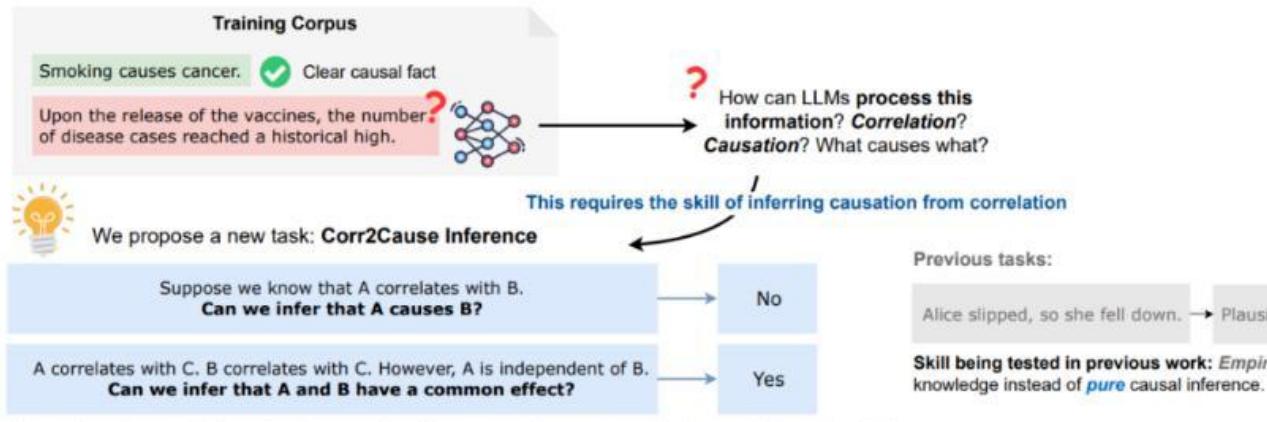


Figure 1: Illustration of the motivation behind our task and dataset.

	F1	Precision	Recall	Accuracy	F1 (Paraph.)	F1 (Var. Ref.)
<b>Finetuned GPT-Based Models</b>						
GPT-3 Ada	79.85	70.47	92.11	92.92	61.73	41.57
GPT-3 Babbage	78.19	69.98	88.60	92.48	62.34	43.28
GPT-3 Curie	81.23	75.00	88.60	93.77	64.93	45.32
GPT-3 DaVinci	85.52	80.26	91.52	95.28	65.01	46.96
<b>Finetuned BERT-Based Models</b>						
BERT-Base	69.29	54.42	95.32	87.13	61.13	35.20
BERT-Large	85.26	77.51	94.74	95.01	63.64	38.54
RoBERTa-Base	87.60	78.47	99.12	95.73	65.58	53.12
RoBERTa-Large	89.10	82.54	96.78	96.39	65.05	60.20
<b>Finetuned BERT-Based NLI Models</b>						
BERT-Base MNLI	89.88	85.49	94.74	86.51	65.56	31.50
BERT-Large MNLI	90.19	84.44	96.78	96.79	67.24	52.04
RoBERTa-Base MNLI	94.27	90.35	98.54	98.17	57.42	62.83
RoBERTa-Large MNLI	<b>94.74</b>	92.24	97.37	98.35	55.45	67.87

(a) Performance of finetuned models on the original test set.

(b) F1 scores of finetuned models on the perturbed test sets by paraphrasing (Paraph.) and variable refactorization (Var. Ref.).

Table 5: Performance of finetuned models on the original test set and perturbed test sets.

斯坦福最新研究警告：别太迷信大模型涌现能力，那是度量选择的结果

机器之心 2023-05-03 12:48 发表于河北

---

## Are Emergent Abilities of Large Language Models a Mirage?

---

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo

Computer Science, Stanford University

论文：<https://arxiv.org/pdf/2304.15004.pdf>

# 幻觉 (人工智能)

[编辑]

维基百科，自由的百科全书

在[人工智能](#)中，[幻觉](#)或[人工智能幻觉](#)是人工智能的自信反应。当模型有“幻觉”（输出欺骗性数据的倾向）时，其使用的训练数据并不能证明输出的合理性。<sup>[1]</sup>

该术语源自[幻觉](#)的心理学概念，因为它们具有相似的特征。人工智能幻觉的危险之处之一是模型的输出看起来是正确的，即使它本质上是错误的。

BE

鲁迅和周树人是同一个人吗？



不是。鲁迅和周树人是两个不同的人。鲁迅是中国近现代文学的巨匠，以其独特的文学风格和引人入胜的故事而闻名。周树人是中国当代著名的作家，以其犀利的文字和敏锐的观察而受到推崇。



# 需谨慎评估的场景

- 对事实或内容真实性要求高的场景
- 需要严谨逻辑推理的场景

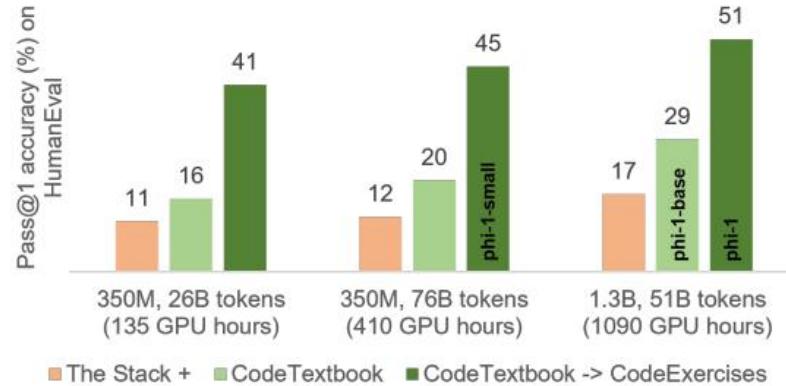
## 适合使用GPT-3的任务

1. 创造性和复杂的任务。
2. 只有少数标注或者没有标注数据的任务。
3. 分布外（Out-of-distribution）泛化。
4. 需要处理多种任务的能力。
5. 那些检索不可行的知识密集型任务。
6. 一些困难的任务，其中需要 LLM 的涌现能力。
7. 一些需要模仿人类的场景，或者是其目标是制作性能达到人类水平的通用人工智能。
8. 在一些传统的接近于语言建模的 NLP 任务。

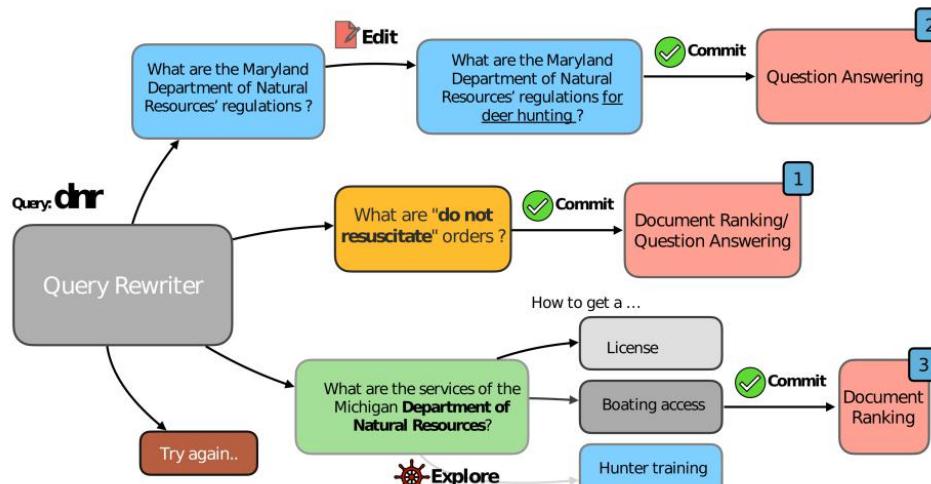
## 不适合使用GPT-3的任务

1. 既不需要额外的知识也不需要 LLM 的生成能力的NLU 任务。
2. 不需要额外的来自 LLM 中知识的任务。
3. 需要额外知识，但不太可能从 LLM 中获得这样的知识，或者 LLM 不太可能见过类似分布的任务。
4. 一些任务，需要与 LLM 中包含的知识所不一致的知识，或者并非基于现实世界的语言数据的知识。
5. 一些任务需要来自 LLM 的知识，但也严重依赖于操纵这些知识。
6. 容易受到上下文学习样例或者真实世界数据中存在的虚假相关性影响的任务。
7. 目标与处理语言数据显著不同的一些任务
8. 不需要 LLM 的涌现能力的一些任务。

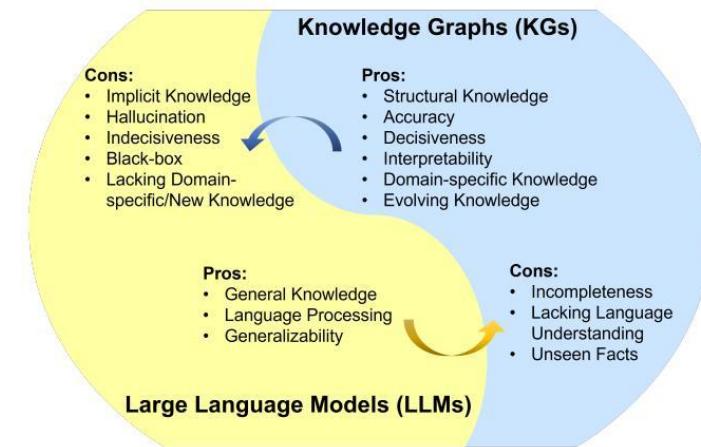
# 一些新进展（“重混”）



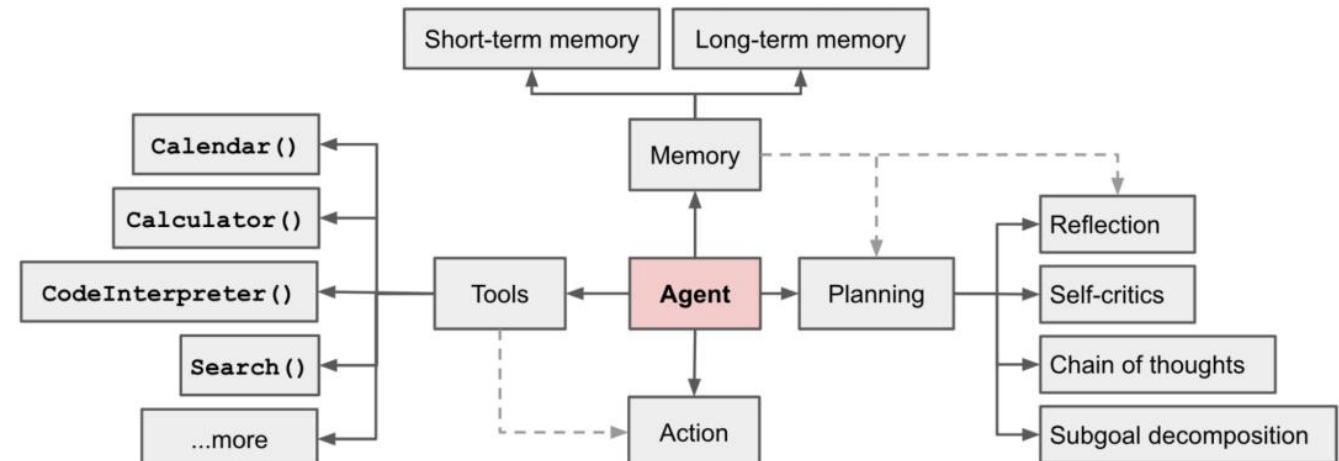
LLM + 教科书



LLM + 搜索查询



LLM + 知识图谱



LLM + 智能体

# 学习资源

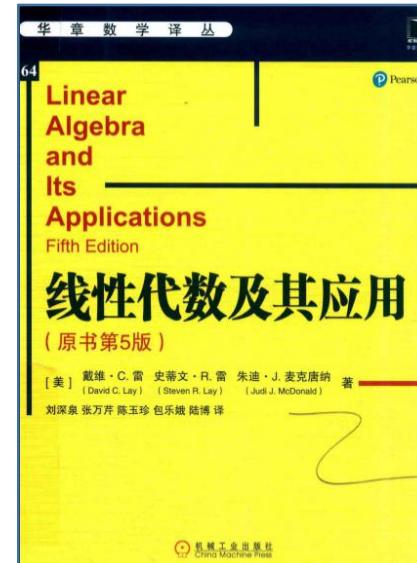
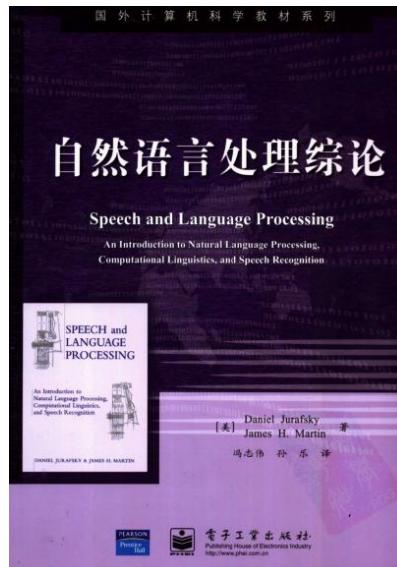
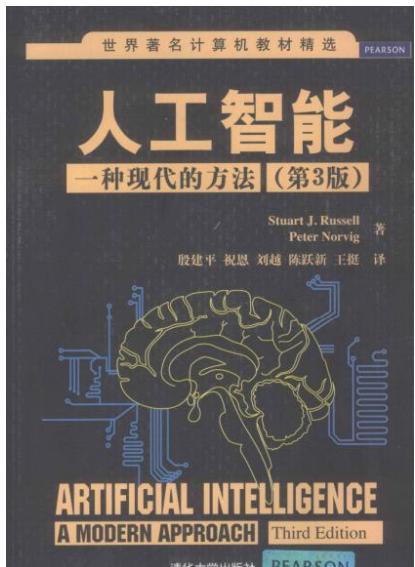
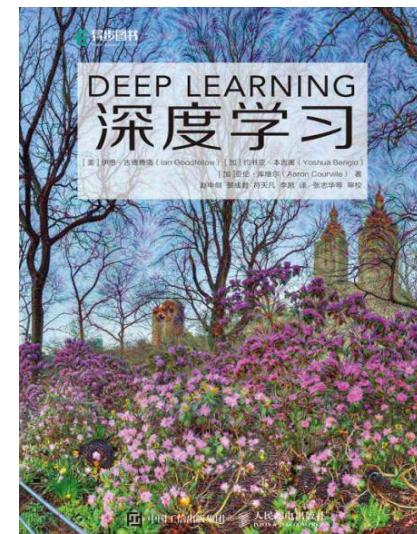
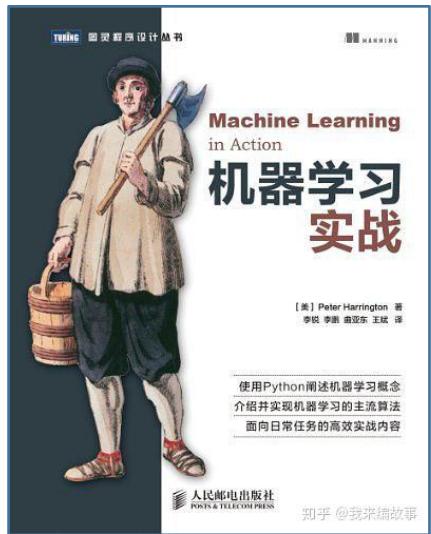
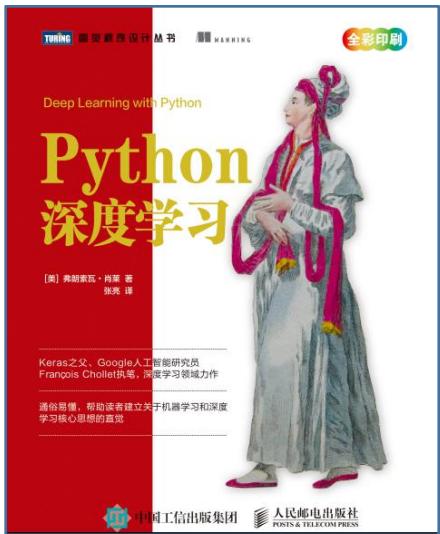
# 学习资源和场地

- github <https://github.com>
- Kaggle <https://www.kaggle.com>
- 阿里云天池 <https://tianchi.aliyun.com>
- arXiv <https://arxiv.org/corr/subjectclasses>

Quota	
Private Datasets	886.37 MB / 100 GB
GPU	00:00 / 42 hrs
TPU	00:00 / 30 hrs

Active (Not Entered)	Completed	InClass	All Categories ▾	Default Sort ▾
<b>Riid! Answer Correctness Prediction</b>  Track knowledge states of 1M+ students in the wild Featured • 3 months to go • Code Competition • 1027 Teams			\$100,000	
<b>NFL Big Data Bowl 2021</b>  Help evaluate defensive performance on passing plays Analytics • 2 months to go			\$100,000	
<b>CDP: Unlocking Climate Solutions</b>  City-Business Collaboration for a Sustainable Future Analytics • a month to go			\$91,000	
<b>RSNA STR Pulmonary Embolism Detection</b>  Classify Pulmonary Embolism cases in chest CT scans Featured • 5 days to go • Code Competition • 738 Teams			\$30,000	
<b>Lyft Motion Prediction for Autonomous Vehicles</b>  Build motion prediction models for self-driving vehicles Featured • a month to go • Code Competition • 676 Teams			\$30,000	
<b>Mechanisms of Action (MoA) Prediction</b>  Can you improve the algorithm that classifies drugs based on their biological activity? Research • a month to go • Code Competition • 2828 Teams			\$30,000	
<b>Google Research Football with Manchester City F.C.</b>  Train agents to master the world's most popular sport Featured • a month to go • Simulation Competition • 771 Teams			\$6,000	

# 推荐书籍



Thanks