



北京航空航天大學  
BEIHANG UNIVERSITY

# 自然語言處理

人工智能研究院

主讲教师 沙磊

# 主讲教师介绍

沙磊

主要研究方向: 可解释大语言模型、大语言模型安全与数据增强、多模态大模型、AI4science。

- 履历

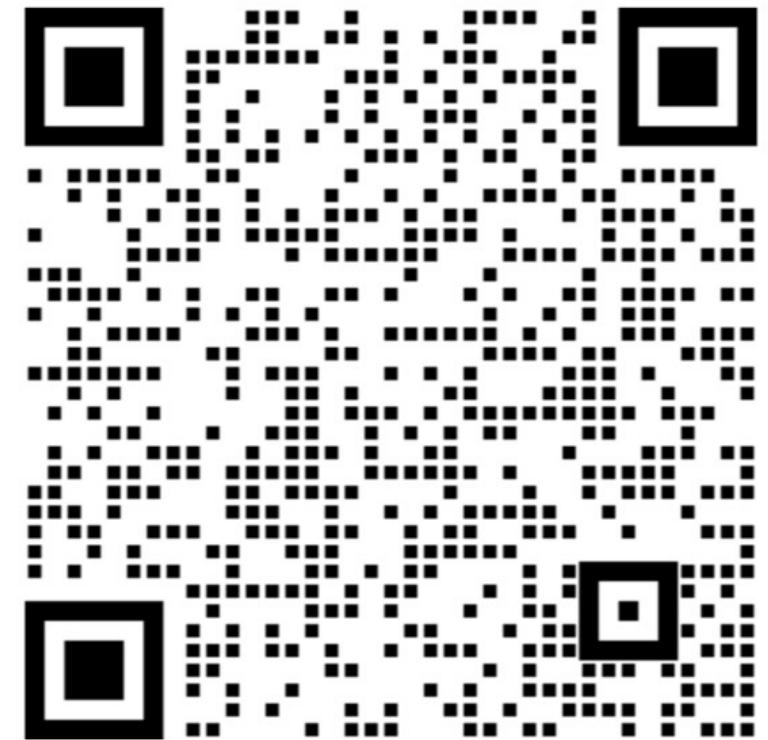
- |                 |                               |                               |
|-----------------|-------------------------------|-------------------------------|
| • 2022.4 至今     | 北京航空航天大学 人工智能研究院              | 教授                            |
| • 2020.3-2022.3 | 牛津大学 智能系统组                    | Research Associate            |
| • 2018.8-2020.3 | Apple Inc. Siri Understanding | Senior NLP Research Scientist |
| • 2013.9-2018.7 | 北京大学 计算语言所                    | 博士                            |

# 考核方式

- 平时成绩 60%
  - 出勤
  - 大作业
- 期末笔试 40%

# 课程信息

- 主页：  
[https://shalei120.github.io/course/NLP/Natural\\_Language\\_Processing.html](https://shalei120.github.io/course/NLP/Natural_Language_Processing.html)
- 课程群



该二维码 3月4日前有效，重新进入将更新

# 主要参考书

- 1. 自然语言处理概论, 俞士汶主编, 商务印书馆, 2003
- 2. Speech and Language Processing, Jurafsky, D. and Martin, J.H., 1st Edition Prentice Hall, 2000(中译本: 自然语言处理综论, 冯志伟等译, 电子工业出版社, 2005)
- 3. Speech and Language Processing, Jurafsky, D. and Martin, 2nd Edition, J.H., Prentice Hall, 2008
- 4. 自然语言处理: 基于预训练模型的方法, 车万翔等, 电子工业出版社, 2021

# 其他参考书

- Machine Learning A Probabilistic Perspective, Kevin P. Murphy, 2012
- Probabilistic Graphical Models, Daphne Koller, 2009
- Reinforcement Learning: An introduction, Richard Sutton
- Geometric Deep Learning, Grids, Groups, Graphs, Geodesics, and Gauges, Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković

# 其他参考书

- Foundations of Statistical Natural Language Processing, Manning,C.D. & Schütze,H., The MIT press, 1999 (有中译本)
- Statistical Language Learning. Charniak, E., The MIT Press. 1996.
- Natural Language Understanding, Allen, J., The Benjamins/Cummins Publishing Co., 1994 (有中译本)
- Natural Language Processing: An Introduction to Computational Linguistics, Gazdar, G. & Mellish, C., AddisonWesley, 1989.
- Introduction to Natural Language Processing, Harris, M.D., Reston Publishing Co. , 1985

# 其他参考书

- 统计自然语言处理，宗成庆，清华大学出版社， 2008
- 自然语言理解，姚天顺，清华大学出版社， 2002
- 自然语言处理技术基础，王小捷、常宝宝，北京邮电大学出版社， 2002
- 自然语言处理，刘颖，清华大学出版社， 2002
- 自然语言处理基础，冯志伟，商务印书馆， 2001
- 自然语言处理导论，翁富良、王野翊，中国社会科学出版社， 1998
- 自然语言的计算机处理，冯志伟，上海外语教育出版社， 1997
- 自然语言处理，刘开瑛、郭炳炎，科学出版社， 1999

# 相关学术期刊和会议

- 1. Computational Linguistics (ACL)
- 2. Transactions of the Association for Computational Linguistics (TACL)
- 3. 中文信息学报 (中文信息学会)
- 4. Annual Meeting of the Association for Computational Linguistics (ACL年会)
- 5. Conference on Empirical methods in natural language processing (EMNLP)
- 6, North American Chapter of the Association for Computational Linguistics (NAACL)
- 6. International Conference on Computational Linguistics(COLING)
- 7. 全国自然语言处理联合学术会议(CCL)
- 8, Natural Language Processing and Chinese Computing (NLPCC)



第一课

# 自然语言处 理概述

# 什么是自然语言处理？

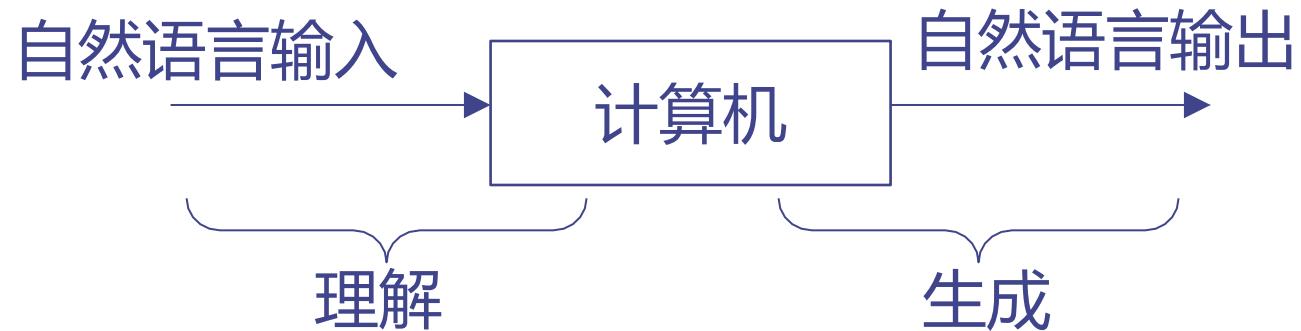
- 自然语言处理是通过建立形式化的计算模型来分析、理解和处理自然语言的学科。
- 什么是自然语言？
- 其它术语
  - 计算语言学(Computational Linguistics)
  - 自然语言理解(Natural Language Understanding)
  - 人类语言技术(Human Language Technology)

# 什么是自然语言处理？

- 自然语言处理是一门交叉学科。自然语言处理研究需要多个学科的知识。
  - 语言学（自然语言是处理对象）
  - 计算机科学（自然语言处理的研究工具）
  - 数学（自然语言的建模工具）

# 为什么要研究自然语言处理？

- 语言障碍
  - 人一人之间的语言障碍（自动翻译）
  - 人一机之间的语言障碍（人机接口）



# 自然语言处理的研究目标

- 终极目标
  - 研制能理解并生成人类语言的计算机系统。
- 当前目标
  - 研制出具有一定人类语言能力的计算机文本或语音处理系统。

# 自然语言处理的研究内容

- 建立形式化的适于计算机处理的语言模型。
- 研制分析、生成以及处理语言的各种算法

# 自然语言处理研究的挑战性

- 大量的词汇、大量的句子。
  - OED收词50万、汉语中有多少词？
- 无法象处理人工语言那样，写出一个完备的、有限的规则系统来进行定义和描述。自然语言的规则很少没有例外。(photo、potato)
- 自然语言中有大量的歧义现象。
- 自然语言的理解不仅和语言本身的规律有关，还和语言之外的知识（例如常识、领域知识）有关。因此语言处理涉及的常是海量知识，知识库的建造维护代价很高。

# 自然语言处理研究的挑战性

- 什么是歧义?
  - 对同一个语言形式有不止一种解读。
- 歧义是自然语言的固有属性，即使对于人类自身而言，也是如此。人工语言没有歧义。
- 语言单位无论大小都有歧义现象。
- 语言学家常把语言研究区分为不同的层次，如：音韵学、形态学、句法学、语义学、语用学等，在这些层面歧义都会有所表现。

# 自然语言处理研究的挑战性

- 歧义举例：

(1)The boy saw the girl with a telescope.

→Who has the telescope?

(2)At last, a computer that understands you like your mother

→The computer understands you as well as your mother  
understands you.

→The computer understands that you like your mother.

→The computer understands you as well as it understands  
your mother.

# 常见对策

- ◆ 由于歧义等因素的存在，自然语言处理的性能还不能满足一般应用的需要，为了满足某些特殊的应用需求，传统上常采用下面的对策
  - 交互式处理策略
    - 人机互助进行处理
  - 子语言策略(sublanguage)
    - 限定处理文本的领域
  - 受控语言策略(controlled language)
    - 限定语言的词汇和句法，降低复杂度
- 做自然语言处理研究时，要避免贪大求全，应限定研究范围和目标。

# 自然语言处理的研究方法

1. 规则驱动的方法
2. 数据驱动的方法
3. 二者融合的方法

# 自然语言处理的研究方法

## 规则驱动的方法（符号主义）

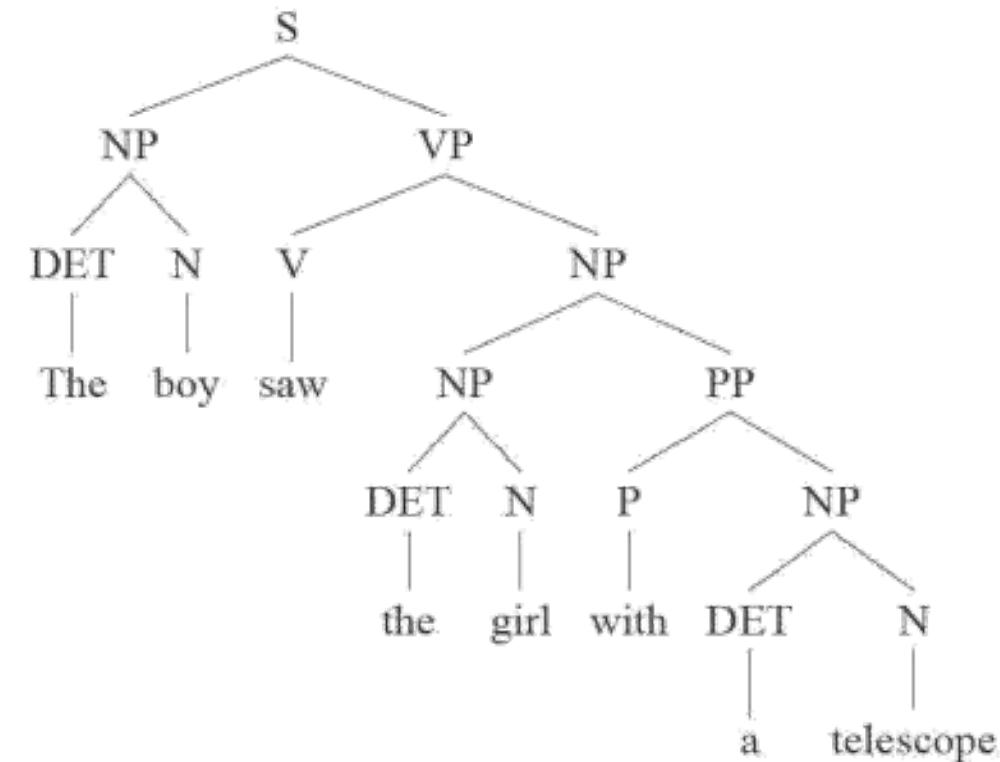
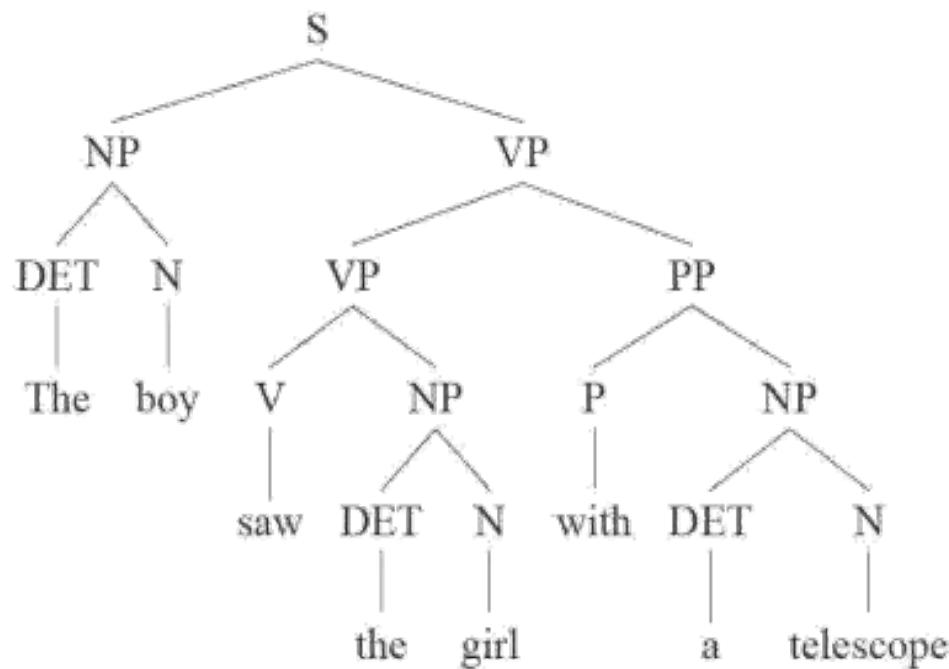
1. 研究人员（例如语言学家）对语言的规律进行总结，形成规则形式的知识库。
2. 研制语言处理算法，利用这些规则对自然语言进行处理。
3. 研究人员根据处理结果，调整规则，改进处理效果。

# 自然语言处理的研究方法

- 规则方法举例
- 例如：
  - $S \rightarrow NP + VP$
  - $NP \rightarrow DET + N$
  - $NP \rightarrow NP + PP$
  - $VP \rightarrow VP + PP$
  - $VP \rightarrow V + NP$
  - $PP \rightarrow P + NP$

# 自然语言处理的研究方法

- 用上述规则分析句子 “the boy saw the girl with an telescope”



# 自然语言处理的研究方法

- All grammar leak (Sapir 1921)
  - 对于自然语言而言，很难写出一部完备的规则集，语言规则有很强的伸缩性。
- 一般而言，很多基于规则的系统不能满足真实语言文本处理的要求，而只能处理真实语言的某个很小的子集。
  - toy system? toy syndrome

# 自然语言处理的研究方法

- 数据驱动的方法（统计方法）
  1. 建立可以反映语言使用情况的语料库。
  2. 研究人员对自然语言进行统计建模。
  3. 利用统计技术或机器学习技术，基于语料库训练语言模型。
  4. 利用得到的模型设计算法对语言进行处理。
  5. 根据处理效果改进模型，提高处理性能。

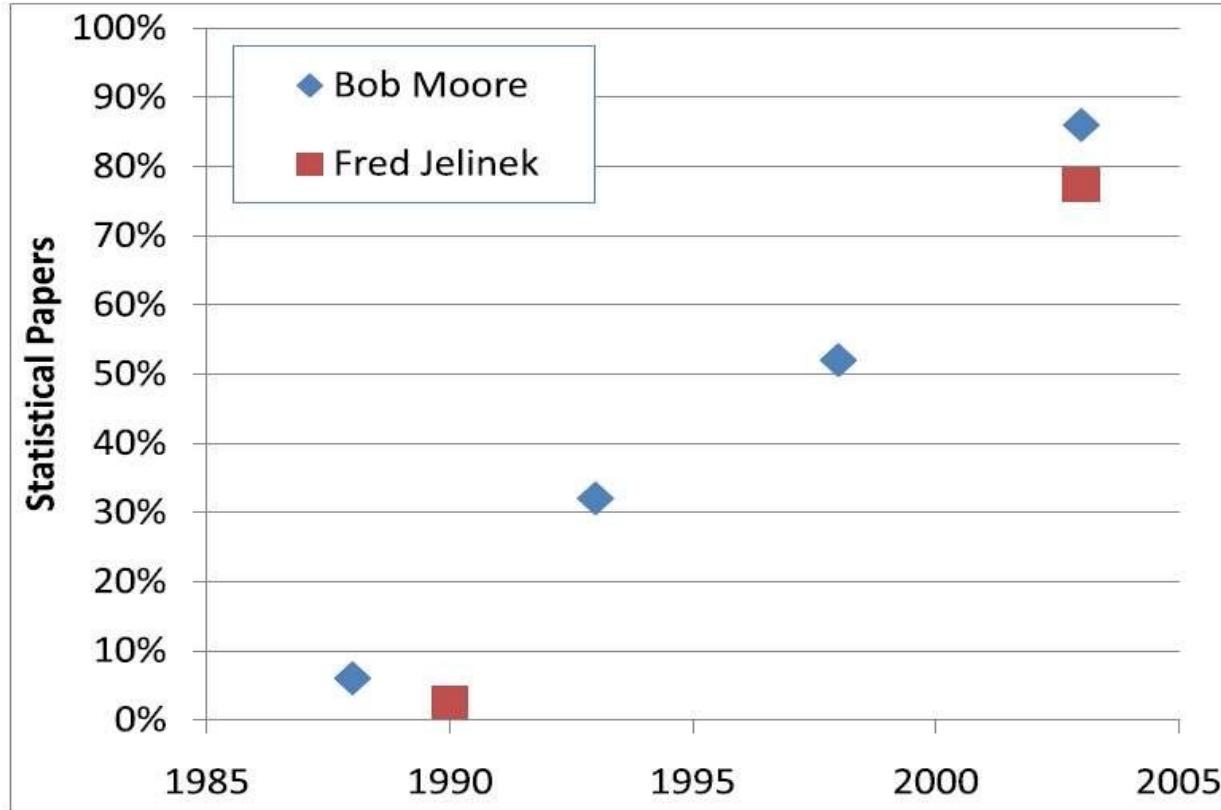
# 自然语言处理的研究方法

- 在数据驱动的方法中，语言模型通常体现为一组参数，这些参数通常表示某个语言形式发生的概率值。例如：
- $P(w_3|w_1 w_2)$
- $P(\text{公鸡}|\text{一只}) > P(\text{供给}|\text{一只})$
- 多项分布？
- 数据驱动的方法忽视了语言的深层结构(?)。

# 自然语言处理的研究方法

- 融合规则驱动和数据驱动的方法
  - 规则驱动、数据驱动都不是完美的方法，都有自己的优势和劣势
  - 综合两种方法有可能扬长避短并达到优势互补的结果
  - 两种方法如何结合需要探索

# 自然语言处理的研究方法



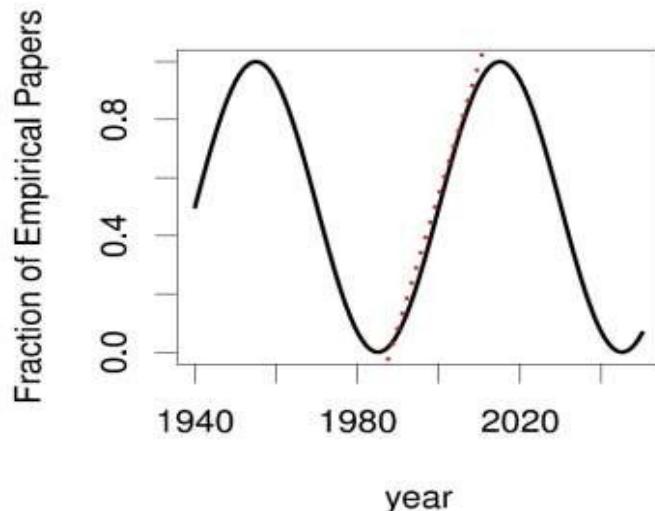
two independent surveys  
of ACL meetings by Bob  
Moore and Fred Jelinek

--- APENDULUM  
SWUNG TOO FAR

- 从学术会议看自然语言处理的研究方法
  - 90年代经验方法开始复苏
  - 机器学习以及统计技术目前是主流研究方法

# 自然语言处理的研究方法

- 如果我们考察更长的时间：
  - 1950s: Empiricism (Shannon, Skinner, Firth, Harris)
  - 1970s: Rationalism (Chomsky, Minsky)
  - 1990s: Empiricism (IBM Speech Group, AT&T Bell Labs)
  - 2010s: A Return to Rationalism?



- The oscillation
- between Rationalism and Empiricism
- --- A PENDULUM SWUNG TOO FAR

# 自然语言处理研究中的评测问题

- 为了评价各种方法的有效性，必须进行客观公正的评测，客观公正的评测有助于引导计算语言学朝着一个健康的方向发展。
- 国内外关于各类自然语言处理任务的性能评价如火如荼。863、973、TREC、MUC、SIGHAN、NIST、SensEval、SemEval等
- 自然语言很复杂、关于语言处理方法和系统的评测也不容易。
- 语言学争议与标准测试集
- 评测有正面作用，但也有负面效果。
  - 评价指标是否合理
  - 模型推广能力

# 自然语言处理的应用

- 自然语言处理有着广阔的应用领域。
  1. 机器翻译
  2. 人机对话
  3. 信息检索
  4. 信息提取
  5. 自动文摘
  6. 文本分类
  7. 拼写检查
  8. 音字转换

# 机器翻译

- 目标是研制能把一种自然语言翻译成另外一种自然语言的计算机软件系统。
  - 例如 汉英机器翻译系统
- 全自动高质量机器翻译(FAHQMT) – 尚须时日，相关研究始于四十年代末（计算机诞生不久）。
- 机器翻译研究经历了曲折的历程，正是对机器翻译的研究导致了自然语言处理的诞生。
- 目前市场上有不少翻译产品，应正确看待。

# 机器翻译

- 著名的例子
  - *the spirit is willing but the flesh is weak.*
  - *the vodka is good but the meat is rotten.*
- 联机机器翻译网站
  - SYSTRAN <http://www.systransoft.com/>
  - 华建<http://www.hjtrans.com/>
  - Google <http://translate.google.cn/#>
  - 微软<http://translator.live.com/Default.aspx>
  - 译星<http://www.transtar.com.cn/cn/index.asp>

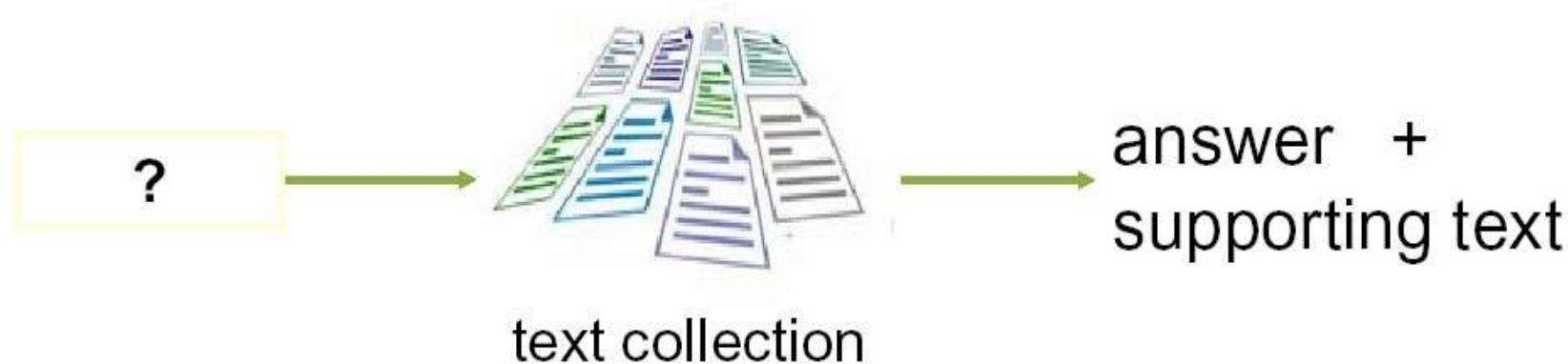
# 人机对话

- 科幻主题
  - 2001:A space odyssey(2001年太空漫游) 1968年奥斯卡奖
  - HAL9000
    - Dave: Open the pod bay doors, HAL.
    - HAL: I'm sorry Dave, I'm afraid I can't do that
    - Dave: What's the problem?
    - HAL: I think you know what the problem is just as well as I do.

# 人机对话

- 自然语言接口
- Question answering system (QA系统)
- 例子：
  - **Question:**
    - ◆ Who is Ronald Reagan's wife?
  - **Possible answers:** XML TXT
    1. Nancy Davis Reagan (1923-...) is the second wife of Ronald Reagan, who served as president of the United States from 1981 to 1989.
    2. Nancy Reagan, wife of President Ronald Reagan, was born Anne Frances Robbins.
    3. .....

# 人机对话



- 联机QA系统
  - AnswerBus <http://www.answerbus.com/>
  - AskJeeves <http://web.ask.com/>
  - START <http://start.csail.mit.edu>

# 信息检索



- Google、百度、Bing、天网
- 信息检索是NLP吗？

# 自动文摘

## About Columbia Newsblaster

Columbia Newsblaster is a system to automatically track the day's news. There are no human editors involved -- everything you see on the main page is generated automatically, drawing on the sources listed on the left side of the screen.

Every night, the system crawls a series of Web sites, downloads articles, groups them together into "clusters" about the same topic, and summarizes each cluster. The end result is a Web page that gives you a sense of what the major stories of the day are, so you don't have to visit the pages of dozens of publications.

- 访问 : Columbia Newsblaster  
<http://www1.cs.columbia.edu/nlp/newsblaster/>  
<http://newsblaster.cs.columbia.edu>

# 自动文摘

**Columbia Newsblaster**  
Summarizing all the news on the Web

Monday, September 10, 2007  
Articles from 09/07/2007 to 09/10/2007  
Last update: 8:56 AM EST

**Poll: More Think The 'Surge' Is Helping, 35% Say Iraq War Strategy Is Making Things Better, 70% Doubt Iraqi Government**  
Summary from multiple countries, from articles in English

By August, the American force in Iraq would be down to 15 combat brigades, the force level before President George W Bush's troop reinforcement plan. ([article 6](#)) The panel said the Iraqis should assume more control of its security and that U.S. forces should step back, emboldening Democrats who want troop withdrawals to start this fall. ([article 27](#)) More people now say Bush's troop buildup in Iraq, the so-called surge, has helped to improve the situation than said so last spring. ([article 31](#)) WASHINGTON Bush's war strategy is failing and the top military commander in Iraq is "dead flat wrong" for warning against major changes, the Democratic chairman of the Senate Foreign Relations Committee said Sunday. ([article 15](#)) Insurgent attacks against Iraqi civilians, their security forces and U.S. troops remain high, according to the document obtained by The Associated Press. ([article 20](#)) (CBS/AP) The Bush administration is leaving the ultimate decision on whether to keep troops in Iraq to the next president, Sen. Edward Kennedy, D-Mass., said Sunday on Face The Nation. ([article 18](#)) In a letter to troops, Army Gen. David Petraeus said Iraq's political leaders had not made the gains hoped for under the US troop

Search for:

[U.S.](#)  
[World](#)  
[Finance](#)  
[Entertainment](#)  
[Sports](#)

[View Today's Images](#)  
[View Archive](#)  
[About Newsblaster](#)  
[About today's run](#)  
[Newsblaster in Press](#)  
[Academic Papers](#)

**Article Sources:**  
[washingtonpost.com](#)



# 信息提取

- 文本数据结构化

BOGOTA, 3 APR 90 (INRAVISION TELEVISION CADENA 1) – [REPORT][JORGE ALONSO]

Liberal senator Federico Estrada Velez was kidnapped on 3 April at the corner of 60<sup>th</sup> and 48<sup>th</sup> streets in western Medellin, only 100 meters from a metropolitan police CAI [Immediate Attention Center]. The Antioquia department liberal party leader had left his house without any bodyguards only minutes earlier. As he waited for the traffic light to change, three heavily armed men forced him to get out of his car and get into a blue Renault.

Hours later, through anonymous telephone calls to the metropolitan police and to the media, the Extraditables claimed responsibility for the kidnapping. In the calls they announced that they will release the senator with a new message for the national government.

Last week, Federico Estrada Velez has rejected talks between the government and the drug traffickers.

# 信息提取

模板编号：

1

事件发生时间：

03 APR 90

事件类型：

Kidnapping

肇事人：

“Three heavily armed men”

肇事组织：

“The Extraditables”

受害人：

“Federico Estrada Velez ”

受害人数：

1

受害人类别：

Political Figure

事件发生地点：

Colombia: Medellin(city)

# 其它应用

- 文本分类（自动判别文本的类别）
- 音字转换（汉字整句输入法）
- 拼写检查和自动勘校系统

# 自然语言处理简史

- 1940年代末—1960年代中期
  - Warren Weaver(49)、GeorgeTown系统(54)
  - Noam Chomsky(57)
  - 统计方法被放弃
- 1966年：ALPAC(66) 语义障碍
- 1970年代中期—1980年代
  - TAUM-METEO(76)
  - SYSTRAN(76)
  - AI繁荣
  - MT产品如Fujitsi、Hitachi、Siemens

# 自然语言处理简史

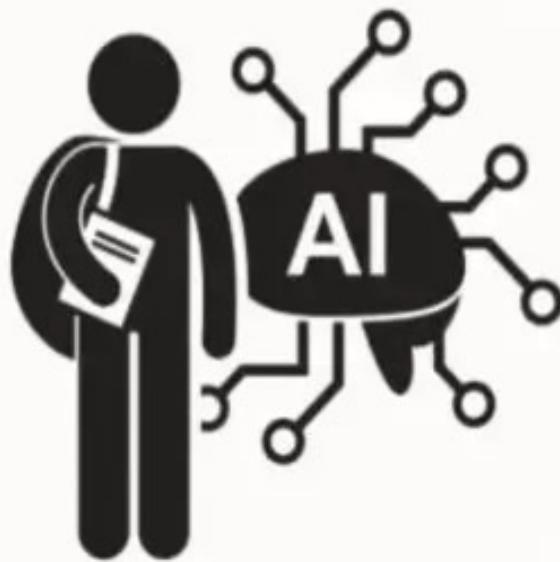
- 1980年代—1990年代前期
  - 欧盟Eurotra 计划(82)
  - 日本Mu系统以及ODA计划(82)
- 1990年代—2000年代中期
  - 统计方法复苏、IBM统计翻译(90)
  - 规则方法、统计方法融合
  - Internet的高速发展为自然语言处理发展注入了新的动力
- 2000年代中期 ----2018
  - 神经网络方法飞速发展，深度学习时代
- 2018 ----
  - 以BERT, GPT为代表的大规模预训练模型时代
  - Dalle, chatGPT

- ~~自然语言处理？~~
- ~~人工智能~~
- 通用人工智能 (AGI)

# 人工智能

## 早期人工智能 (AI)

制造智能机器和程序的工程



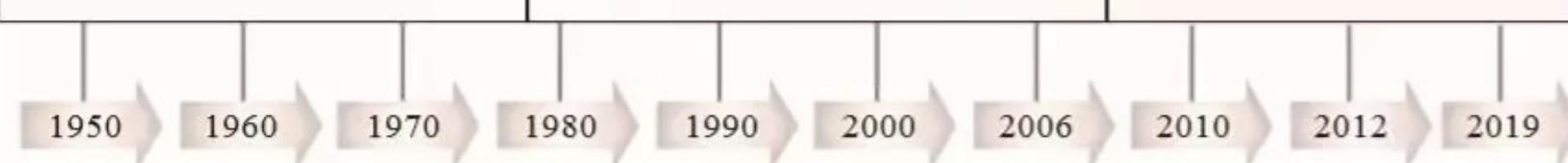
## 机器学习 (ML)

没有明确编程的学习能力

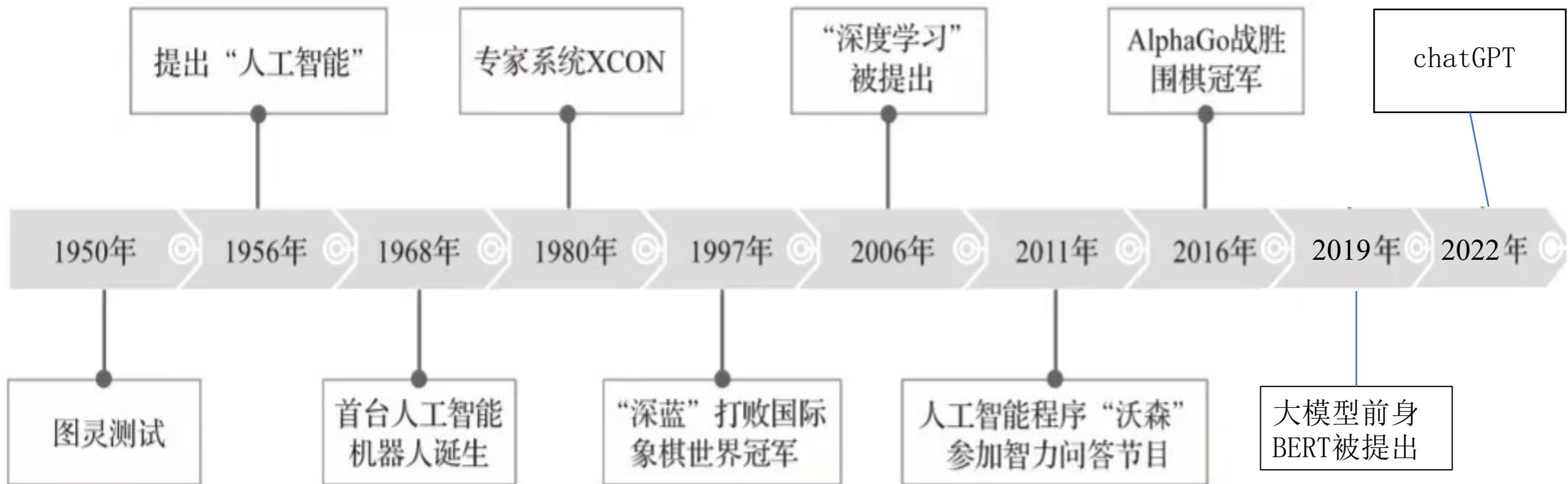


## 深度学习 (DL)

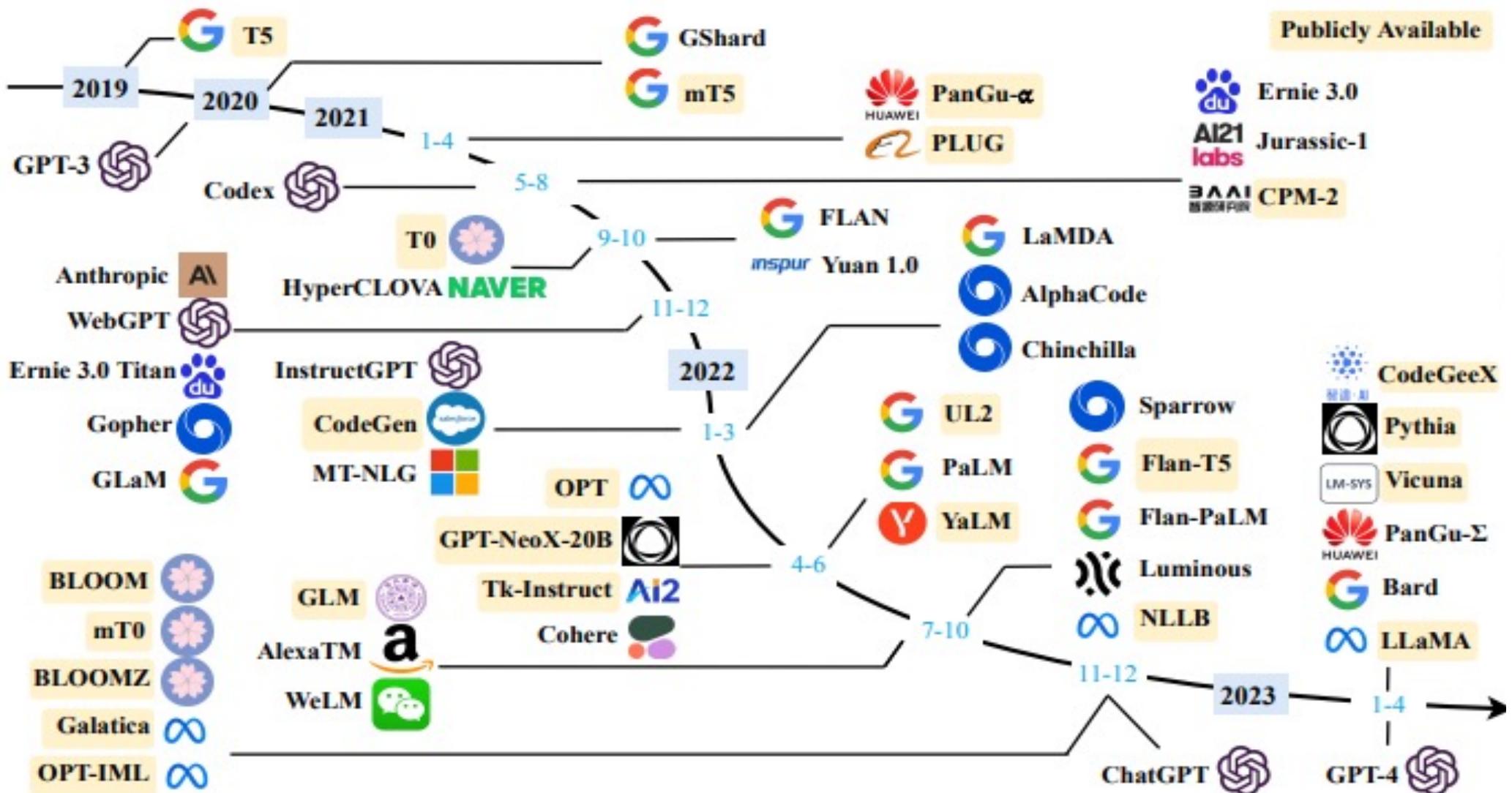
基于深度神经网络的学习



# 人工智能重大事件



# 大模型AIGC时代



# 颠覆性趋势1：生成式人工智能

- 文本生成
  - ChatGPT- 3.5, 4.0, 5.0 .....、LLaMa、Gemini
  - 个人AI助手
- 代码生成
  - 重复任务的自动化（代码生成、文档格式化、应用程序测试），
  - 优化创意流程，
  - 提高代码质量，
  - 支持解决问题。

# 颠覆性趋势1：生成式人工智能

- 图像生成
  - Dalle-2, 百度文心一格, GPT4
- 音频生成
- 视频生成



生成一张日落的照片



好的，以下是您要求的日落的照片：

``

预览：



知乎 @chatgpt账号  
CSDN @苏书QAQ

# 颠覆性趋势2：AI4Science

AI for Quantum Mechanics

AI for Density Functional Theory

AI for Small Molecules

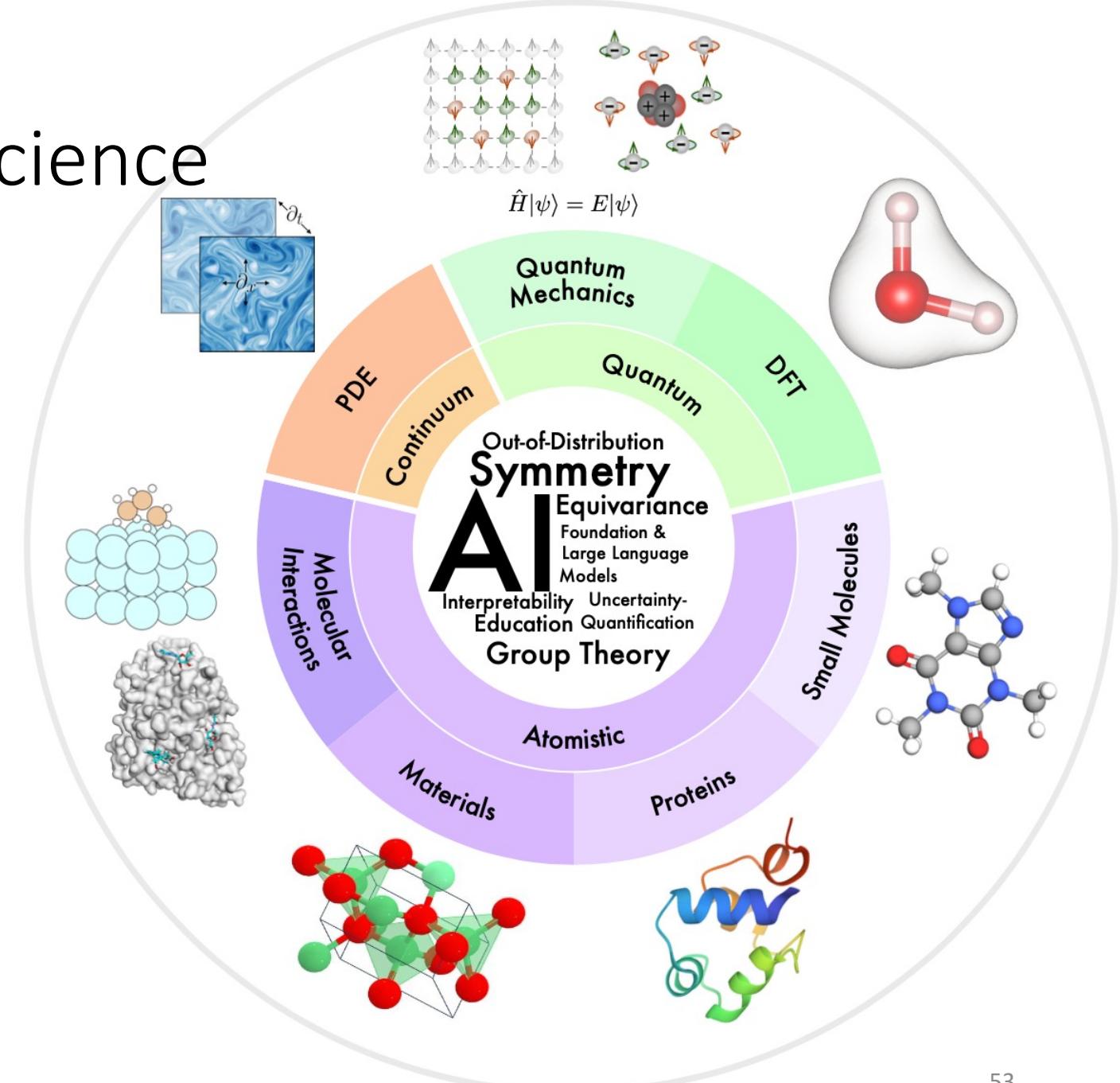
AI for Protein Science

AI for Materials Science

AI for Molecular Interactions

AI for Partial Differential Equations

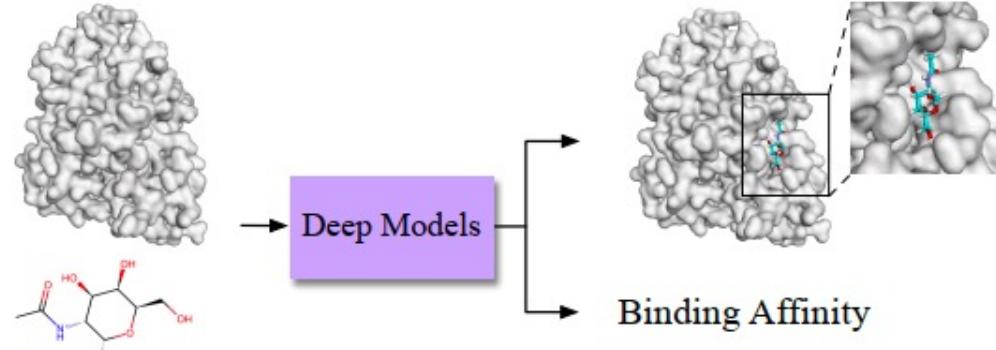
.....



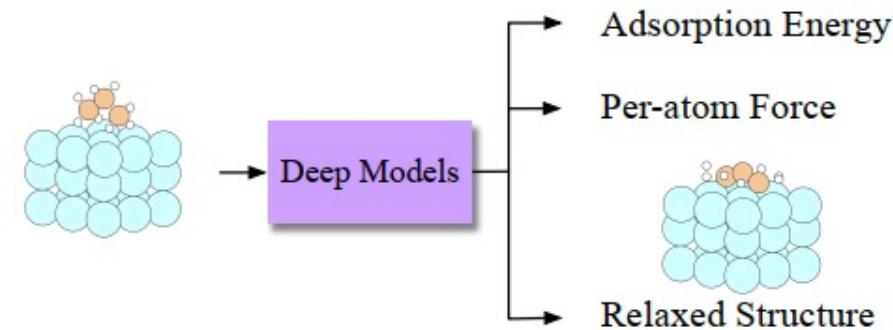
# 人工智能用于分子交互

Predictive Task

## Protein-Ligand Binding Prediction

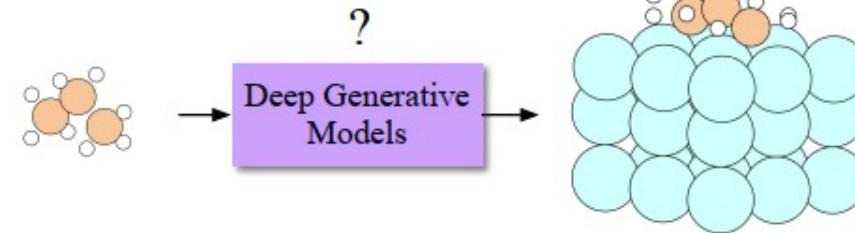
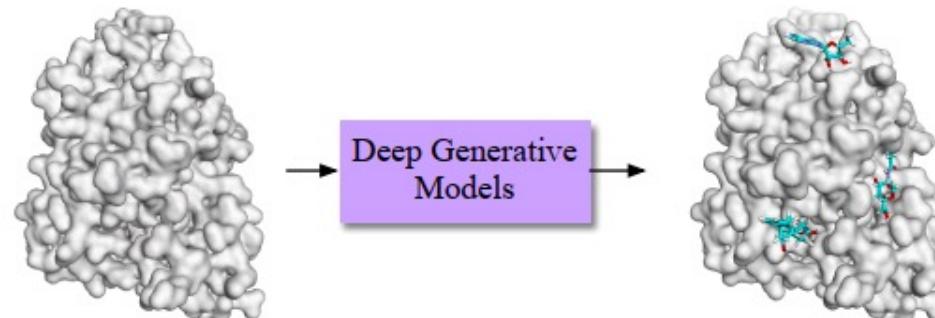


## Energy, Force, and Position Prediction



Generative Task

## Structure-Based Drug Design



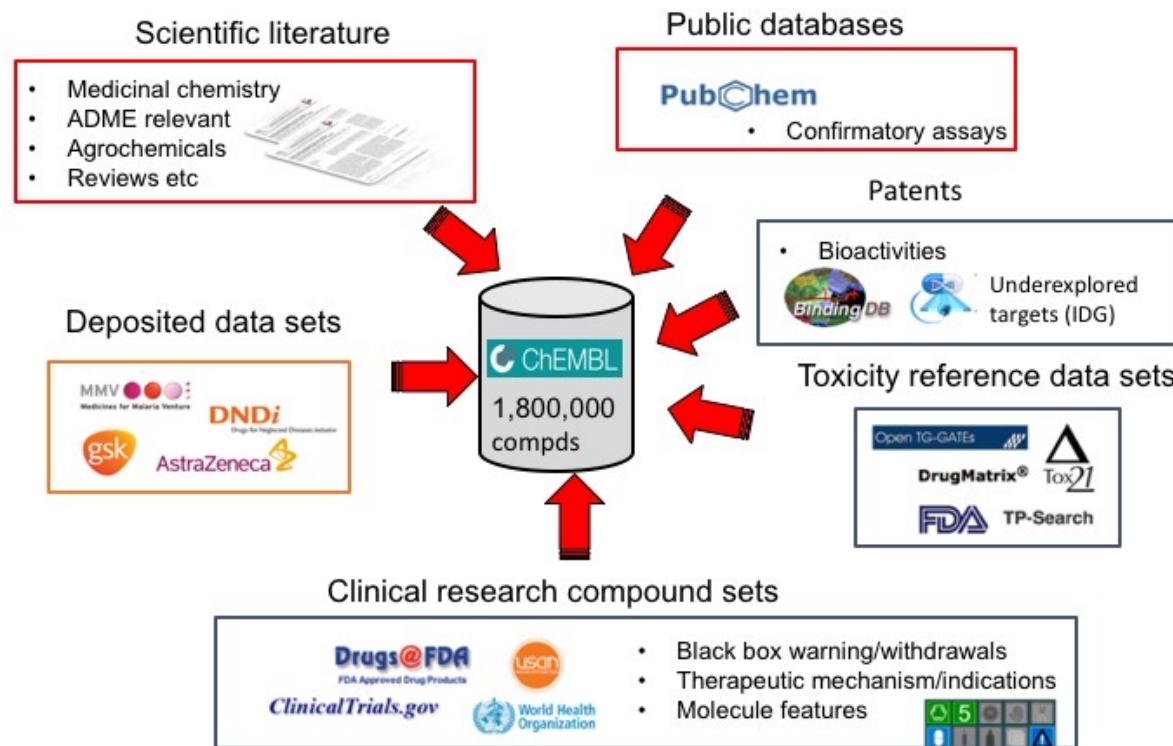
Small Molecule  $\longleftrightarrow$  Protein Molecule

Small Molecule  $\longleftrightarrow$  Material Molecule

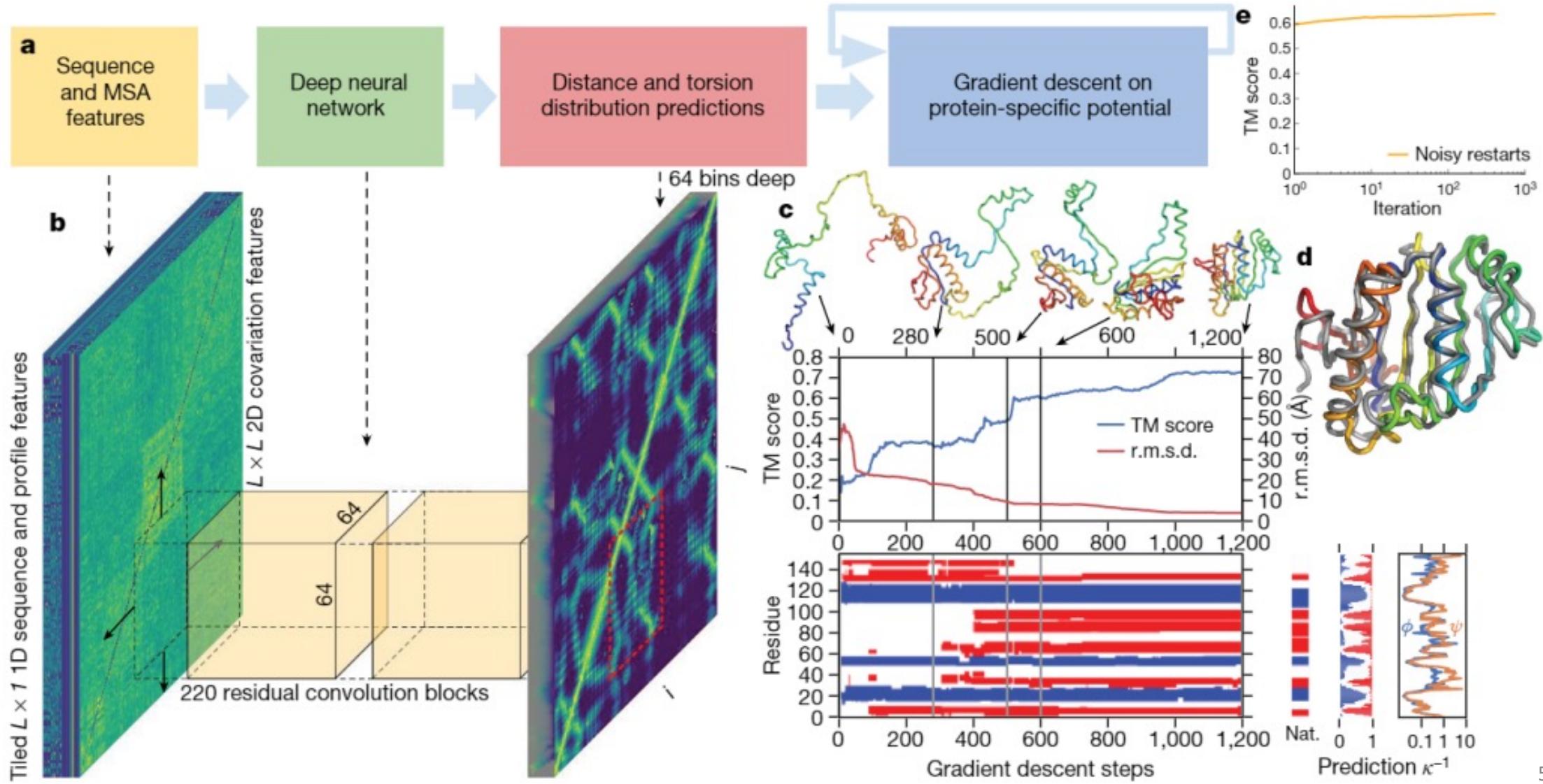
# Big Opportunity for Artificial Intelligence

- A huge amount of data is generated in the biomedical domain

## ChEMBL Database Content



# AlphaFold-1



# 神经网络&反向传播

# Classification and Regression

- 对于分类问题，我们有训练数据集：它由一些样本组成
- $\{x_i, y_i\}_{i=1}^N$
- $x_i$  是输入，例如单词(索引或是向量)，句子，文档等等(维度为  $d$  )
- $y_i$  是我们尝试预测的标签( $C$ 个类别中的一个)，例如：
  - 类别：感情，命名实体，购买/售出的决定
  - 其他单词
  - 多词序列(之后会提到)

# Classification intuition

- 训练数据  $\{x_i, y_i\}_{i=1}^N$
- 用一个最简单的2维词向量分类问题作为案例
  - 使用softmax / logistic回归
  - 构建线性决策边界
- 传统的机器学习/统计学方法：假设  $x_i$  是固定的，训练 softmax/logistic 回归的权重来决定决定边界(超平面)
- 预测阶段，对每个  $x$ ，预测：

$$p(y|x) = \frac{\exp(W_y \cdot x)}{\sum_{c=1}^C \exp(W_c \cdot x)}$$

# Training with “cross entropy loss”

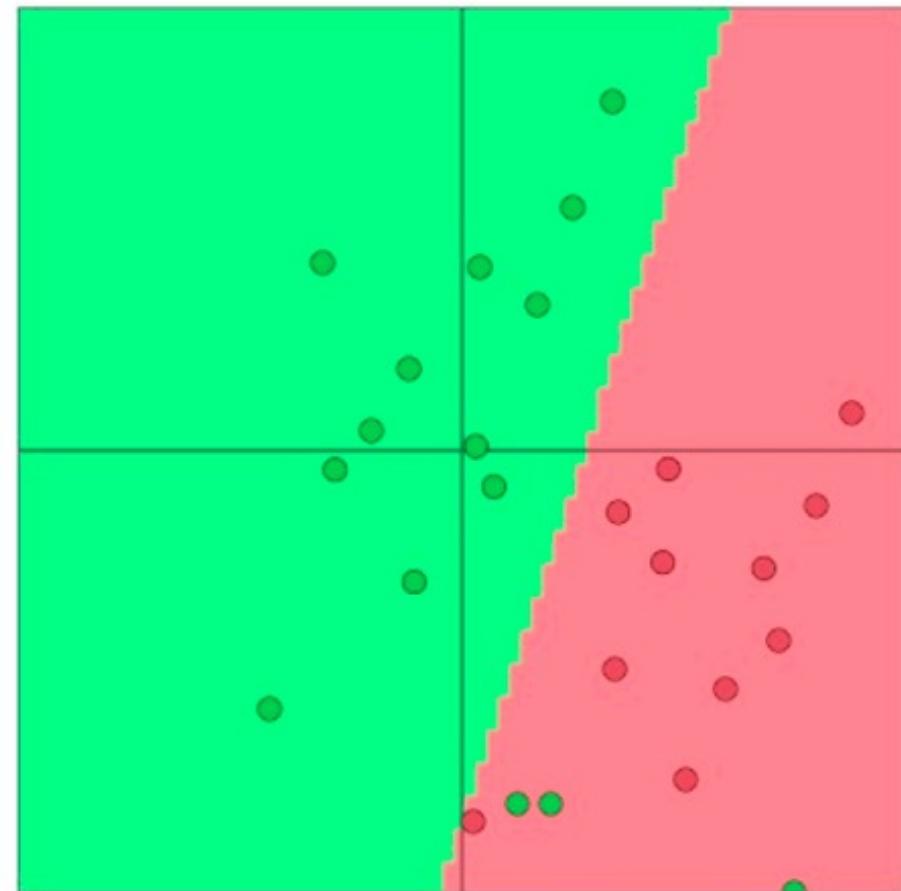
- 交叉熵- 信息论中的概念
- 真实概率分布  $p$
- 模型计算出的概率 $q$  
$$H(p, q) = - \sum_{c=1}^C p(c) \log q(c)$$
- 在softmax分类器中最常用到交叉熵损失，也是负对数概率形态。
- 对于每个训练样本 $(x, y)$ ，我们的目标是最大化正确类 $y$ 的概率，或者我们可以最小化该类的负对数概率
- 正确类 $y$ 的概率 (ground truth (or true or gold or target) probability distribution) 一般以one-hot的形式来表示：  $p = [0, \dots, 0, 1, 0, \dots, 0]$
- 所以交叉熵函数只剩下：  $-\log p(y_i|x_i)$

# Classification over a full dataset

- 整个数据集上的cross-entropy loss

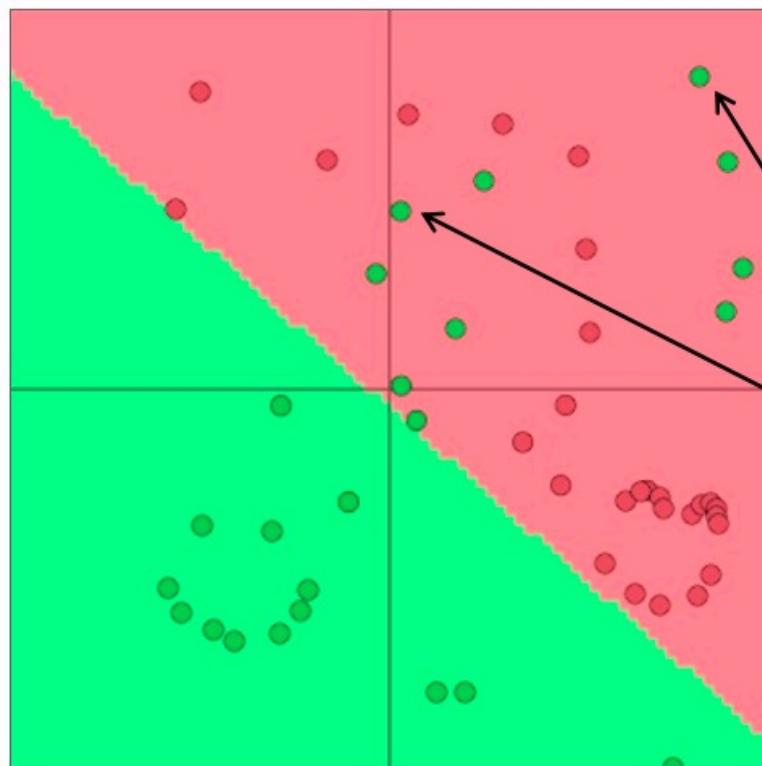
$$J(\theta) = \frac{1}{N} \sum_{i=1}^N -\log \left( \frac{e^{f_{y_i}}}{\sum_{c=1}^C e^{f_c}} \right)$$

$$f_{y_i} = W_{y_i} x$$



# Neural Network Classifiers

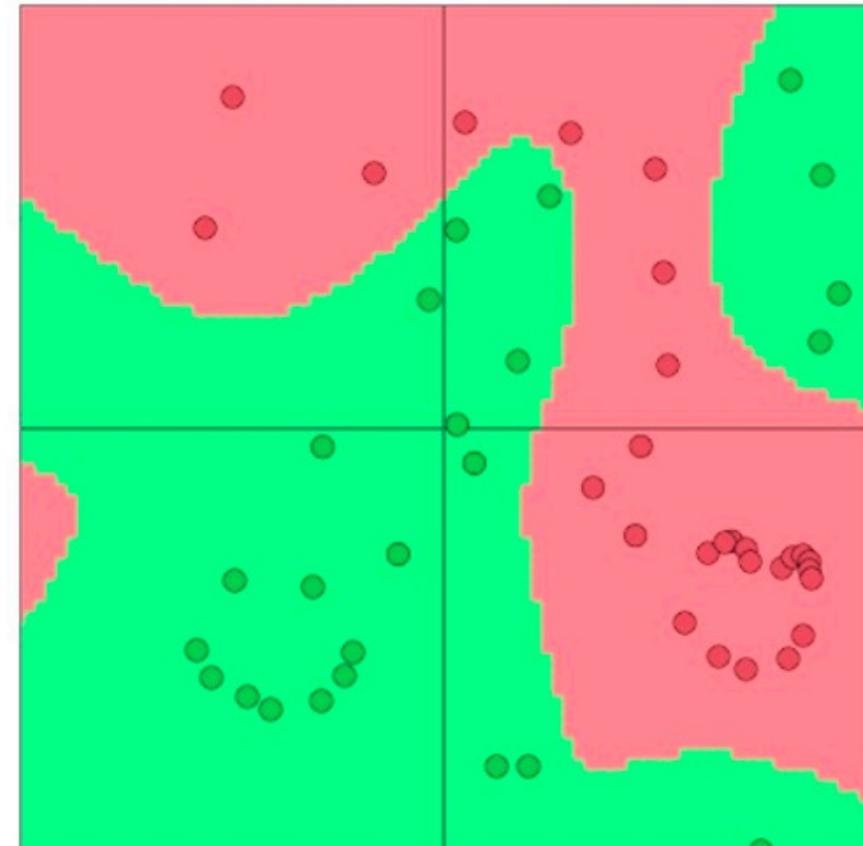
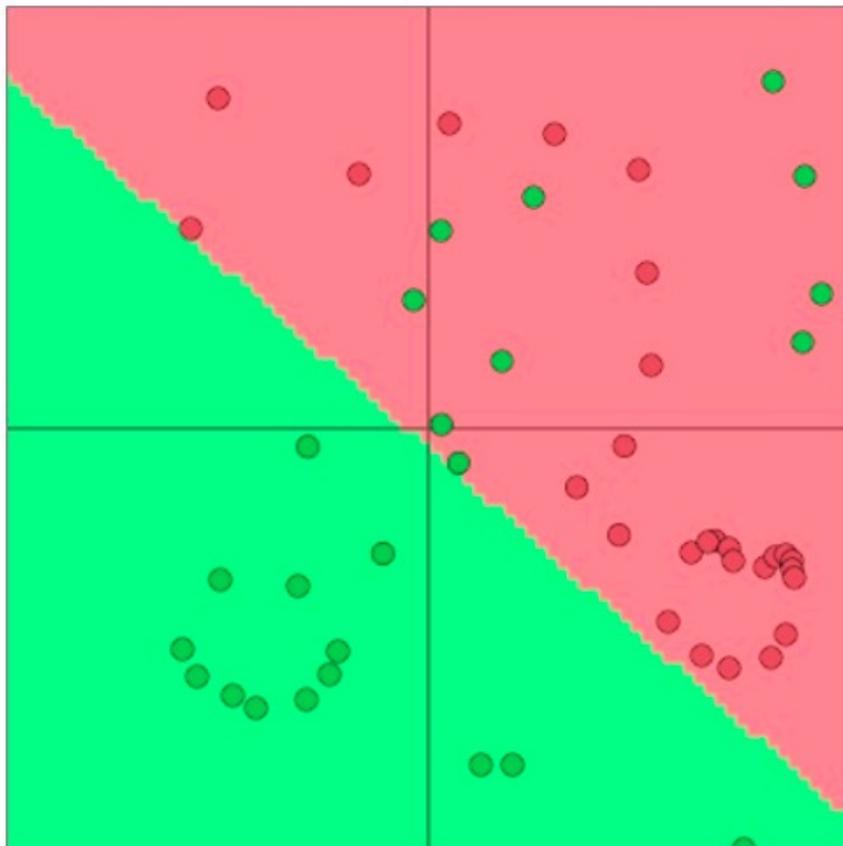
- 单独Softmax ( $\approx$  logistic regression) 能力有限
- 只能给出线性决策平面



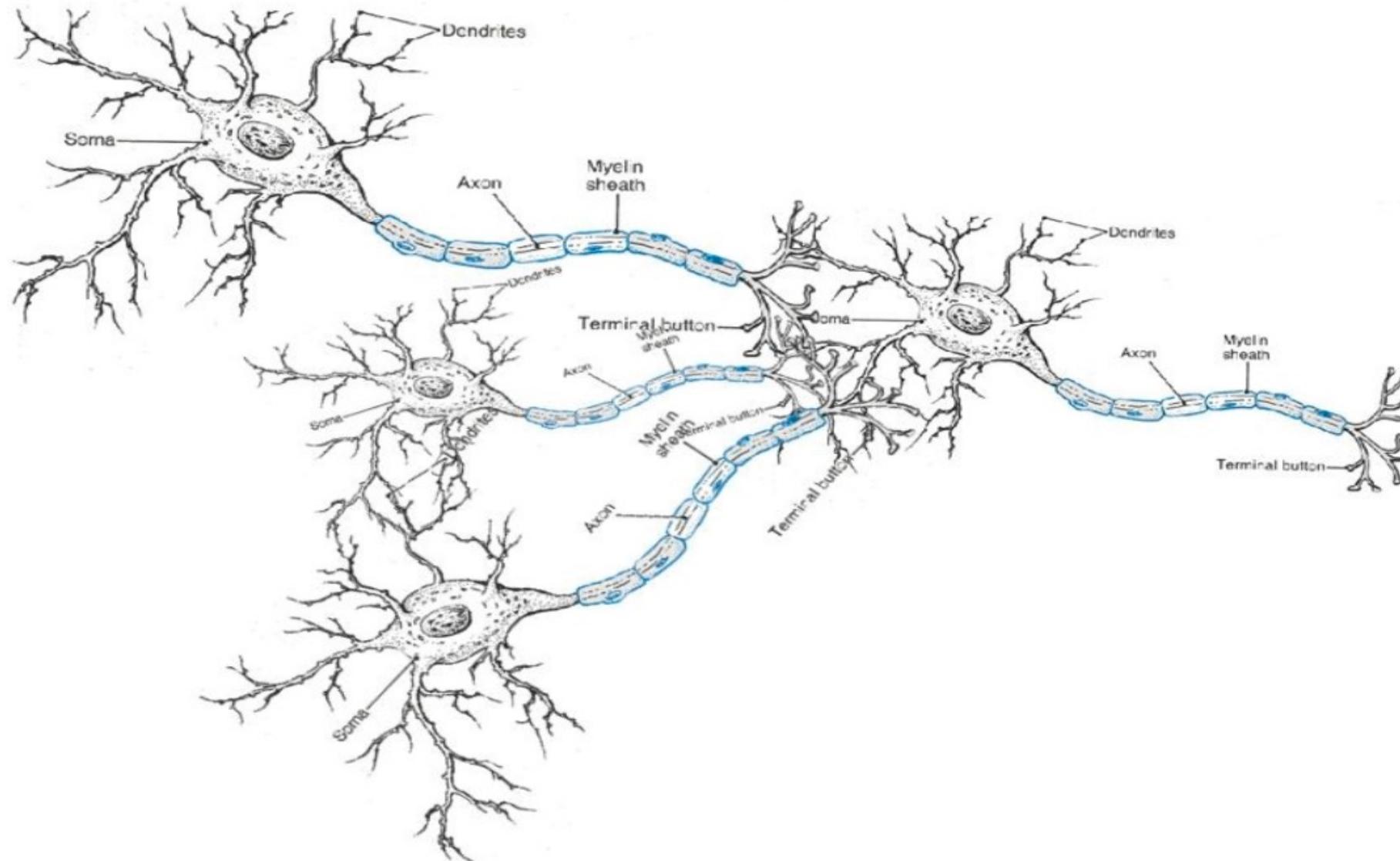
This can be quite limiting  
→ Unhelpful when a problem is complex  
Wouldn't it be cool to get these data points correct too?

# Neural Nets

- 神经网络可以学习到更复杂的非线性决策平面

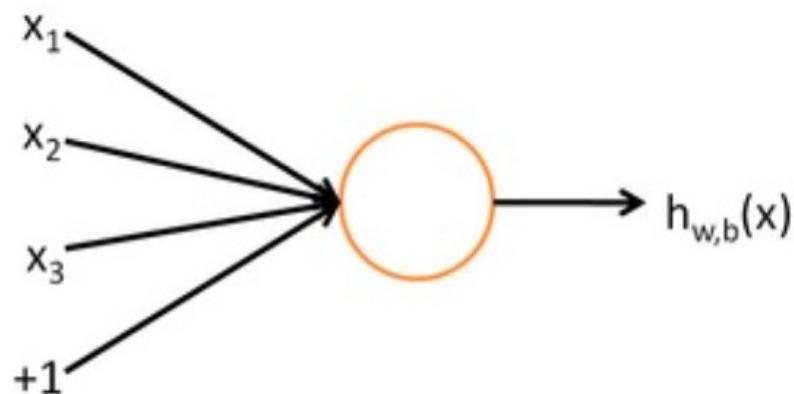


# Neural computation



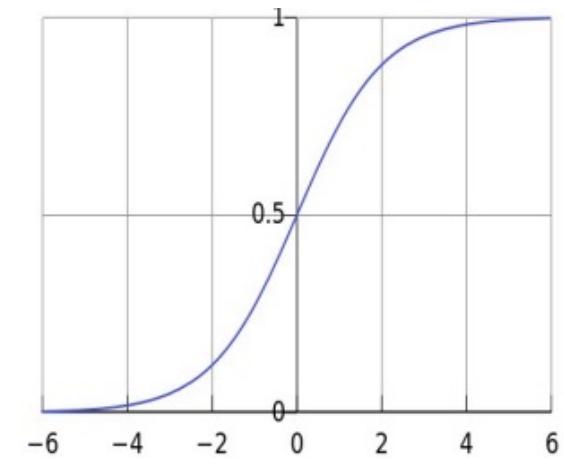
A binary logistic regression unit is a bit similar to a neuron

- f: 非线性激活函数 (sigmoid)
- W: 权重
- b: 偏置项
- h: 隐藏层
- x: 输入



$$h_{w,b}(x) = f(w^T x + b)$$

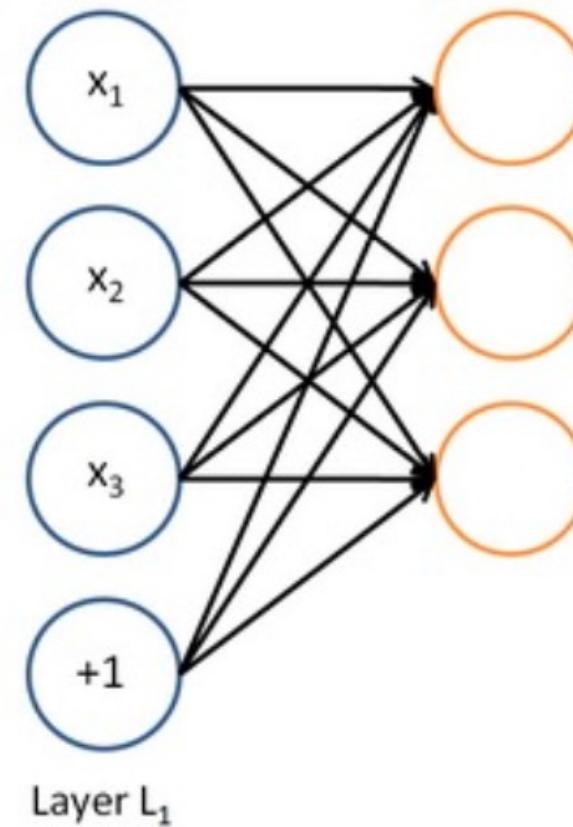
$$f(z) = \frac{1}{1 + e^{-z}}$$



W, b是这个神经元的参数  
Logistic regression

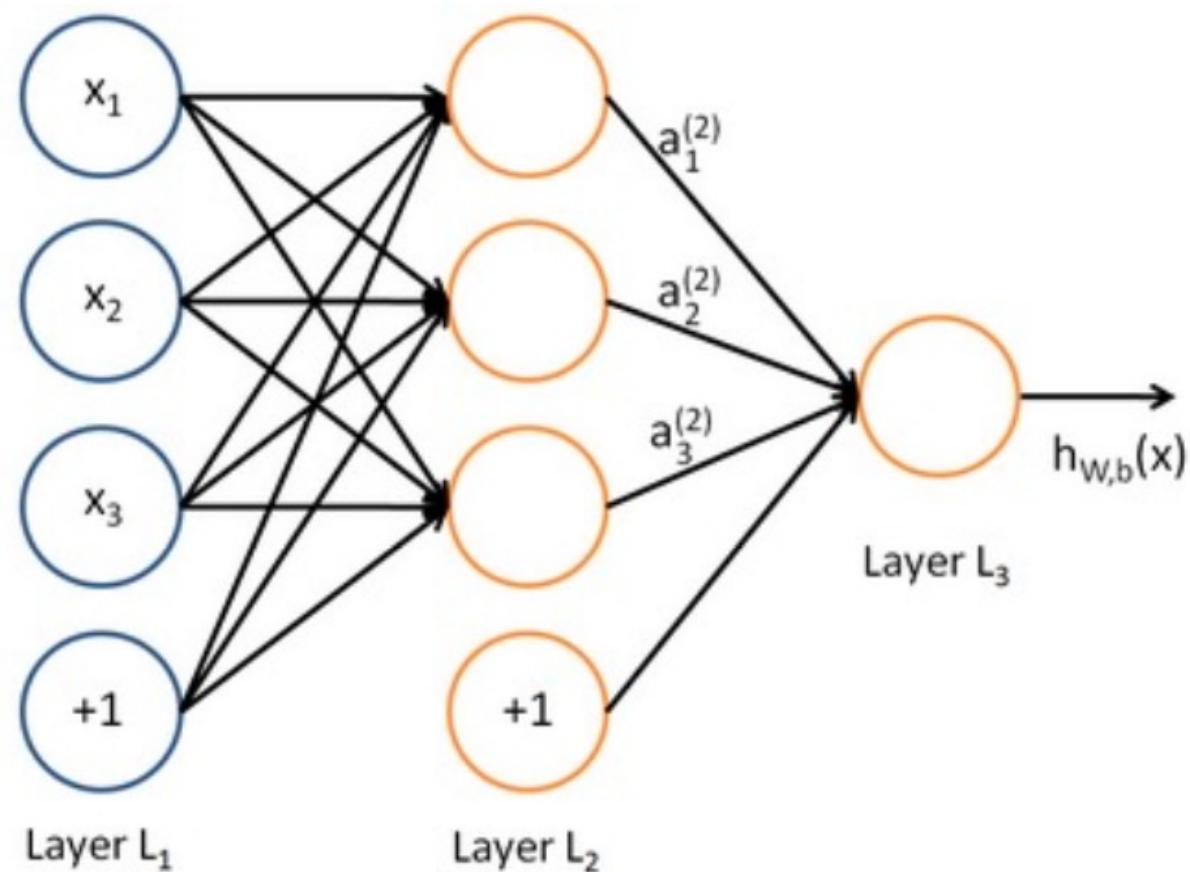
Difference: A neural network  
= running several logistic regressions at the same time

- 如果我们输入一个向量通过一系列逻辑回归函数，那么我们得到一个输出向量。
- 但是我们不需要提前决定这些逻辑回归试图预测的变量是什么。



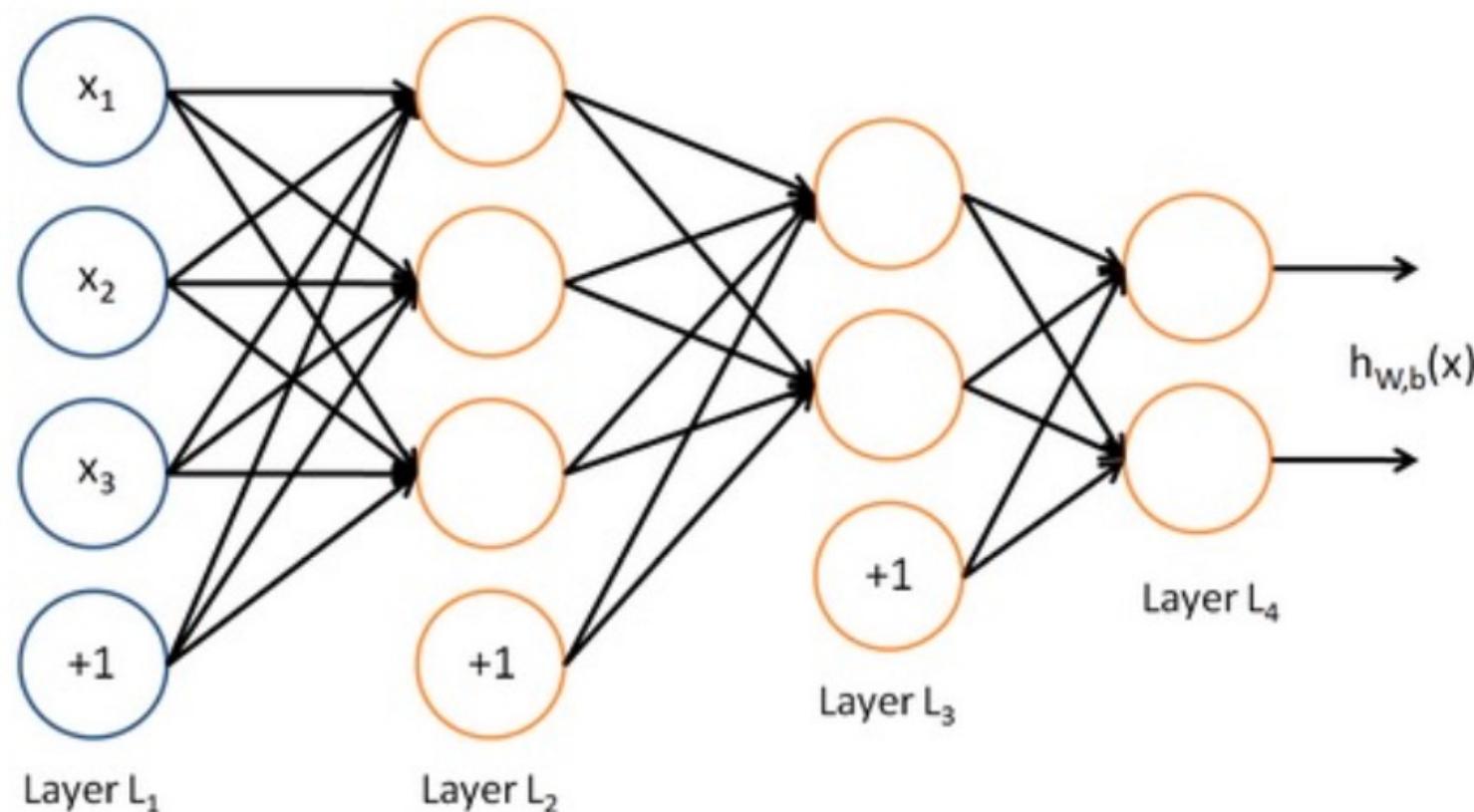
Neural Classification Difference: A neural network  
= running several logistic regressions at the same time

- 我们可以输入另一个 logistic 回归函数。
- 损失函数将指导中间隐藏变量应该是什么，以便更好地预测下一层的目标。



# Neural Classification Difference: A neural network = running several logistic regressions at the same time

- 我们添加更多层的神经网络，就得到了多层感知器。
- 这样我们可以多次重新组合数据特征，得到高度非线性的决策平面



# Matrix notation for a layer

- 一般的，我们有

$$a_1 = f(W_{11}x_1 + W_{12}x_2 + W_{13}x_3 + b_1)$$

$$a_2 = f(W_{21}x_1 + W_{22}x_2 + W_{23}x_3 + b_2)$$

etc.

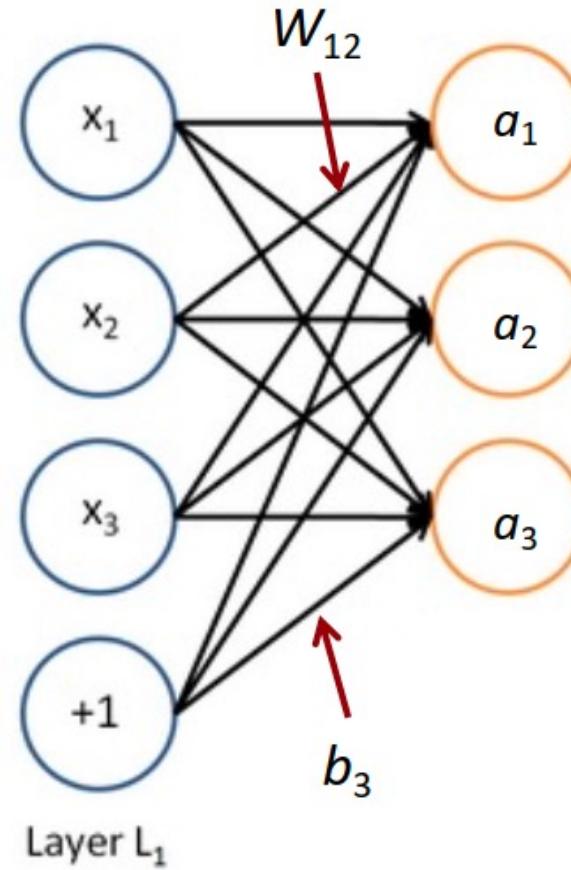
- 用矩阵表示就是

$$z = Wx + b$$

$$a = f(z)$$

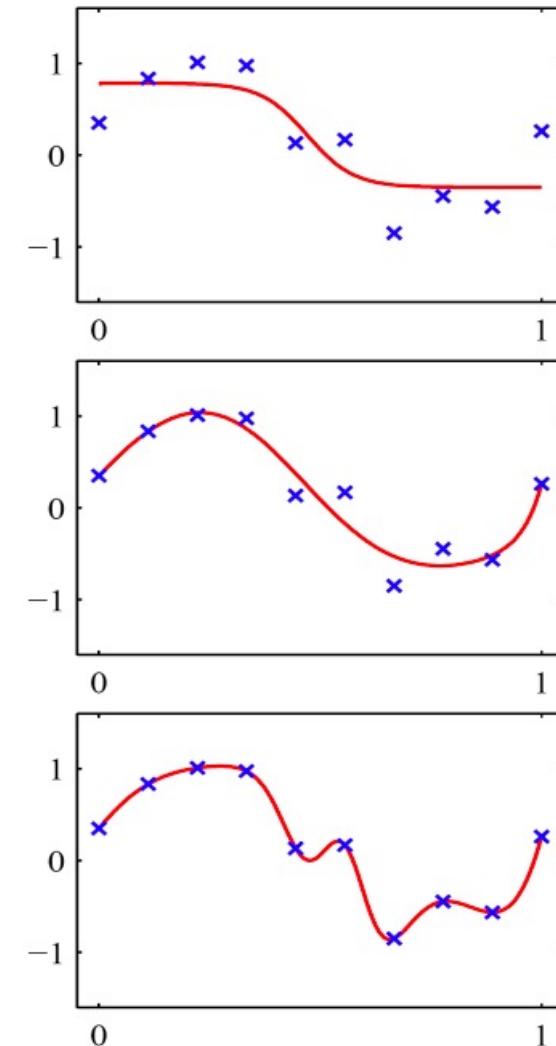
- 激活函数是element-wise的

$$f([z_1, z_2, z_3]) = [f(z_1), f(z_2), f(z_3)]$$



# 思考：为什么需要non-linear activation function?

- 例如：函数近似，如回归或分类
  - 没有非线性，深度神经网络只能做线性变换
  - 多个线性变换，也还是组成一个线性变换  $W_1 W_2 x = Wx$
- 对于非线性函数而言，使用更多的层，他们可以近似更复杂的函数



# 求梯度回顾

- 1个输出， n个输入的函数

$$f(\boldsymbol{x}) = f(x_1, x_2, \dots, x_n)$$

- 其梯度是对每个输入的梯度组成的向量

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]$$

# 求梯度回顾

- $m$ 个输出， $n$ 个输入的函数

$$\mathbf{f}(\mathbf{x}) = [f_1(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n)]$$

- 其梯度是 $m \times n$ 的偏导数矩阵  $\rightarrow$  雅可比矩阵

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\left( \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)_{ij} = \frac{\partial f_i}{\partial x_j}$$

# Example Jacobian: Elementwise activation Function

- $\mathbf{h} = f(\mathbf{z})$  如何计算  $\frac{\partial \mathbf{h}}{\partial \mathbf{z}}$
- 对于每个元素  $h_i = f(z_i)$

$$\begin{aligned}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{z}}\right)_{ij} &= \frac{\partial h_i}{\partial z_j} = \frac{\partial}{\partial z_j} f(z_i) \\ &= \begin{cases} f'(z_i) & \text{if } i = j \\ 0 & \text{if otherwise} \end{cases}\end{aligned}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \begin{pmatrix} f'(z_1) & & 0 \\ & \ddots & \\ 0 & & f'(z_n) \end{pmatrix} = \text{diag}(\mathbf{f}'(\mathbf{z}))$$

# 求梯度回顾

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{W}\mathbf{x} + \mathbf{b}) = \mathbf{W}$$

$$\frac{\partial}{\partial \mathbf{b}}(\mathbf{W}\mathbf{x} + \mathbf{b}) = \mathbf{I} \text{ (Identity matrix)}$$

$$\frac{\partial}{\partial \mathbf{u}}(\mathbf{u}^T \mathbf{h}) = \mathbf{h}^T$$

- 更多矩阵求梯度的内容可以参考《矩阵分析》课程或书籍

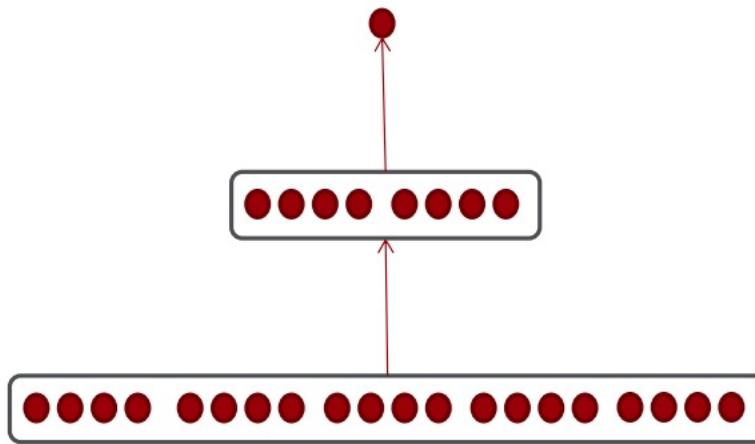
# Back to our Neural Net!

- 示例：如何计算  $\frac{\partial s}{\partial b}$

$$s = \mathbf{u}^T \mathbf{h}$$

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b})$$

$\mathbf{x}$  (input)



# Write out the Jacobians

$$s = \mathbf{u}^T \mathbf{h}$$

$$\mathbf{h} = f(\mathbf{z})$$

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

$\mathbf{x}$  (input)

$$\begin{aligned}\frac{\partial s}{\partial \mathbf{b}} &= \frac{\partial s}{\partial \mathbf{h}} \quad \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \quad \frac{\partial \mathbf{z}}{\partial \mathbf{b}} \\ &= \mathbf{u}^T \text{diag}(f'(\mathbf{z})) \mathbf{I}\end{aligned}$$

$$= \mathbf{u}^T \odot f'(\mathbf{z})$$

Useful Jacobians from previous slide

$$\frac{\partial}{\partial \mathbf{u}} (\mathbf{u}^T \mathbf{h}) = \mathbf{h}^T$$

$$\frac{\partial}{\partial \mathbf{z}} (f(\mathbf{z})) = \text{diag}(f'(\mathbf{z}))$$

$$\frac{\partial}{\partial \mathbf{b}} (\mathbf{W}\mathbf{x} + \mathbf{b}) = \mathbf{I}$$

# Backpropagation

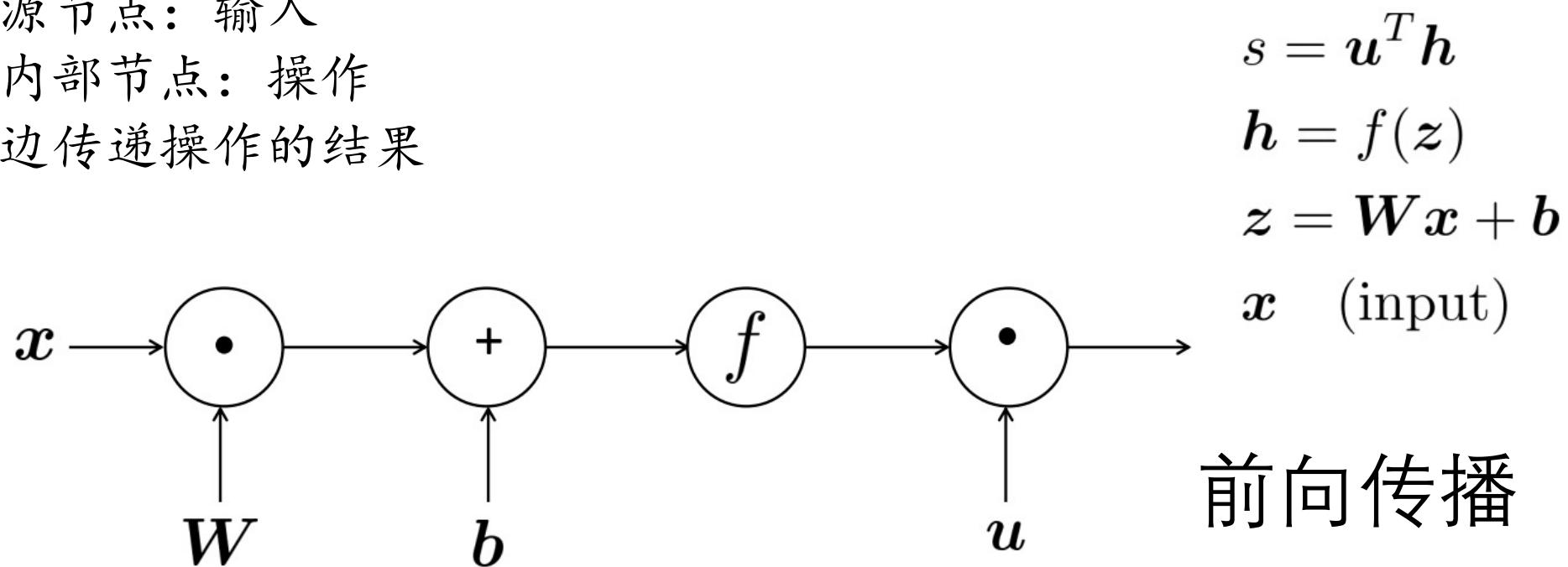
- 我们几乎已经展示了反向传播
  - 求导并使用(广义)链式法则
- 另一个技巧：在计算较低层的导数时，我们重用对较深层计算的导数，以减小计算量

$$\frac{\partial s}{\partial \mathbf{W}} = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial z} \frac{\partial z}{\partial \mathbf{W}}$$

$$\frac{\partial s}{\partial \mathbf{b}} = \frac{\partial s}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial z} \frac{\partial z}{\partial \mathbf{b}}$$

# Computation Graphs and Backpropagation

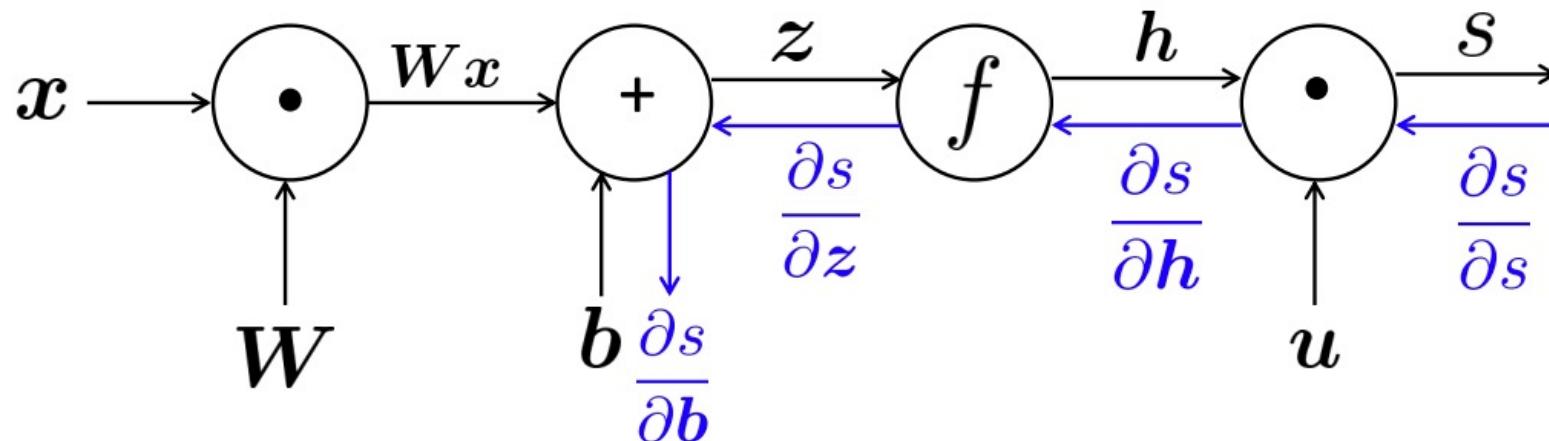
- Theano, Tensorflow, pyTorch, Dynet, Paddlepaddle, Mindspore.....
- 神经网络自动求导工具把神经网络方程表示成一个图
  - 源节点：输入
  - 内部节点：操作
  - 边传递操作的结果



# Backpropagation

- 反向传播

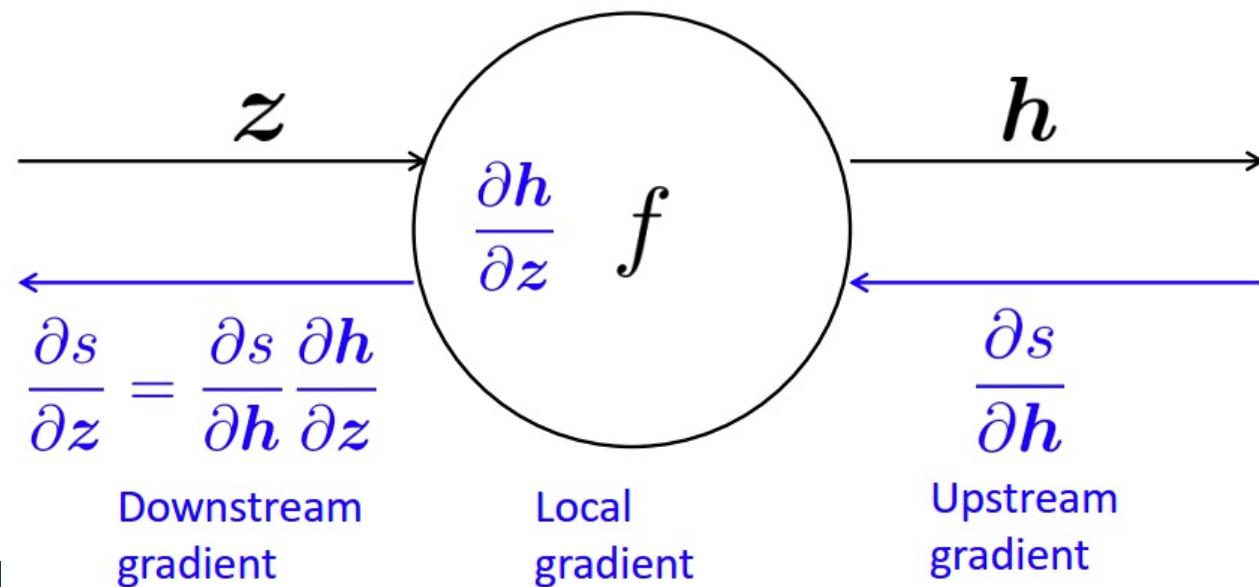
- 前向传播结束，计算loss function，然后开始对每一个训练参数求梯度
- 反向传播就是求梯度的过程



(此处做了简化，直接用最后一层 $s$ 进行了求梯度操作，实际上应利用 $s$ 与真实标注进行对比后求出loss  $L$ 再进行求梯度操作)

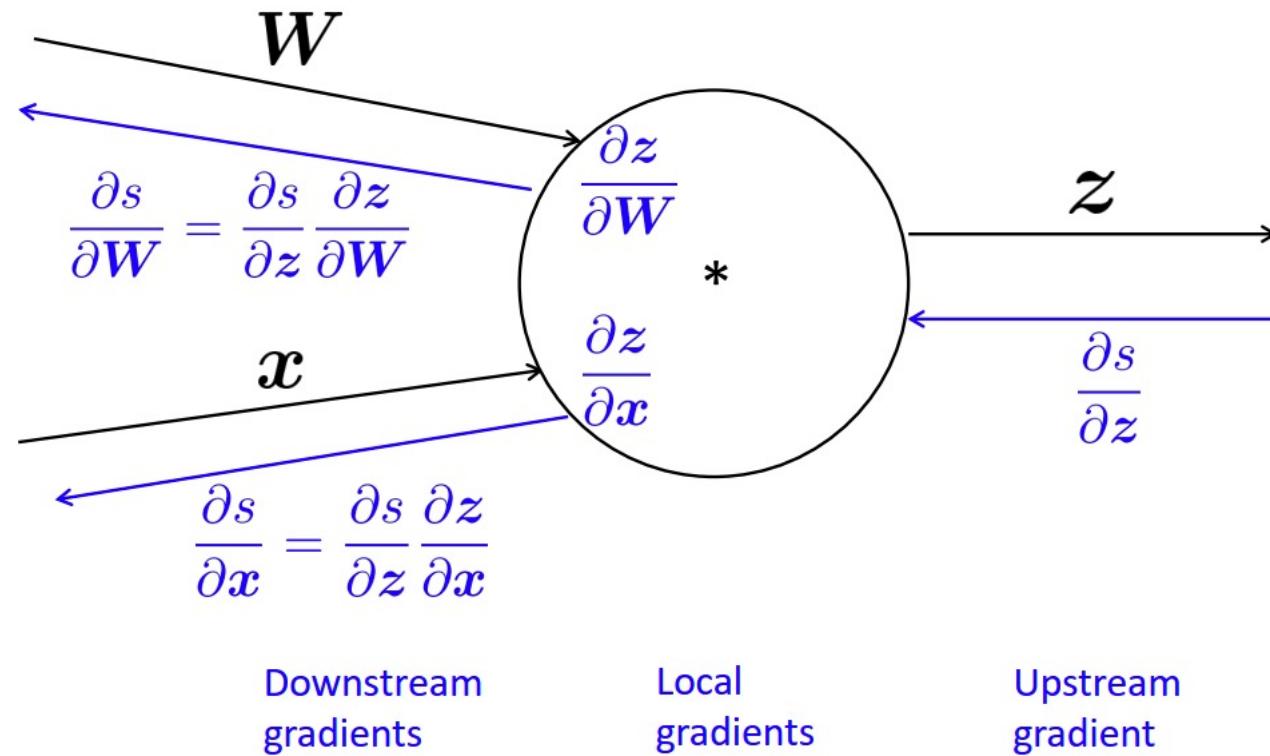
# Backpropagation: Single Node

- 节点接收“上游梯度”
  - 目标是传递正确的“下游梯度”
- 每个节点都有局部梯度 local gradient
  - 它输出的梯度是与它的输入有关
  - [downstream gradient] = [upstream gradient] x [local gradient]



# Backpropagation: Single Node

- 多个输入?
- 多个输入  $\rightarrow$  多个局部梯度



# An Example

$$f(x, y, z) = (x + y) \max(y, z)$$
$$x = 1, y = 2, z = 0$$

Forward prop steps

$$a = x + y$$

$$b = \max(y, z)$$

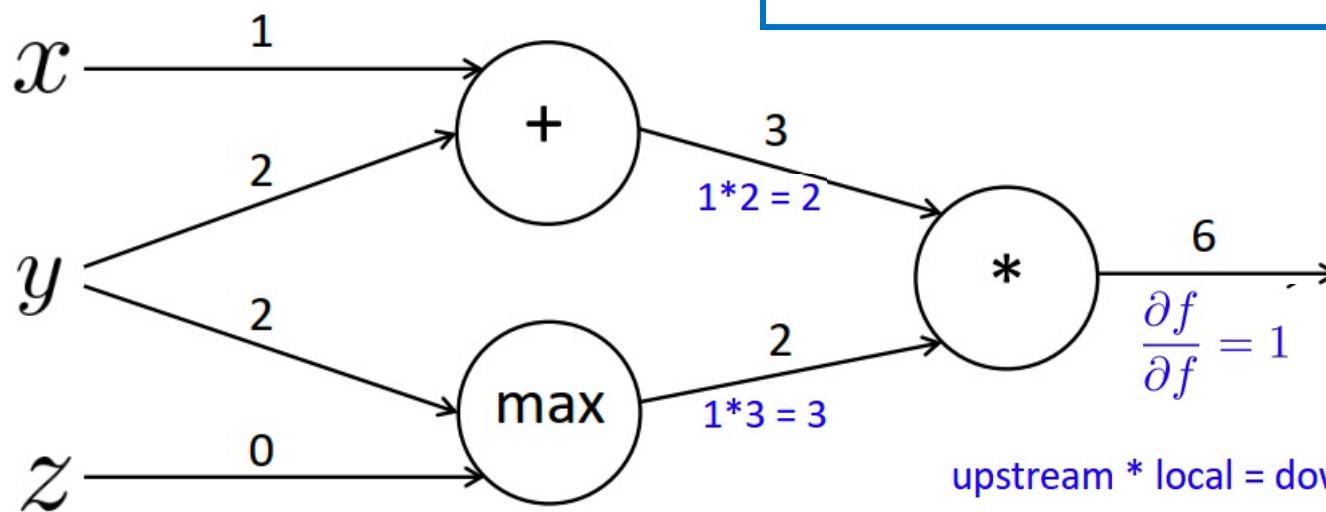
$$f = ab$$

Local gradients

$$\frac{\partial a}{\partial x} = 1 \quad \frac{\partial a}{\partial y} = 1$$

$$\frac{\partial b}{\partial y} = \mathbf{1}(y > z) = 1 \quad \frac{\partial b}{\partial z} = \mathbf{1}(z > y) = 0$$

$$\frac{\partial f}{\partial a} = b = 2 \quad \frac{\partial f}{\partial b} = a = 3$$



# An Example

$$f(x, y, z) = (x + y) \max(y, z)$$
$$x = 1, y = 2, z = 0$$

Forward prop steps

$$a = x + y$$

$$b = \max(y, z)$$

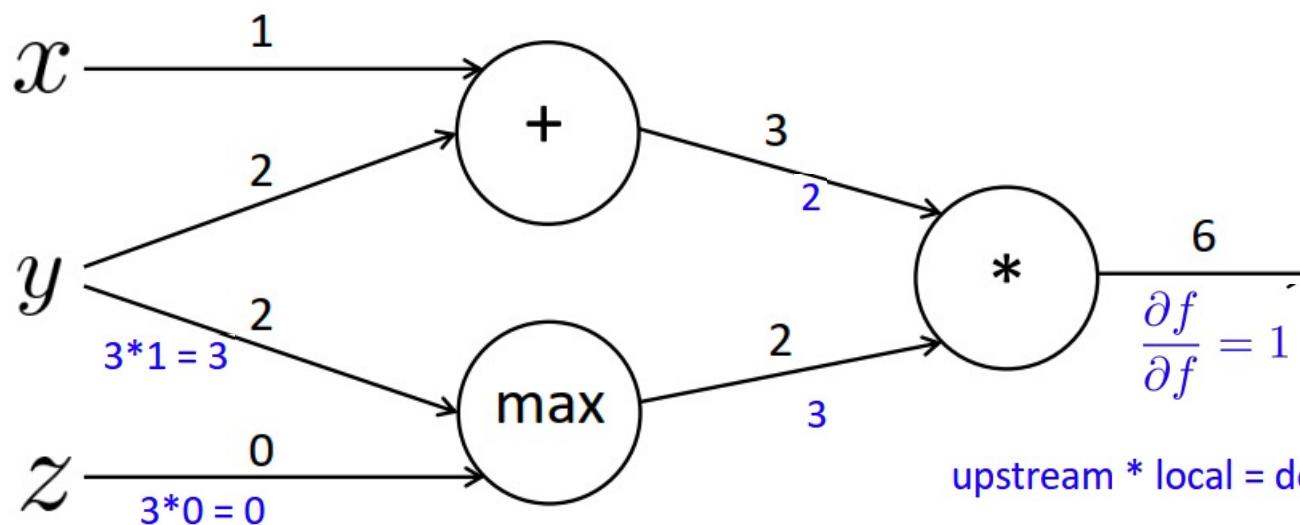
$$f = ab$$

Local gradients

$$\frac{\partial a}{\partial x} = 1 \quad \frac{\partial a}{\partial y} = 1$$

$$\frac{\partial b}{\partial y} = \mathbf{1}(y > z) = 1 \quad \frac{\partial b}{\partial z} = \mathbf{1}(z > y) = 0$$

$$\frac{\partial f}{\partial a} = b = 2 \quad \frac{\partial f}{\partial b} = a = 3$$



# An Example

$$f(x, y, z) = (x + y) \max(y, z)$$
$$x = 1, y = 2, z = 0$$

Forward prop steps

$$a = x + y$$

$$b = \max(y, z)$$

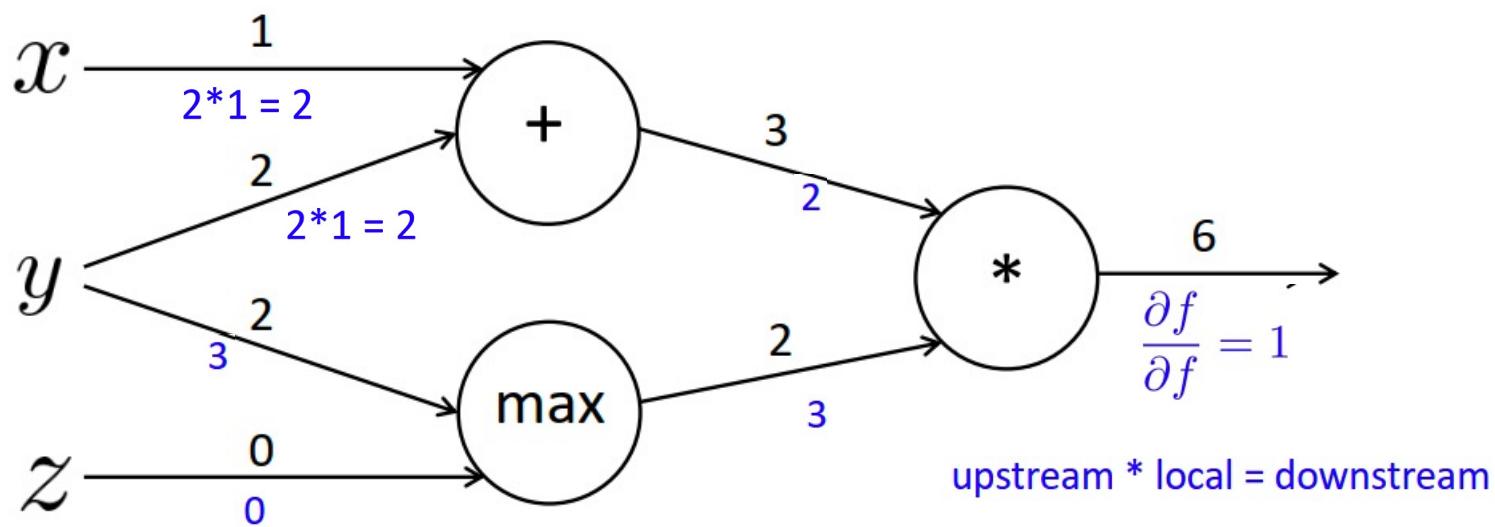
$$f = ab$$

Local gradients

$$\frac{\partial a}{\partial x} = 1 \quad \frac{\partial a}{\partial y} = 1$$

$$\frac{\partial b}{\partial y} = \mathbf{1}(y > z) = 1 \quad \frac{\partial b}{\partial z} = \mathbf{1}(z > y) = 0$$

$$\frac{\partial f}{\partial a} = b = 2 \quad \frac{\partial f}{\partial b} = a = 3$$



# An Example

$$f(x, y, z) = (x + y) \max(y, z)$$

$$x = 1, y = 2, z = 0$$

Forward prop steps

$$a = x + y$$

$$b = \max(y, z)$$

$$f = ab$$

$$\frac{\partial f}{\partial x} = 2$$

$$\frac{\partial f}{\partial y} = 3 + 2 = 5$$

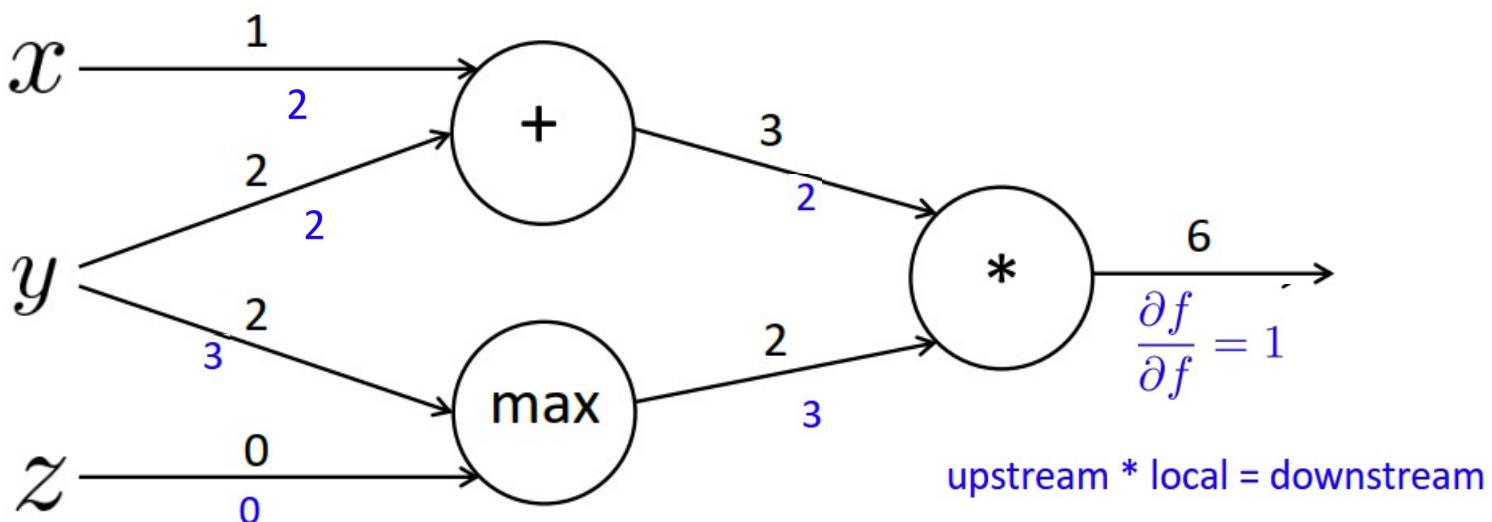
$$\frac{\partial f}{\partial z} = 0$$

Local gradients

$$\frac{\partial a}{\partial x} = 1 \quad \frac{\partial a}{\partial y} = 1$$

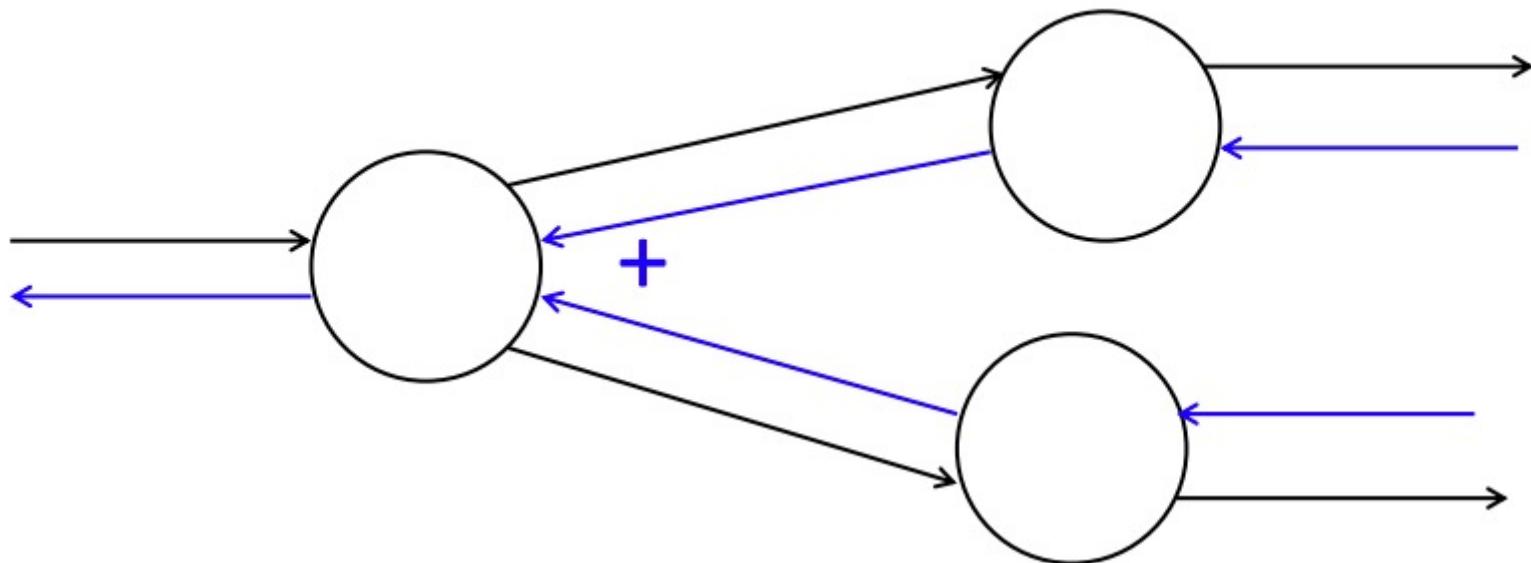
$$\frac{\partial b}{\partial y} = \mathbf{1}(y > z) = 1 \quad \frac{\partial b}{\partial z} = \mathbf{1}(z > y) = 0$$

$$\frac{\partial f}{\partial a} = b = 2 \quad \frac{\partial f}{\partial b} = a = 3$$



upstream \* local = downstream

# Gradients sum at outward branches



$$a = x + y$$

$$b = \max(y, z)$$

$$f = ab$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial a} \frac{\partial a}{\partial y} + \frac{\partial f}{\partial b} \frac{\partial b}{\partial y}$$

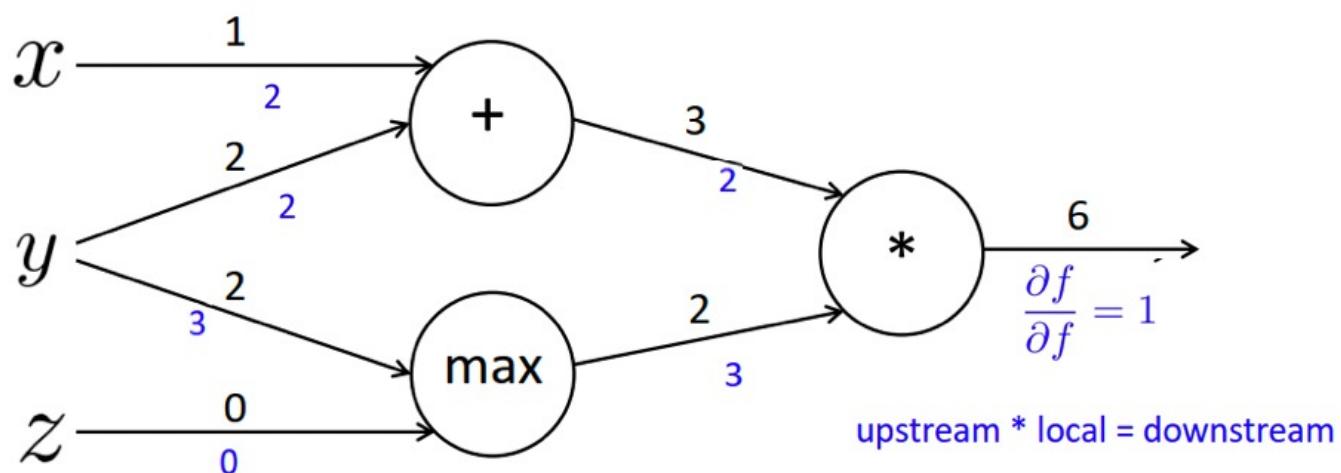
# Node Intuitions

- $+$  : “分发” 上游梯度
- $\max$  : “路由” 上游梯度，将梯度发送到最大的方向
- $*$  : “切换” 上游梯度

$$\frac{\partial f}{\partial x} = 2$$

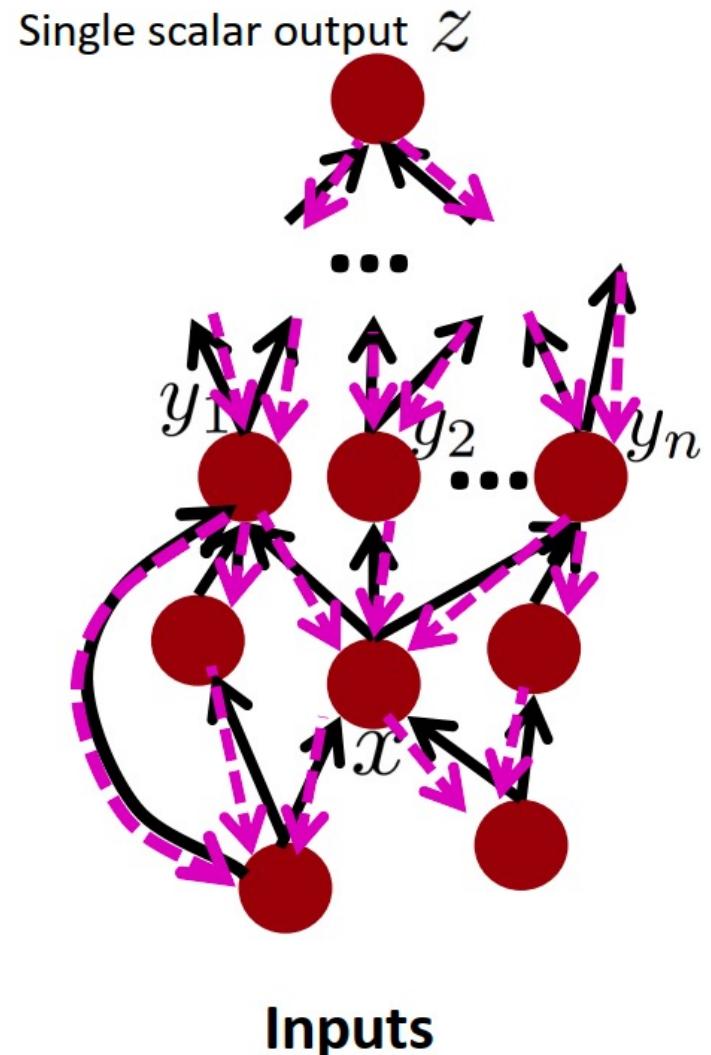
$$\frac{\partial f}{\partial y} = 3 + 2 = 5$$

$$\frac{\partial f}{\partial z} = 0$$



# Back-Prop in General Computation Graph

- 1, Fprop: 按拓扑排序顺序访问节点
  - 计算给定父节点的节点的值
- 2, Bprop:
  - 初始化输出梯度为 1
  - 以相反的顺序访问节点，使用节点的后继的梯度来计算每个节点的梯度
  - $\{y_1, y_2, \dots, y_n\}$  是  $x$  的后继
- 正确地说，Fprop 和 Bprop 的计算复杂度是一样的
- 一般来说，我们的网络有固定的层结构，所以我们可以使用矩阵和雅可比矩阵



# Why learn all these details about gradients?

- In PyTorch:
  - `loss = model(x)` → 前向计算
  - `loss.backward()` → 反向传播
  - `optimizer.step()` → 参数更新
- 为什么要学习编译器或操作系统原理?
  - 了解底层原理是很有帮助的
- 反向传播并不是时时有效的
  - 理解为什么对调试和改进模型至关重要
  - 参见 [Karpathy文章](#)
    - <https://medium.com/@karpathy/yes-you-should-understand-backprop-e2f06eab496b>
    - 未来课程的例子:爆炸和消失的梯度

Thank you