



符号学习简介

绪论

(Press ? for help, n and p for next and previous slide)

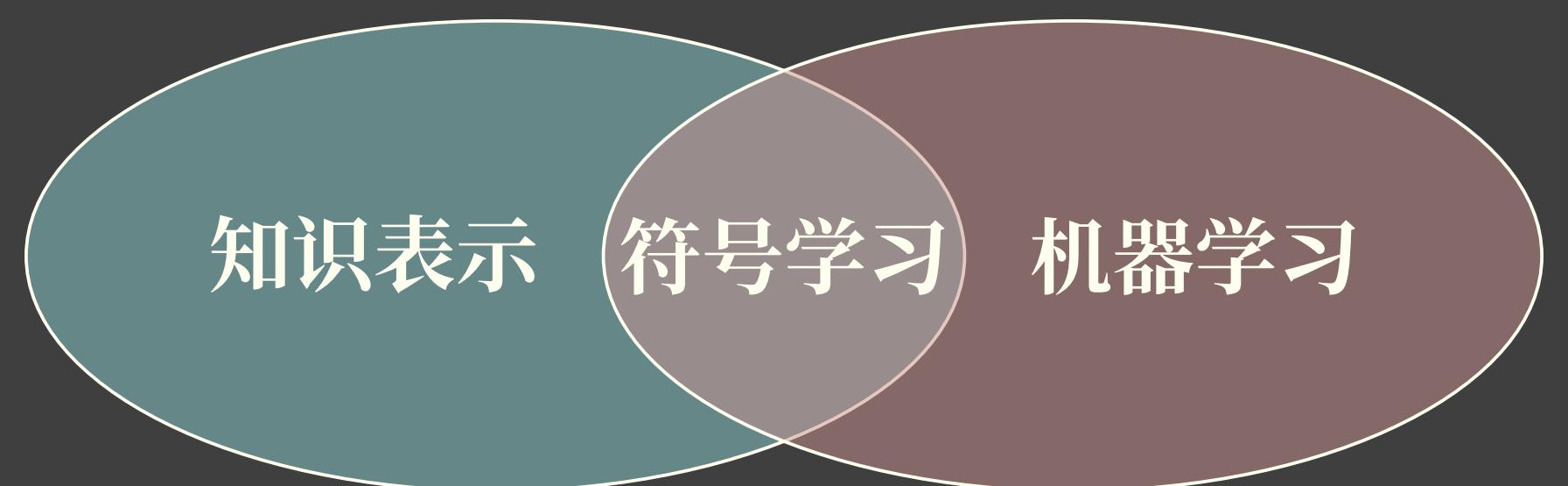
戴望州

南京大学智能科学与技术学院
2025年-秋季

<https://daiwz.net>



课程简介





课程安排

授课：戴望州（daiwz@lamda.nju.edu.cn）

时间：32学时（周三下午1、2节）

考核：大作业

经验不足，请多多反馈



课程目标

『俱往矣，数风流人物，还看今朝。』

——《沁园春·雪》

回顾符号学习的发展史，展望机器学习的未来。

具体而言，尝试回答以下几个大问题：

1. 符号主义凭什么和联结主义分庭抗礼？
2. 纯符号学习为什么曾经一蹶不振？
3. 为什么神经—符号学习重新崛起？
4. 大模型年代，符号学习未来的出路在何方？



符号学习 · 绪论

1. 为什么?
2. 是什么?
3. 怎么做?

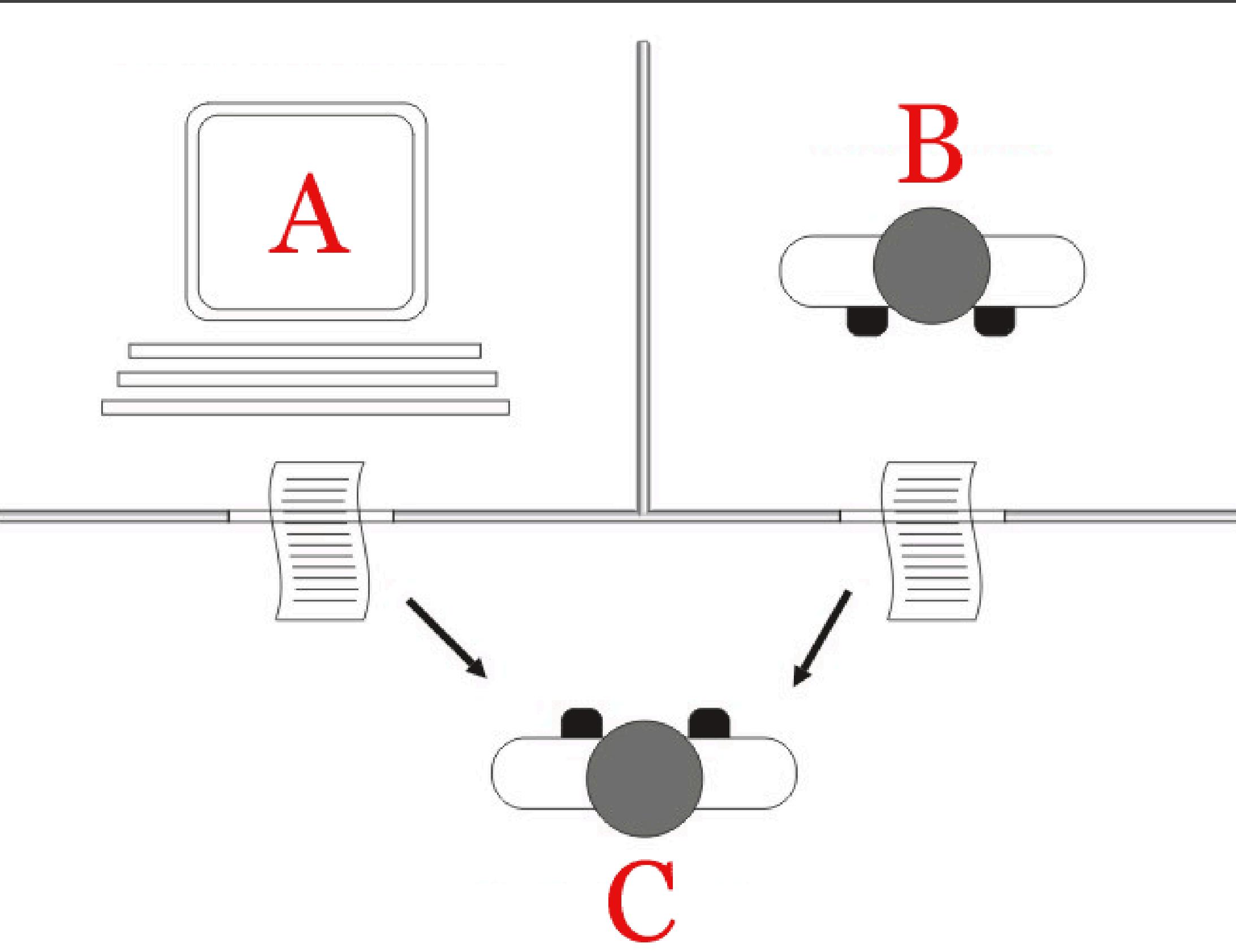


智能是一个复杂概念

- > 感知 (perception)
- > 抽象 (abstraction)
- > 逻辑 (logic)
- > 知识 (knowledge)
- > 推理 (reason)
- > 规划 (planning)
- > 创造 (creativity)
- > 学习 (learning)
- > 自觉 (self-consciousness)
- >



什么是人工智能?

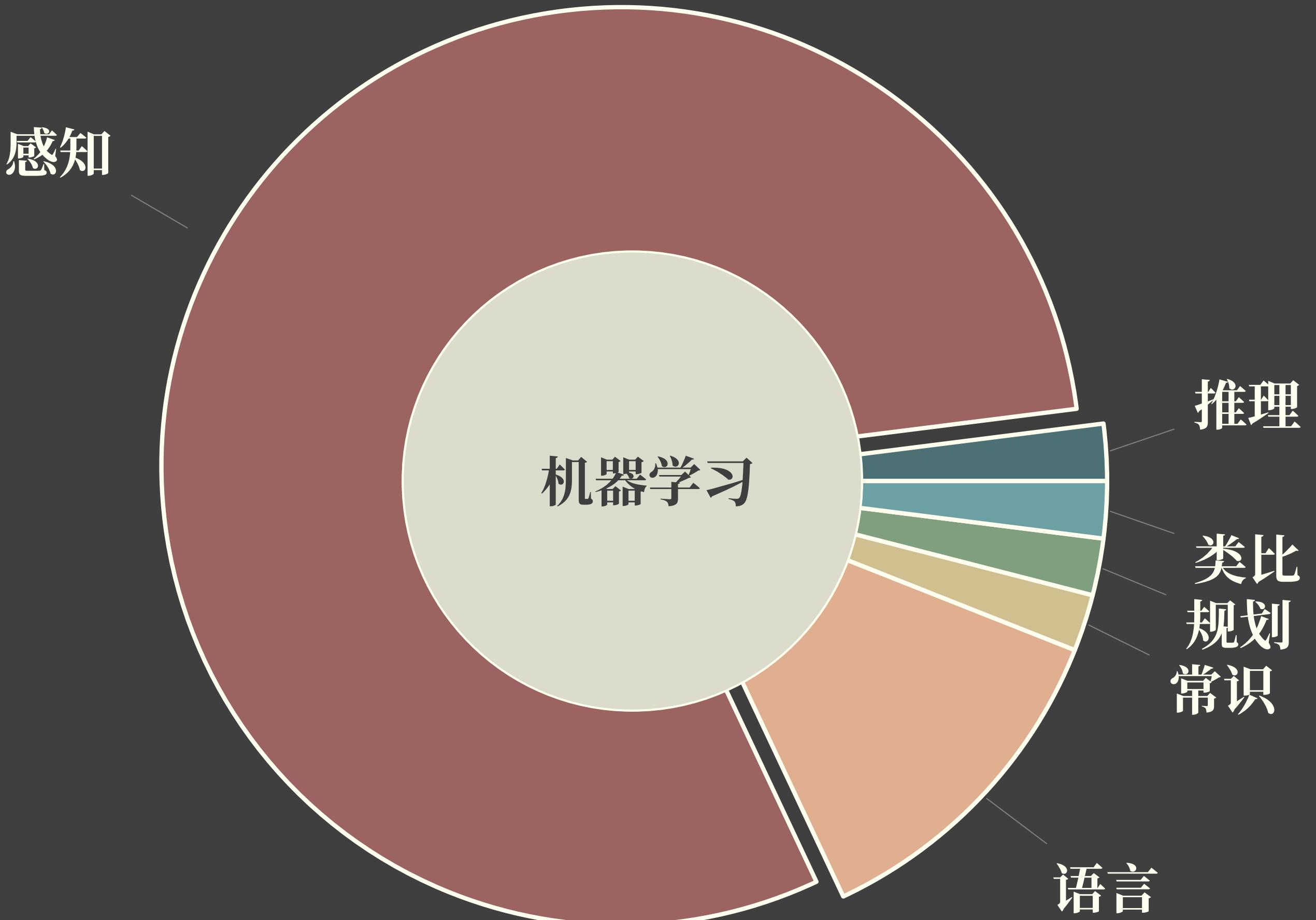




什么是人工智能?

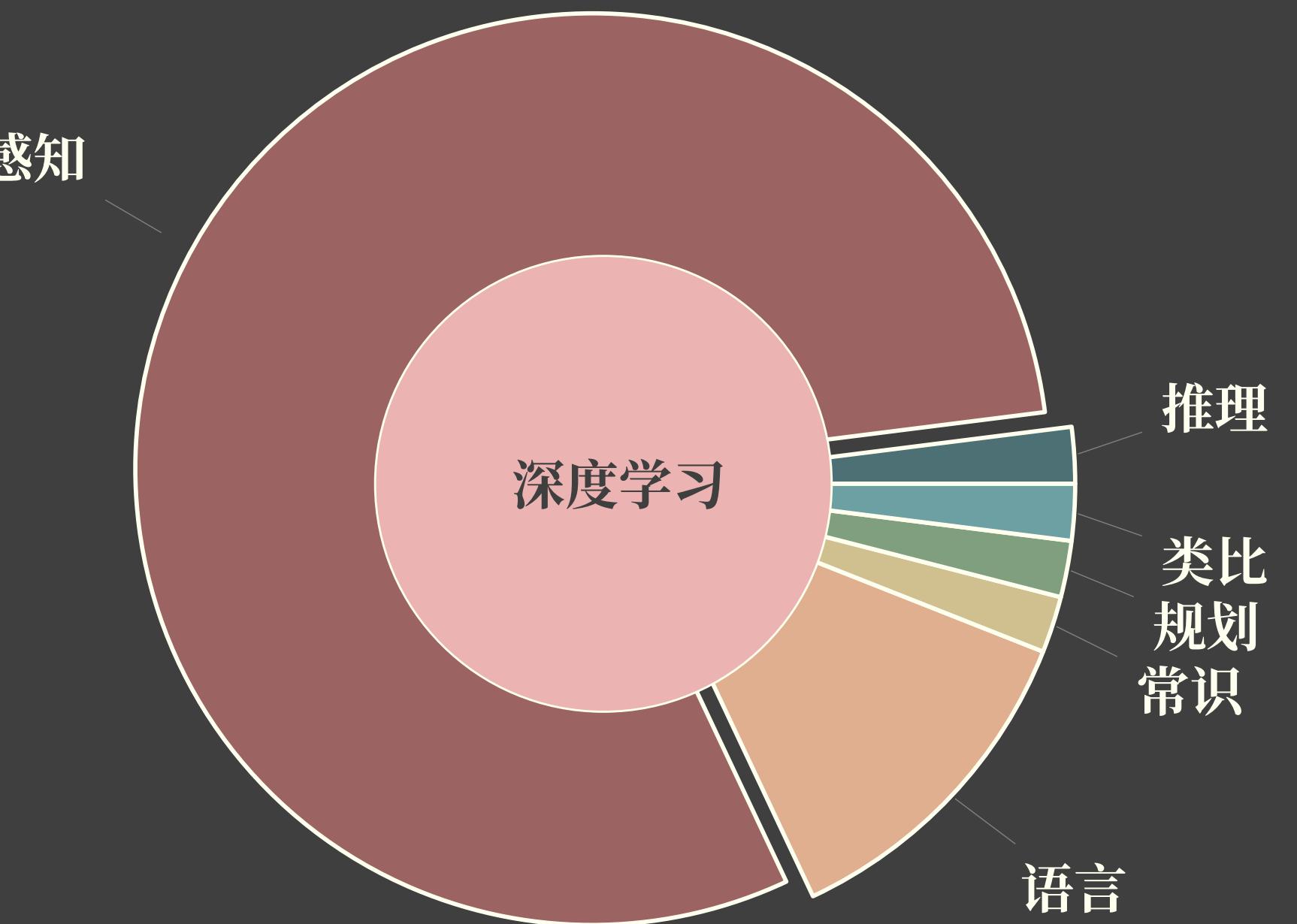


人工智能研究现状





人工智能研究之现状





DNN能/不能做什么?

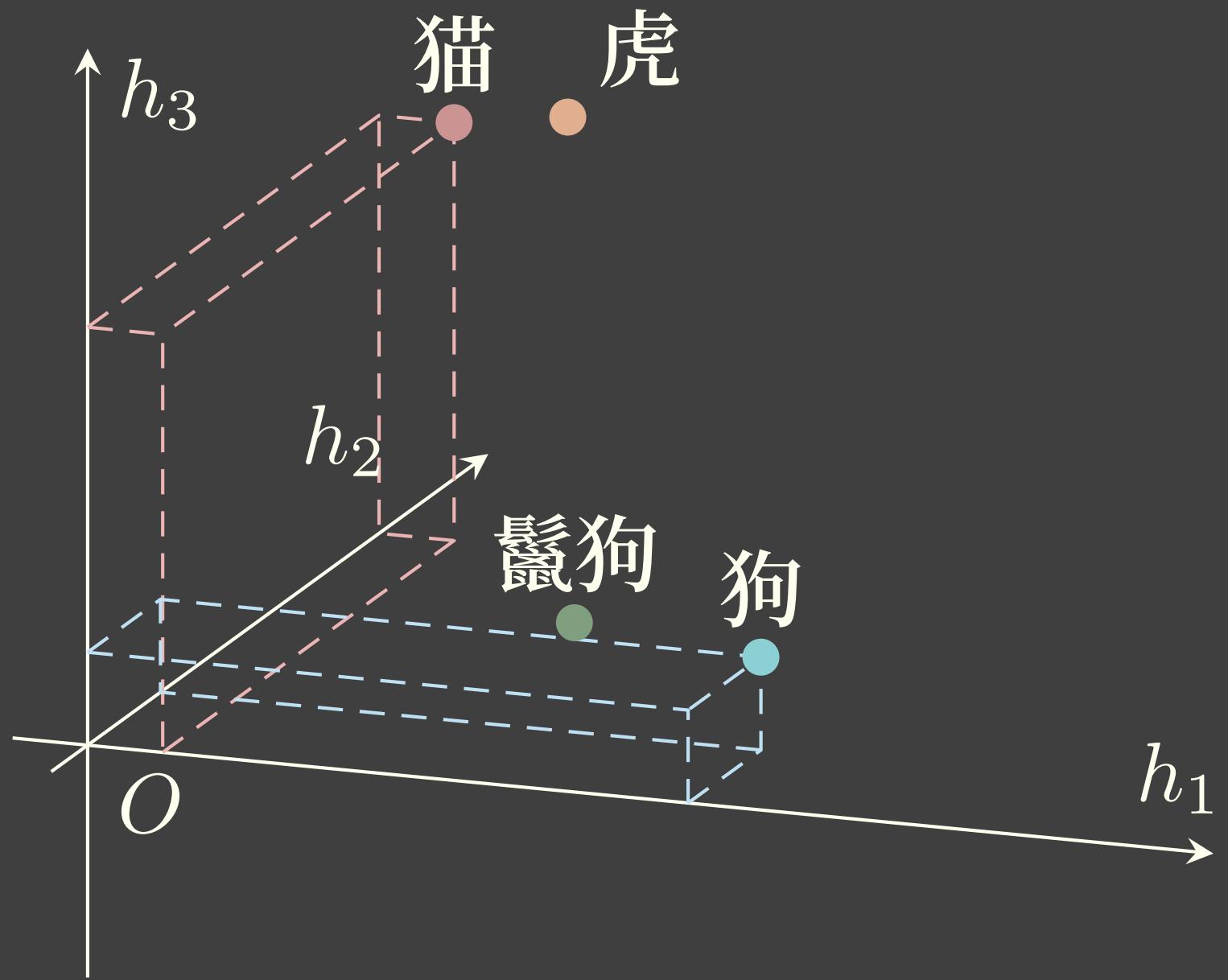
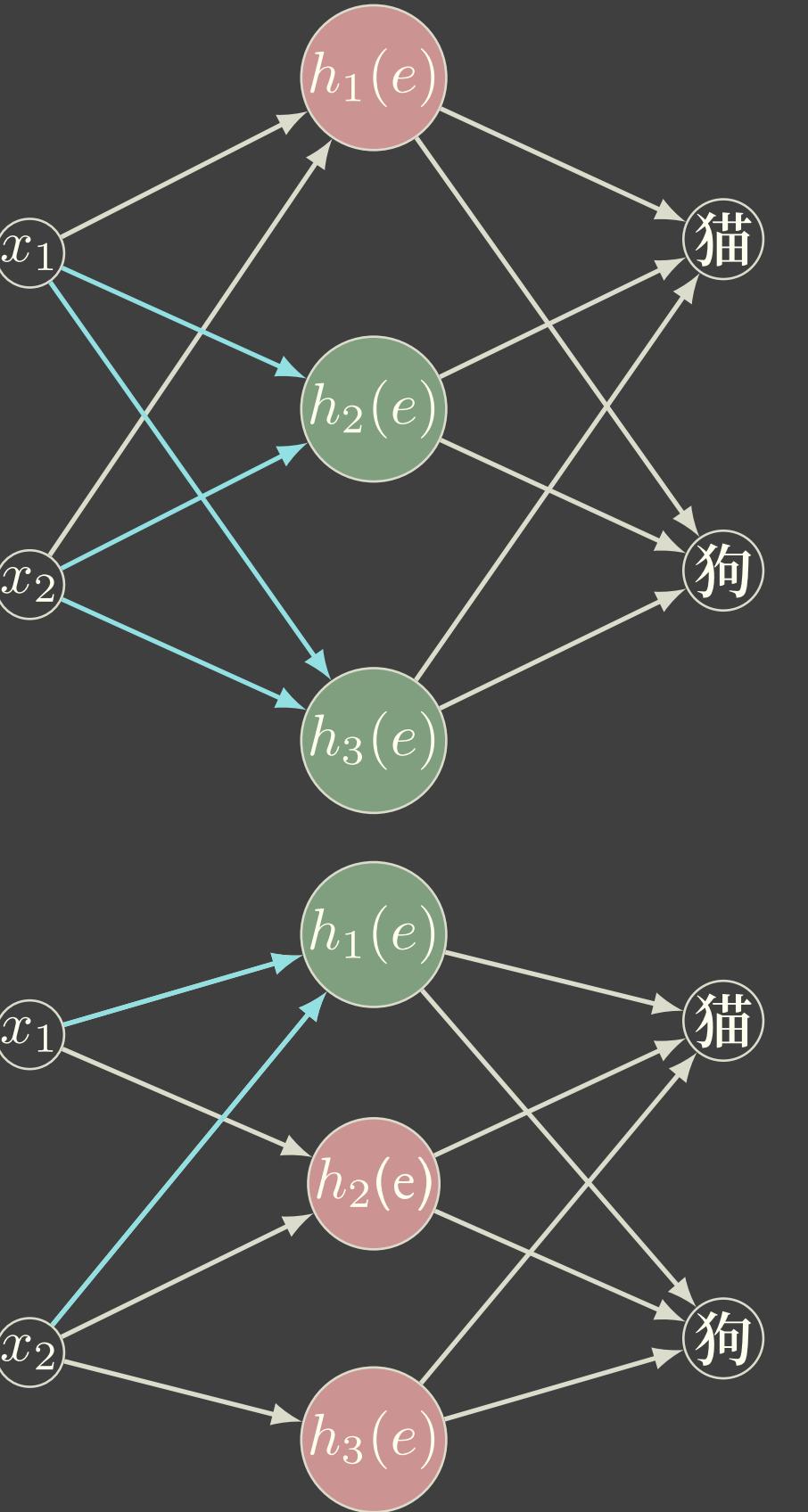
分布式表示（distributed representation）以及基于它的计算能/不能做什么？

- > Each neuron must represent something, so this must be a *local representation*.
- > “**Distributed representation**” means a *many-to-many* relationship between two types of representation (such as concepts and neurons).
 - » Each concept is represented by many neurons
 - » Each neuron participates in the representation of many concepts

Geoffrey Hinton. “Introduction to Neural Networks and Machine Learning”



分布式表示



嵌入式表示空间 (Embedding space)



分布式表示的优点

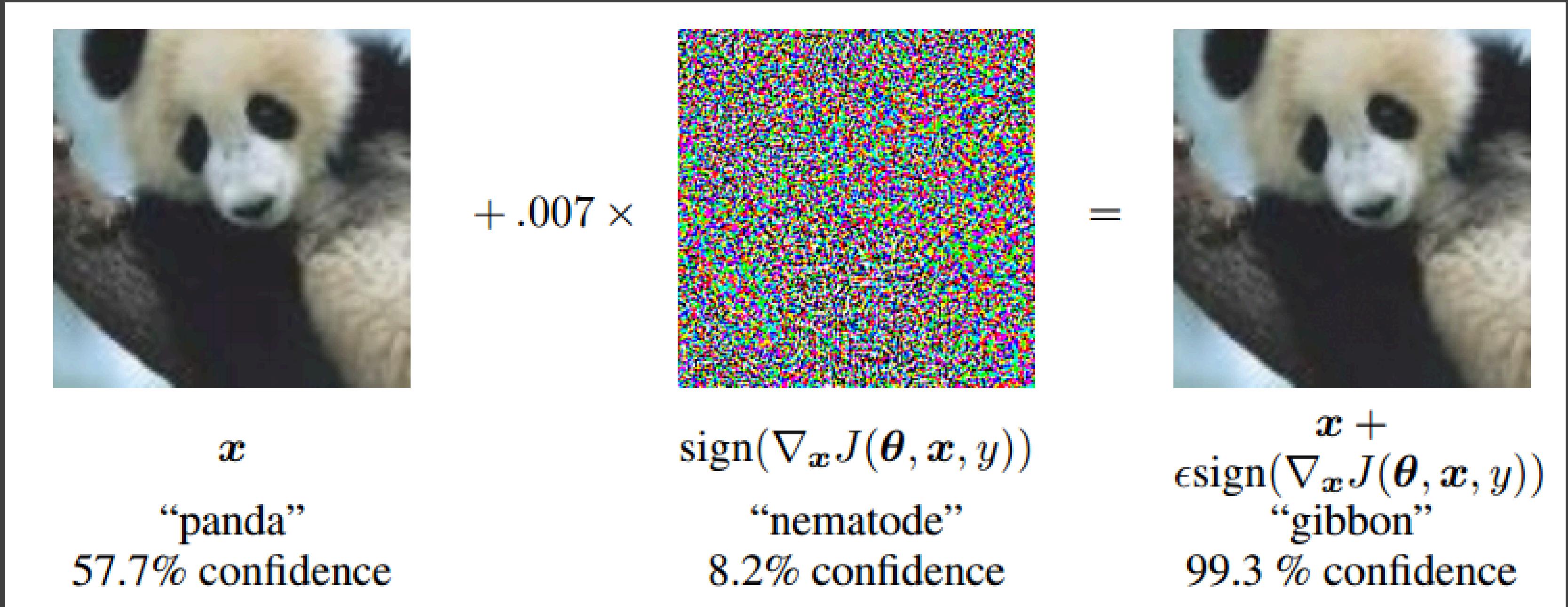
- I. 学习海量样本，用参数化的嵌入式表示空间逼近数据真实分布
 - » 大数定律
2. 学习、推理算法复杂度低，速度快
 - » 梯度下降 + 四则运算
3. 无需人工输入知识（？）
 - » 端到端建模（？）



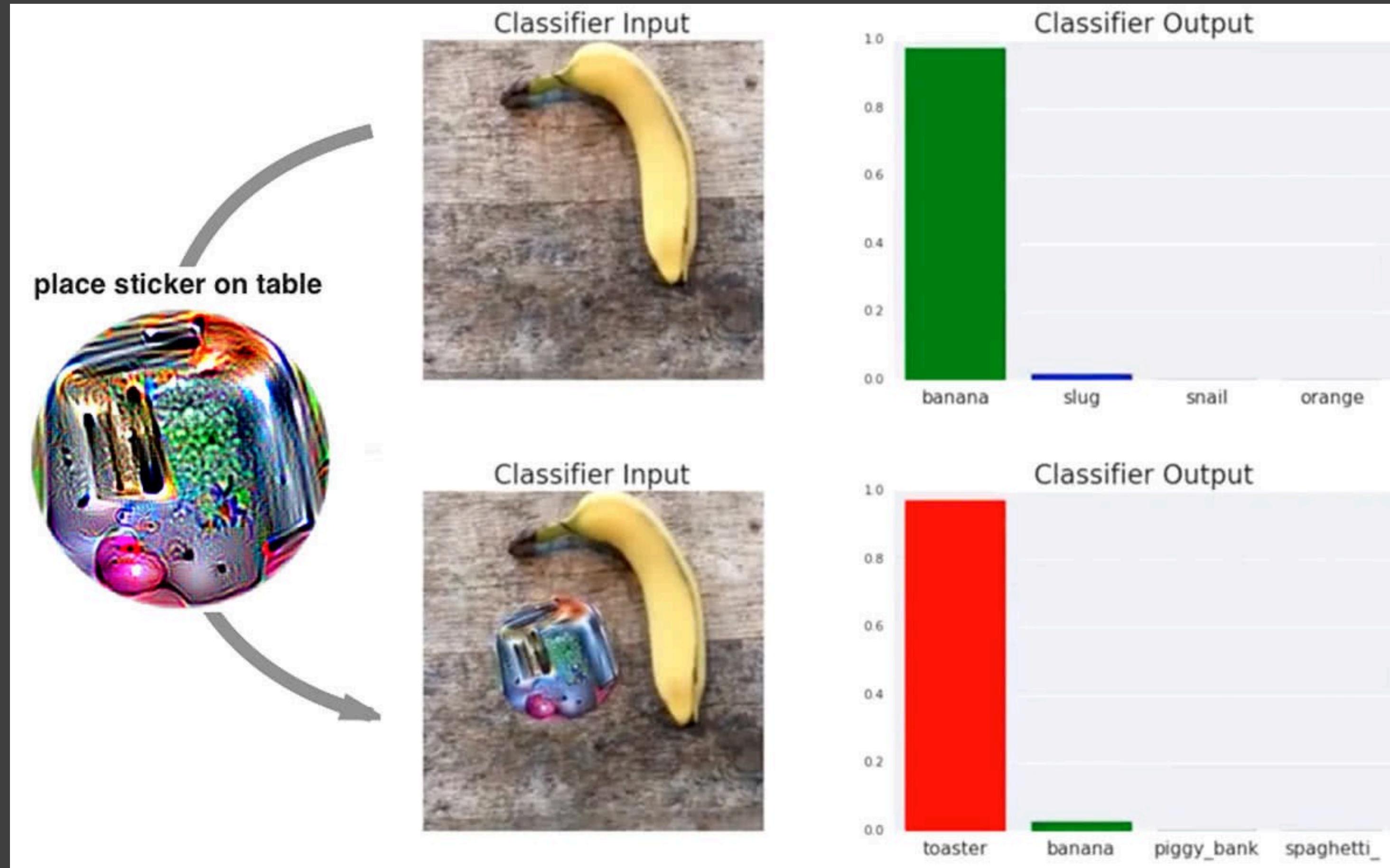
那么，还有必要采用符号表示（symbolic representation）吗？



分布式表示的不足



分布式表示的不足



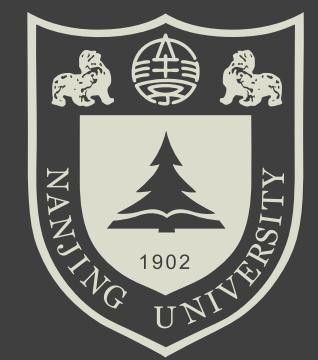


分布式表示的不足



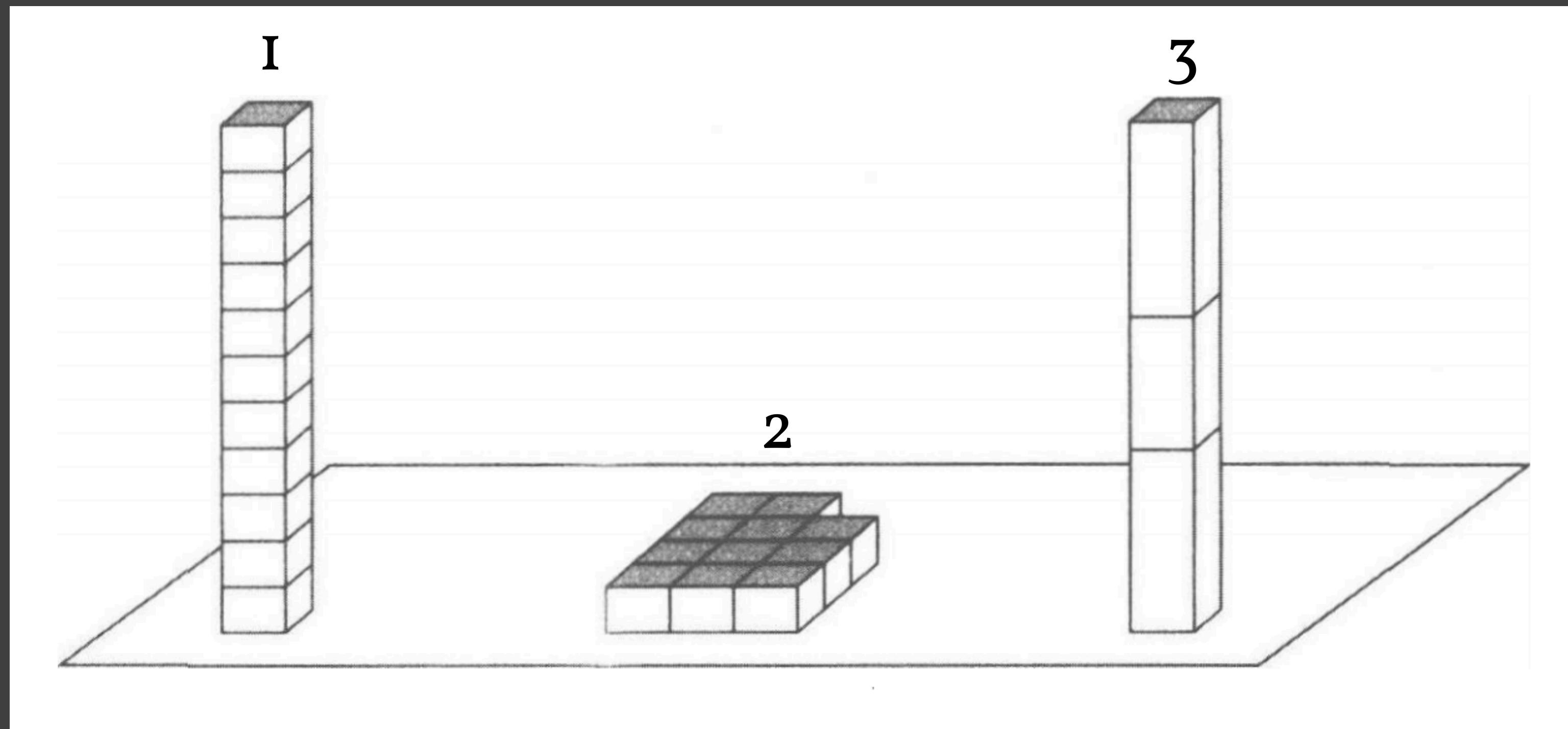
符号学习简介

<https://daiwz.net>



为什么我们不会这样？

认知科学中的猜想：人的认知分为**系统I**和**系统II**。[Kahneman, 2011]



视觉上的例子



系统I、系统II

你能多快判断出以下命题的正误？

- I. $1 + 1 = 2$ 是对的。
2. 窗外的树有些是绿色的。
3. ○○○○○ ←这里有5个圈。
4. ○○○○○○○○○○○○○○○○○○ ←这里有18个圈。
5. 若 $p \rightarrow q$ 且 $q \rightarrow r$, 那么 $p \rightarrow r$ 。
6. $1 + \frac{1}{4} + \frac{1}{9} + \cdots + \frac{1}{n^2} + \cdots = \frac{\pi^2}{6}$ 。
7. 如果教室里有一百只大象, 那么我的符号学习课就能满分。
8. “我只给自己不理发的人理发, 所以我可以给自己理发”。
9. 乌克兰在一年之内应该会停火。
10. $P = NP$ 是错的。



系统I、系统II

系统I:

- I. 直觉思维、一步到位
2. 大量训练实现肌肉记忆、熟能生巧
3. 速度快、几乎不消耗注意力
4. 失败时放弃并转向系统II

系统II:

- I. 程序思维、逐步推理
2. 必须依赖背景知识、举一反三
3. 速度慢、消耗大量注意力
4. 经常（盲目）相信系统I提供的信息
5. 过于困难时放弃并转向系统I

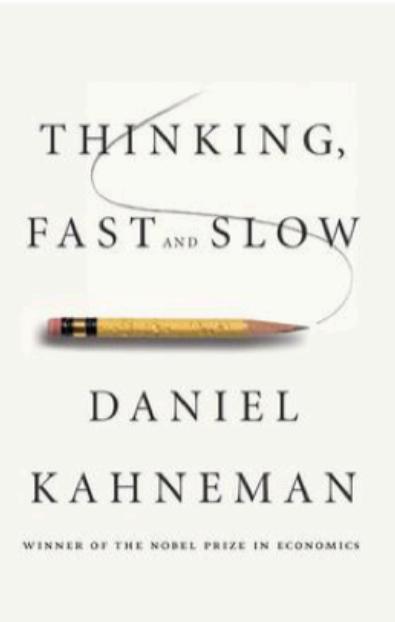


SYSTEM 1 VS. SYSTEM 2 COGNITION

2 systems (and categories of cognitive tasks):

System 1

- Intuitive, fast, **UNCONSCIOUS**, non-linguistic, habitual
- Current DL



Manipulates high-level / semantic concepts, which can be recombined combinatorially

System 2

- Slow, logical, sequential, **CONSCIOUS**, linguistic, algorithmic, planning, reasoning
- Future DL



3

Yoshua Bengio, “From System 1 to System 2 Deep Learning”



符号学习 · 绪论

1. 为什么?
2. 是什么?
3. 怎么做?



什么是符号?

- > 公孙龙：离坚白
- > 墨子：“坚白，不相外也。”
- > 毕达哥拉斯学派：万物皆数
- > 亚里士多德：目的论、Automaton
- > 莱布尼茨：“精炼我们的推理的唯一方式是使它们同数学一样切实”
- > 图灵：通用图灵机
- > (人工) 神经元的工作模式：命题逻辑演算

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. McCULLOCH and WALTER H. PITTS

Because of the “all-or-none” character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.



符号：认知解释的词汇

“知行工夫，本不可离。”

——王守仁

“What I cannot create, I do not understand.”

——Richard Feynman

科学中，探索理解的过程是解释：

- > 为什么会发生这种现象？
- > 什么该为此负责？
- > 它是怎样引起该现象？

解释就是要获得概括，而不同的词汇体现不同的概括。



符号：认知解释的词汇

假定你站在大街的一个角落，看到描述如下的一系列事件：

1. 一个行人正沿着人行道走着。
2. 这个人突然转身穿过街道。
3. 与此同时，一辆小汽车飞快地冲着这个行人疾驰而来。
4. 小汽车司机踩了急刹车。
5. 汽车打滑，冲向路边撞向电线杆。
6. 行人犹豫片刻，走到汽车司机坐位那边，看了看车里。
7. 他跑到转角处的电话亭，拨了电话9和1。

请问：出了什么事？为什么行人跑向电话亭？下一步他将做什么？为什么那样做？



符号：认知解释的词汇

请描述一下你下课以后打算做什么。

1. 这一系列行为之间有**物理上的因果性**吗？
2. 这些行为**合理**吗？
3. 这些行为具有**普遍性**吗？
4. 这些行为之间的关联**唯一**吗？
5. 你的这些描述**完全准确**吗？
6. 如果不借助任何**符号化词汇**，你还能够描述吗？



符号：认知计算的表征

一个大胆，但又不得不做的假设：**认知是一种计算**

I. “计算”具有普适性：

- » 通用图灵机
- » λ 演算
- » 原始递归函数
- » C语言

2. “计算”是一种抽象过程，与承载它的物理设备无关：

- » 大脑
- » 电子计算机
- » 量子计算机

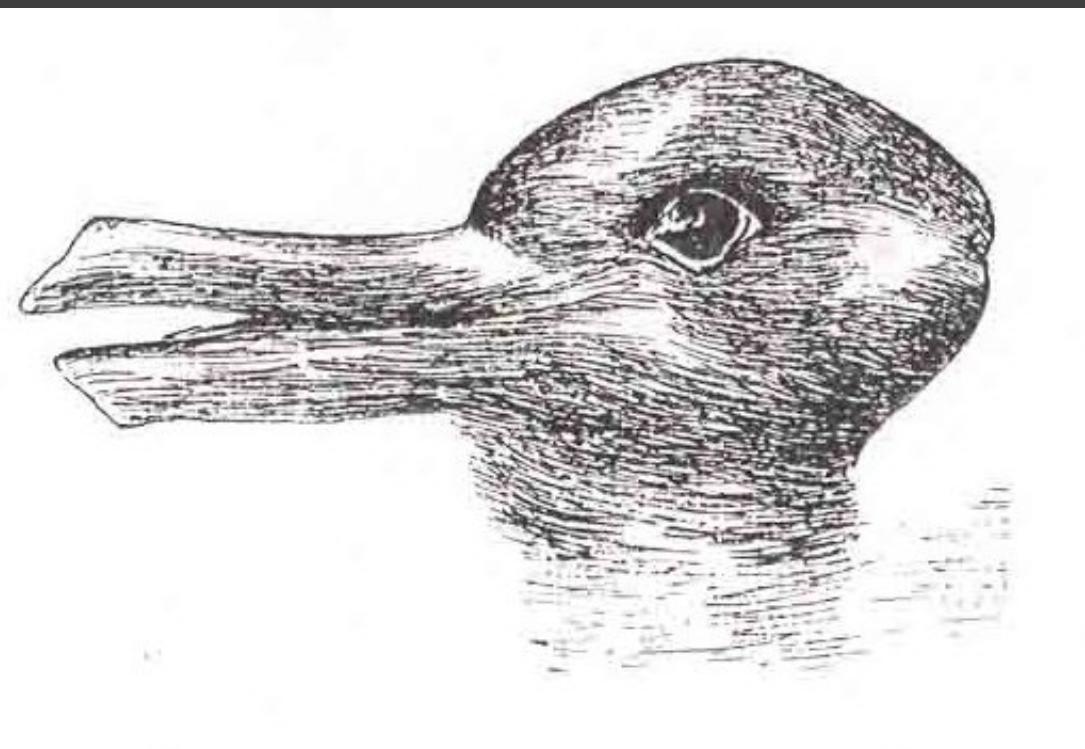
3. 我们最终希望用计算机实现一个认知体（弱/强人工智能）



符号：认知计算的表征

一个大胆，但又无法避免的结论：**认知是关于符号的计算**

1. 没有无表征的计算：包括用代码实现的DNN
2. 计算是由规则支配的过程：函数、证明、形式语言、逻辑推理
3. 所谓的黑箱——直觉、情绪乃至神经网络：输出的表征仍然是符号
4. 认知是可穿透的：可解释、可调试





符号学习的例子：程序归纳

A	B
1 Email	Column 2
2 Nancy.FreeHafer@fourthcoffee.com	nancy freehafer
3 Andrew.Cencici@northwindtraders.com	andrew cencici
4 Jan.Kotas@litwareinc.com	jan kotas
5 Mariya.Sergienko@gradicdesigninstitute.com	mariya sergienko
6 Steven.Thorpe@northwindtraders.com	steven thorpe
7 Michael.Neipper@northwindtraders.com	michael neipper
8 Robert.Zare@northwindtraders.com	robert zare
9 Laura.Giussani@adventure-works.com	laura giussani
10 Anne.HL@northwindtraders.com	anne hl
11 Alexander.David@contoso.com	alexander david
12 Kim.Shane@northwindtraders.com	kim shane
13 Manish.Chopra@northwindtraders.com	manish chopra
14 Gerwald.Oberleitner@northwindtraders.com	gerwald oberleitner
15 Amr.Zaki@northwindtraders.com	amr zaki
16 Yvonne.McKay@northwindtraders.com	yvonne mckay
17 Amanda.Pinto@northwindtraders.com	amanda pinto

```
Concatenate(ToLower(Substring(v,WordToken,1)),  
           " ", ToLower(SubString(v,WordToken,2)))
```



符号学习的直观意义

通俗来讲：符号学习就是选择“词汇”加寻找“解释”。

> “词汇”：背景知识、领域特定语言（DSL）

» FlashFill语言中的语法和关键字

» 开普勒第三定律→万有引力的发现→广义相对论

> “解释”：假设模型

» FlashFill中自动合成的程序

» 从观测中总结出来的科学定律

```
language FlashFill;

@output string e := f | Concat(f, e);
string f := ConstStr(s)
    | let string x = std.Kth(v, k) in SubStr(x, pp);
Tuple<int, int> pp := std.Pair(p, p);
int p := AbsPos(x, k) | RegPos(x, rr, k);
Tuple<Regex, Regex> rr := std.Pair(r, r);

@values[FlashFill.Semantics.StaticTokens] Regex r;
@input string[] vs;      string s;      int k;

string Concat(string f, string e) { return f + e; }      (b)
string ConstStr(string s) { return s; }
string SubStr(string x, Tuple<int, int> pp) {
    int l = pp.Item1, r = pp.Item2;
    return (l < 0 || r > x.Length) ? null : x.Substring(l, r-l);
}
int AbsPos(string x, int k) { return k < 0 ? x.Length+k+1 : k; }
int? RegPos(string x, Tuple<Regex, Regex> rr, int k) {
    Regex r = new Regex("(?<=" + rr.Item1 + ")" + rr.Item2);
    MatchCollection ms = r.Matches(x);
    int i = k > 0 ? (k - 1) : (k + ms.Count);
    return (i < 0 || i >= ms.Count) ? null : ms[i].Index;
}
static readonly Regex[] StaticTokens = new[] {
    new Regex("\d+"), new Regex("[a-zA-Z]+"), /*more tokens... */
}
```

```
(a)   Grammar ff = new Grammar("FlashFill.grammar");
ProgramNode p = ff.ParseAST(@"let x = std.Kth(v, 0) in
    SubStr(x, RegPos(x, std.Pair(new Regex(""),
        new Regex(@""[A-Z]+"))), -1),
    AbsPos(-1));
State input = new State(ff.Symbol("vs"),
    new string[] {"Leslie Lamport"});
Assert.Equals(p.Invoke(input), "Lamport");

(c)
language FlashExtract.Text;
using namespace std; using namespace std.list;

StringRegion[] N1 :=
    Map(λx ⇒ Pair(p, p), Ls)           //LinesMap
    | Map(λt ⇒ let string x = GetSuffix(d, t) in
        Pair(t, p), Ps)               //StartSeqMap
    | Map(λt ⇒ let string x = GetPrefix(d, t) in
        Pair(p, t), Ps);             //EndSeqMap
int[] Ps := FilterInt(i0, k, RRs);
int[] RRs := RegexMatches(d, rr);
StringRegion[] Ls := FilterInt(i0, k, FltLs);
StringRegion[] FltLs := Filter(λℓ ⇒ B, AllLs);
StringRegion[] AllLs := SplitLines(d);

(d)
@extern[FlashFill] int p;      //FV(p) = {x: string}
@extern[text.matches] bool B; //FV(B) = {ℓ: StringRegion}
@input StringRegion d;       int i0;      int k;
```



符号学习的形式化定义

通过机器学习算法，从样本 E 中学会一个符号模型 H ，使得在给定背景知识 B 的情况下：

$$B \cup H \vdash E.$$

1. 这里的“ \vdash ”是一个通用计算过程，表示“**有限步推出**”
2. B 的表征形式多种多样，一般与 H 和 E 相同



符号学习的特点

表征	分布式/向量	符号
样本量	许多	较少
数据形式	表格	灵活
模型形式	固定形式	通用程序
可解释性	较难	较简单
知识迁移	较难	较简单
复杂度	较低	较高

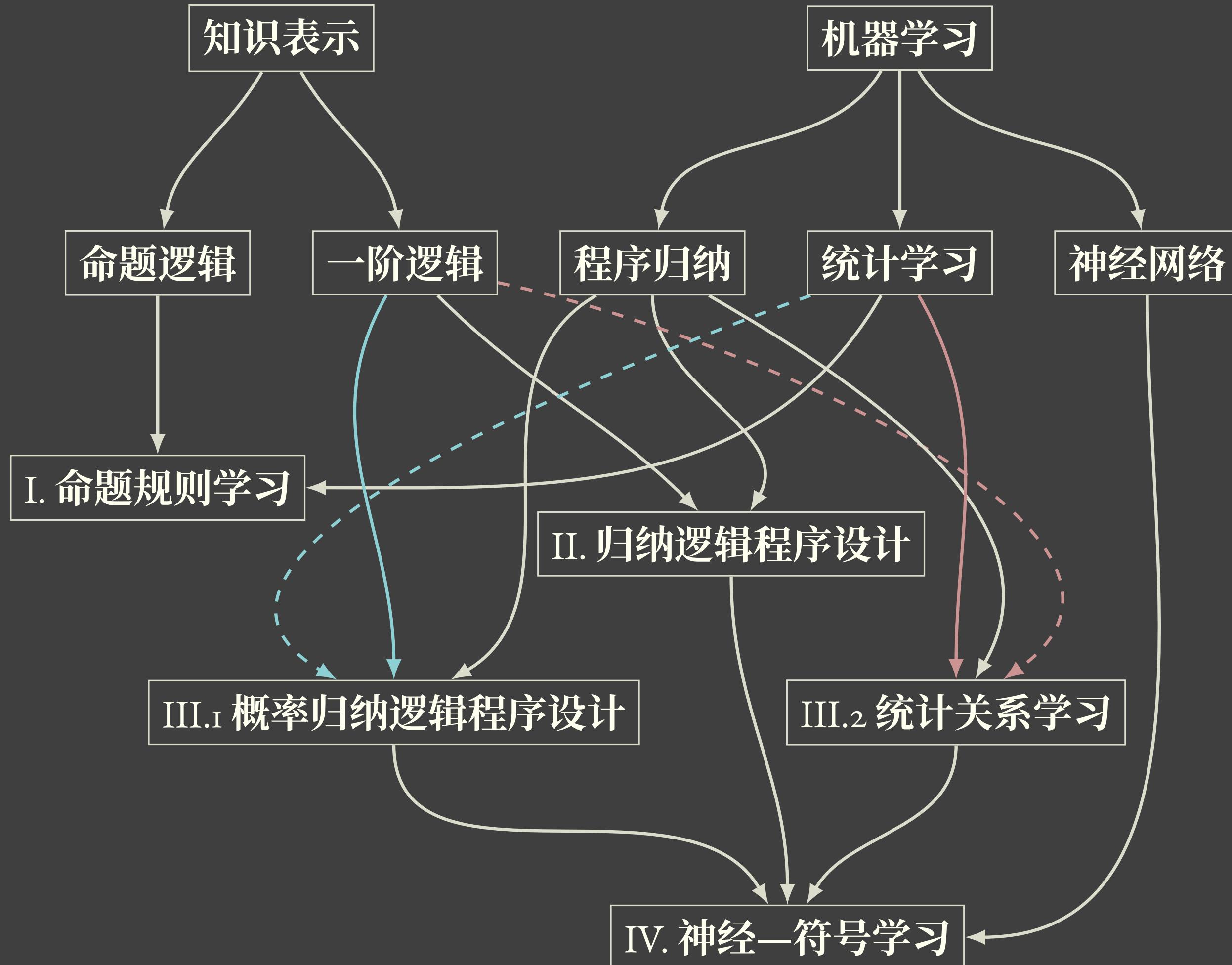


符号学习 · 绪论

1. 为什么?
2. 是什么?
3. 怎么做?



路径图





参考书目

命题规则学习、归纳逻辑程序设计：

1. 周志华.《机器学习》.清华大学出版社.
2. Johannes Fürnkranz, Dragan Gamberger and Nada Lavrač. *Foundations of Rule Learning*. Springer.
3. John Wylie Lloyd. *Foundations of Logic Programming*. Springer.

概率归纳逻辑程序设计、统计关系学习：

1. Luc de Raedt, Paolo Frasconi, Kristian Kersting and Stephen Muggleton. *Probabilistic Inductive Logic Programming*. Springer.
2. Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press.

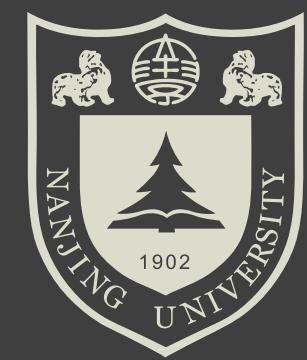
神经—符号学习：

1. **New!** Pascal Hitzler, Md Kamruzzaman Sarker. *Neuro-Symbolic Artificial Intelligence: The State of the Art*. IOS Press.

符号学习简介

<https://daiwz.net>

会议



IJCLR 2021

Home CfP Dates Program Accepted Papers Awards Keynote Speakers Registration Organization

1st International Joint Conference on Learning & Reasoning

25 - 27 October 2021
Virtual

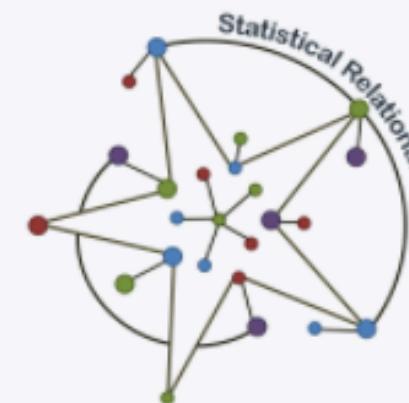
ILP 2020-21
30th International Conference on Inductive Logic Programming



NeSy 2020-21
15th International Workshop on Neural-Symbolic Learning and Reasoning



StarAI 2020-21
10th International Workshop on Statistical Relational AI



AAIP 2020-21
10th International Workshop on Approaches and Applications of Inductive Programming



POPPER



<https://github.com/logic-and-learning-lab/Popper/tree/main/examples/kinship-ancestor>