自然语言处理

2022年秋季

黄河燕、鉴萍

北京理工大学 计算机学院

hhy63, pjian@bit.edu.cn

知识体系、问题及方法论

(二) 与知识工程、与机器学习

黄河燕，鉴萍

北京理工大学 计算机学院

hhy63, pjian@bit.edu.cn

大纲

- NLP的几个问题
- NLP的几个研究范式
- NLP与语言学
- NLP与知识工程 ——留到课程最后做知识图谱的专题
- NLP与机器学习
- NLP与贝叶斯理论和概率图模型
- NLP与信息论

大纲

- NLP的几个问题
- NLP的几个研究范式
- NLP与语言学
- NLP与知识工程
- NLP与机器学习
- NLP与贝叶斯理论和概率图模型
- NLP与信息论

NLP与机器学习

- NLP的第二、第三范式均建立在机器学习基础上，系统性能极大程度上取决于数据和机器学习模型的质量

不“学习”，只需人来设定规则 (规则系统)

词的情感打分加和平均，得到句子的情感打分

	A	B
1	word	strength
2	百分之百	3
3	倍加	3
4	备至	3
5	不得了	3

full_pos_dict_sougou

1 清莹
2 轻倩
3 晴丽
4 求索
5 热潮

full_neg_dict_sougou

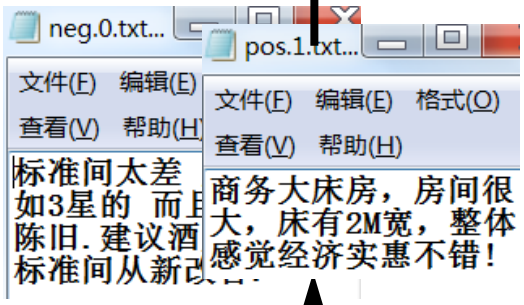
1 脏乱
2 糟报
3 早衰
4 责备
5 贼眼

查情感词典

这 汽水 太凉 。

有一丝“学习”的意味，人来定义模型(概率模型)

$$C^* = \underset{c}{\operatorname{argmax}} P(C_i|W)$$



标注语料中查$P(w_k|C)$

$$P(C|W) = \frac{P(W|C)P(C)}{P(W)}$$

$$= \frac{\prod_k P(w_k|C)\,P(C)}{P(W)}$$

Naïve Bayes

这 汽水 太凉 。

“学习”占有了更大比重，人来定义模型和特征(统计学习模型)

Learning:

Label Feature sequence 词汇特征 词性特征 句长特征

0 [0 0 1 0 0 0…, 0 0 0 0 1 0…, 1 0 0 0 0 0…, …; 0 0 0 1, 0 1 0 0, …; 0 1; …….]

1 [1 0 0 0 0 0…, 0 1 0 0 0 0…, 0 0 0 0 1 0…, …; 0 0 0 1, 1 0 0 0, …; 1 0; …….]

1 [1 0 0 0 0 0…, 0 0 0 0 0 0…, 0 0 0 0 0 1…, …; 0 1 0 0, 0 0 1 0, …; 1 0; …….]

0 …

…… 送入SVM等统计机器学习模型

Inference:

Feature sequence：

[0 0 0 1 0 0…, 0 0 0 0 1 0…, 0 0 1 0 0 0…, …; 0 0 0 1, 0 1 0 0, …; 0 1; …….]

⇩

Label ？

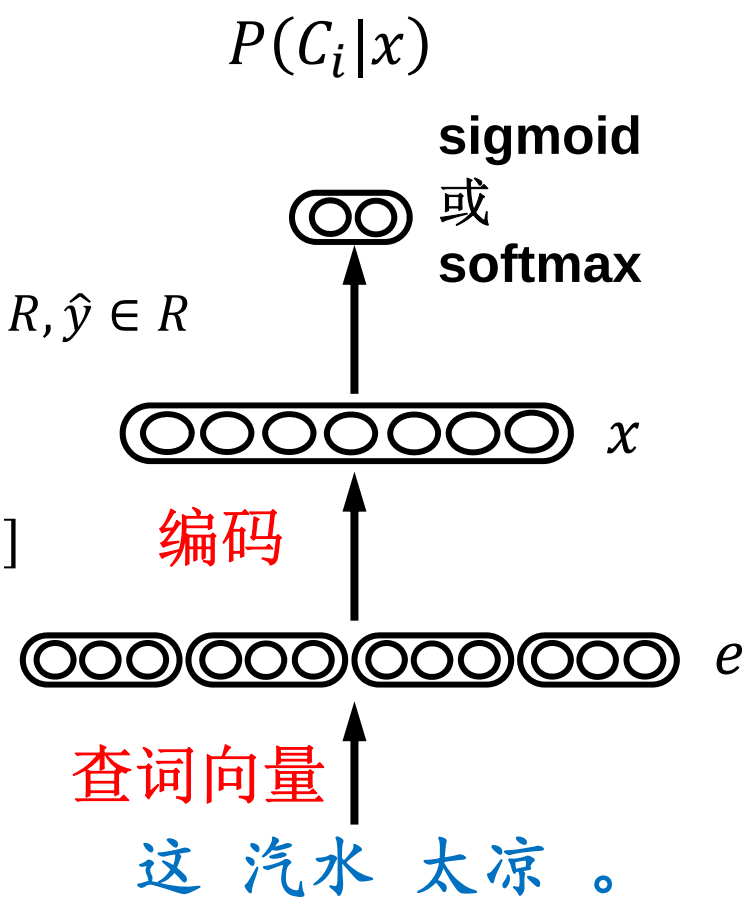
“学习”非常重要，人来定义架构和objective (深度学习模型)

$$\text{类别} = \begin{cases} 0, & \hat{y} < 0.5 \\ 1, & else \end{cases}$$

$$\hat{y} = sigmoid\left(b + \sum_{i=0}^{m-1} W_i * x_i\right), \quad W \in R^m, b \in R, \hat{y} \in R$$

$$x = \sum_{i=1}^{4} e_i, \; x \in R^m, x = [x_0, x_1, \ldots, x_{m-1}]$$

$$e_1 \quad e_2 \quad e_3 \quad e_4$$
$$e_i \in R^m, e_i = [e_{i0}, e_{i1}, \ldots, e_{i(m-1)}]$$



机器学习

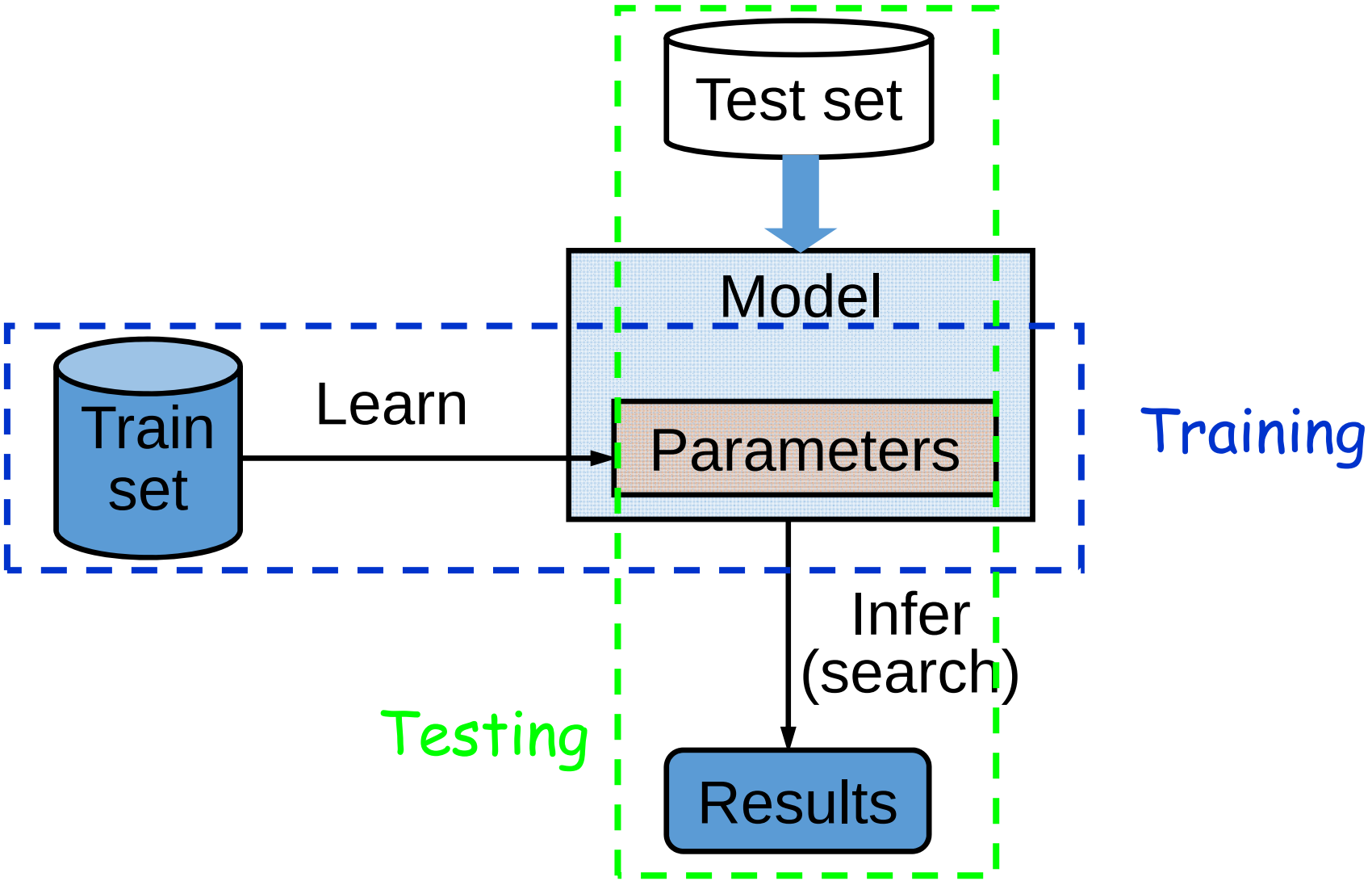
- 机器学习基础
- 人工神经网络概述
- 深度学习基础
- 典型神经网络模型

机器学习

- 机器学习基础
  - 机器学习
  - 机器学习示例——线性回归
  - 机器学习的一个主要挑战
  - 机器学习基本问题
  - 机器学习中的非线性问题
- 人工神经网络概述

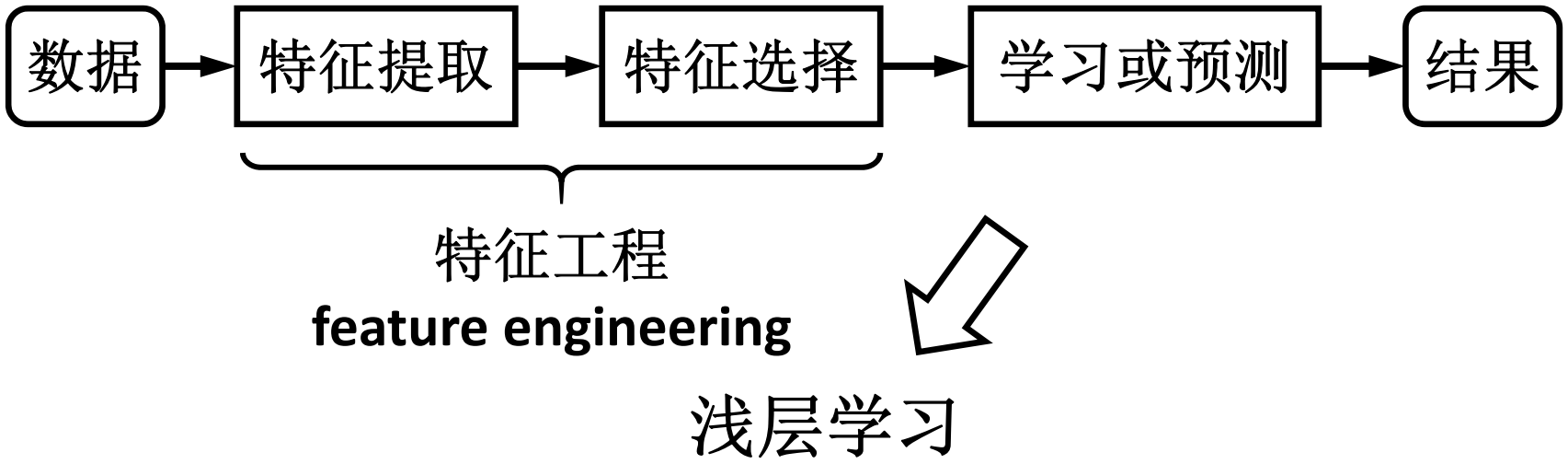
机器学习基础

- 机器学习(machine learning)
  - 从有限的观测数据中学习出具有一般性的规律，并将这些规律应用到未观测样本上的方法——用数据编程
  - 是一门讨论各式各样的适用于不同问题的函数形式，以及如何使用数据来有效地获取函数参数具体值的学科
  - Modeling, learning, inference
  - Feature


Test set
Model
Train set
Learn
Parameters
Training
Infer (search)
Testing
Results

机器学习基础

传统机器学习



机器学习基础

机器学习类型

- 监督学习(supervised learning)
  - 观察$(\boldsymbol{x}, \boldsymbol{y})$，从$\boldsymbol{x}$预测$\boldsymbol{y}$，通常是估计$p(\boldsymbol{y}|\boldsymbol{x})$
  - 逻辑回归，**SVM**，决策树，***k*-**最近邻
- 无监督学习(unsupervised learning)
  - 观察$\boldsymbol{x}$的多个样本，隐式或显式学习$p(\boldsymbol{x})$
  - **PCA**，***k*-**均值聚类
  - 目前的热点是自监督学习
- 强化学习(reinforcement learning)

机器学习基础

机器学习示例——线性回归

- 向量$\boldsymbol{x} \in \mathbb{R}^n$为输入，预测标量$y \in \mathbb{R}$为输出，假设输出是其输入的线性函数，预测值

$$\hat{y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

- 度量模型性能：

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{m}\left\|\hat{y}^{(\mathrm{test})} - y^{(\mathrm{test})}\right\|_2^2$$ 均方误差

- 优化方法：

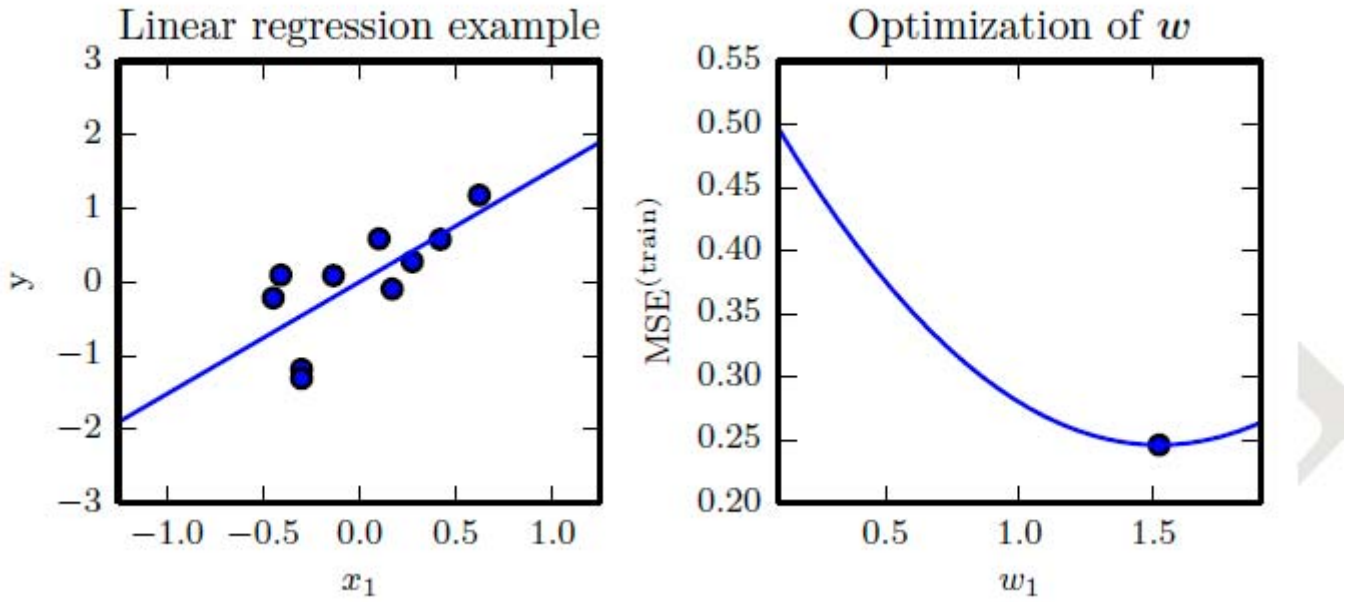
$$\nabla_{\boldsymbol{w}}\mathrm{MSE}_{\mathrm{train}} = 0$$ 最小二乘估计
经验风险最小

机器学习基础

机器学习示例——线性回归

- 解正规方程得参数估计结果：

$$\boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\text{T}}\boldsymbol{X}^{(\text{train})}\right)^{-1}\left(\boldsymbol{X}^{(\text{train})\text{T}}y^{(\text{train})}\right)$$



机器学习基础

机器学习示例——线性回归

- 以上是①直接建模$\boldsymbol{x}$和y之间的函数关系。还可以通过②建模$p(y|\boldsymbol{x})$来进行参数估计：

  假设为随机变量y，服从均值为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$，方差为$\sigma^2$的高斯分布，最大似然估计**(maximum likelihood estimate, MLE)**指找到一组$\boldsymbol{w}$使似然函数$p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w},\sigma)$最大

  令$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w},\sigma)}{\partial \boldsymbol{w}}=0$得

  $$\boldsymbol{w}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right)$$

  最大似然估计
  经验风险最小

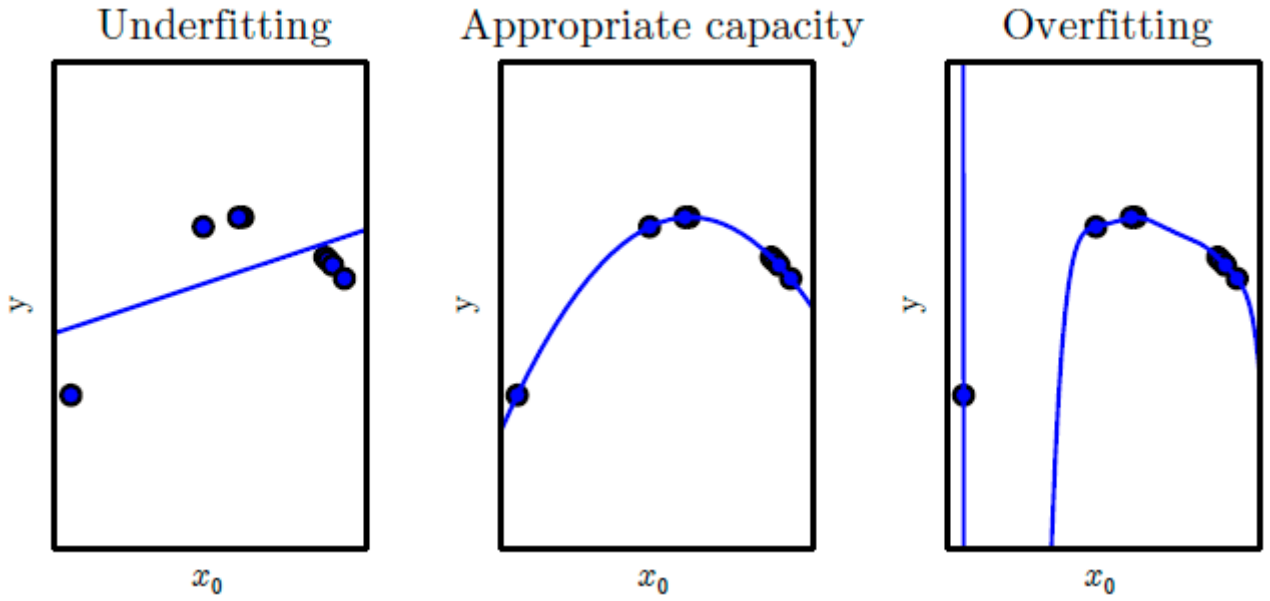
机器学习基础

机器学习的一个主要挑战

- 我们一直在降低训练误差，但真正的目标是降低测试误差，也即泛化误差
- 机器学习的目的：
  - 降低训练误差——防止欠拟合
  - 缩小训练误差和测试误差的差距——防止过拟合
- 可以通过控制模型容量来达到目的
  - 容量指模型拟合各种函数的能力
  - 容量低的模型可能很难拟合训练集
  - 容量高的模型可能会过拟合

机器学习基础

- 机器学习的一个主要挑战



$$\hat{y} = b + wx \quad \hat{y} = b + w_1 x + w_2 x^2 \quad \hat{y} = b + \sum_{i=1}^{9} w_i x^i$$

机器学习基础

- 机器学习的一个主要挑战——线性回归为例
  - 控制模型容量可以从两个角度考虑
    - 矩阵论
    - 概率论

机器学习基础

- 机器学习的一个主要挑战——线性回归为例
  - 矩阵论——使用伪逆来稳定欠定问题

$$\boldsymbol{w} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right) \rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right), \lambda > 0$$

    - 在ML中也称为岭回归(1970年)
    - 是一种正则化方法
    - 岭回归的解$\boldsymbol{w}$可以看做是结构风险最小化准则下的最小二乘估计

$$\mathcal{R}(\boldsymbol{w}) = \frac{1}{2}\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - y\right\|^2 + \frac{1}{2}\lambda\|\boldsymbol{w}\|^2$$

机器学习基础

机器学习的一个主要挑战——线性回归为例

- 概率论—最大后验估计(贝叶斯估计的点估计)
  - 假设参数$\boldsymbol{w}$为一个随机向量，并服从一个先验分布

$$p(\boldsymbol{w}|v) = \mathcal{N}(\boldsymbol{w}|0, v^2\boldsymbol{I})$$

  - 最大后验估计**(maximum a posteriori estimation, MAP)**是指最优参数为下述后验分布中概率密度最高的参数$\boldsymbol{w}$：

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}, v, \sigma)$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \sigma)\, p(\boldsymbol{w}|v)$$

最大似然估计

最大后验估计

机器学习基础

- 机器学习的一个主要挑战——线性回归为例
  - 概率论——最大后验估计(贝叶斯估计的点估计)

$$\log p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y},v,\sigma) \propto -\frac{1}{2\sigma^2}\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}-y\right\|^2-\frac{1}{2v^2}\|\boldsymbol{w}\|^2$$

    - 等价于均方损失的结构风险最小化

机器学习基础

机器学习基本问题

- 模型
  - 分类(二分类，多分类)，回归
  - 线性模型$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

  非线性模型$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b$

机器学习基础

机器学习基本问题

- 学习准则
  - 损失函数：**0-1**损失，均方损失，交叉熵，**KL**散度，**Hinge**损失**(**二分类时，$L(\mathrm{y}) = \max(0\,,\,1 - \mathrm{t}\cdot\mathrm{y})$ **)**
  - 估计方法：
    - 期望风险最小
    - 经验风险最小——最大似然估计：最小二乘**(**最小均方误差**)**
    - 结构风险最小——最大后验估计：**Ridge(**岭**)**回归**(**带L_2范数**)**，**Lasso**回归**(**带L_1范数**),**L_0范数

机器学习基础

- ✓ 范数常用来充当损失函数的正则化(Regularization)项——防止模型过拟合

L_2范数：$\|\boldsymbol{w}\|_2 = \sqrt{\sum_{i=1}^{n} w_i^2}$　　L_1范数：$\|\boldsymbol{w}\|_1 = \sum_{i=1}^{n} |w_i|$　　L_0范数：$\|\boldsymbol{w}\|_0 = \#(i|w_i \neq 0)$

- ■ $\boldsymbol{L_2}$范数正则化**(Ridge回归)**

偏差　　正则化项

$$\boldsymbol{J(w)} = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - y_i\right)^2 + \lambda\|\boldsymbol{w}\|_2^2 \quad (\lambda > 0)$$

$\boldsymbol{\lambda}$**:**平衡正则化项和经验风险的系数

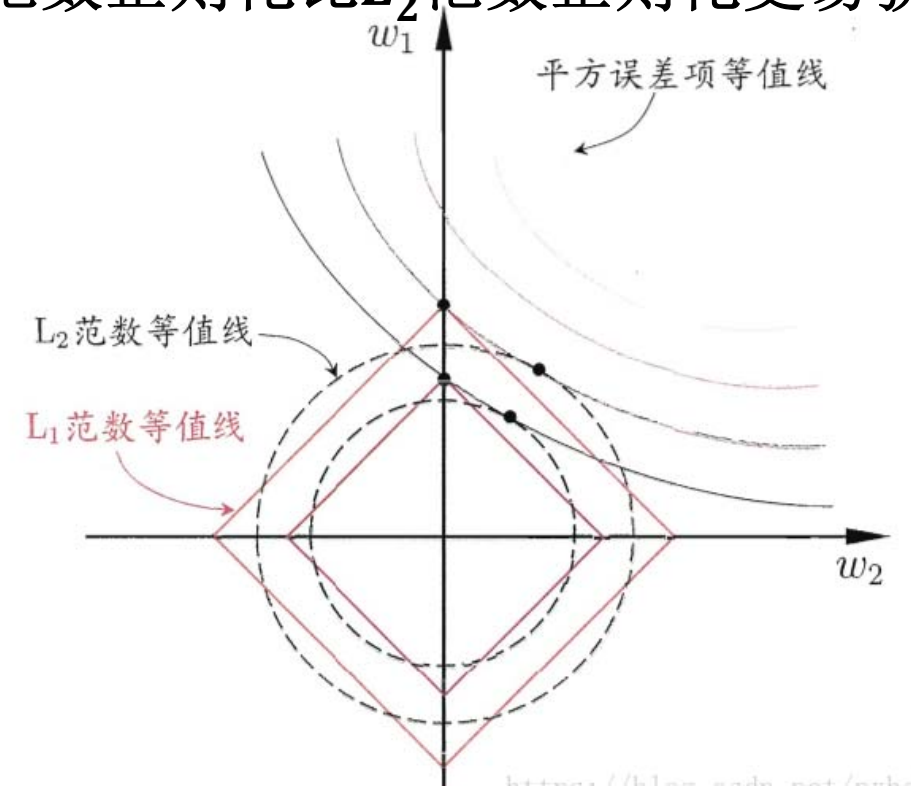
- ■ $\boldsymbol{L_1}$范数正则化**(Lasso回归)**

$$\boldsymbol{J(w)} = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - y_i\right)^2 + \lambda\|w\|_1 \quad (\lambda > 0)$$

偏差越小，模型越复杂(容量越大)，正则化项的值越大。
那么同时限制正则化项的值，能够防止过拟合。

机器学习基础

- ✓ 范数常用来充当损失函数的正则化(Regularization)项
  - ■ L_1范数正则化比L_2范数正则化更易获得“稀疏”解



机器学习基础

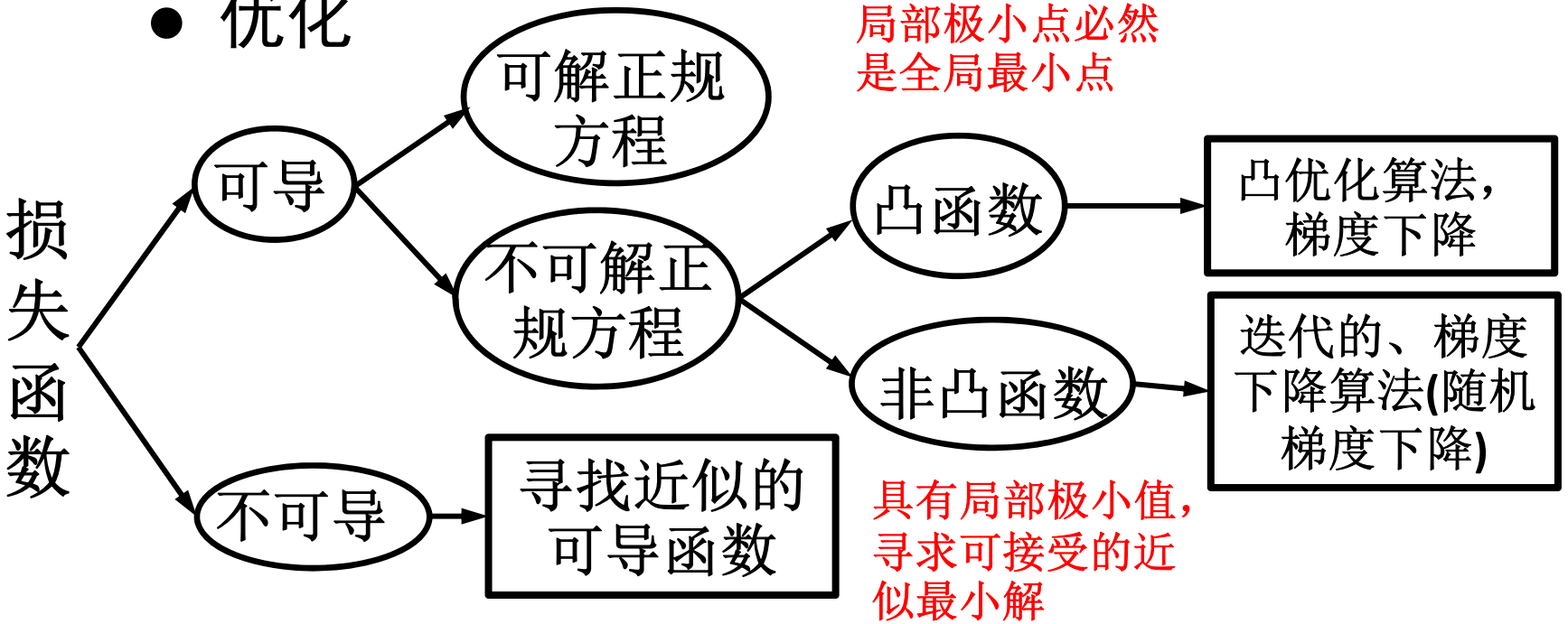
机器学习基本问题

- 优化
  - 正规方程
  - 随机梯度下降，共轭梯度，牛顿法，**BFGS**

机器学习基础

机器学习基本问题

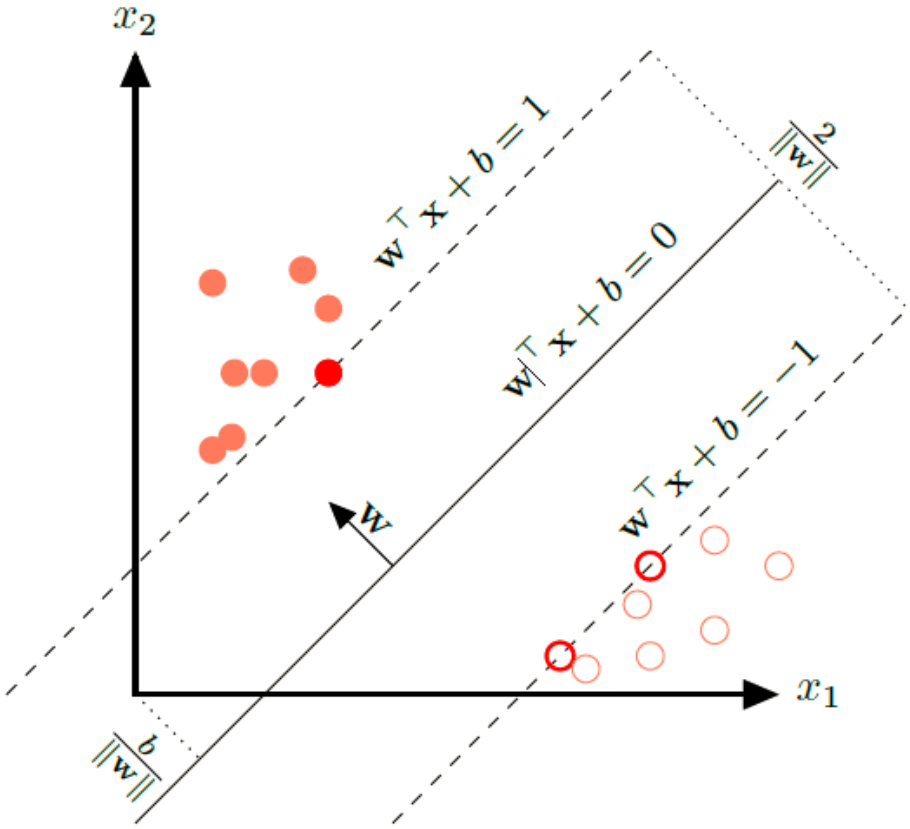
- 优化



机器学习基础

机器学习——非线性

- 支持向量机(support vector machine, SVM)，经典两类分类算法



针对“线性可分”
Margin算法

机器学习基础

机器学习——非线性

- 但它使用核函数隐式地将样本从原始特征空间映射到更高维的空间，解决原始特征空间中的线性不可分问题
- 决策函数：

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\phi(\boldsymbol{x}) + b = \sum_{i=1}^{n} \alpha_i y_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) + b$$

$$= \sum_{i=1}^{n} \alpha_i y_i \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i) + b$$

线性→非线性by 特征映射

在深度学习之前，学习非线性模型的主要方法是结合核策略的线性模型

机器学习基础

回顾

- 线性模型$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$
- 非线性模型$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b$

  其中，$\phi(\boldsymbol{x})=[\phi_{\boldsymbol{1}}(\boldsymbol{x}),\phi_{\boldsymbol{2}}(\boldsymbol{x}),\dots,\phi_K(\boldsymbol{x})]^{\mathrm{T}}$为$\boldsymbol{K}$个非线性基函数组成的向量。

  如果$\phi(\boldsymbol{x})$本身为可学习的基函数，如

  $$\phi_k(\boldsymbol{x})=h\left(\boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{T}}\phi'(\boldsymbol{x})+b_k\right),\forall 1\le k\le K$$

  则$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})$等价于神经网络模型。

机器学习基础

- 可以看出，如果希望模型能力更强(比如解线性不可分问题)，可以借助特征映射；
- 而特征映射只是可以看作神经网络这种模型的一个特殊方面；
- 神经网络可以建模的函数空间很大——实际上，两层神经网络就可以拟合任意函数
- 神经网络可以进行多层的特征映射；
- 机器学习是一门讨论各种适用于不同问题的函数形式的学科，深度学习是ML中的一类函数，而这类函数通常是多层神经网络。

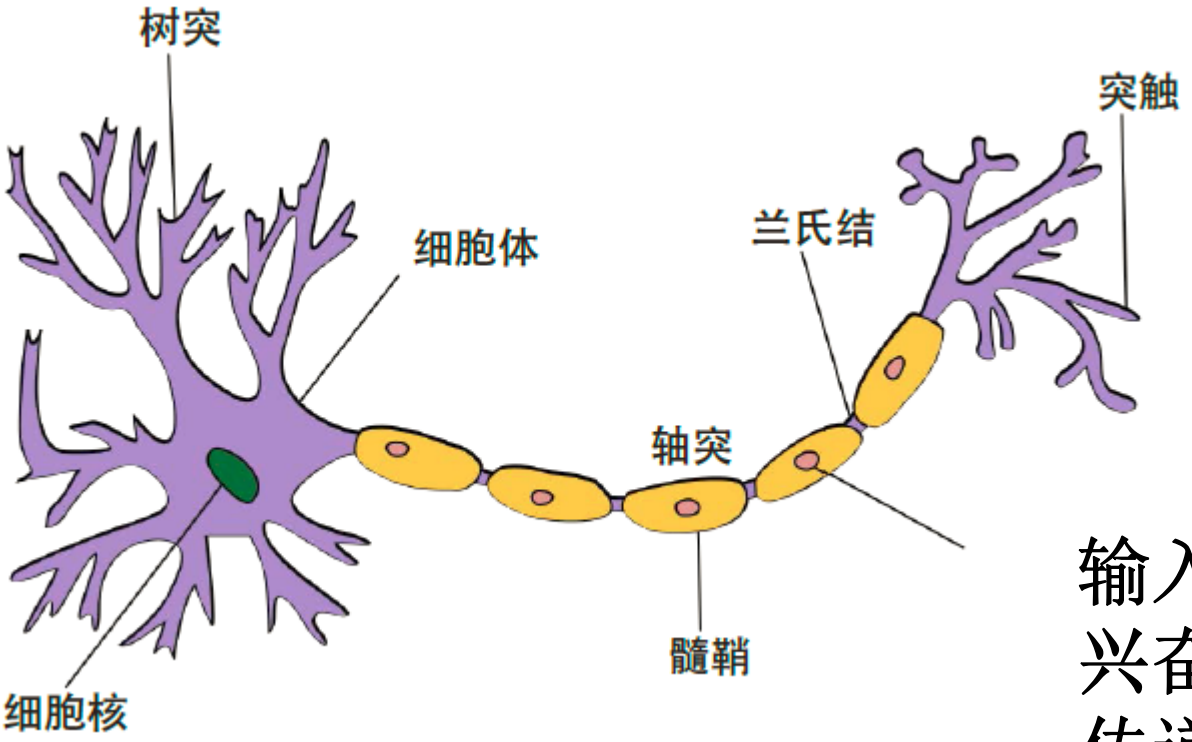
表示学习

大纲

- 机器学习基础
- 人工神经网络概述
  - 人脑神经网络
  - 人工神经网络
- 深度学习基础
- 典型神经网络模型

人工神经网络概述

- 人脑神经网络



输入信息量，
兴奋或抑制，
传递(连接)

图片来自**Wikipedia**

人工神经网络概述

人脑神经网络的学习

- 神经元之间突触的强度是可塑(学习、训练)的，不同的连接形成了不同的记忆印痕。
- Hebb (1949)
  - **Hebb's Rule:** 当神经元**A**的一个轴突和神经元**B**很近，足以对它产生影响，并且持续地、重复地参与了对**B**的兴奋，那么在这两个神经元或其中之一会发生某种生长过程或新陈代谢变化，以致于**A**作为能使**B**兴奋的细胞之一，它的效能加强了。
  - **Hebb's learning:**如果两个神经元总是相关联地受到刺激，它们之间的突触强度增加。

人工神经网络概述

人工神经网络

- 从结构、实现机理和功能上模拟人脑神经网络
- 人工神经网络是由大量神经元通过极其丰富和完善的连接而构成的自适应非线性动态系统

人工神经网络概述

人工神经网络

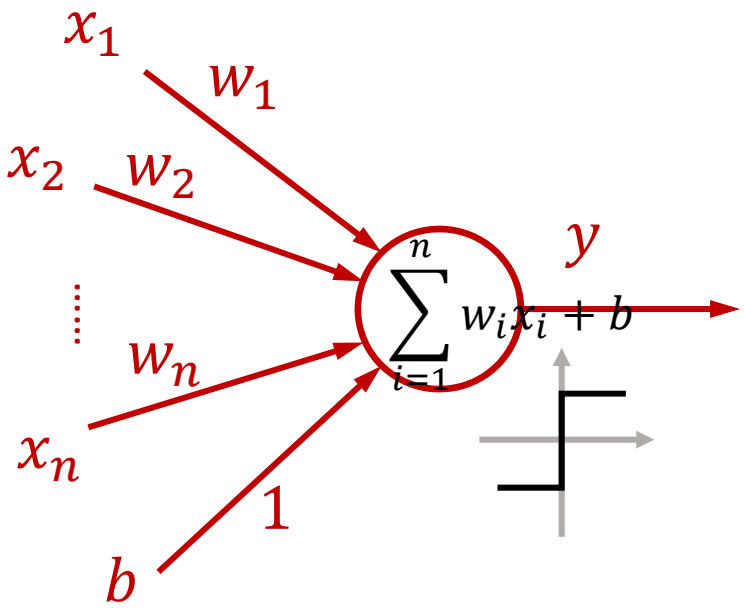
- 1943年，MP网络，理想化
- 1949年，Hebb网络，第一个可学习的网络，基于Hebb规则的无监督学习
- 1958年，感知机(perceptron)，第一个具有机器学习思想的神经网络，学习方法无法扩展到多层
- 1969年，明斯基发表《感知机》

✓ 控制论

人工神经网络概述

- Perceptron

$$y = \mathrm{sgn}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b\right)$$



- 线性模型，有监督学习
- 为了逻辑分类，使用了符号函数
- 损失函数使用均方误差，不可导

  ——可以使用错误分类点到分类面的距离

人工神经网络概述

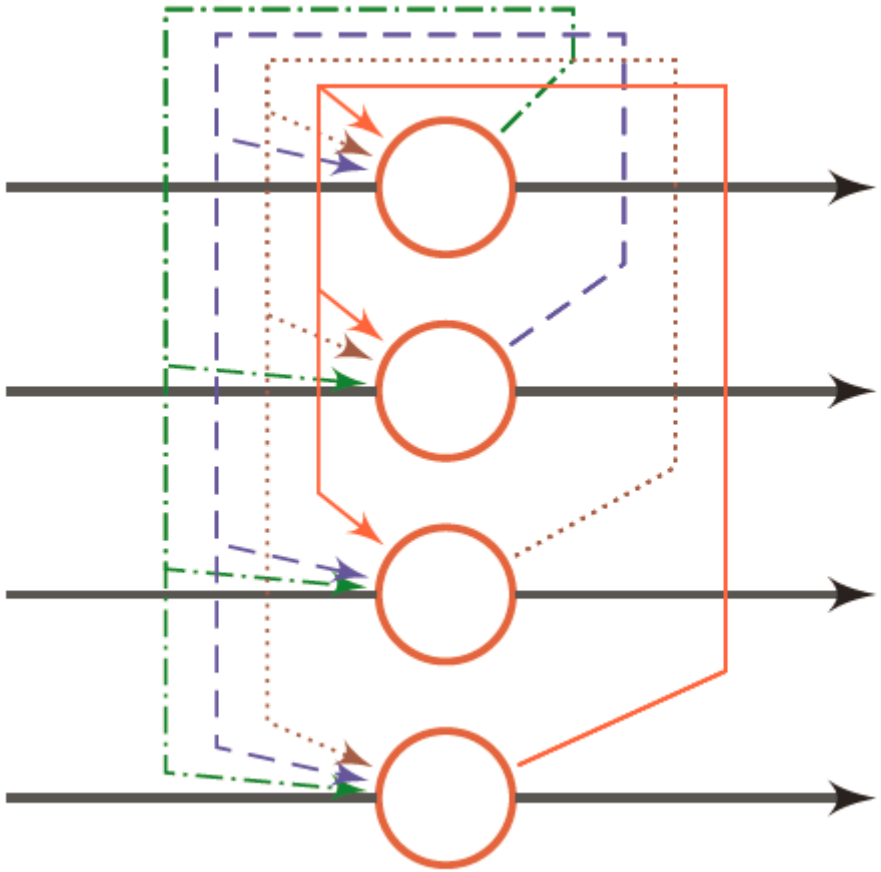
人工神经网络

- 1974年，Paul Werbos，反向传播(backpropagateion, BP)，未受到重视
- 1983年，Hopfield网络，一种用于联想记忆的循环神经网络
- 1984年，Hinton，玻尔兹曼机
- 1986年，BP，并解决多层感知机的学习问题

✓ 连接主义(联结主义)——当网络将大量简单的计算单元连接在一起时可以实现智能行为

人工神经网络概述

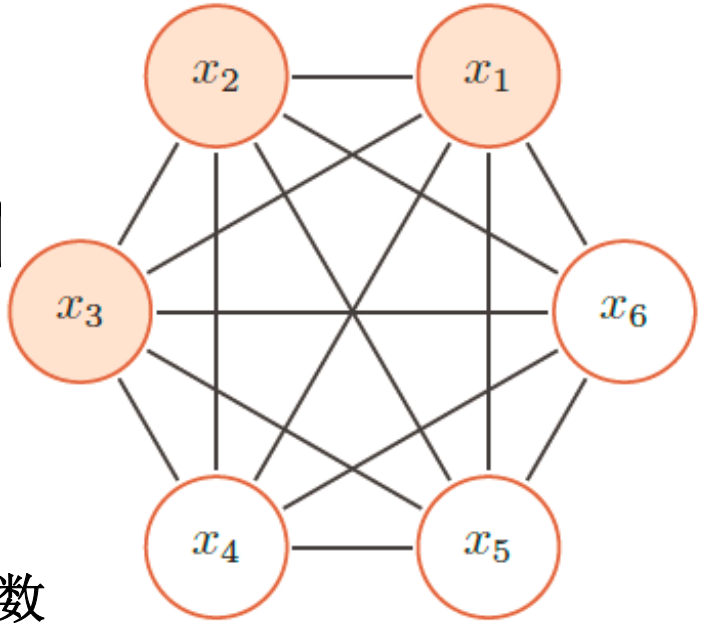
- Hopfield网络



- 所有神经元都互连接的不分层的神经网络；
- 权值按规则计算；
- 神经元的状态不断更新，网络演变到稳定时各神经元的状态便是问题之解

人工神经网络概述

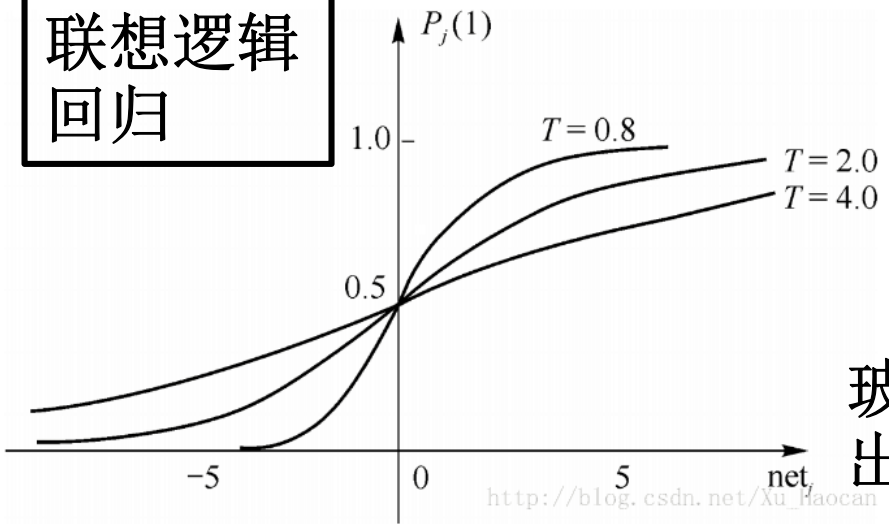
- 玻尔兹曼机——一种随机1
  络



$$\mathrm{net} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - T$$

$$P_j(1) = \frac{1}{1 + \mathrm{e}^{-\mathrm{net}_j/T}}$$ Sigmoid函数

联想逻辑回归



$$E(t) = -\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\boldsymbol{x}(t) + \boldsymbol{x}^{\mathrm{T}}(t)T$$

$$\frac{P(\alpha)}{P(\beta)} = \frac{\mathrm{e}^{-E_\alpha/T}}{\mathrm{e}^{-E_\beta/T}}$$

玻尔兹曼分布：网络中任意两个状态出现的概率与之对应能量之间的关系

人工神经网络概述

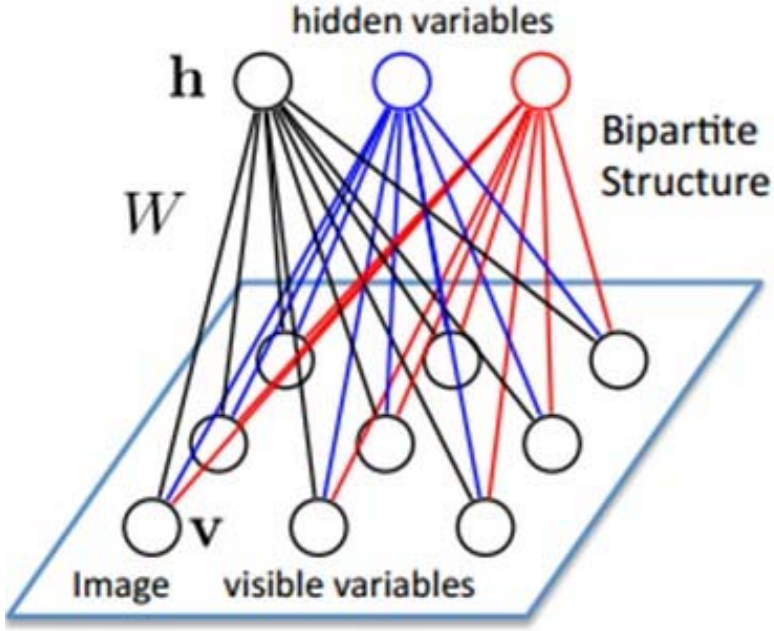
人工神经网络

- 2006年，Hinton，逐层预训练
- 2012年，在语音、图像领域取得巨大成功，全面进入NLP
- 2017年，在语音、图像领域趋于饱和
- 2018年，预训练语言模型，进入预训练+微调时代

✓ 深度学习

人工神经网络概述

- 受限玻尔兹曼机(RBM)——基于能量的概率分布模型



$$E(\boldsymbol{v},\boldsymbol{h}) = -\boldsymbol{h}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{v} - a^{\mathrm{T}}\boldsymbol{v} - b^{\mathrm{T}}\boldsymbol{h}$$

$$P(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\mathrm{e}^{-E(v,h)}$$

$$P(h_j = 1|\boldsymbol{v}) = \mathrm{sigmoid}\left(b_j + \sum_i v_i w_{ij}\right)$$

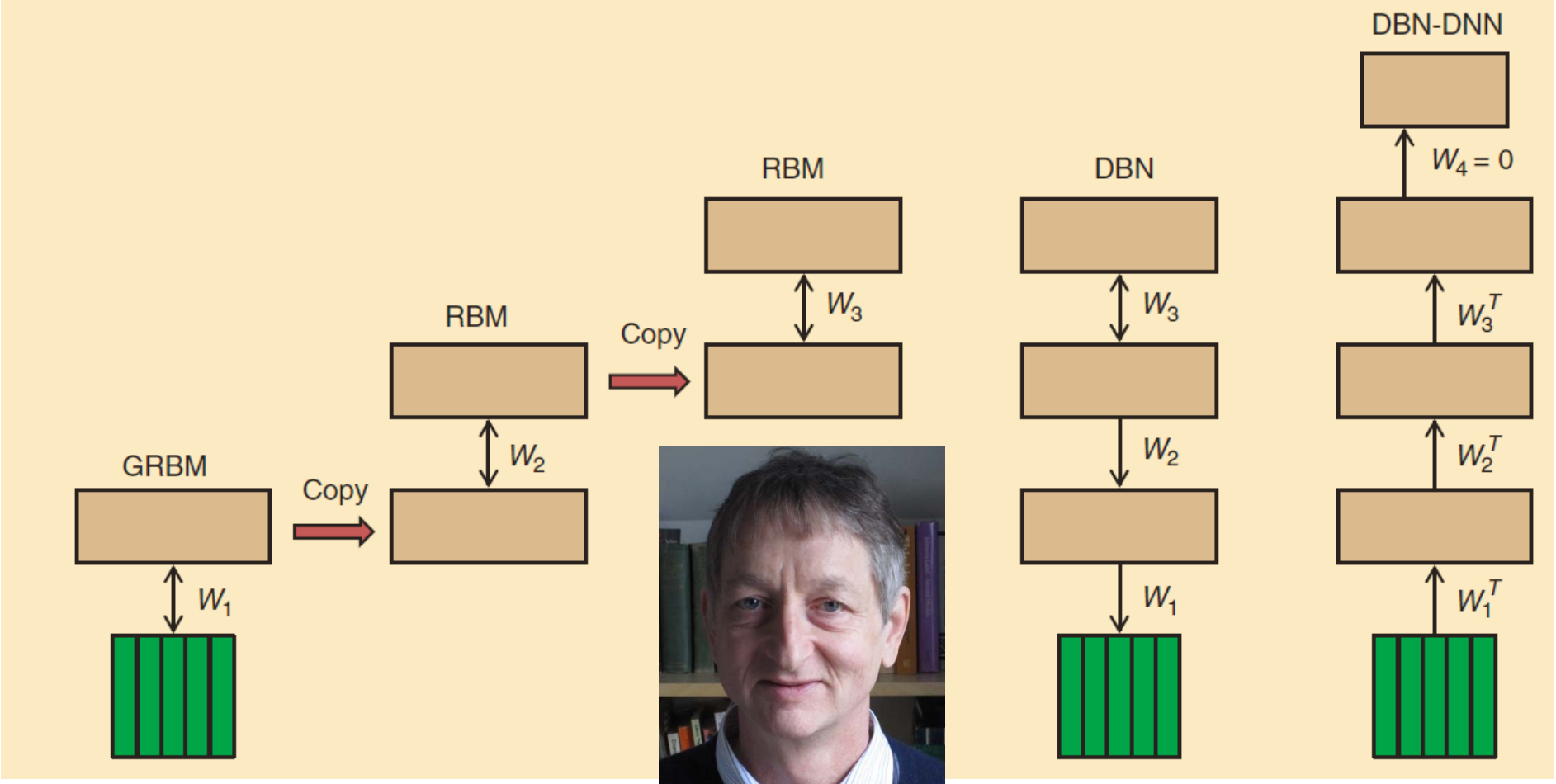
$$P(v_i = 1|\boldsymbol{h}) = \mathrm{sigmoid}\left(a_i + \sum_j h_j w_{ij}\right)$$

$$L(\boldsymbol{W},a,b) = -\sum_{i=1}^{m} \ln\left(P\left(\boldsymbol{v}^{(i)}\right)\right)$$

$$p(\mathbf{v};\mathbf{W}) = \sum_h p(\mathbf{h};\mathbf{W})\, p(v|\mathbf{h};\mathbf{W}).$$

人工神经网络概述

- 深度信念网络(DBN)，逐层预训练



人工神经网络概述

- 基于神经网络的深度学习之所以成立
  1. NN可学习——Hebb学习规则，感知机
  2. 多层NN可学习——BP
  3. 多层NN可有效学习——玻尔兹曼机，DBN

大纲

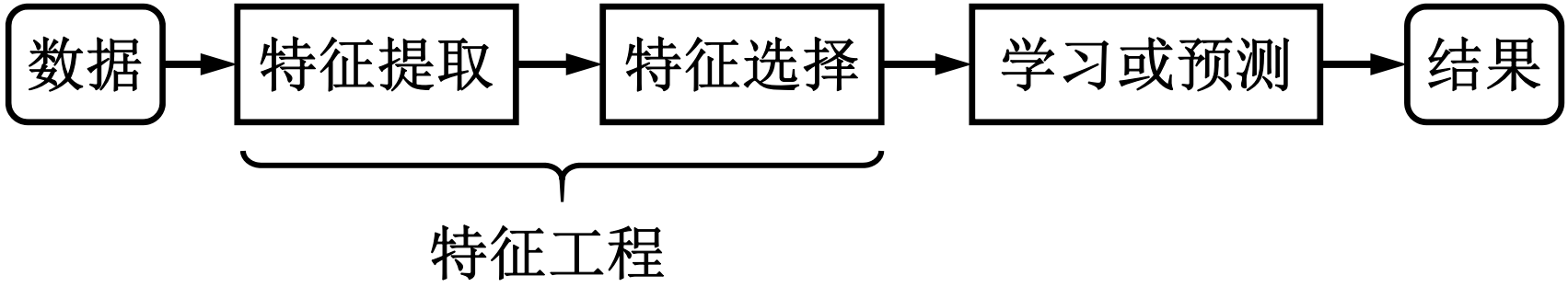
- 机器学习基础
- 人工神经网络概述
- 深度学习基础
  - 表示学习
  - 深度学习
  - 深度前馈网络
  - 基于梯度的学习
  - 反向传播
  - 正则化

重点掌握深度学习与传统学习的差别

深度学习基础

表示学习

- 传统机器学习：



- 但很多时候，很难知道提取哪些特征

深度学习基础

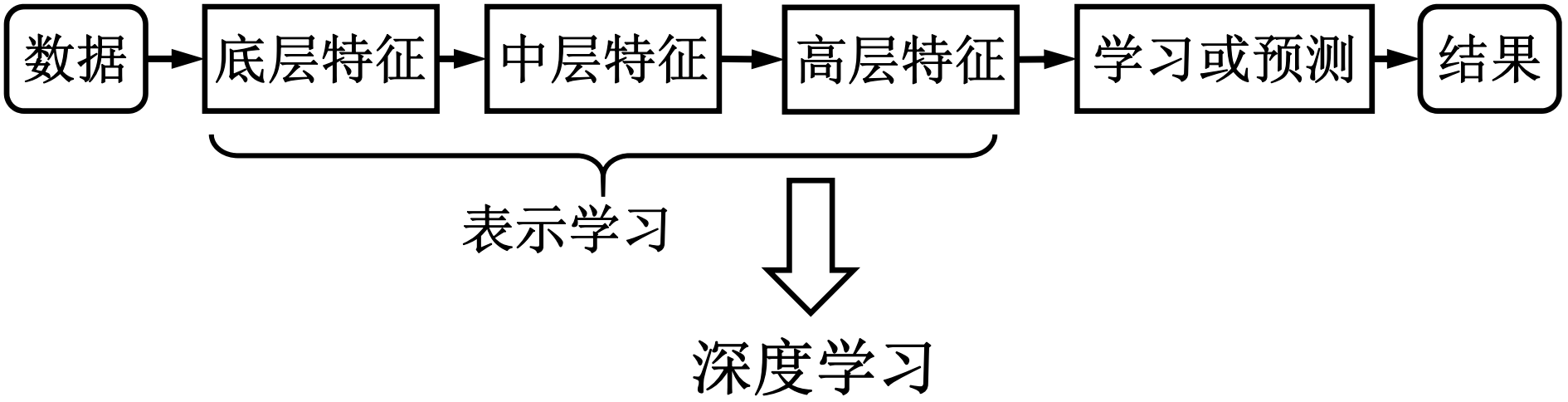
表示学习

- 途径之一：使用机器学习来发掘表示本身，而不仅仅把表示映射到输出——表示学习
- 典型模型：自编码器
- 要学习到一种好的高层语义表示，通常需要从底层特征开始，经过多步非线性转换才能得到——表示学习的关键是构建具有一定深度的多层次特征表示

深度学习基础

深度学习

- 通过其他较简单的表示来表达复杂表示，解决了表示学习中的核心问题



- 深度：原始数据进行非线性特征转换的次数

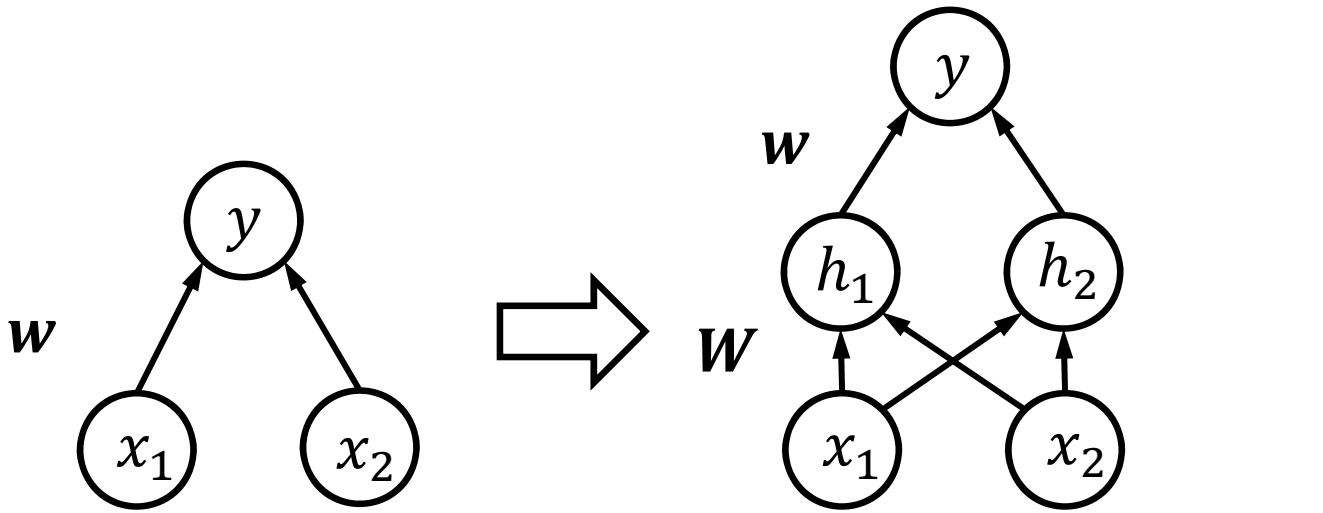
深度学习基础

深度前馈网络——学习XOR

- 线性模型(包括感知机)不能拟合XOR
  - 样本是闭集：$\boldsymbol{X} = \{[0,0]^{\mathrm{T}}, [0,1]^{\mathrm{T}}, [1,0]^{\mathrm{T}}, [1,1]^{\mathrm{T}}\}$——只需拟合训练集
  - 模型：$f(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$
  - 利用均方误差，解正规方程得：$\boldsymbol{w} = 0$，$b = 0.5$ X
  - 原因：线性模型中不能使用x_1的值来改变的x_2系数
- 解决办法：特征变换——用非线性变换将输出必须为1的两个点折叠到特征空间中的单个点 $\boldsymbol{x} \rightarrow \boldsymbol{h}$

深度学习基础

深度前馈网络——学习XOR



$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

$$\boldsymbol{h} = g\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}\right),$$

g为非线性函数ReLU

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}$$

深度学习基础

基于梯度的学习

- 梯度(gradient)：一个向量，第i个元素是f关于x_i的偏导数，记作：$\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$
- 向量有方向，梯度的方向就指出了函数在给定点的上升最快的方向
  - 方向导数是函数f在某一点沿某一方向l的倾斜程度(坡度)，是梯度在该方向上的投影
  - 当该方向与梯度方向一致时，方向导数取到最大

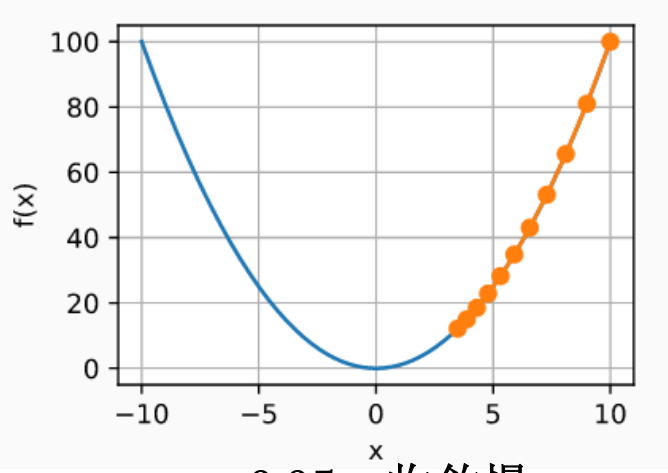
$$\frac{\partial f}{\partial l} = |\nabla_{\boldsymbol{x}} f(\boldsymbol{x})| \cos\theta$$

深度学习基础

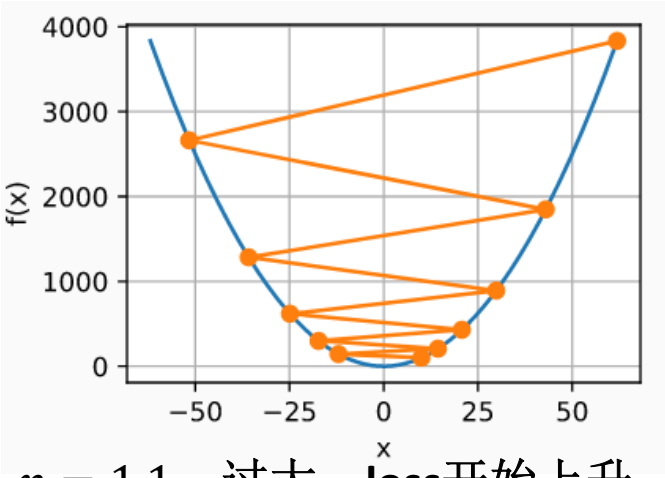
- 梯度下降(gradient descent)
  - 在负梯度方向上移动，可以最快地减小f
  - 梯度下降建议新的点为：

$$\boldsymbol{w}' = \boldsymbol{w} - \eta \nabla_{\boldsymbol{x}} f(\boldsymbol{w})$$

η：学习率
(leaning rate)



$\eta = 0.05$，收敛慢



$\eta = 1.1$，过大，**loss**开始上升

  - 在梯度的每一个元素为**0(**或很接近**0)**时收敛

深度学习基础

- 梯度下降
  - 举例：$J(\boldsymbol{\theta}) = w_1^2 + w_2^2$

设：$w^{(0)} = [1,3]$，$\eta = 0.1$

$\nabla J(\boldsymbol{w}) = [2\theta_1, 2\theta_2]$,

迭代：

$w^{(1)} = w^{(0)} - \eta \nabla J(\boldsymbol{w}) = [1,3] - 0.1[2,6] = [0.8,2.4]$

$w^{(2)} = w^{(1)} - \eta \nabla J(\boldsymbol{w}) = [0.8,2.4] - 0.1[1.6,4.8]$

$= [0.64,1.92]$

……

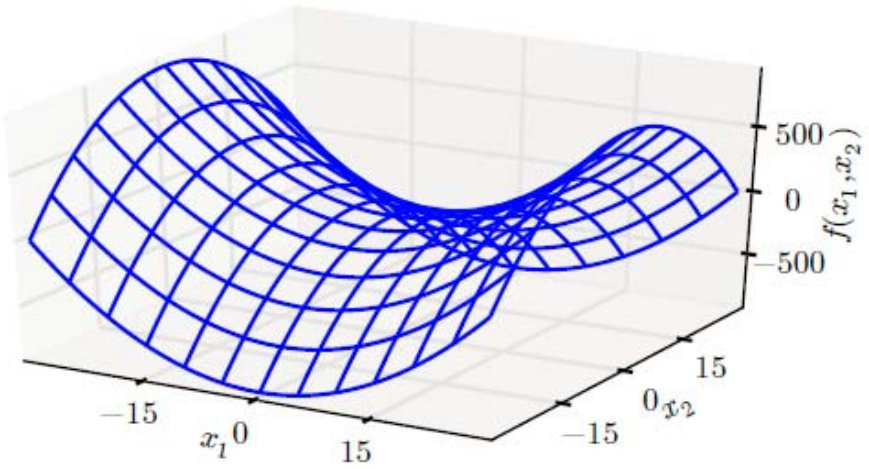
深度学习基础

- 梯度下降是典型的一阶(first order)优化算法
  - 它使用**Jacobian**矩阵：如果函数$f: \mathbb{R}^m \to \mathbb{R}^n$，$f$的**Jacobian**矩阵$\boldsymbol{J} \in \mathbb{R}^{n \times m}$定义为$J_{i,j} = \frac{\partial}{\partial w_j} f(\boldsymbol{x})_i$
- 二阶导数(曲率)能够衡量梯度下降的预期
  - **Hessian**矩阵：$H(f)(\boldsymbol{x})_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} f(\boldsymbol{x})$
  - 在$x^{(0)}$近似二阶泰勒级数展开f：

$$f\left(x^{(0)} - \eta \boldsymbol{g}\right) = f\left(x^{(0)}\right) - \eta \boldsymbol{g}^{\mathrm{T}} \boldsymbol{g} + \frac{1}{2} \eta^2 \boldsymbol{g}^{\mathrm{T}} \boldsymbol{H} \boldsymbol{g}$$

深度学习基础

- Hessian矩阵可以用来确定一个临界点是否是局部极值或鞍点



$$f(\boldsymbol{x}) = x_1^2 - x_2^2$$

- 使用Hessian矩阵的优化算法称为二阶(second-order)优化算法，如牛顿法

深度学习基础

- 用梯度下降训练神经网络
  - 与其他使用梯度下降的**ML**模型区别不大。
  - 但由于神经网络的非线性导致大多数损失函数非凸，算法失去了全局收敛的保证——梯度下降往往使函数达到一个非常小的近似值，同时对初始值敏感。
  - 深度学习任务通常有非常大的训练数据规模，因此更多地使用随机梯度下降。

深度学习基础

- 随机梯度下降
  - 整个训练集上的梯度

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L\left(x^{(i)}, y^{(i)}, \boldsymbol{\theta}\right)$$

  - 梯度是期望，可使用小规模的样本近似估计

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{m'}\sum_{i=1}^{m'} \nabla_{\boldsymbol{\theta}} L\left(x^{(i)}, y^{(i)}, \boldsymbol{\theta}\right)$$

m'个样本组成一个**Minibatch**

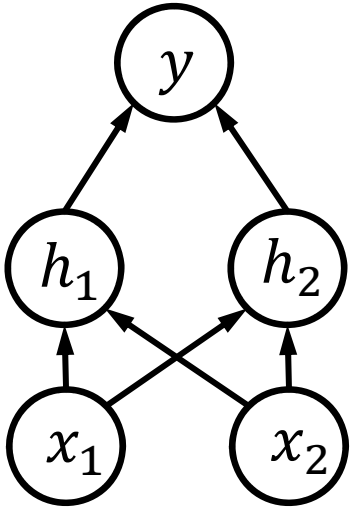
深度学习基础

典型深度前馈网络结构

- 模型

$$\boldsymbol{y} = g(\boldsymbol{z})$$

  - 回归，二分类，多分类——由输出单元决定
  - 隐藏层： $\boldsymbol{z} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h} + \boldsymbol{b}$



$$\boldsymbol{h}^{(1)} = g^{(1)}\left(\boldsymbol{w}^{(1)\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right) \quad \text{隐藏层}$$

$$\boldsymbol{h}^{(2)} = g^{(2)}\left(\boldsymbol{w}^{(2)\mathrm{T}}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}\right)$$

$$\cdots$$

$$\cdots$$

$$\boldsymbol{y} = g^{(l)}\left(\boldsymbol{w}^{(l)\mathrm{T}}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}\right) \quad \text{输出层}$$

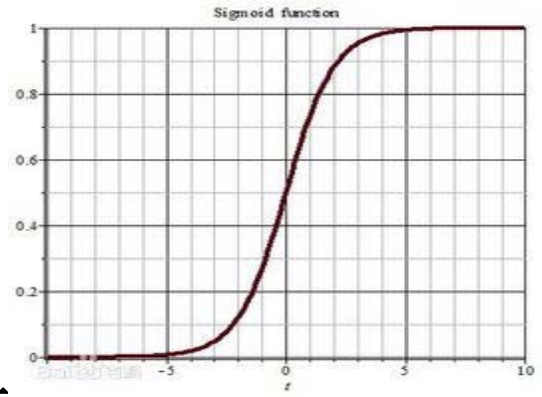
深度学习基础

$$\boldsymbol{y} = g(\boldsymbol{z})$$

- 输出单元
  - 回归——采用线性输出层：$\boldsymbol{y} = \boldsymbol{z} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{h} + \boldsymbol{b}$

  假设$\boldsymbol{y}$的后验$p(\boldsymbol{y}|\boldsymbol{x})$服从均值为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h} + \boldsymbol{b}$的高斯分布，最大化对数似然等价于最小化均方误差。

深度学习基础



- 二分类——通常采用**sigmoid**单元

$p(y=1|\boldsymbol{x})$必须在**[0,1]**，一种做法是：

$$p(y=1|\boldsymbol{x})=\max\left\{0,\min\{1,\boldsymbol{w}^{\mathrm{T}}\boldsymbol{h}+b\}\right\}$$

但是，无法用梯度下降有效训练——饱和

$$p(y=1|\boldsymbol{x})=\frac{\exp(yz)}{\sum_{y'=0}^{1}\exp(y'z)}$$

引入**sigmoid**单元：$p(y=1|\boldsymbol{x})=\sigma(z)=\frac{1}{1+\exp^{-z}}$

同时采用最大似然估计$\log p(y|\boldsymbol{x})$，抵消**sigomid**中的**exp**。

Sigmoid+最大似然估计——二分类标配

深度学习基础

- 多分类——通常采用**softmax**单元

$$p(y=i|\boldsymbol{x}) = \text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

例如：$z_1 = 3$，$z_2 = 1$，$z_3 = -3$

$$p(y=1) = {e^3}/{e^3+1+e^{-3}} = 0.88$$

采用最大似然估计：

$$\log p(y=i|\boldsymbol{x}) = z_i - \log \sum_j \exp(z_j)$$

总是强烈地惩罚最活跃的不正确预测

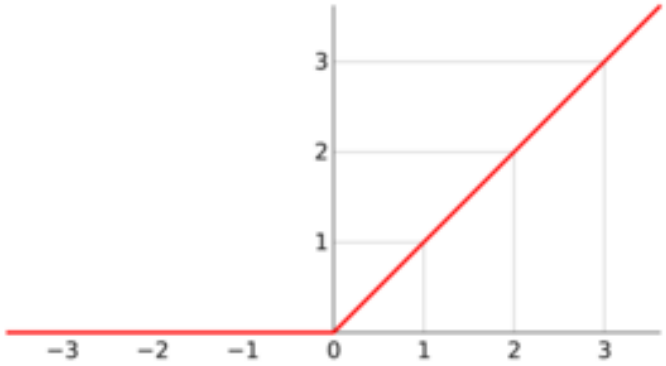
Softmax+最大似然估计——多分类标配

深度学习基础

- 隐藏单元

$$\boldsymbol{h} = g\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}\right)$$

  - **Sigmoid**函数：$g(z) = \sigma(z)$**(**不推荐使用在隐藏单元**)**
  - 双曲正切函数：$g(z) = \tanh(z)$
  - 整流线性单元**(ReLU)**：$g(z) = \max\{0, z\}$



深度学习基础

反向传播(back propagation, BP)

- 网络接收输入$\boldsymbol{x}$，输出估计值$\hat{\boldsymbol{y}}$时，发生的是前向传播。
- 训练时，允许来自损失函数的信息通过网络向后流动以便计算梯度，称为反向传播。
- 梯度下降等算法用来训练网络(学习)，BP是一种计算梯度的方法，它基于链式法则

深度学习基础

- 链式法则
  - 设x是实数，f和g是从实数到实数的映射函数，$z = g(x)$，$y = f(z)$：

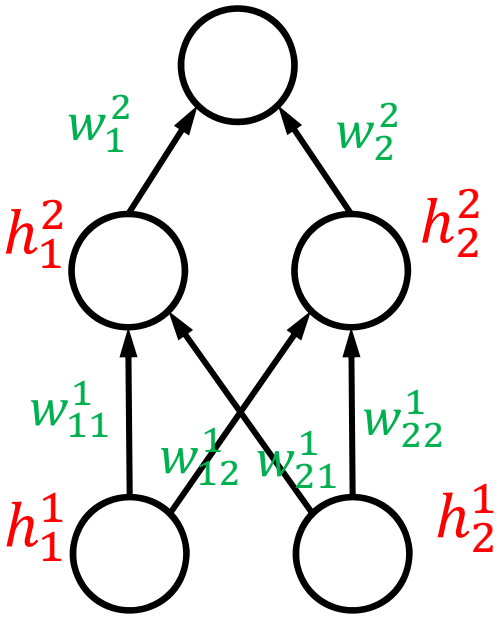
$$\frac{dy}{dx} = \frac{dy}{dz}\frac{dz}{dx}$$

  - 设$\boldsymbol{x}$ $\boldsymbol{z}$是向量：$\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{z} \in \mathbb{R}^n$，$g$是从$\mathbb{R}^m$到$\mathbb{R}^n$的映射，$f$是从$\mathbb{R}^n$到$\mathbb{R}$的映射，$\boldsymbol{z} = g(\boldsymbol{x})$，$y = f(\boldsymbol{z})$：

$$\nabla_{\boldsymbol{x}} y = \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \nabla_{\boldsymbol{z}} y$$

深度学习基础

- 利用链式法则实现BP



$$e_o = \frac{1}{2}(\hat{y} - y)^2$$

$$\hat{y} = h^3 = sigmoid(z^3)$$

$$z^3 = w_1^2 \cdot h_1^2 + w_2^2 \cdot h_2^2 + b^3$$

分配w_1^2：

$$\frac{\partial e_o}{\partial w_1^2} = \frac{\partial e_o}{\partial h^3} \cdot \frac{\partial h^3}{\partial z^3} \cdot \frac{\partial z^3}{\partial w_1^2}$$
$$= (h^3 - y) \cdot \mathrm{s}(z^3) \cdot \left(1 - \mathrm{s}(z^3)\right) \cdot h_1^2$$

分配w_2^2：

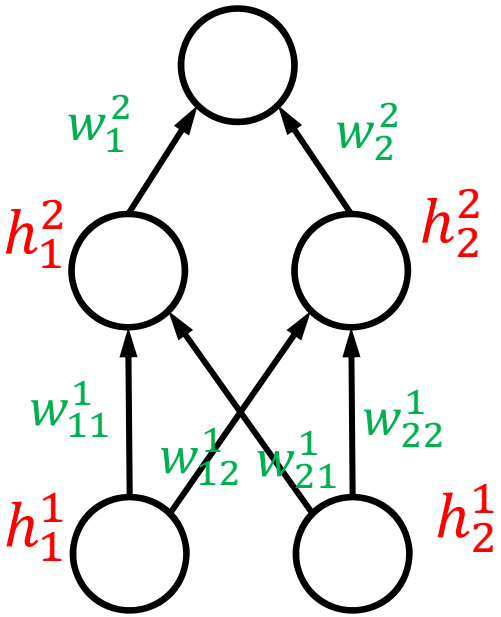
$$\frac{\partial e_o}{\partial w_2^2} = \frac{\partial e_o}{\partial h^3} \cdot \frac{\partial h^3}{\partial z^3} \cdot \frac{\partial z^3}{\partial w_2^2}$$
$$= (h^3 - y) \cdot \mathrm{s}(z^3) \cdot \left(1 - \mathrm{s}(z^3)\right) \cdot h_2^2$$

分配b^3：

$$\frac{\partial e_o}{\partial b^3} = \frac{\partial e_o}{\partial h^3} \cdot \frac{\partial h^3}{\partial z^3} \cdot \frac{\partial z^3}{\partial b^3} = (h^3 - y) \cdot \mathrm{s}(z^3) \cdot \left(1 - \mathrm{s}(z^3)\right)$$

深度学习基础

- 利用链式法则实现BP



$$e_o = \frac{1}{2}(\hat{y} - y)^2$$

$$h_1^2 = sigmoid(z_1^2)$$

$$z_1^2 = w_{11}^2 \cdot h_1^1 + w_{21}^2 \cdot h_2^1 + b_1^2$$

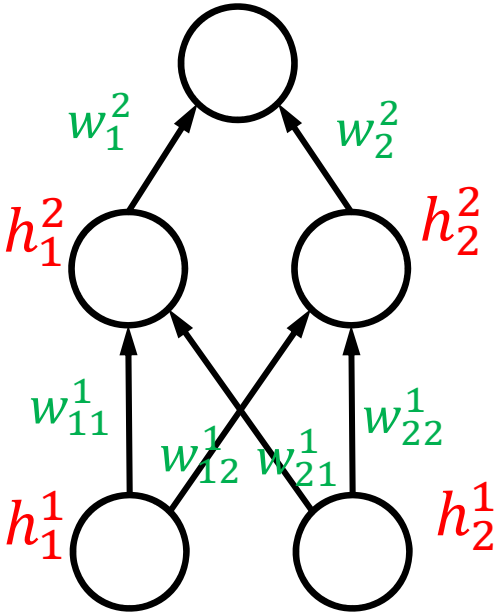
分配w_{11}^1：

$$\frac{\partial e_o}{\partial w_{11}^1} = \frac{\partial e_o}{\partial h^3} \cdot \frac{\partial h^3}{\partial z^3} \cdot \frac{\partial z^3}{\partial h_1^2} \cdot \frac{\partial h_1^2}{\partial z_1^2} \cdot \frac{\partial z_1^2}{\partial w_{11}^1}$$

……

深度学习基础

- 利用链式法则实现BP



更新：$$w = w - \eta \frac{\partial e}{\partial w}$$

$$b = b - \eta \frac{\partial e}{\partial b}$$

深度学习基础

- 梯度下降+反向传播→NN的训练(DL的学习)
- 这是一个优化问题，通常是非线性非凸优化
- 在深度学习框架下，优化存在的挑战
  - 病态
  - 局部极小
  - 平坦区域：高原、鞍点等
  - 悬崖
  - 梯度消失和梯度爆炸

深度学习基础

- 梯度消失和梯度爆炸
  - 当模型很深的时候，可能会出现反复与$\boldsymbol{W}$相乘的路径

    $$\boldsymbol{W}^t = (\boldsymbol{V}\mathrm{diag}(\lambda)\boldsymbol{V}^{-1})^t = \boldsymbol{V}\mathrm{diag}(\lambda)^t\boldsymbol{V}^{-1}$$
  - 特征值λ_i若在量级上大于$\mathbf{1}$则会爆炸；若小于$\mathbf{1}$时则会消失
  - 梯度消失使得难以知道参数朝哪个方向移动能够改进代价函数
  - 梯度爆炸会使得学习不稳定
  - ✓ 实际上，不合适的激活函数也造成梯度的消失(饱和)

深度学习基础

- 梯度消失和梯度爆炸
  - 预训练+微调，缓解梯度消失
  - 采用**ReLU**等具有线性性质的激活函数，缓解梯度消失
  - 梯度截断，缓解梯度爆炸
  - 正则化，缓解梯度爆炸

深度学习基础

正则化——增强模型的泛化能力

- 范数惩罚

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

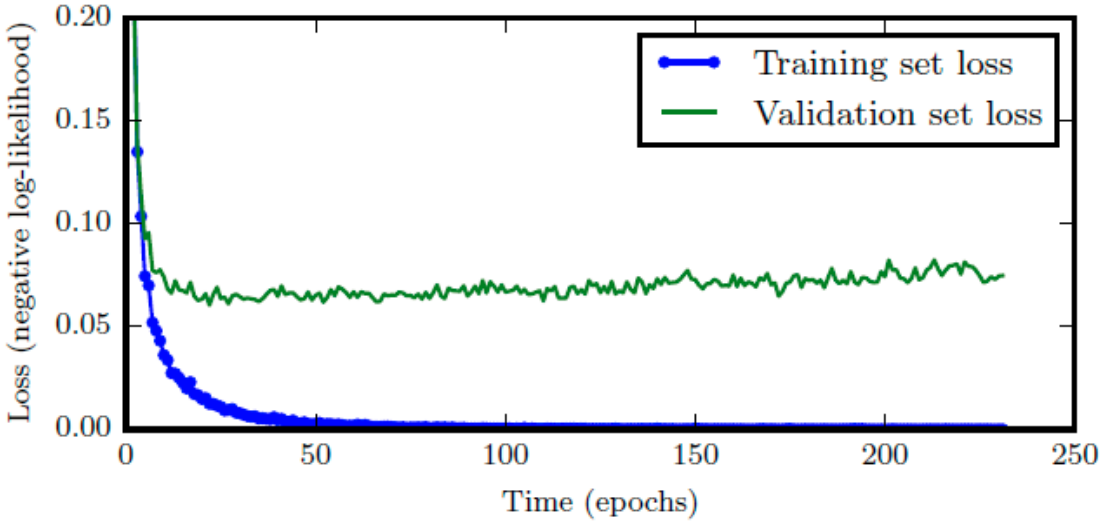
$$\mathcal{L}(\boldsymbol{\theta}, \alpha; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha(\Omega(\boldsymbol{\theta}) - k)$$

- 数据增强(data augmentation)
  - 通过生成同分布的伪数据扩大训练集，在图像和语音领域有通用做法。
- 加入噪声
  - 数据中加入噪声属于数据增强，等价于权重衰减；
  - 还可以在参数中加入噪声，看作关于权重的贝叶斯推断的随机实现

深度学习基础

正则化——增强模型的泛化能力

- 稀疏激活
  - 不是衰减权重，而衰减激活
- Dropout
  - 通过随机行为训练网络并平均多个随机决定进行预测，通过参数共享实现了**bagging**
- 提前终止
  - 高效的超参



深度学习基础

关于深度学习你要理解的

- 问题的动机和特点；
- 将大量不同类型神经网络层通过特定方式组合在一起的模型背后的数学原理；
- 在原始数据上拟合极复杂的深层模型的优化算法；
- 有效训练模型、避免数值计算陷阱以及充分利用硬件性能所需的工程技能；
- 为解决方案挑选合适的变量（超参数）组合的经验 ——《动手学深度学习》

深度学习基础

- 深度学习中的挑战源自
  - 非线性
  - 非凸
  - 模型复杂——深，宽
  - 参数多
  - 数据量大
  - ✓ 实际上，就是模型复杂度和数据规模的均衡

数据过拟合与拟合不够之间需要达到一种平衡，这种中间状态是我们想要的

大纲

- 机器学习基础
- 人工神经网络概述
- 深度学习基础
- 典型神经网络模型
  - 卷积神经网络
  - 循环神经网络

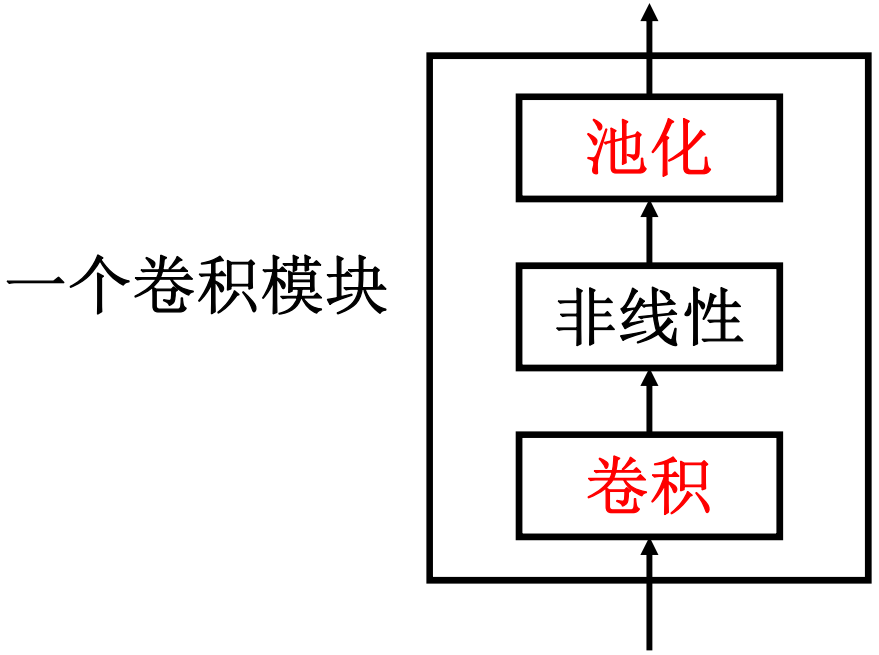
典型神经网络模型

- 卷积神经网络
- 循环神经网络

卷积神经网络

卷积神经网络(convolutional NN)

- 至少在网络的一层中用卷积运算来代替一般矩阵乘法运算的神经网络



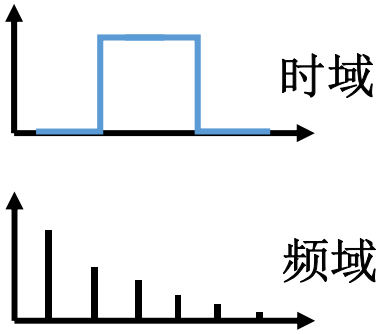
卷积神经网络

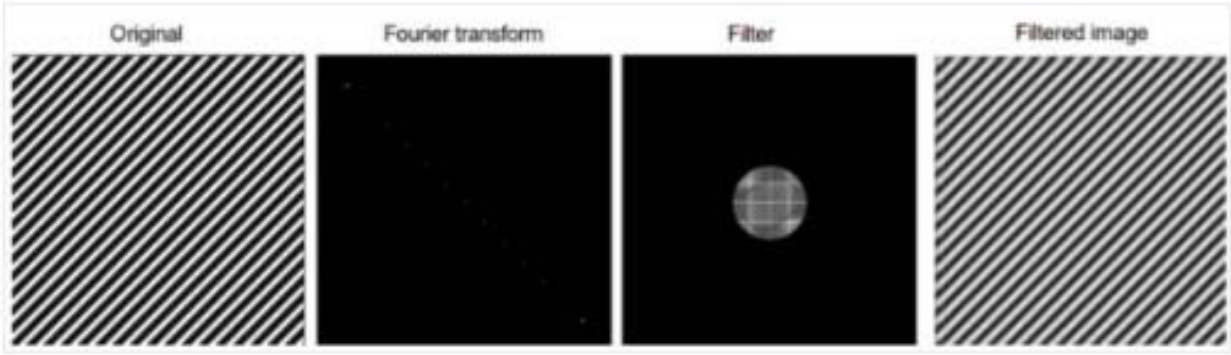
卷积

- 卷积过程是在混合两种信息

$$s(t) = \int_{-\infty}^{\infty} x(\tau)w(t-\tau)\,d\tau, \qquad s(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)w(t-\tau)$$

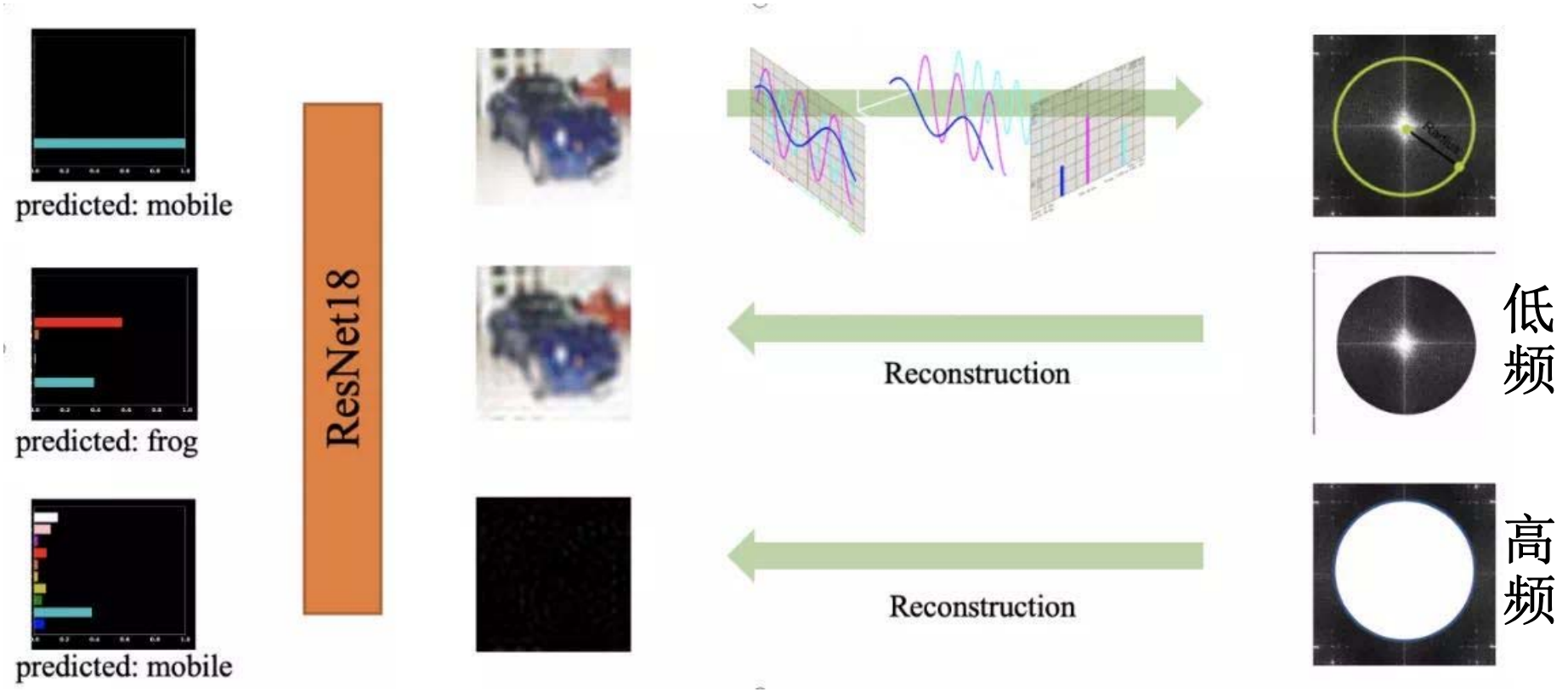
- 目的是滤波 $\int_{-\infty}^{\infty} x(\tau)w(t-\tau)\,d\tau = \mathcal{F}^{-1}\left(\sqrt{2\pi}\mathcal{F}|x|\mathcal{F}|w|\right)$





原始图像　　傅里叶变换后　　卷积核(滤波器)　　卷积后图像(变模糊,滤去了高频信息)

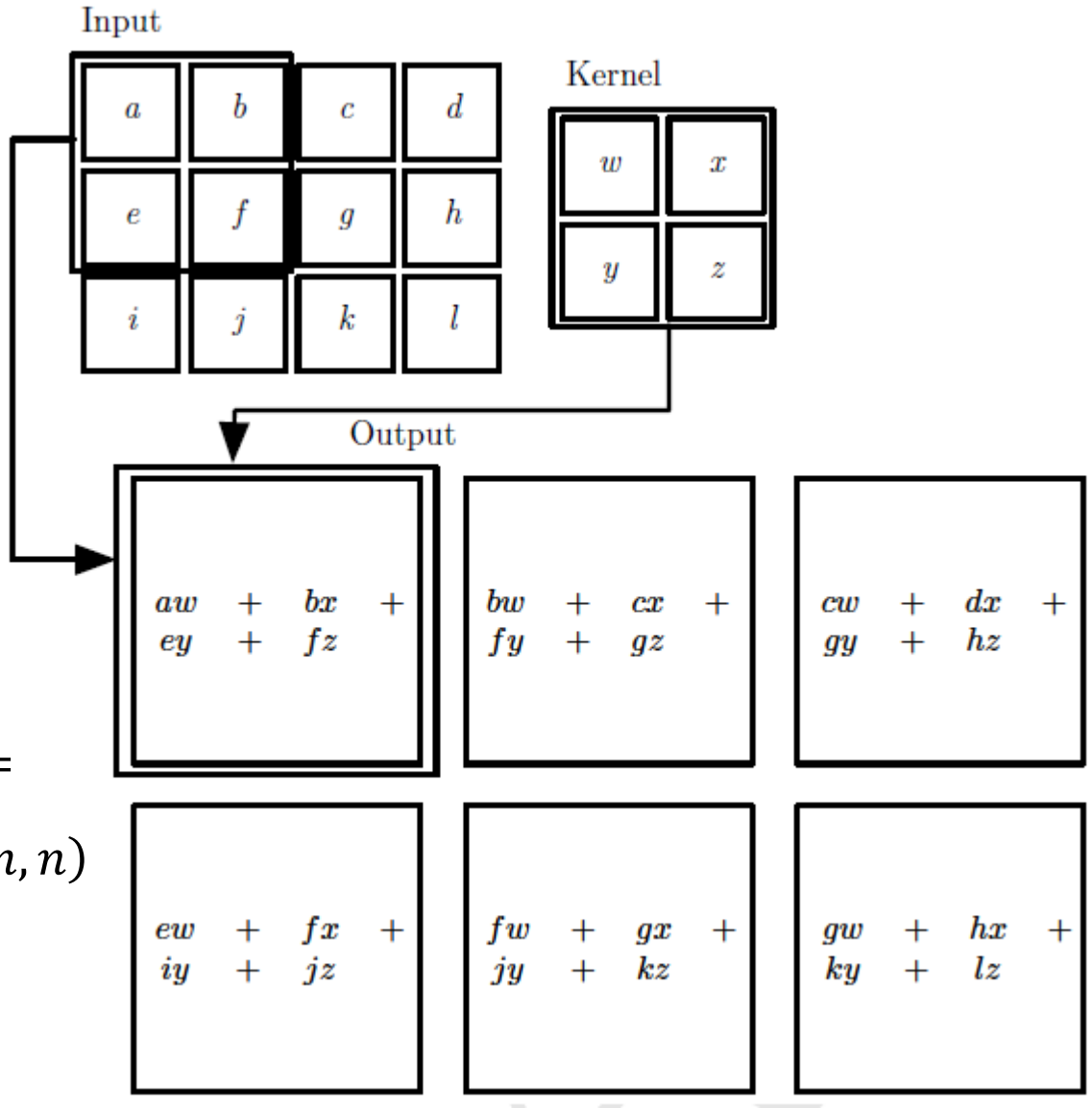
卷积神经网络



2020: High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks

卷积神经网络

- 神经网络中的卷积操作

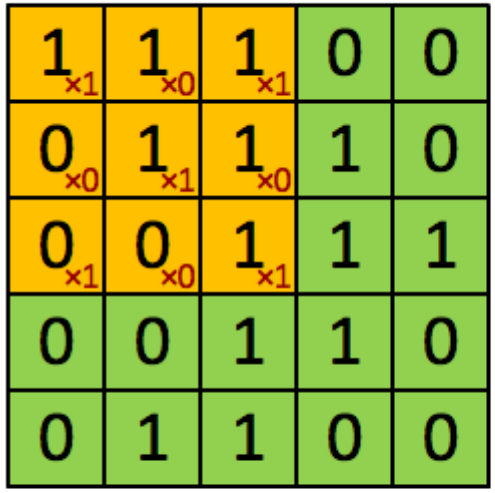


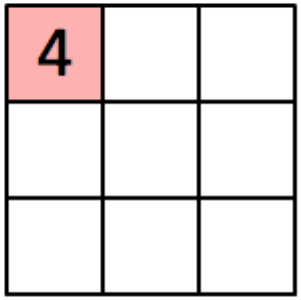
一个**2D**卷积的例子

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

卷积神经网络

- 神经网络中的卷积操作





一个**2D**卷积的例子

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

卷积神经网络

- ML为什么用卷积
  - 卷积可以保留“有用”的信息
    - 例如边缘检测，锐化，模糊化
  - 通过学习一个卷积核(滤波器)，学习器可以自动“抽取”最有效的特征

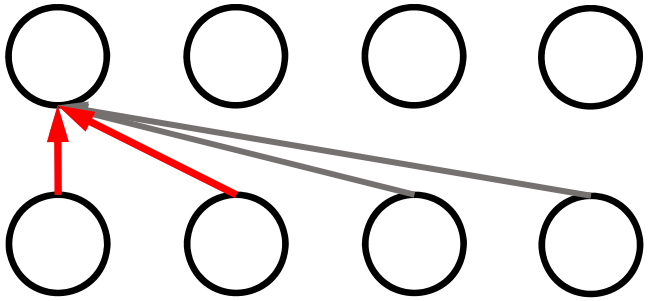
卷积神经网络

- 三个动机
  - 稀疏交互
  - 参数共享
  - 等变表示

卷积神经网络

三个动机

- 稀疏交互
  - 通过交互的局部化来探索一些小的有意义的特征
  - 直接连接的局部性，间接连接的全局性
  - 提高统计效率
- 参数共享
- 等变表示



卷积神经网络

- 三个动机
  - 稀疏交互
  - 参数共享
    - 提高统计效率
    - 保证对平移等变
    - 可扩展到不同形式的样本
  - 等变表示

卷积神经网络

三个动机

- 稀疏交互
- 参数共享
- 等变表示

$$f\big(g(x)\big) = g\big(f(x)\big)$$

  - 则说函数f对变换g等变
  - 卷积对平移变换等变——如果移动输入中的对象，它的表示也会在输出中移动相同的量

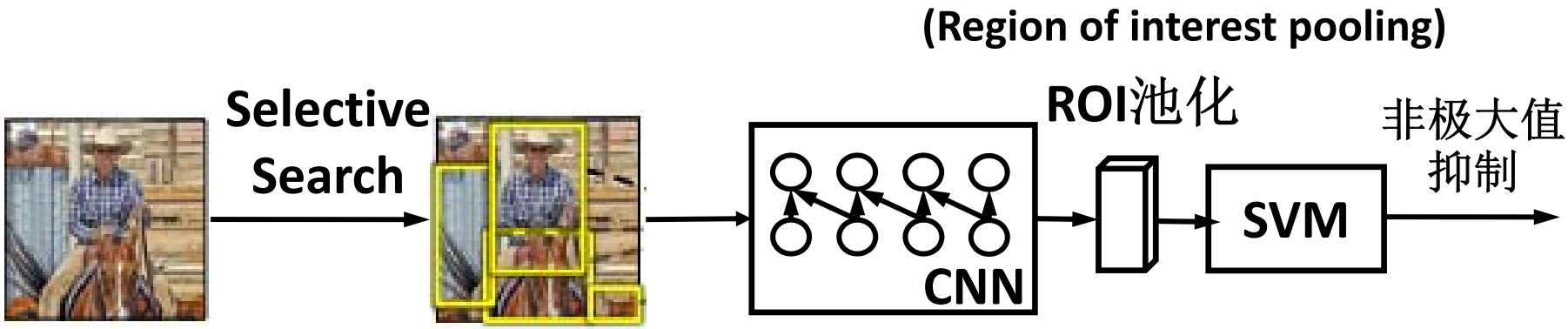
卷积神经网络

池化

- 使用某一位置的相邻输出的总体统计特征来代替网络在该位置的输出
  - 如最大池化，平均池化，L_2范数池化等
- 动机：对输入的少量平移、缩放不敏感

举例

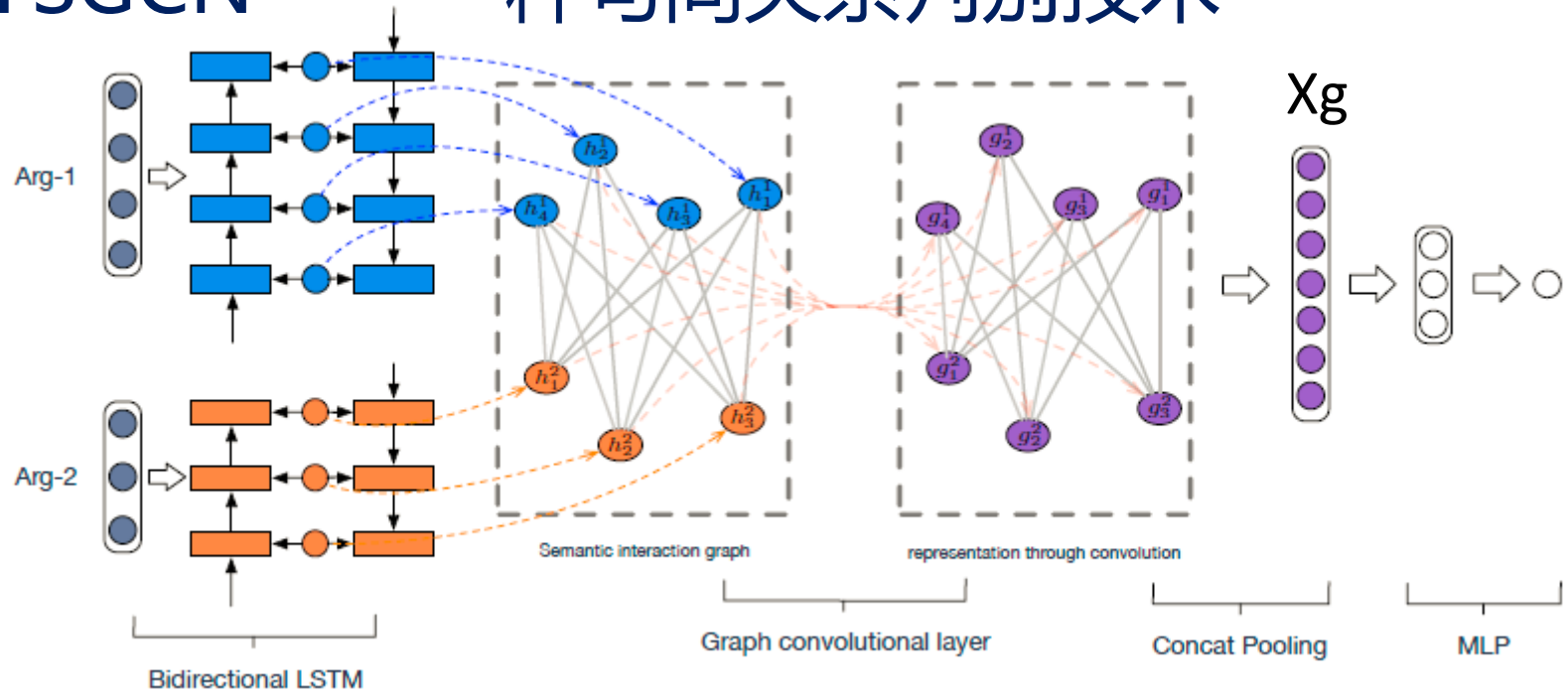
- SPPNet (Spatial Pyramid Pooling Net)，一种目标检测技术



- 针对不同候选框，用池化保证输出大小一致

举例

- SGCN——一种句间关系判别技术

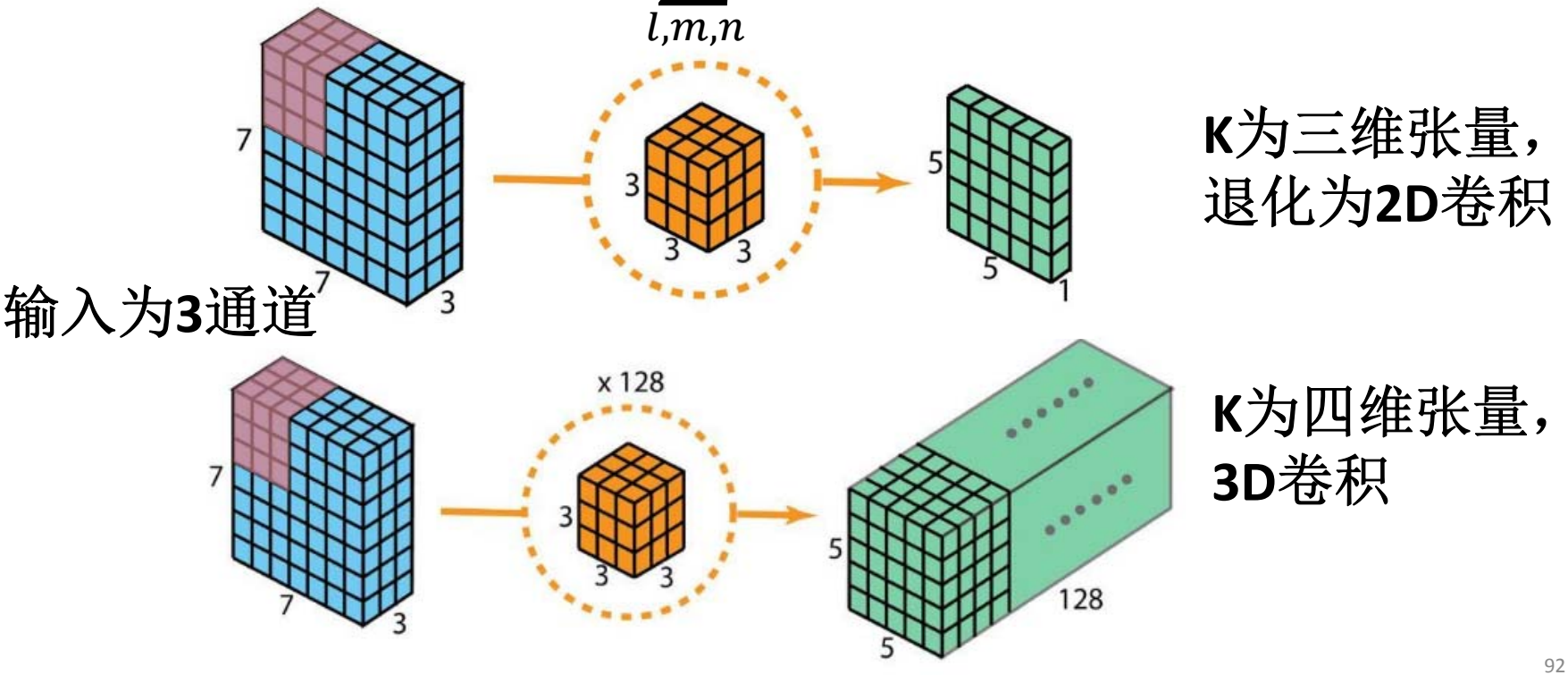


- 两个论元之间的关系由Xg 中包含的一些强语义信号确定，使用最大池化和平均池化对Xg 降维

卷积神经网络

- 卷积网络的其它问题
  - 常用的是多通道卷积—包含通道到通道的连接

$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n}$$



输入为3通道

K为三维张量，
退化为2D卷积

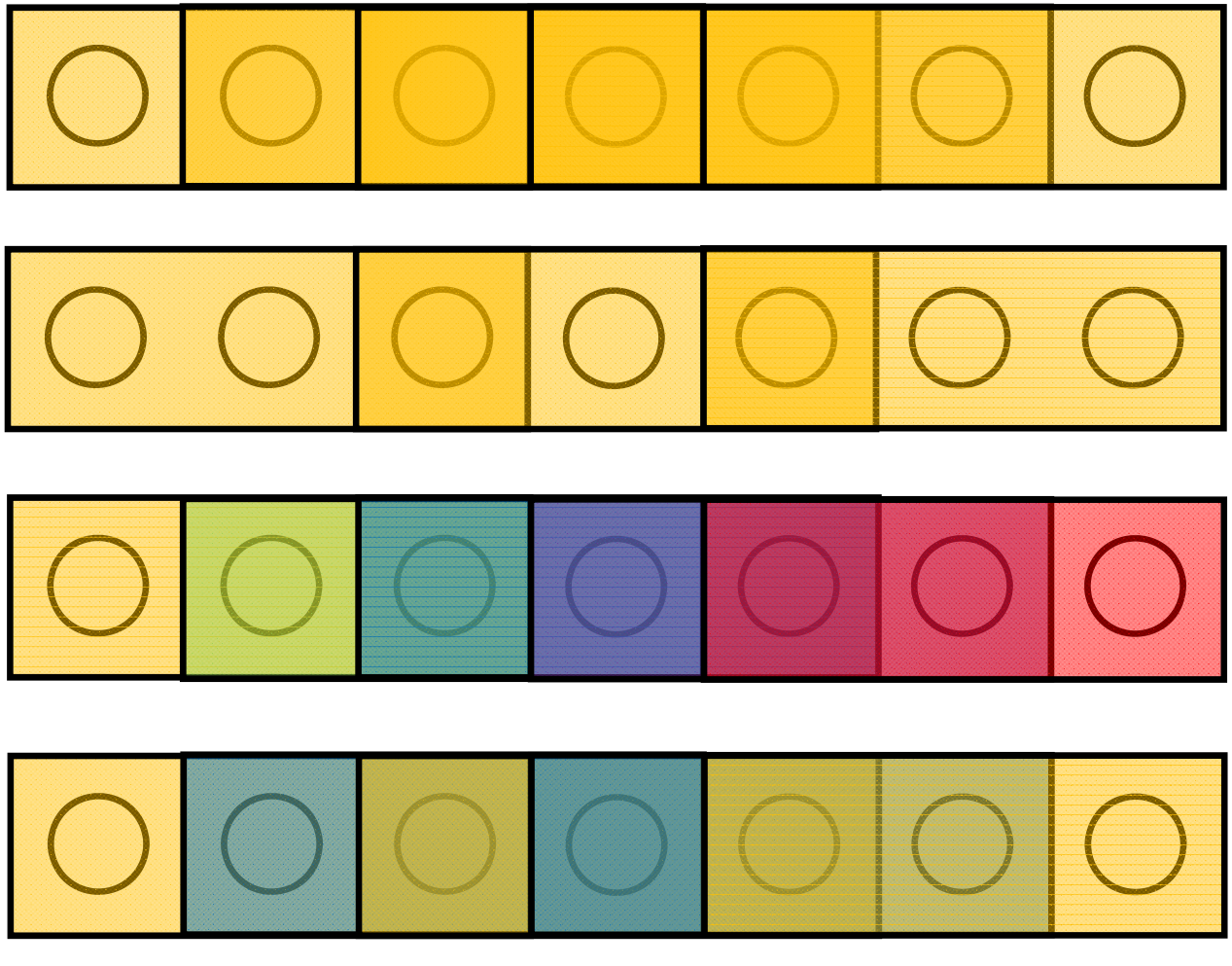
K为四维张量，
3D卷积

卷积神经网络

- 卷积网络的其它问题
  - 卷积导致的网络宽度缩减——可使用pad
    - 有效卷积、同维卷积、完全卷积
  - 带有步幅的卷积，相当于降采样
  - 局部连接——不共享参数
  - 平铺卷积——卷积和局部连接的折衷

卷积神经网络

- 卷积网络的其它问题



普通卷积

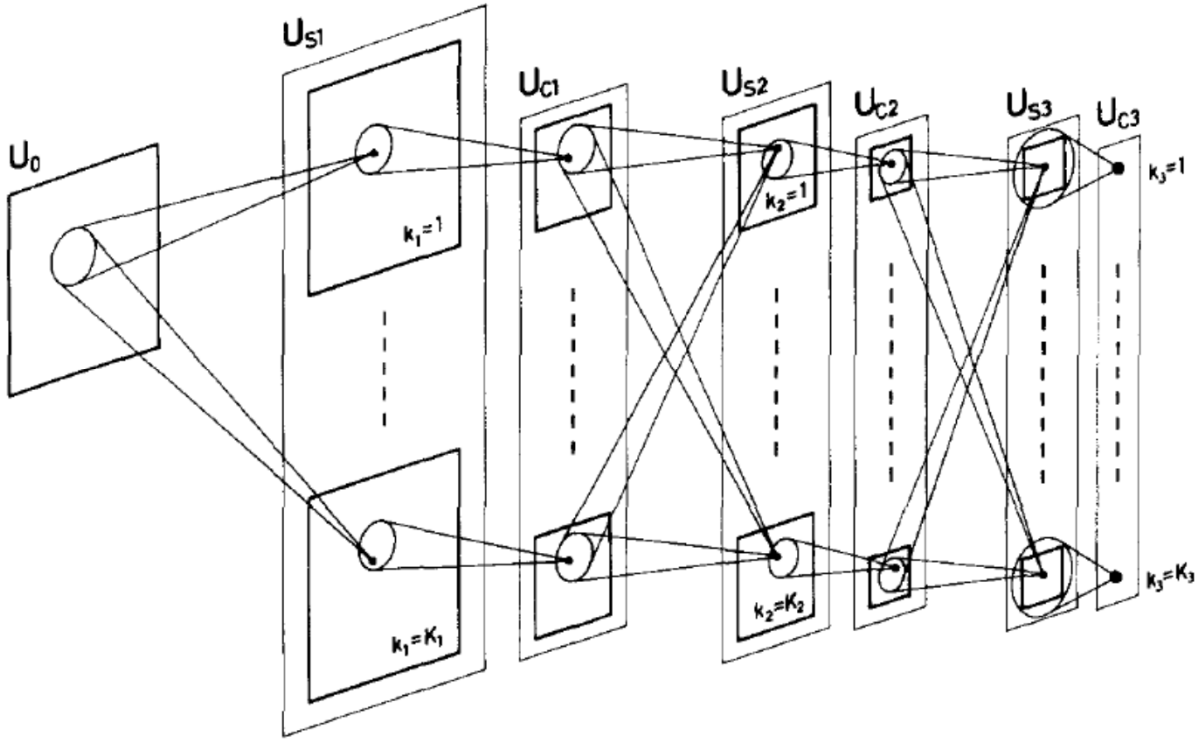
步幅卷积

局部连接—每一个卷积核都不一样

平铺卷积—相邻若干卷积核一样

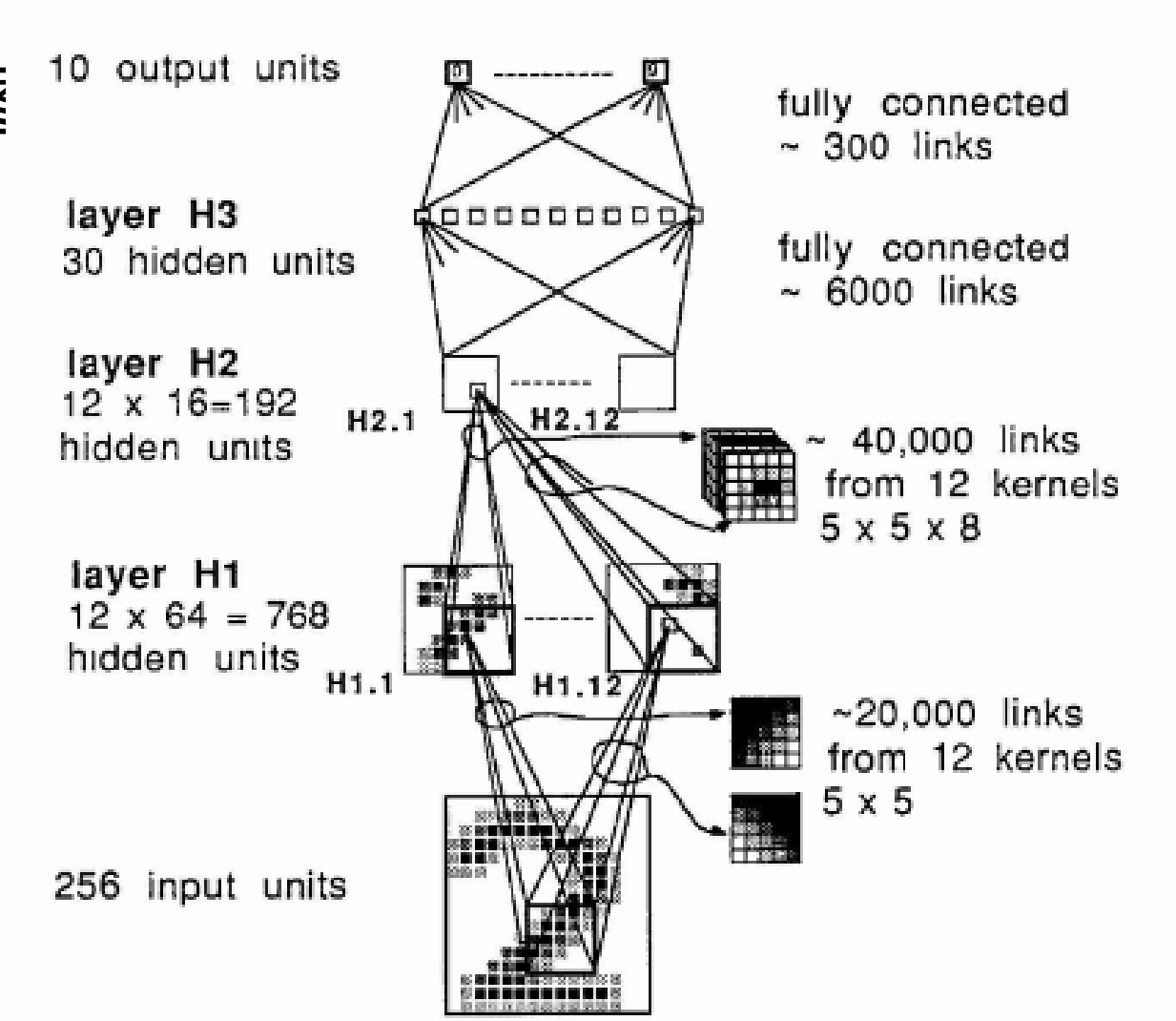
卷积神经网络

- CNN用于图像识别的发展
  - Neocognitron 1980



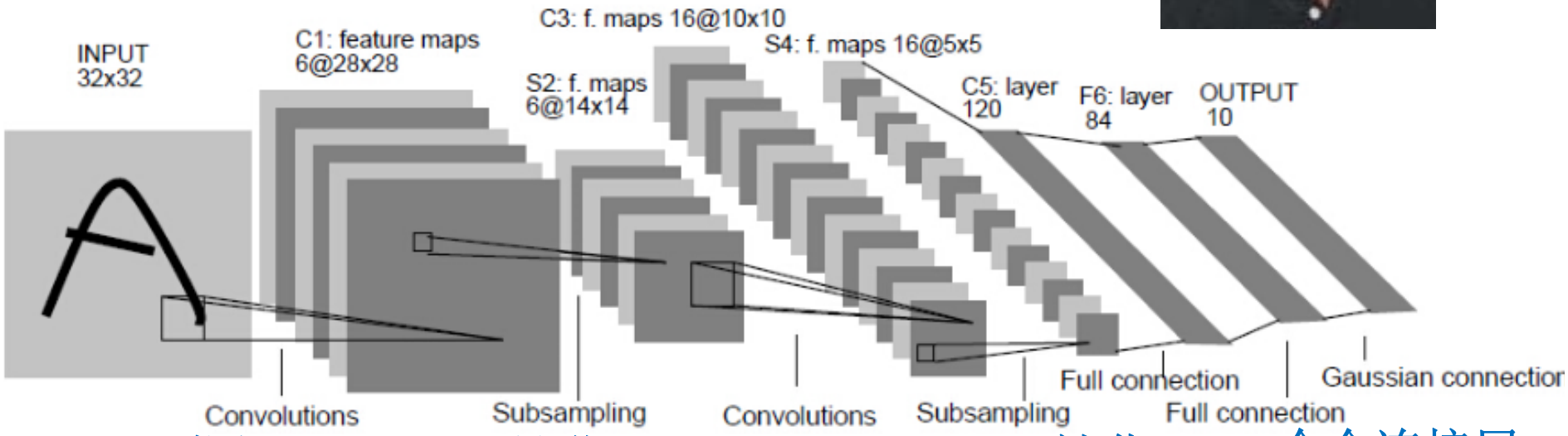
卷积神经网络

- CNN用于图
  - LeCun 1989



卷积神经网络

- CNN用于图像识别的发展
  - LeCun 1998 LeNet (LeNet5)



卷积层：
- 6输出通道，5×5卷积，2padding
- 一个sigmoid激活

池化：
- 2×2到1的平均池化

卷积层：
- 16输出通道，5×5卷积
- 一个sigmoid激活

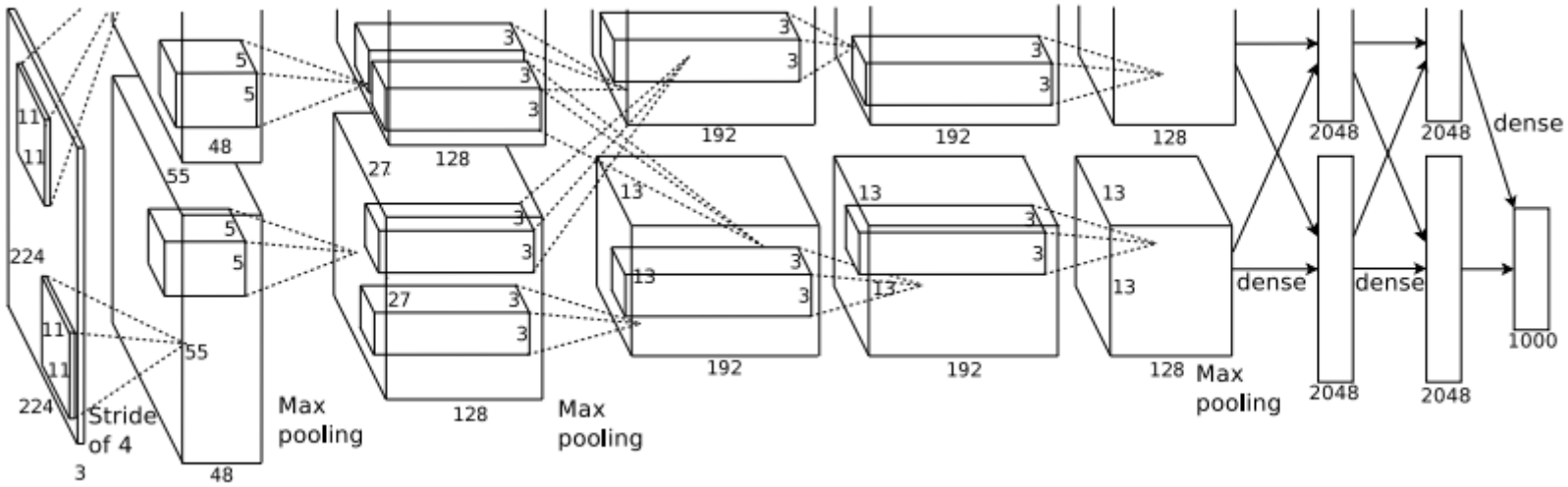
池化：
- 2×2到1的平均池化

3个全连接层

算是第一个CNN

卷积神经网络

- CNN用于图像识别的发展
  - AlexNet 2012 (Alex是Hinton的学生)



DL在图像分类任务上飞速发展的起点

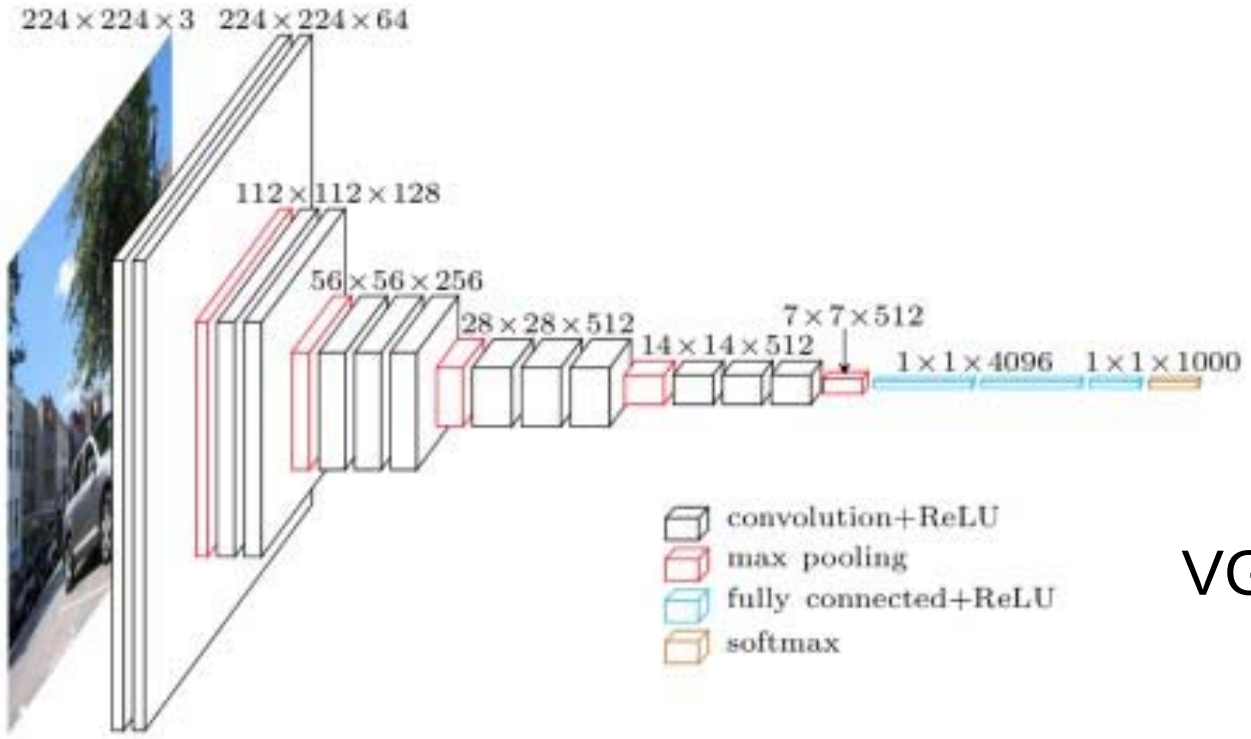
卷积神经网络

- CNN用于图像识别的发展
  - ILSVRC(ImageNet Large Scale Visual Recognition Competition)

ImageNet识别错误率

卷积神经网络

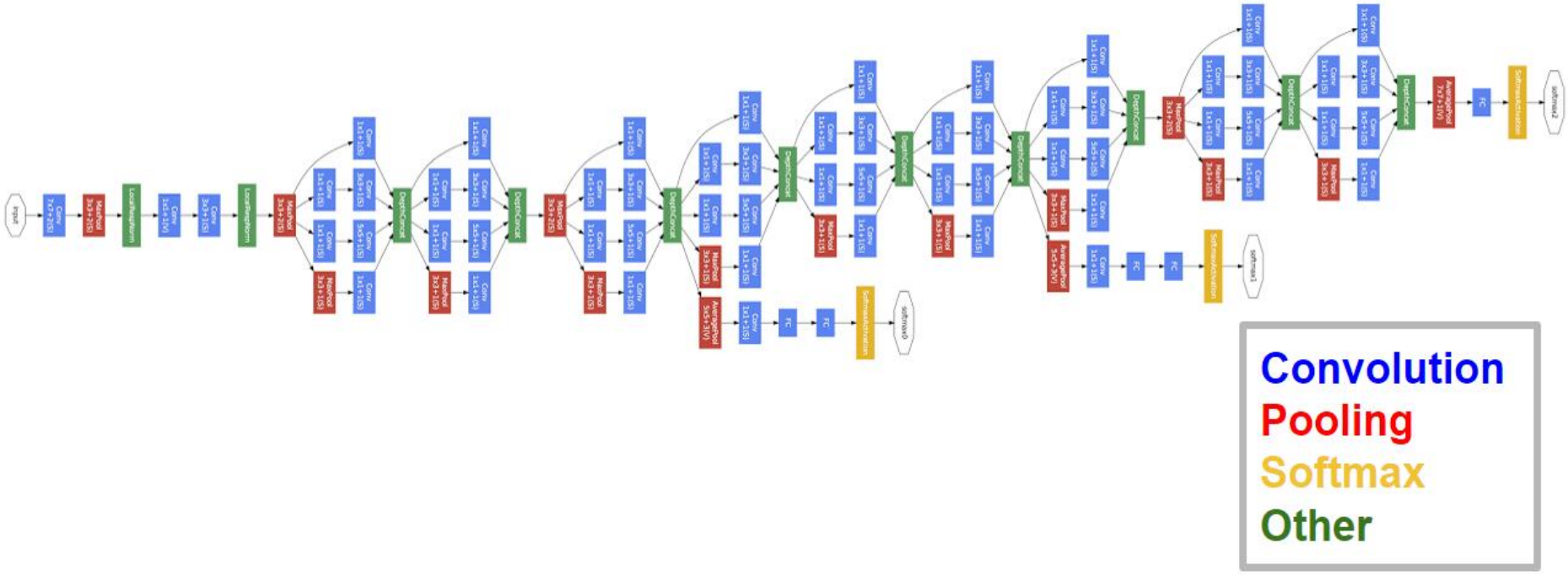
- CNN用于图像识别的发展
  - VGG



VGG16

卷积神经网络

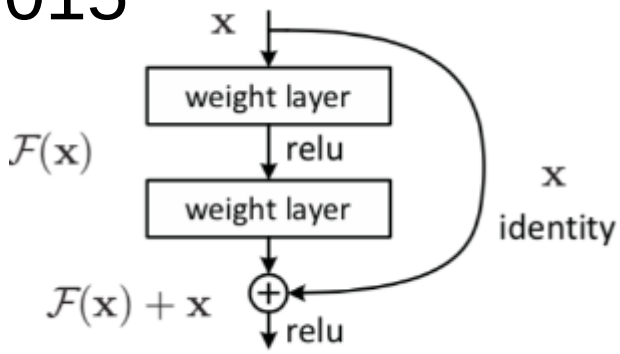
- CNN用于图像识别的发展
  - GoogLeNet 2014



卷积神经网络

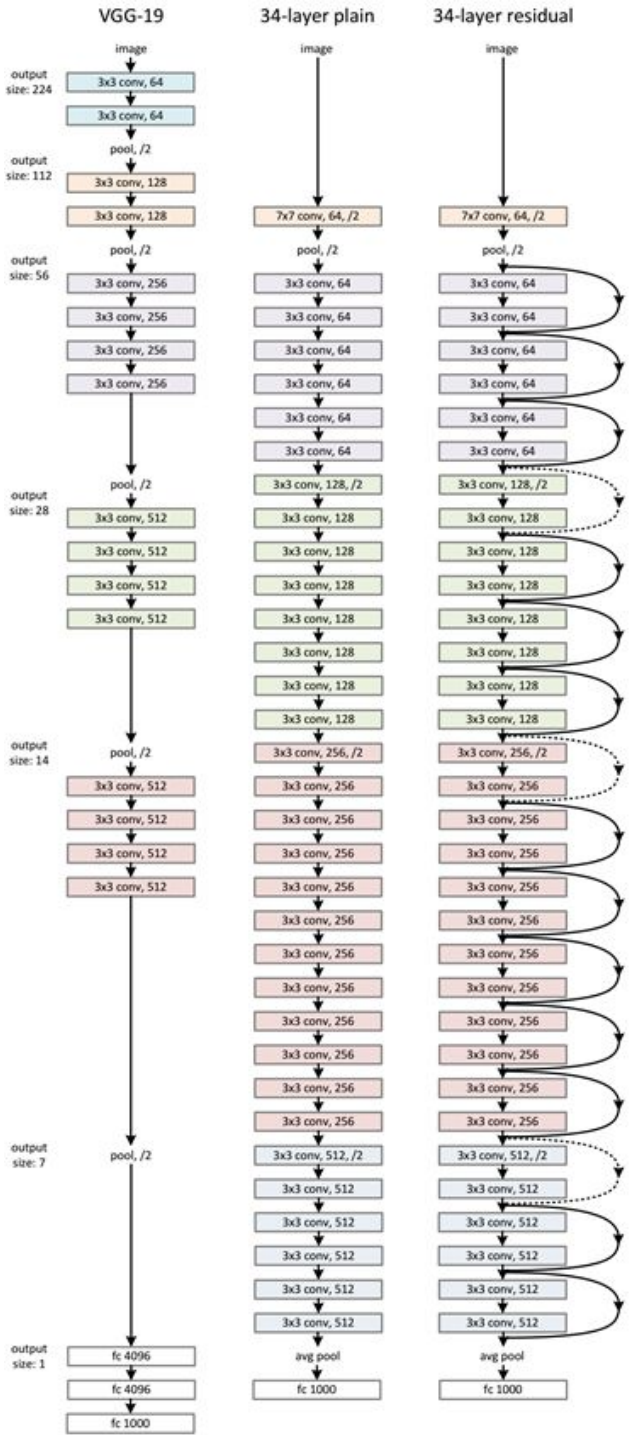
- CNN用于图像识别的发展
  - ResNet(残差网络)

2015



一个残差块

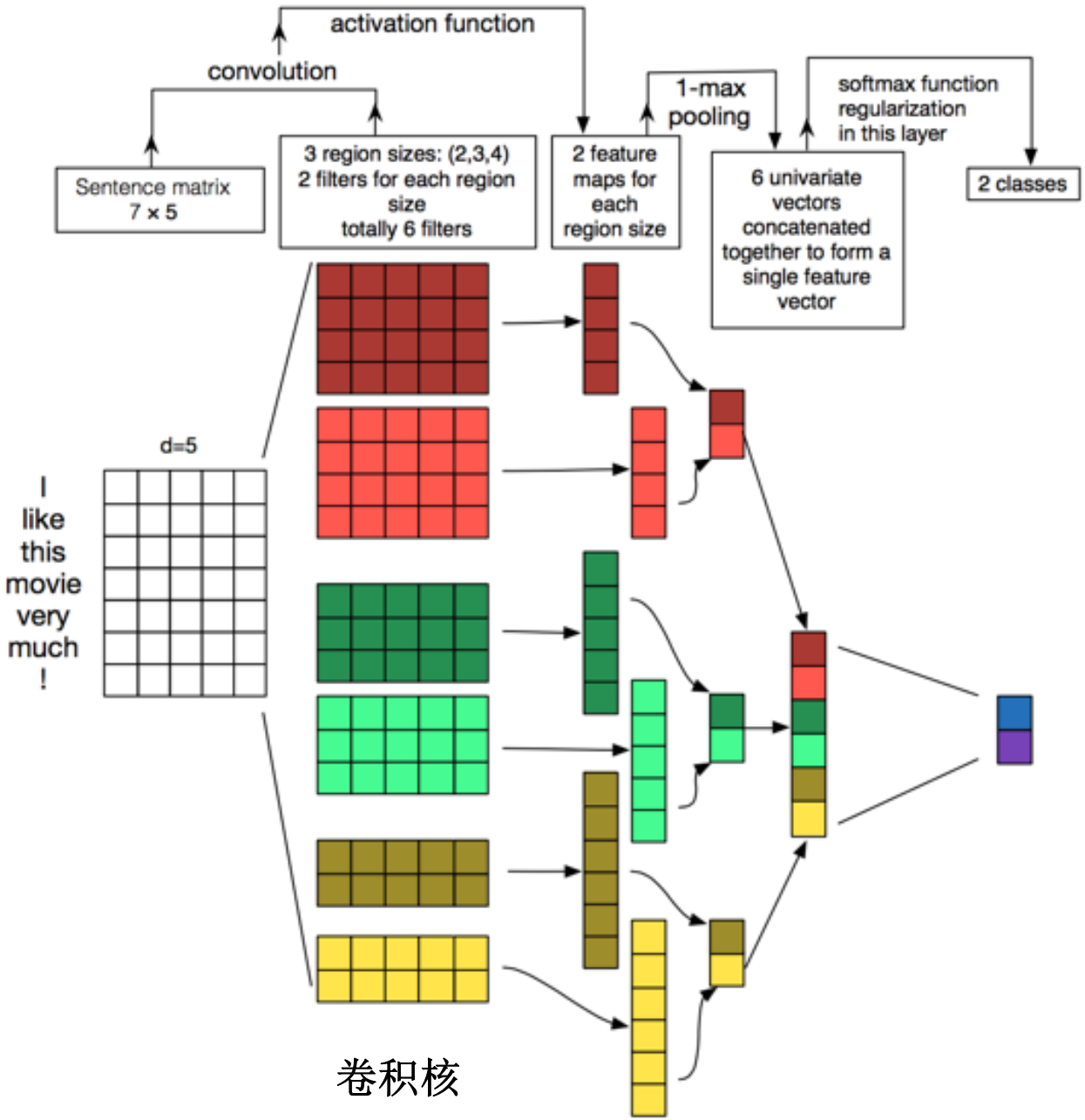
- 为了缓解模型退化——网络越深，训练越难，训练效果越差(网络可能存在冗余)



卷积神经网

- CNN用于自然语言处理

CNN用于文本分类——这里是句子情感分析

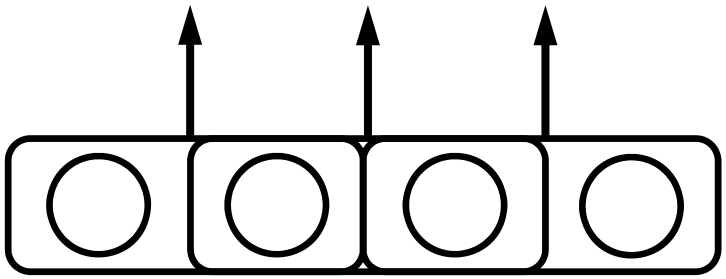


典型神经网络模型

- 卷积神经网络
- 循环神经网络

循环神经网络

- 循环神经网络(recurrent NN)
  - CNN适合于网格化数据
    - 也可以在一维时间序列上使用卷积

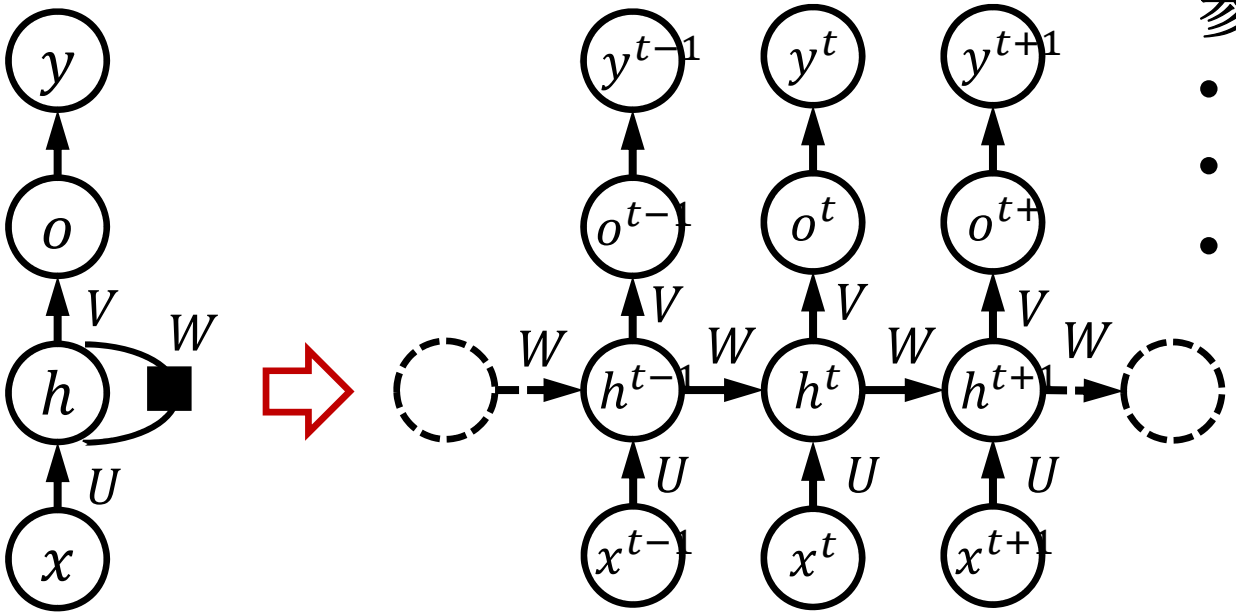


    - 但，是浅层的
  - RNN适合于序列化数据

循环神经网络

N to N → 预测，语言模型，序列标注

RNN



参数随序列共享：

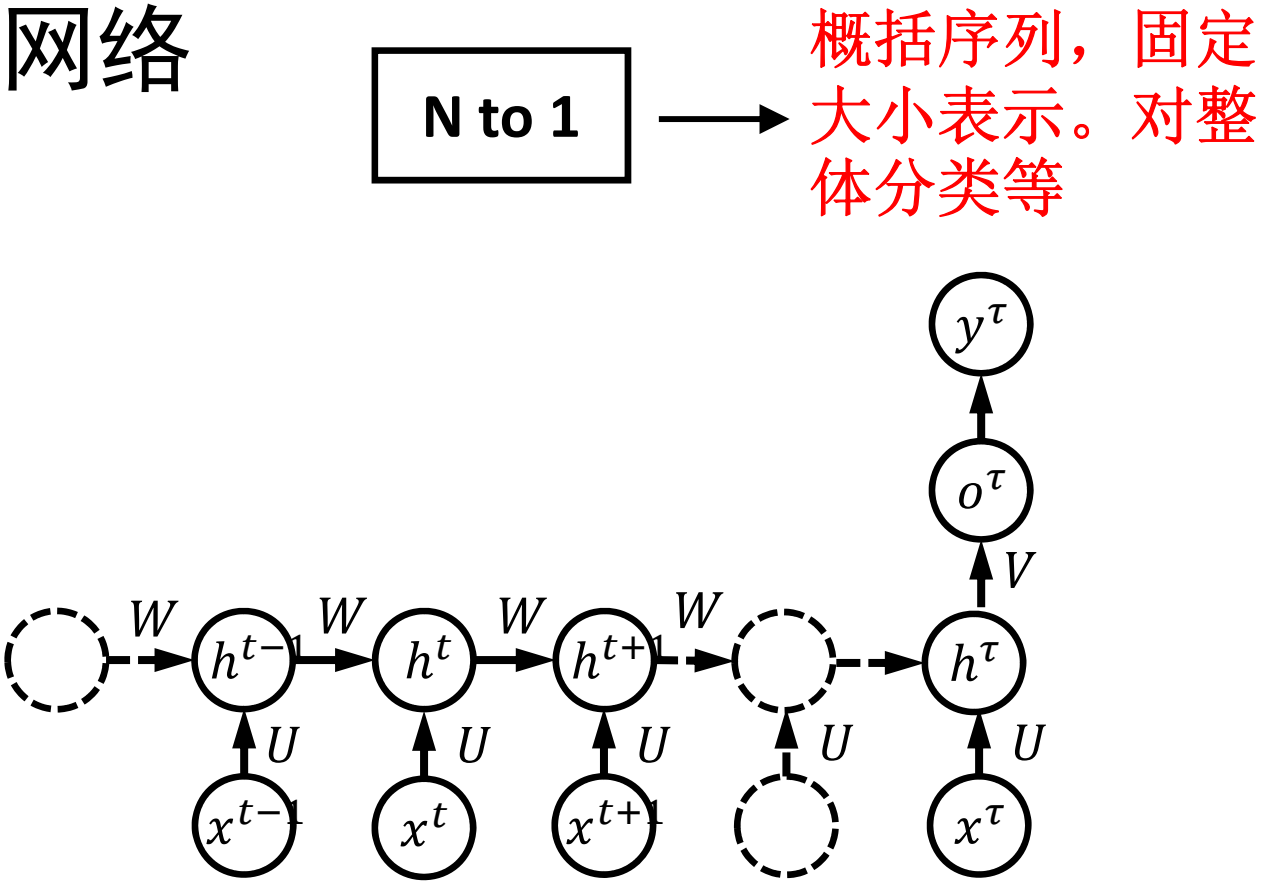
- 提高统计效率
- 对平移等变
- 可扩展到不同形式的样本

$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}$$
$$\boldsymbol{h}^{(t)} = \tanh\left(\boldsymbol{a}^{(t)}\right)$$
$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + V\boldsymbol{h}^{(t)}$$
$$\hat{y}^{(t)} = \mathrm{softmax}\left(\boldsymbol{o}^{(t)}\right)$$

通常情况下，会在表示层(*h*)之上加全连接层(*o*)和**Softmax**以用于分类

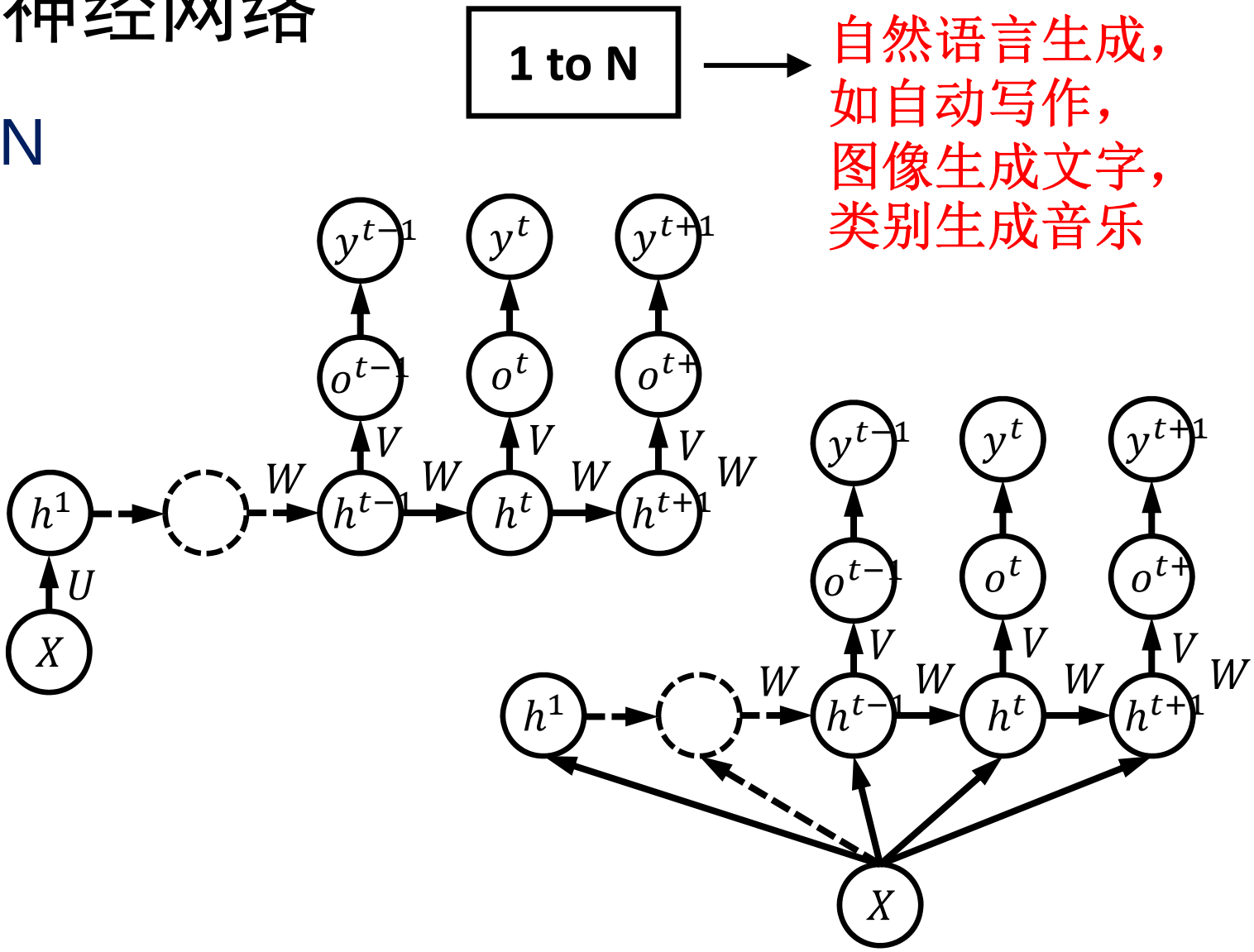
循环神经网络

- RNN


N to 1
概括序列，固定大小表示。对整体分类等
y^τ
o^τ
V
h^τ
W
h^{t-1}
W
h^t
W
h^{t+1}
W
U
U
U
U
U
x^{t-1}
x^t
x^{t+1}
x^τ

循环神经网络

- RNN


1 to N
自然语言生成，
如自动写作，
图像生成文字，
类别生成音乐
y^{t-1}
y^t
y^{t+1}
o^{t-1}
o^t
o^{t+}
V
W
h^1
h^{t-1}
h^t
h^{t+1}
U
X

循环神经网络

- LSTM(long short-term memory)RNN的改进
  - RNN的状态计算公式决定了反向传播的导数包含每一步梯度的连乘——梯度消失或爆炸

$$\boldsymbol{h}^{(t)} = f\left(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}\right)$$

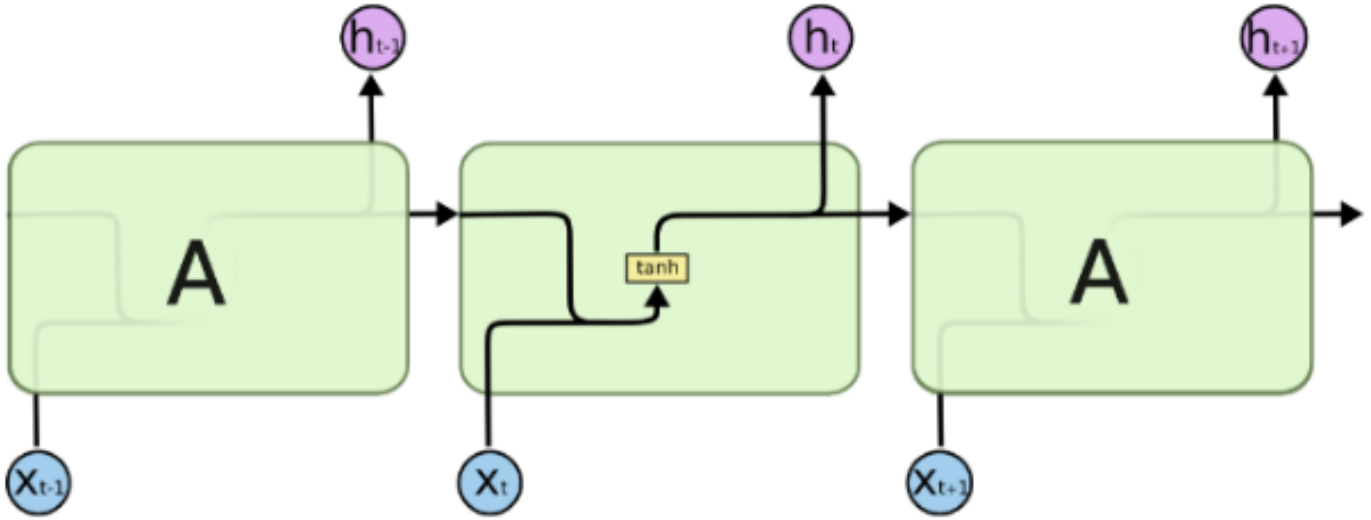
  - 因此难以处理“远程依赖”

循环神经网络

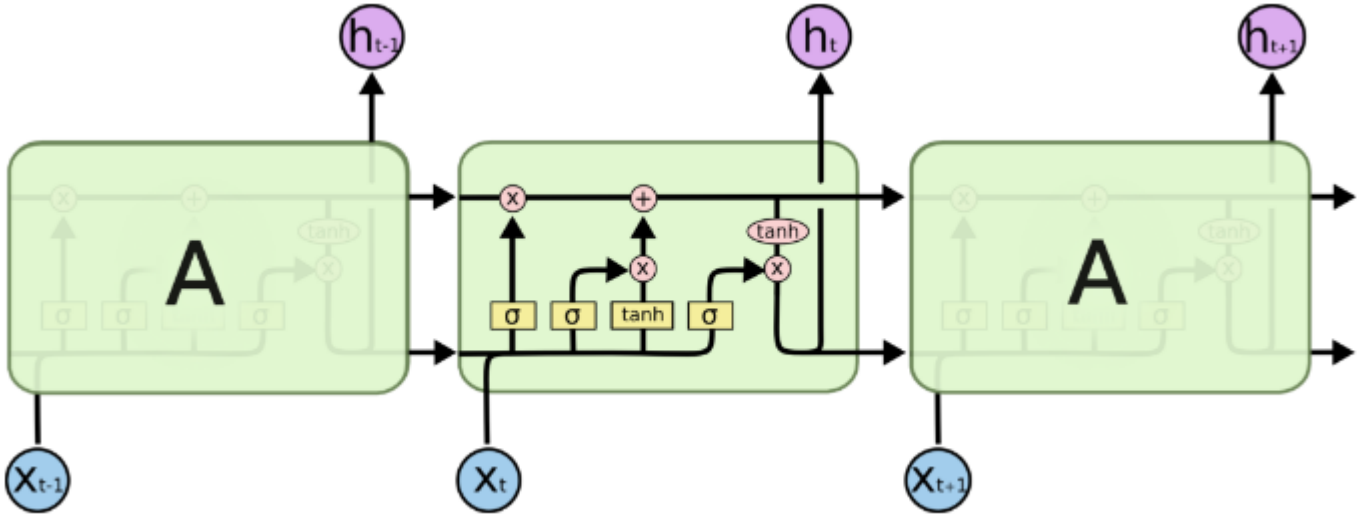
RNN: $\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}$

$\boldsymbol{h}^{(t)} = \tanh\left(\boldsymbol{a}^{(t)}\right)$

RNN



LSTM



循环神经网络

LSTM

$$\boldsymbol{f}^{(t)} = \sigma\left(\boldsymbol{b}_f + \boldsymbol{W}_f \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_f \boldsymbol{x}^{(t)}\right)$$

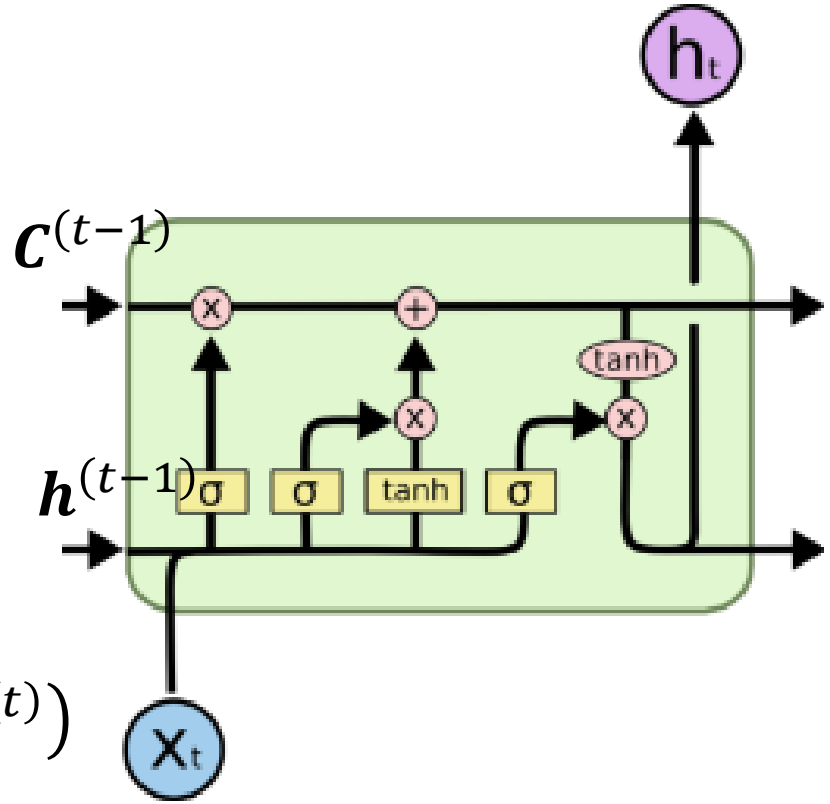
$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{b}_i + \boldsymbol{W}_i \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_i \boldsymbol{x}^{(t)}\right)$$

$$\widetilde{\boldsymbol{C}}^{(t)} = \tanh\left(\boldsymbol{b}_C + \boldsymbol{W}_C \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_C \boldsymbol{x}^{(t)}\right)$$

$$\boldsymbol{C}^{(t)} = \boldsymbol{f}^{(t)} * \boldsymbol{C}^{(t-1)} + \boldsymbol{i}^{(t)} * \widetilde{\boldsymbol{C}}^{(t)}$$

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{b}_o + \boldsymbol{W}_o \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_o \boldsymbol{x}^{(t)}\right)$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} * \tanh(\boldsymbol{C}_t)$$

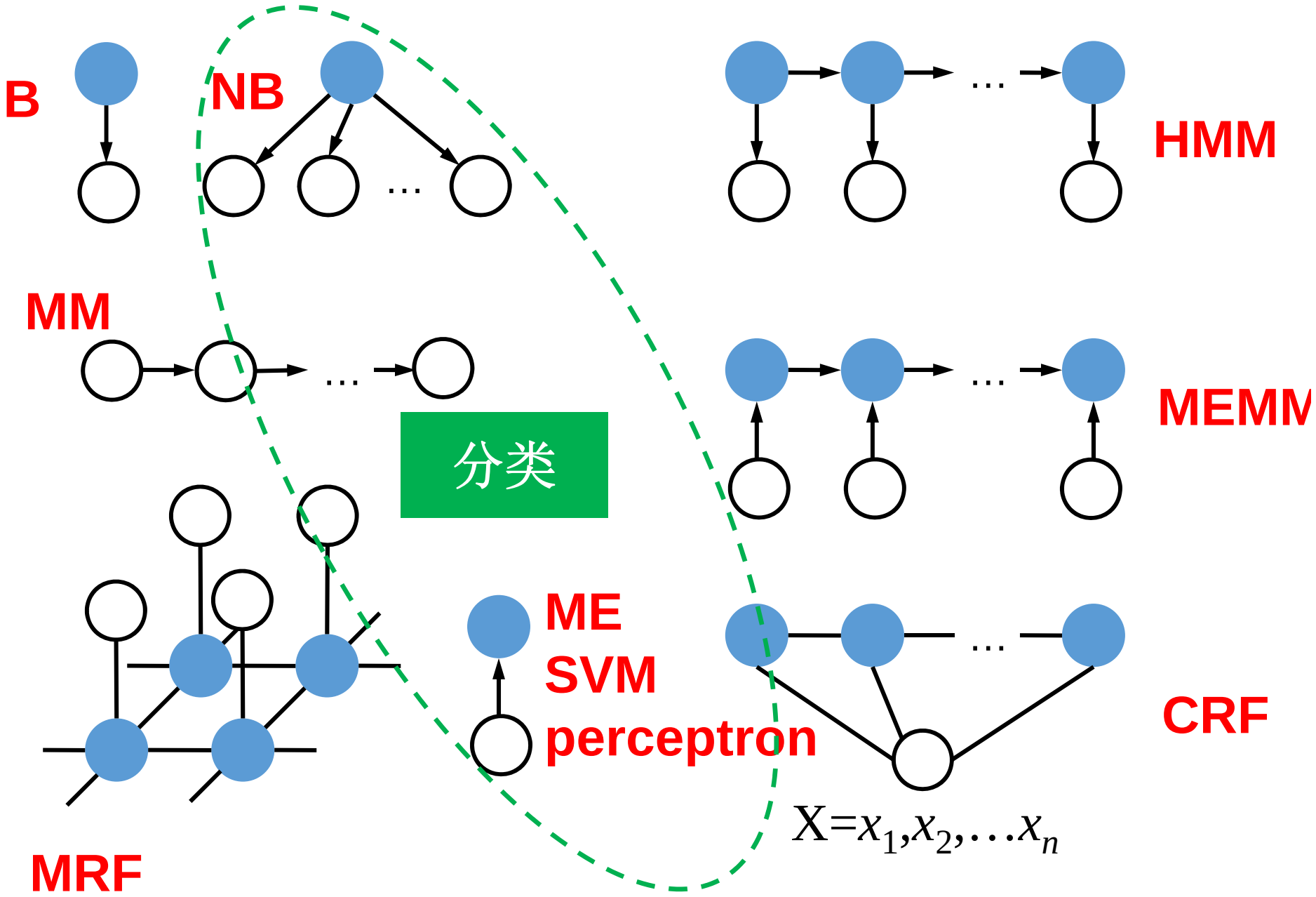


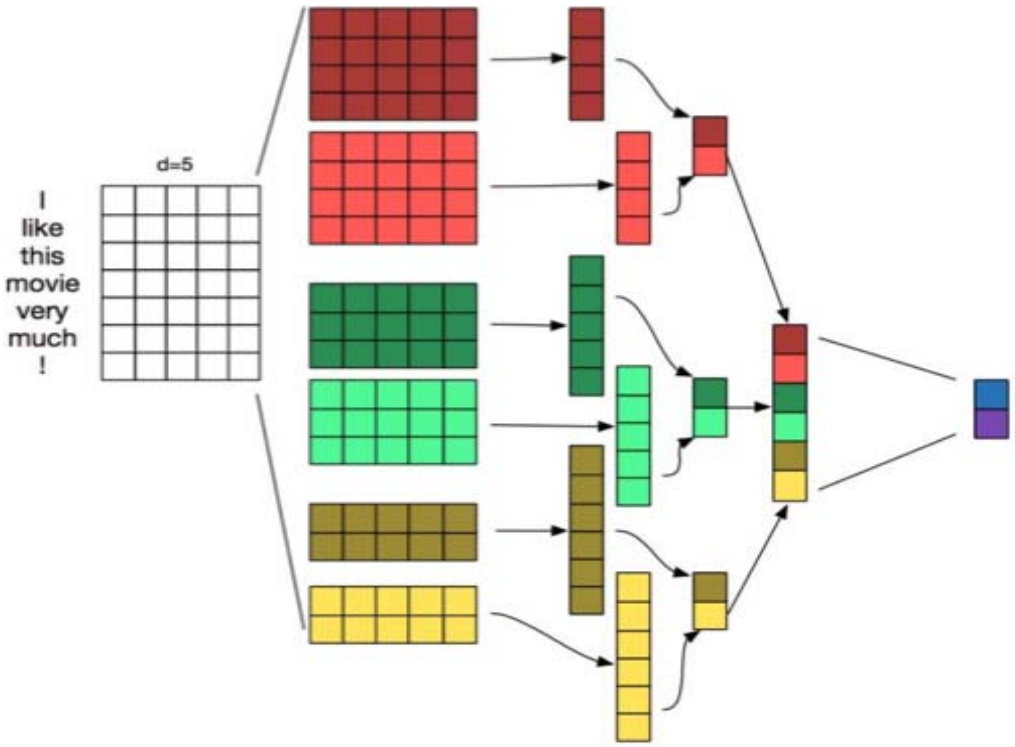
核心思想：

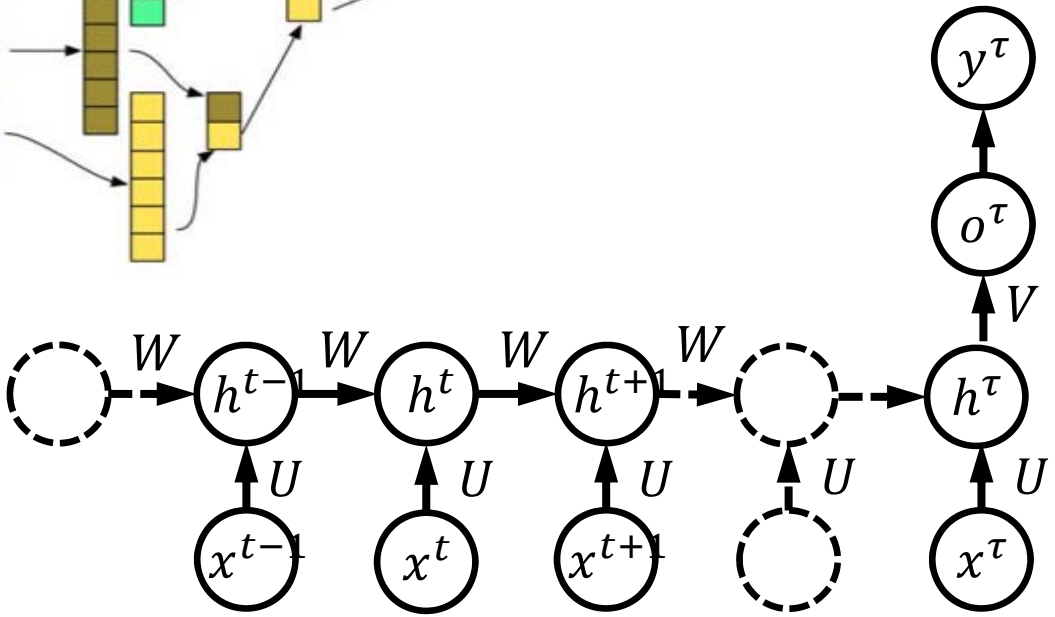
- 梯度长时间持续流动的路径——$\boldsymbol{C}^{(t)}$，其自循环权重可动态改变

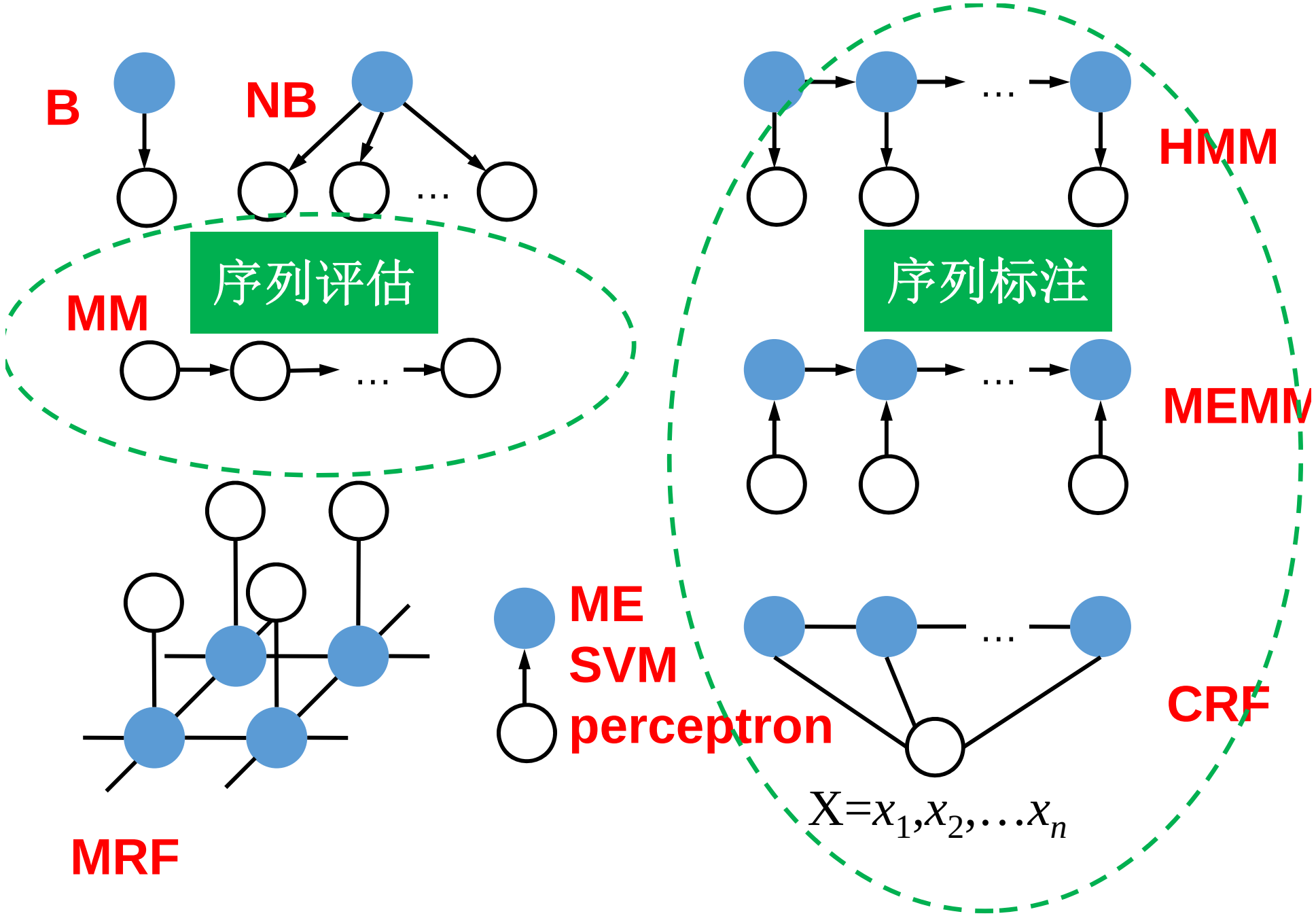
多层NN可用以后

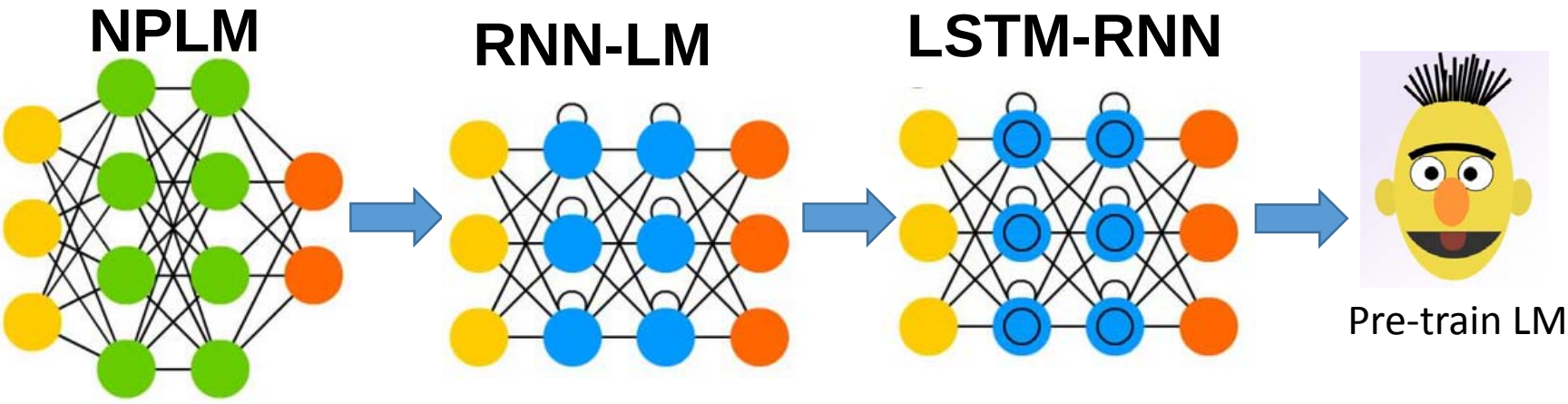
- NLP的分类问题
  - NB, ME, SVM → FNN, CNN, RNN
- NLP的序列评估问题
  - n-gram → NPLM, LSTMLM, BERT
- NLP的序列标注问题
  - HMM, CRF → RNN, Bi-LSTM, BERT

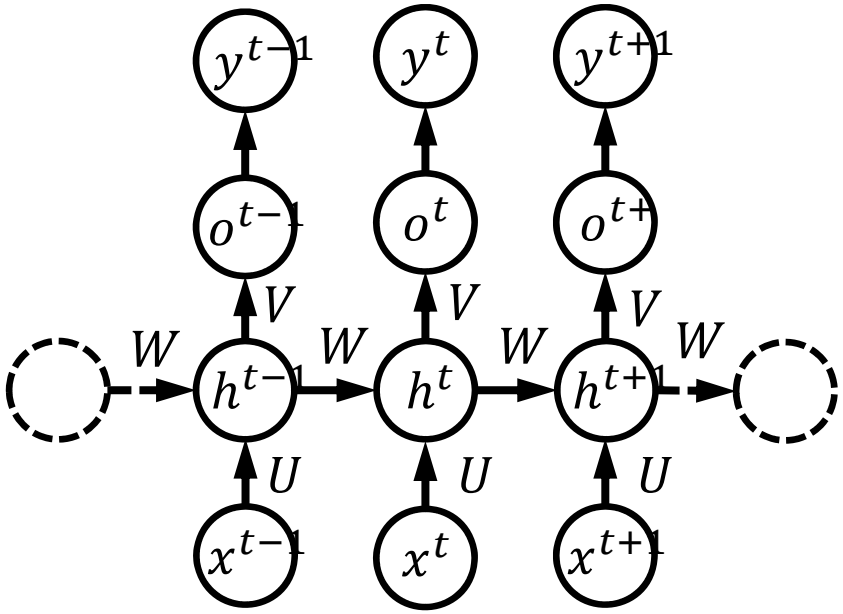

B
NB
HMM
MM
MEMM
分类
ME
SVM
perceptron
CRF
MRF
X=x1,x2,…xn


d=5
I
like
this
movie
very
much
!


y^τ
o^τ
V
W
W
W
W
h^{t-1}
h^t
h^{t+1}
h^τ
U
U
U
U
U
x^{t-1}
x^t
x^{t+1}
x^τ

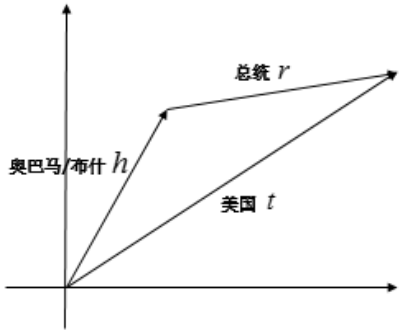

B
NB
…
序列评估
MM
…
HMM
序列标注
…
MEMM
…
CRF
ME
SVM
perceptron
MRF
X=x1,x2,…xn


NPLM
RNN-LM
LSTM-RNN
Pre-train LM

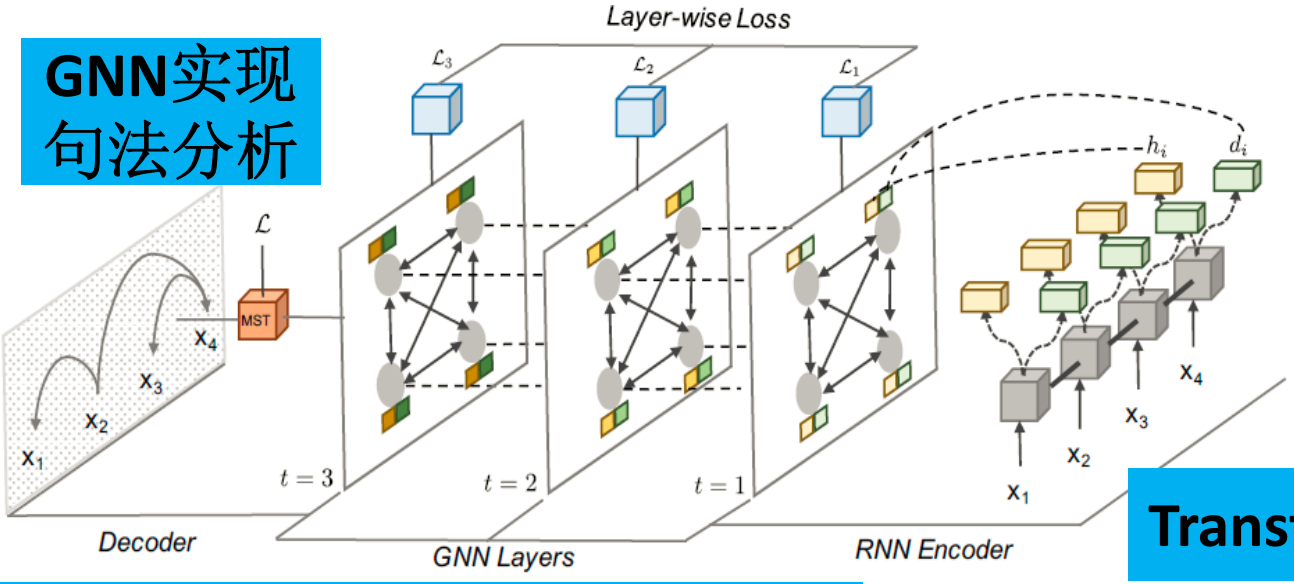

y^{t-1}
y^t
y^{t+1}
o^{t-1}
o^t
o^{t+1}
V
W
h^{t-1}
h^t
h^{t+1}
U
x^{t-1}
x^t
x^{t+1}

多层NN可用以后

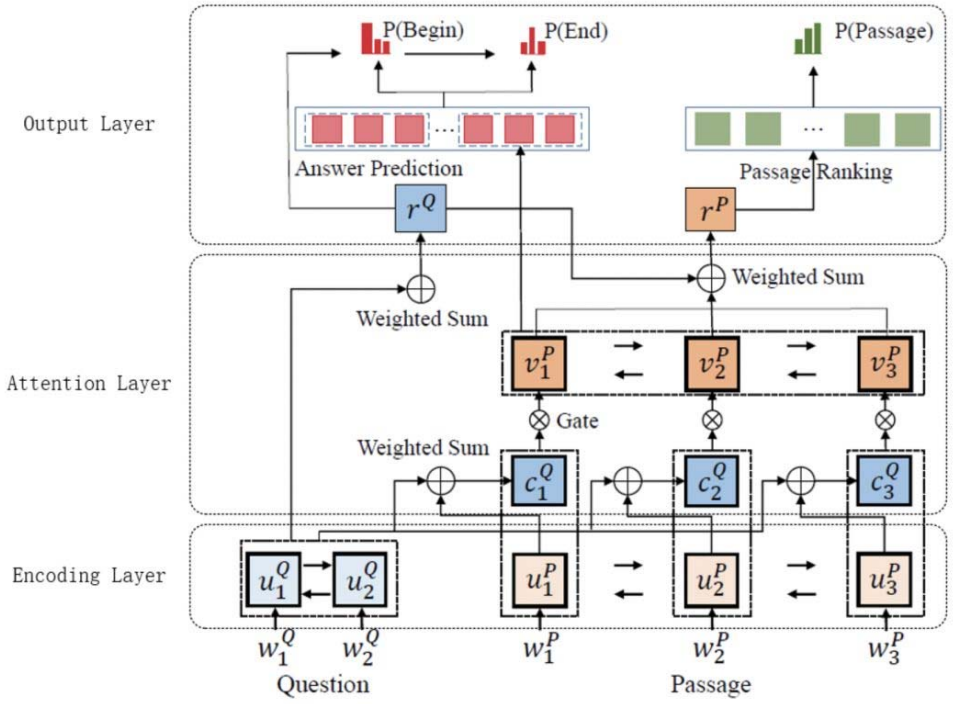
- NLP的结构预测问题
  - CYK, MST, seq-classifier → RecursiveNN, GNN, 基于NN的分类
- NLP的序列转换问题
  - Noisy channel, 基于ME的判别式模型→ Seq2Seq
- NLP的匹配问题
  - 基于logic form的知识匹配→
  - 编辑距离、字符匹配→



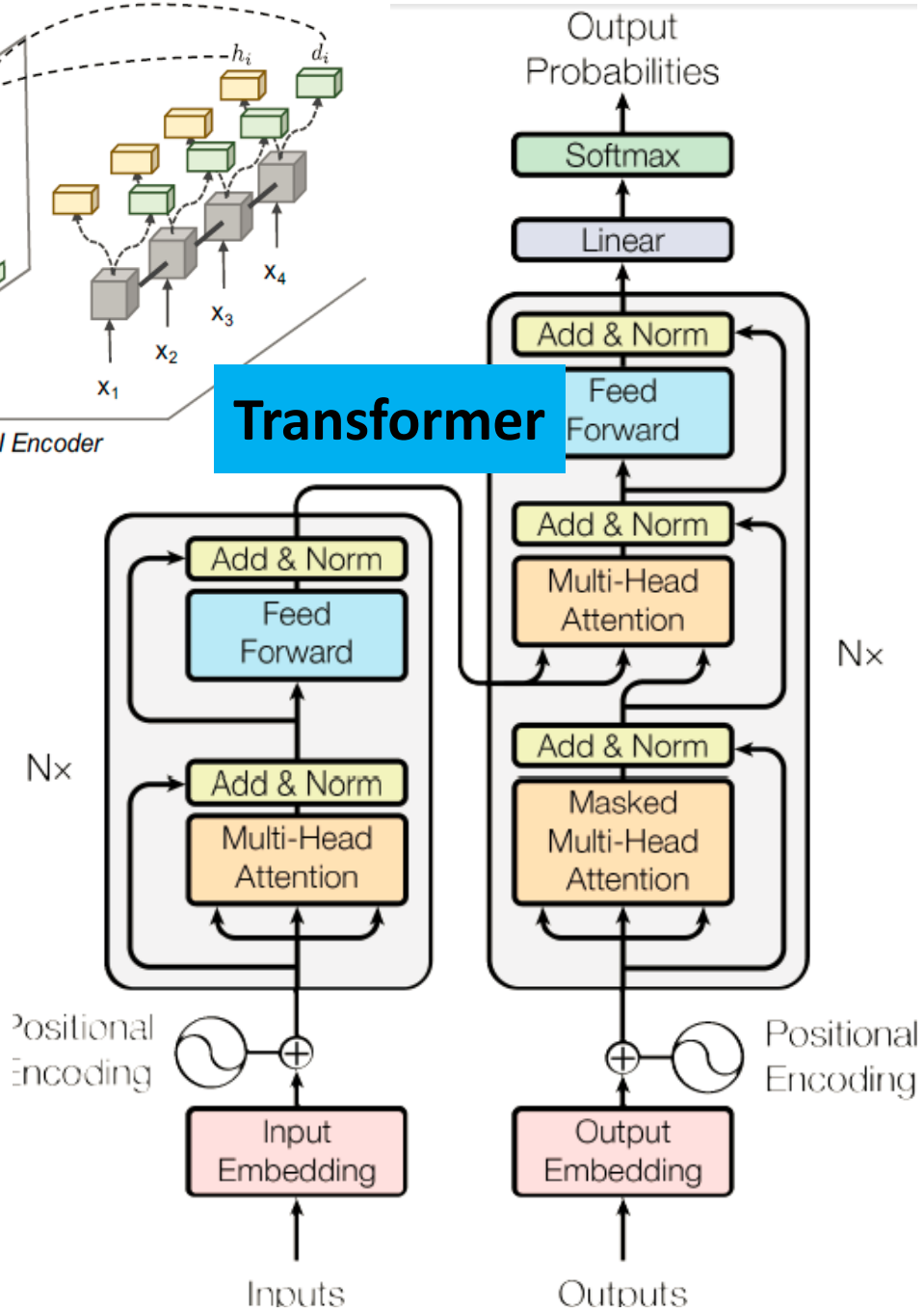
GNN实现句法分析



Point网络实现基于文档的问答



Transformer



参考书目

- **Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learing*. MA: MIT Press. 2016.**

  赵申剑等译. 深度学习. 北京: 人民邮电出版社. 2017.

FNN（逻辑回归）实现句子情感分类

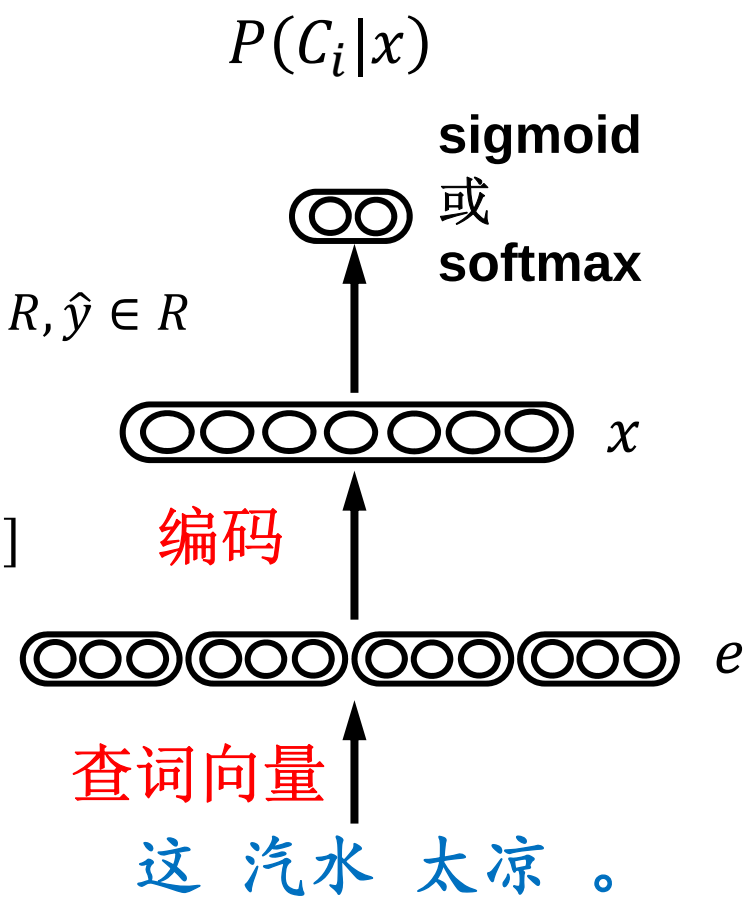
$$类别 = \begin{cases} 0, & \hat{y} < 0.5 \\ 1, & else \end{cases}$$

$$\hat{y} = sigmoid\left(b + \sum_{i=0}^{m-1} W_i * x_i\right),\ W \in R^m, b \in R, \hat{y} \in R$$

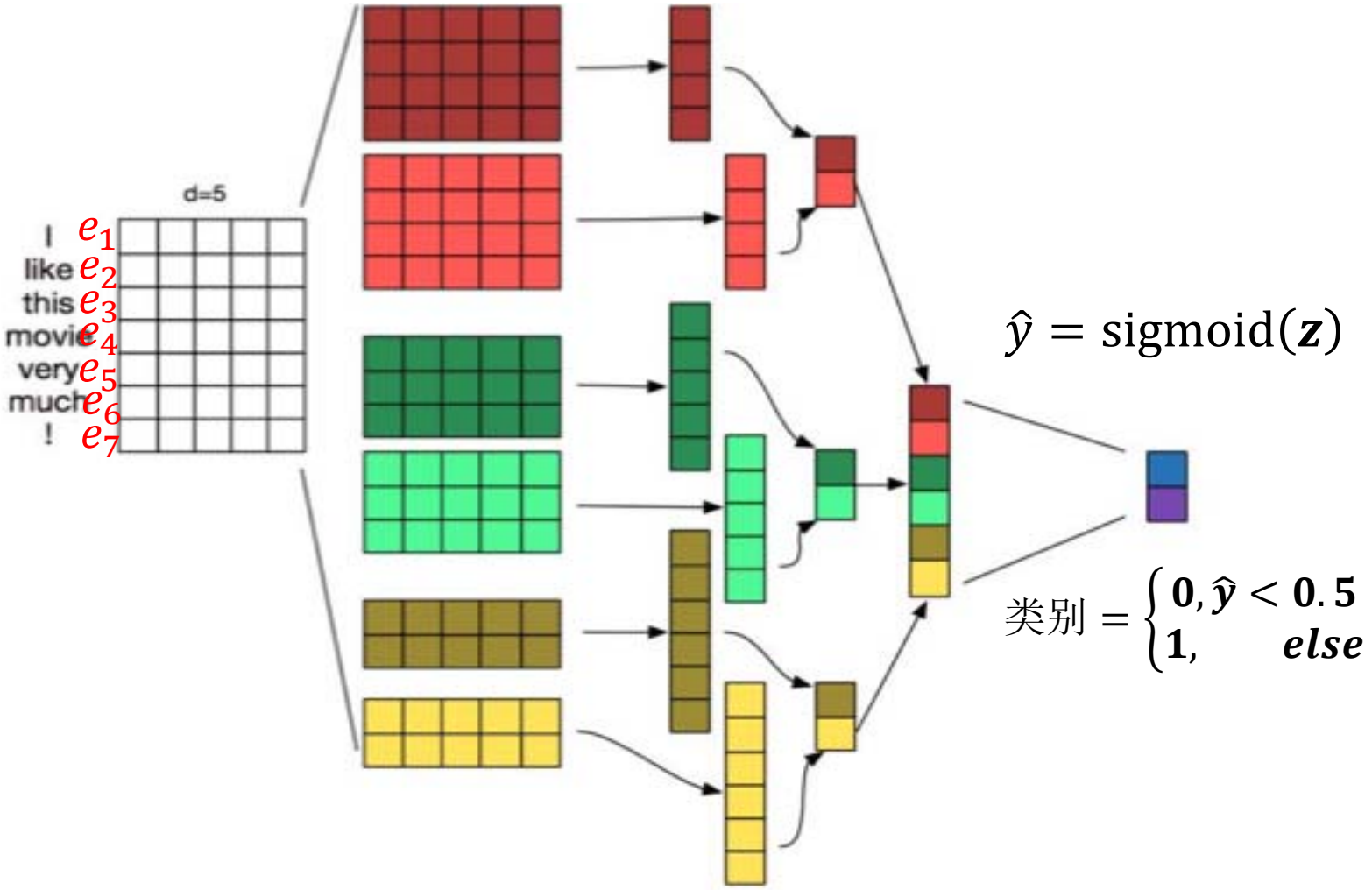
$$x = \sum_{i=1}^{4} e_i,\ x \in R^m, x = [x_0, x_1, \ldots, x_{m-1}]$$

$$e_1 \quad e_2 \quad e_3 \quad e_4$$

$$e_i \in R^m, e_i = [e_{i0}, e_{i1}, \ldots, e_{i(m-1)}]$$



CNN实现句子情感分类



RNN实现句子情感分类

$$\text{类别} = \begin{cases} \mathbf{0}, \hat{\boldsymbol{y}} < \mathbf{0.5} \\ \mathbf{1}, \quad \boldsymbol{else} \end{cases} \qquad \hat{y} = \text{sigmoid}(o^{\tau})$$