Deep Learning I

4. 多层感知机

WU Xiaokun 吴晓堃

xkun.wu [at] gmail

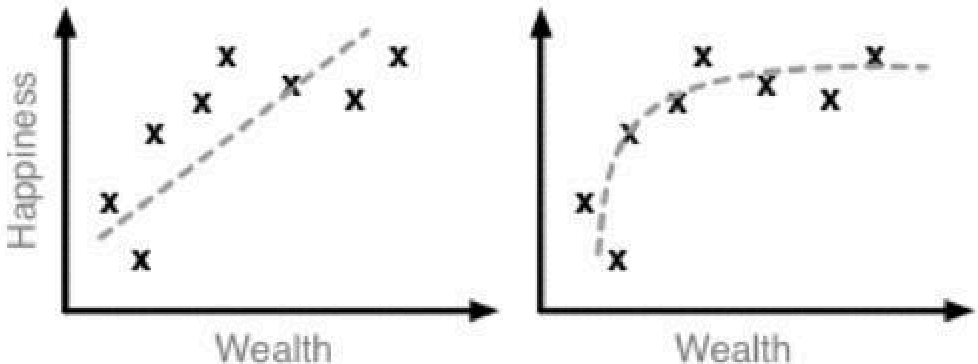
2021/03/14

感知机

线性的问题

线性意味着**单调性假设**

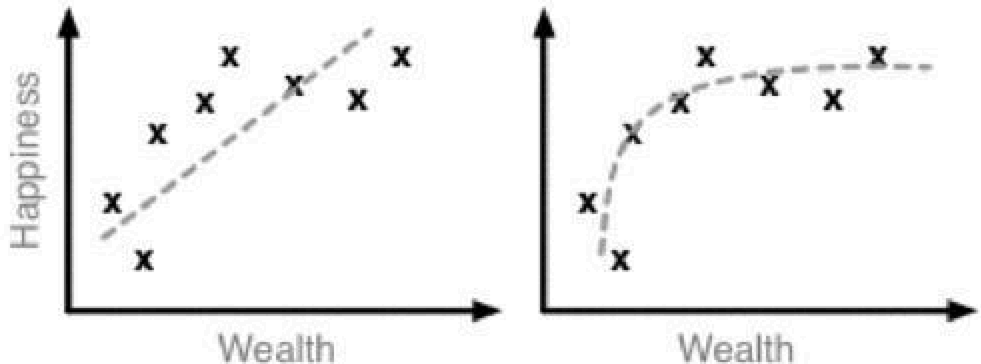
- 财富和幸福：更接近对数关系



线性的问题

线性意味着单调性假设

- 财富和幸福：更接近对数关系

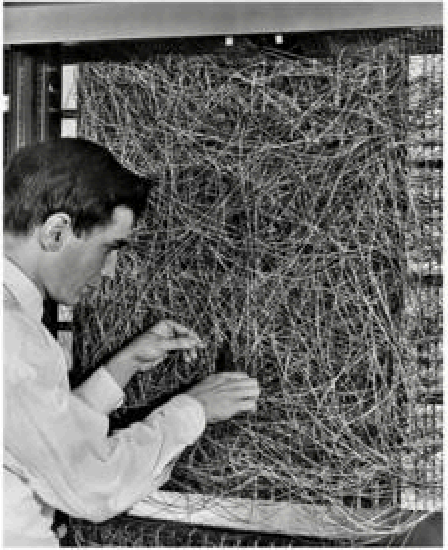


- 体温和健康：37°
- 倒立的猫：不，是图片拿反了

I型感知机

Frank Rosenblatt 和I型感知机，1960

- **感知机算法，1957**



感知机

- “感”：感受，输入
- “知”：知识，输出
- 对输入进行处理，输出结果的机器

灵感来源 I

生物神经系统

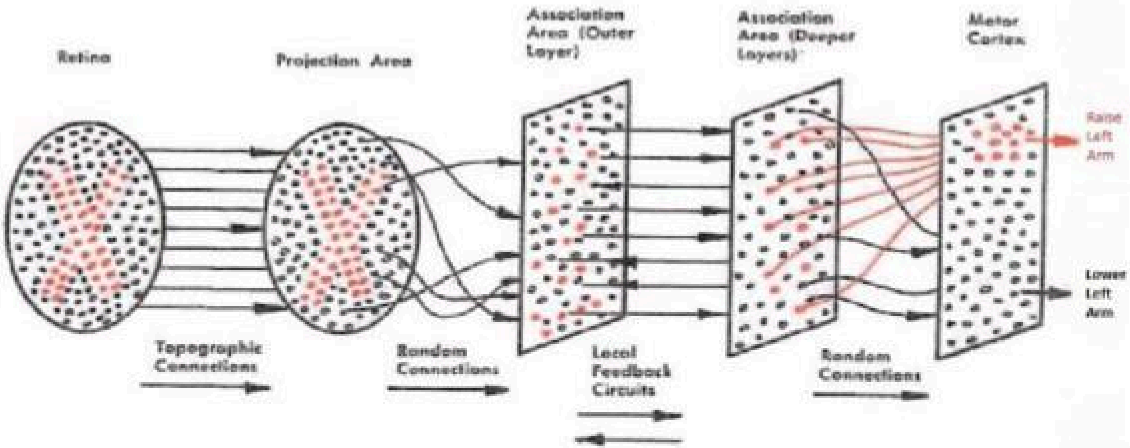


FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

灵感来源 II

感知机的组织结构

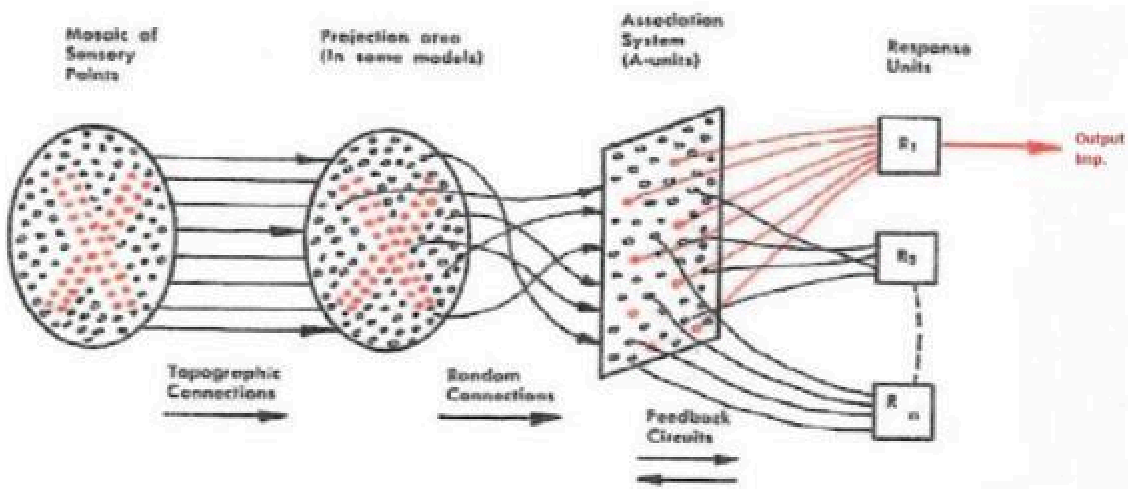


FIG. 2 — Organization of a perceptron.

灵感来源 III

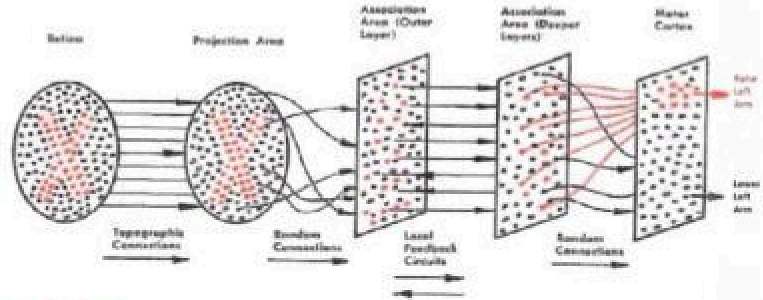


FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

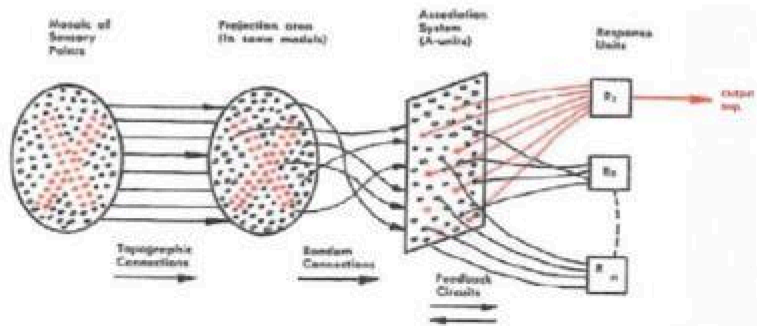


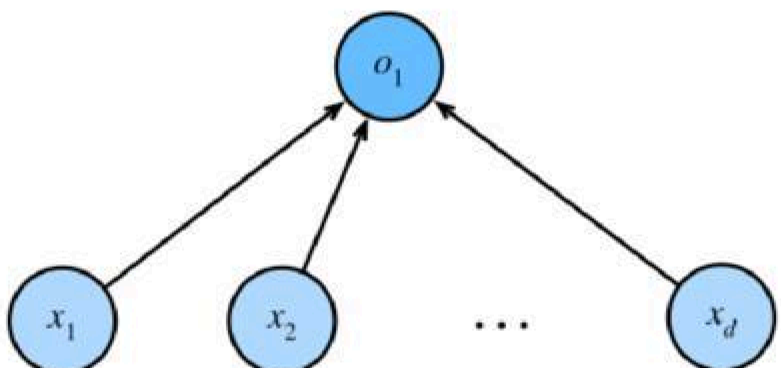
FIG. 2 — Organization of a perceptron.

感知机原型

输入：$\mathbf{x}$；参数：$\mathbf{w}, b$；输出：

$$o = \sigma\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right), \sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

激活函数σ：输出并传递电信号，激活/未激活



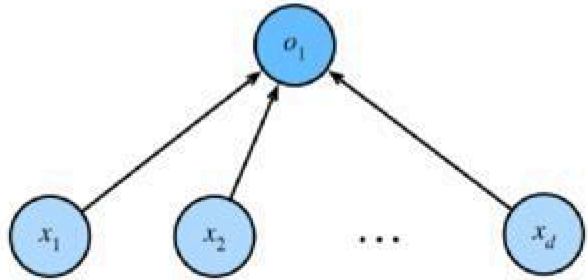
感知机二分类模型

输入：$\mathbf{x}$；参数：$\mathbf{w}, b$；输出：

$$o = \sigma\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right), \sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases}$$

二分类问题，激活函数通常输出：$\{-1, 1\}$

- 回归：实数；softmax回归：概率
- 线性部分不变，**只改变激活函数**



o_1
x_1
x_2
. . .
x_d

训练感知机

训练过程等价于使用**批量大小为1**的梯度下降

1. 初始化： $\mathbf{w} = \mathbf{0}, b = 0$;
2. 直到所有数据都被分类正确:
   1. IF $y_i\left(\langle \mathbf{w_i}, \mathbf{x_i} \rangle + b\right) \leq 0$:
      1. $\mathbf{w_i} = \mathbf{w_i} + y_i \mathbf{x_i}$;
      2. $b = b + y_i$;

- 二分类输出$\{-1, 1\}$： $\hat{y}_i = \langle \mathbf{w_i}, \mathbf{x_i} \rangle + b$为正时输出1
  - 所以可以用$y_i\left(\langle \mathbf{w_i}, \mathbf{x_i} \rangle + b\right) = y_i \hat{y}_i$判断分类是否正确

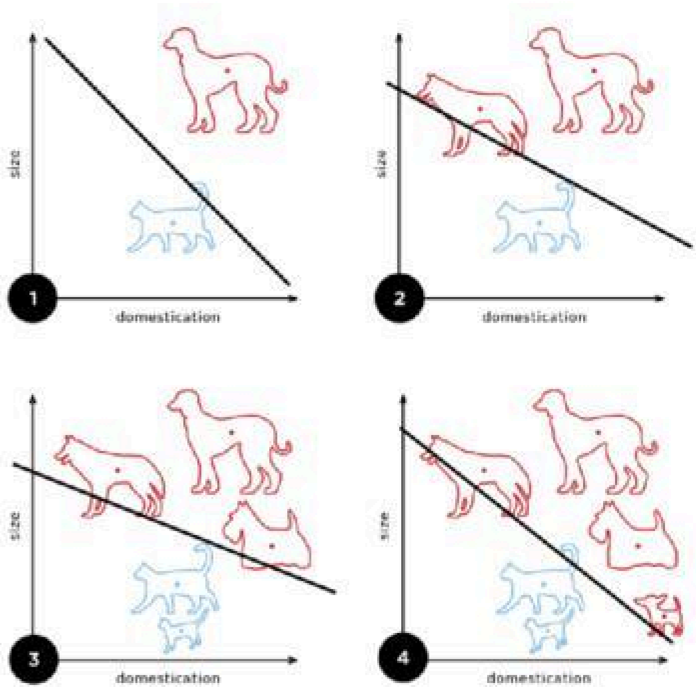
训练感知机

训练过程等价于使用**批量大小为1**的梯度下降

1. 初始化：$\mathbf{w}=\mathbf{0}, b=0$;
2. 直到所有数据都被分类正确:
   1. IF $y_i\left(\langle\mathbf{w_i},\mathbf{x_i}\rangle+b\right)\leq 0$:
      1. $\mathbf{w_i}=\mathbf{w_i}+y_i\mathbf{x_i}$;
      2. $b=b+y_i$;

- 二分类输出$\{-1,1\}$：$\hat{y}_i=\langle\mathbf{w_i},\mathbf{x_i}\rangle+b$为正时输出1
  - 所以可以用$y_i\left(\langle\mathbf{w_i},\mathbf{x_i}\rangle+b\right)=y_i\hat{y}_i$判断分类是否正确
- **损失函数**：$l(y,\mathbf{x};\mathbf{w})=\max(0,-y\langle\mathbf{w},\mathbf{x}\rangle)$
  - 梯度：$y_i\mathbf{x_i}$，进而得到步长为1的更新公式

感知机训练示例

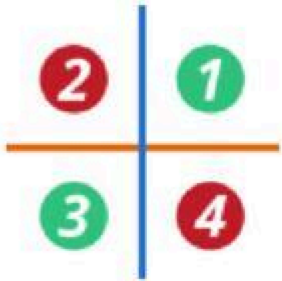


size
1
domestication
size
2
domestication
size
3
domestication
size
4
domestication

XOR问题

[Minsky 1969] 感知机不能拟合XOR函数

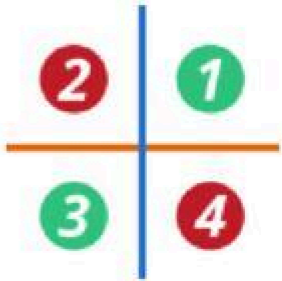
	x_1	x_2
x_1	1	0
x_2	0	1



XOR问题

[Minsky 1969] 感知机不能拟合**XOR函数**

	x_1	x_2
x_1	1	0
x_2	0	1

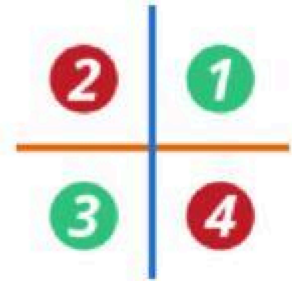


- 感知机只能产生**线性分割面**
  - 思考：为什么?

XOR问题

[Minsky 1969] 感知机不能拟合**XOR函数**

	x_1	x_2
x_1	1	0
x_2	0	1

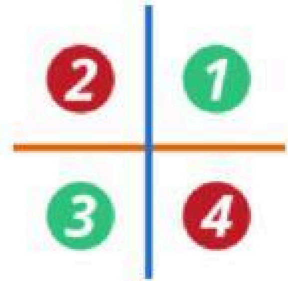


- 感知机只能产生**线性分割面**
  - 思考：为什么?
- 本质：判断线性模型的输出是否> 0

XOR问题

[Minsky 1969] 感知机不能拟合**XOR函数**

	x_1	x_2
x_1	1	0
x_2	0	1



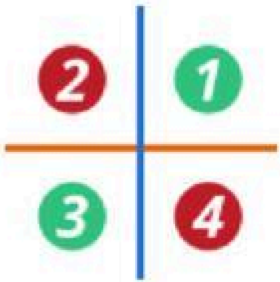
- 感知机只能产生**线性分割面**
  - 思考：为什么?
- 本质：判断线性模型的输出是否> 0
- 理论缺陷间接导致第一次AI寒冬

小结

- 感知机是二分类模型，最早的AI模型之一
- 训练过程等价于使用批量大小为1的梯度下降
- 不能拟合XOR函数，间接导致第一次AI寒冬

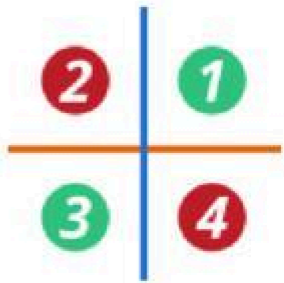
多层感知机

学习XOR：辅助函数

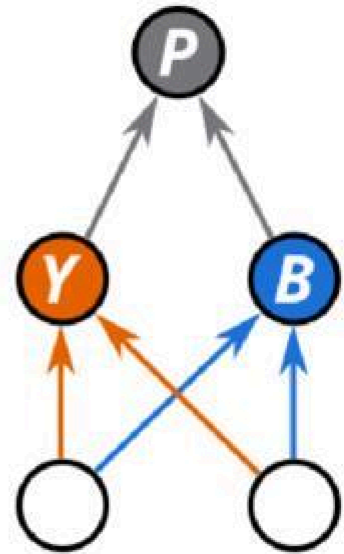


	1	2	3	4
Y	+	+	-	-
B	+	-	-	+

学习XOR

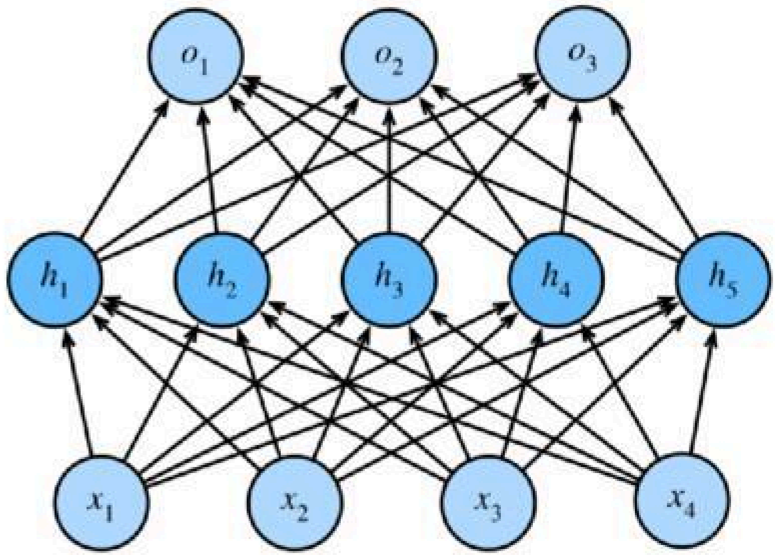


	1	2	3	4
Y	+	+	-	-
B	+	-	-	+
P	+	−	+	−



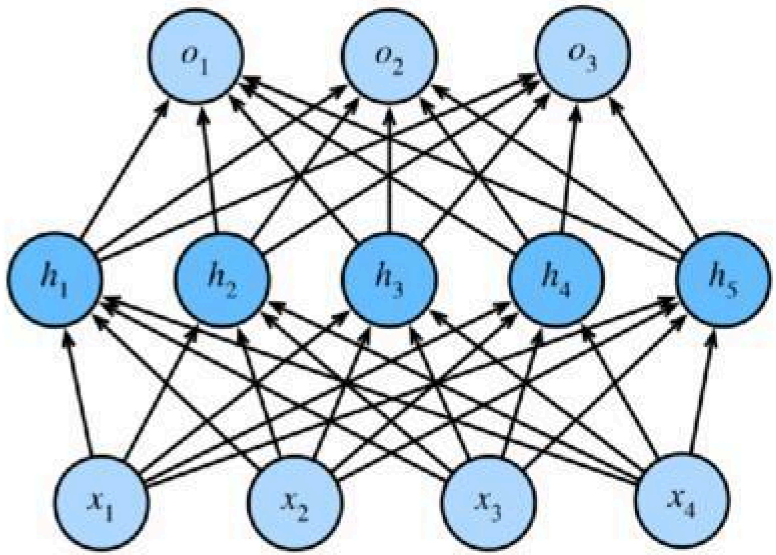
单隐藏层

隐藏层：将辅助函数推广到多个



单隐藏层

隐藏层：将辅助函数推广到多个



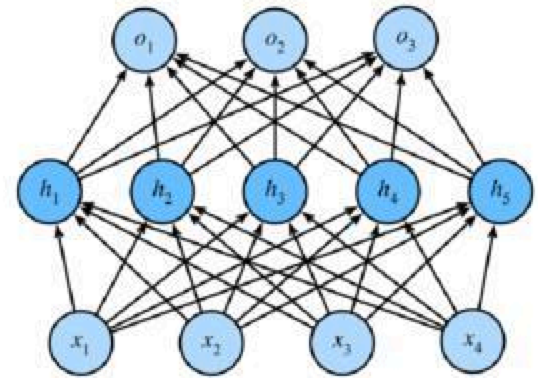
- 隐藏层的大小是**超参数**，取决于设计

单隐藏层：隐函数

首先考虑隐藏层$\mathbf{h}$

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

- $\mathbf{x} \in \mathbb{R}^{d\times 1}, \mathbf{W}^h \in \mathbb{R}^{l\times d}, \mathbf{b}^h \in \mathbb{R}^{l\times 1}$

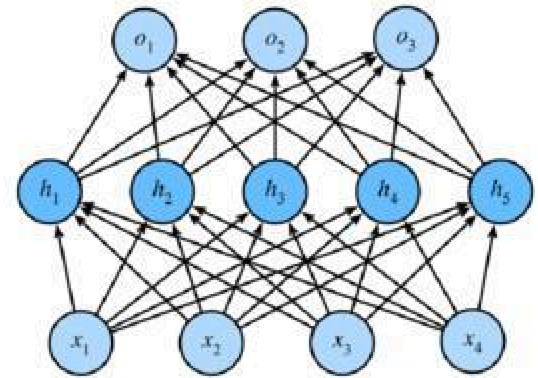


单隐藏层：隐函数

首先考虑隐藏层$\mathbf{h}$

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

- $\mathbf{x} \in \mathbb{R}^{d\times 1}, \mathbf{W}^h \in \mathbb{R}^{l\times d}, \mathbf{b}^h \in \mathbb{R}^{l\times 1}$



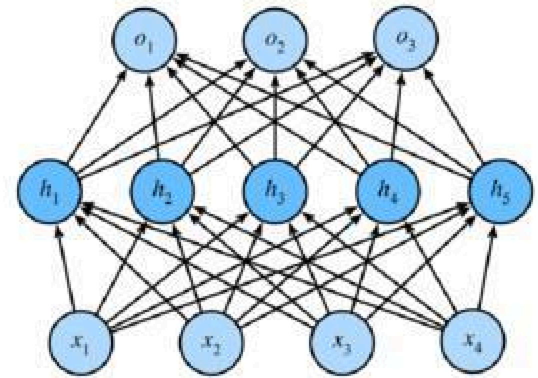
- σ是**非线性**函数，按元素计算激活值

单隐藏层：输出函数

其次考虑输出层$\mathbf{o}$

$$\mathbf{o} = \mathbf{W}^o \mathbf{h} + \mathbf{b}^o$$

- $\mathbf{h} \in \mathbb{R}^{l \times 1}, \mathbf{W}^o \in \mathbb{R}^{C \times l}, \mathbf{b}^o \in \mathbb{R}^{C \times 1}$

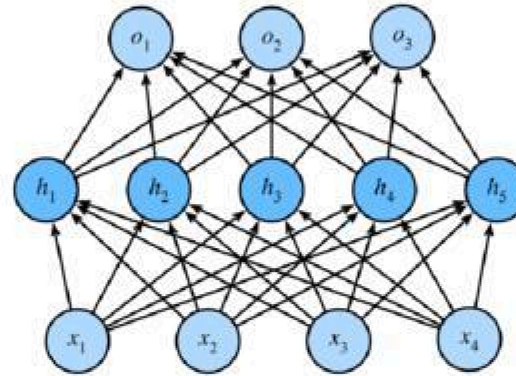


单隐藏层：输出函数

其次考虑输出层$\mathbf{o}$

$$\mathbf{o} = \mathbf{W}^o \mathbf{h} + \mathbf{b}^o$$

- $\mathbf{h} \in \mathbb{R}^{l \times 1}, \mathbf{W}^o \in \mathbb{R}^{C \times l}, \mathbf{b}^o \in \mathbb{R}^{C \times 1}$

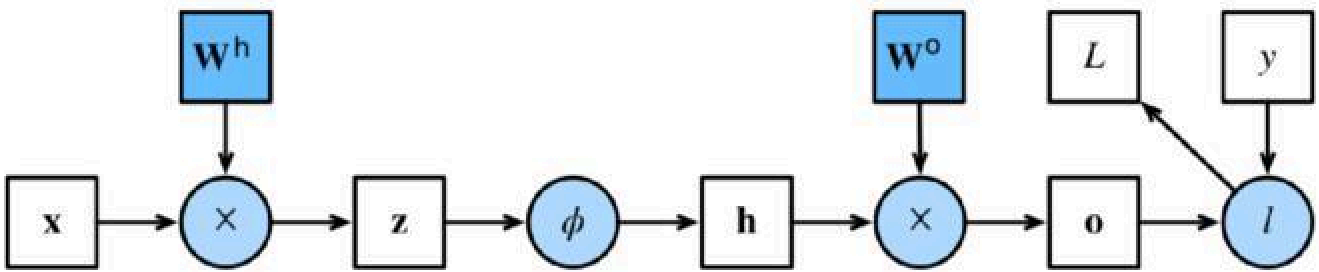


- 暂时输出实数：最终输出的构造取决于实际应用

自动微分计算图

$$\mathbf{h} = \sigma(\mathbf{W}^h \mathbf{x} + \mathbf{b}^h)$$
$$\mathbf{o} = \mathbf{W}^o \mathbf{h} + \mathbf{b}^o$$

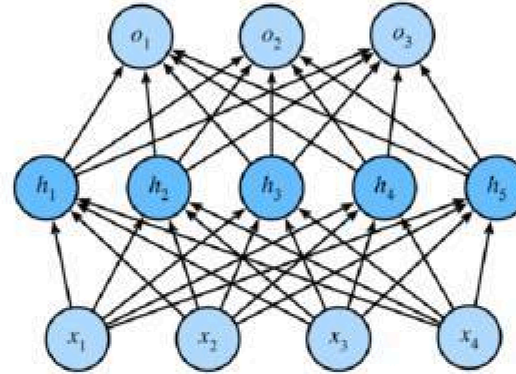
$$\mathbf{z} = \mathbf{W}^h \mathbf{x}$$
$$\mathbf{h} = \phi(\mathbf{z})$$
$$\mathbf{o} = \mathbf{W}^o \mathbf{h}$$
$$\mathcal{L} = l(O, y)$$



单隐藏层：小批量计算

$$\mathbf{H} = \sigma(\mathbf{X}\mathbf{W}^h + \mathbf{b}^h)$$
$$\mathbf{O} = \mathbf{H}\mathbf{W}^o + \mathbf{b}^o$$

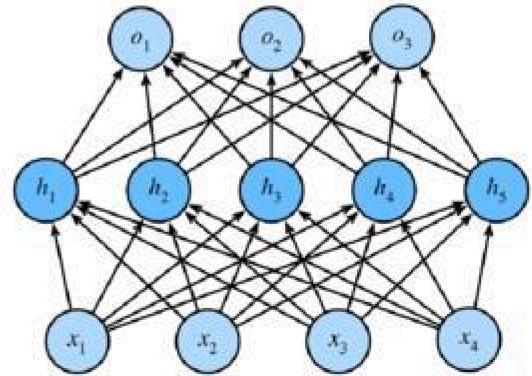
- $\mathbf{X} \in \mathbb{R}^{B \times d}, \mathbf{W}^h \in \mathbb{R}^{d \times l}, \mathbf{b}^h \in \mathbb{R}^{1 \times l}$
- $\mathbf{H} \in \mathbb{R}^{B \times l}, \mathbf{W}^o \in \mathbb{R}^{l \times C}, \mathbf{b}^o \in \mathbb{R}^{1 \times C}$



为什么需要非线性激活函数?

假设σ是线性函数

$$\begin{aligned}\mathbf{H} &= \sigma(\mathbf{X}\mathbf{W}^h + \mathbf{b}^h)\\ &= \mathbf{W}(\mathbf{X}\mathbf{W}^h + \mathbf{b}^h) + \mathbf{b}\\ \mathbf{O} &= \mathbf{H}\mathbf{W}^o + \mathbf{b}^o\end{aligned}$$

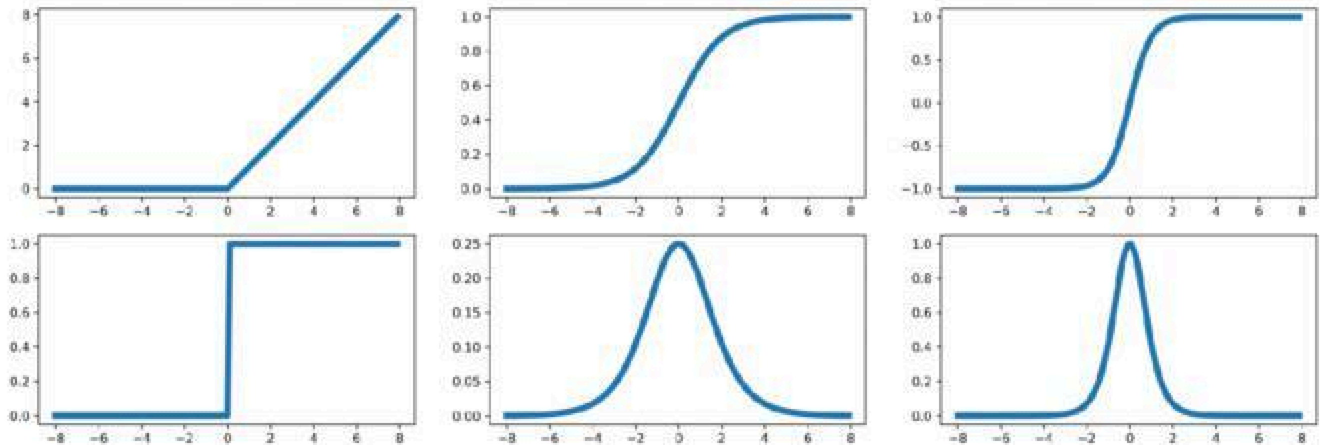


- 只能学习线性映射

激活函数

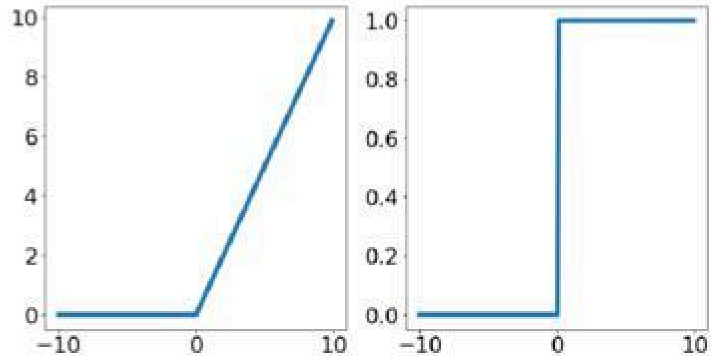
三种基本激活函数

ReLU, sigmoid, tanh.



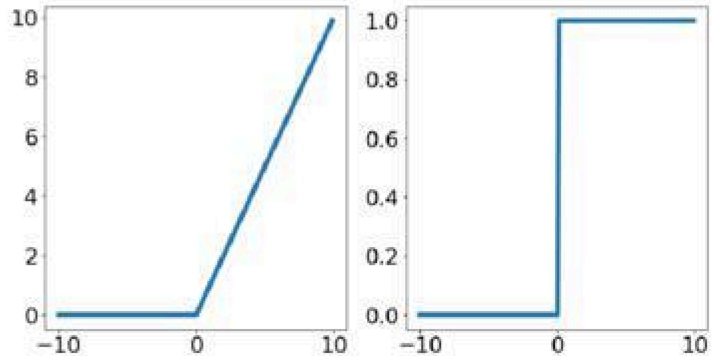
- 注意：**分段线性**也是非线性！

修正线性单元 Rectified Linear Unit (ReLU)



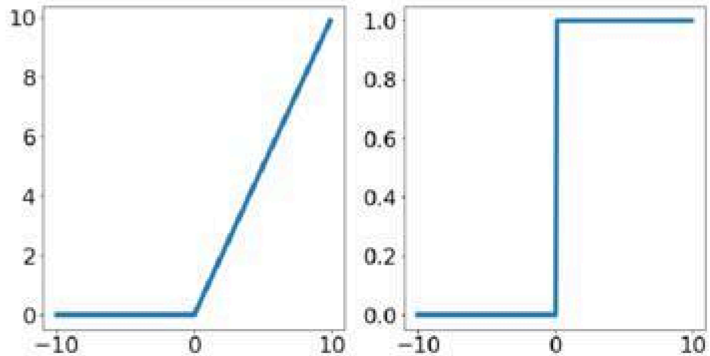
- 输入值需要达到**阈值**：例如疼痛、电流
  - 减少无效信号干扰：否则坐着也难受！

修正线性单元 Rectified Linear Unit (ReLU)



- 输入值需要达到**阈值**：例如疼痛、电流
  - 减少无效信号干扰：否则坐着也难受！
- 最常用，减轻**梯度消失**问题

修正线性单元 Rectified Linear Unit (ReLU)



- 输入值需要达到**阈值**：例如疼痛、电流
  - 减少无效信号干扰：否则坐着也难受！

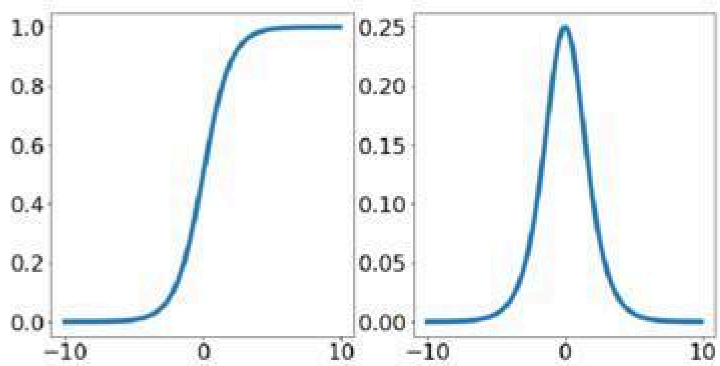
- 最常用，减轻**梯度消失**问题

问题：0点处的导数怎么办？简单选取一个值

> *“如果微妙的边界条件很重要，我们很可能是在研究数学而非工程”*

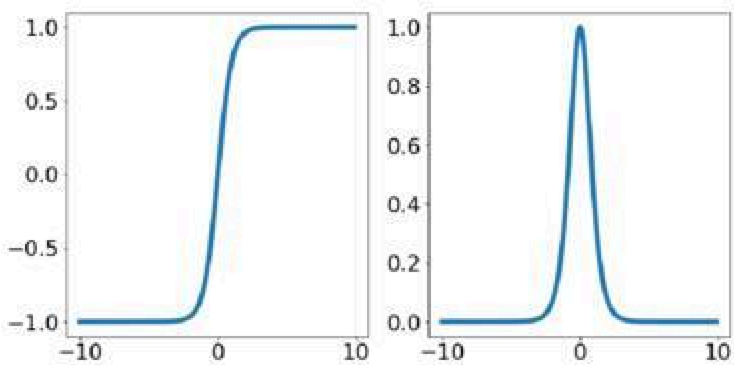
- 激活值为0？最好**先检查bug**

S曲线 Sigmoid



- 压缩到$[0, 1]$
  - 模拟**激发/不激发**两种状态
  - 常用于构造**概率**

双曲正切 tanh



- 压缩到$[-1,1]$
  - 常用于**分类**问题
    - 对比S曲线：不确定范围更小，**容易突破阈值**

输出映射

输出是实数，但分类任务需要将输出映射到**类别标签**

$$\mathbf{o} = \mathbf{W}^o \mathbf{h} + \mathbf{b}^o$$

- 在**标准化区间**才能比较：例如输出100，究竟是多高的置信度?

输出映射

输出是实数，但分类任务需要将输出映射到**类别标签**

$$\mathbf{o} = \mathbf{W}^o\mathbf{h} + \mathbf{b}^o$$

- **在标准化区间**才能比较：例如输出100，究竟是多高的置信度?

转换的关键是**激活函数**

单、二分类

- `Sigmoid`：映射到$[0,1]$，是/否
- `Tanh`：映射到$[-1,1]$，两极

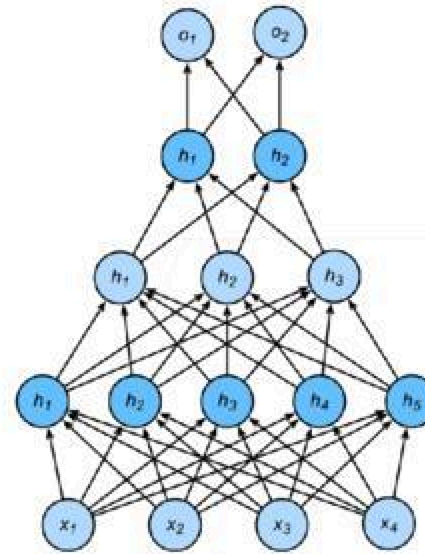
多分类

- `softmax`：映射到对应类别概率值

实验：激活函数

多隐藏层

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$
$$\mathbf{h}_3 = \sigma(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3)$$
$$\mathbf{o} = \mathbf{W}^o\mathbf{h}_3 + \mathbf{b}^o$$

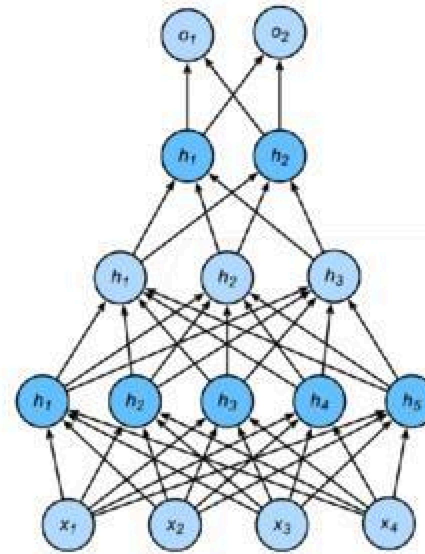


超参数：取决于模型的设计

- 隐藏层数
- 每个隐藏层的大小

多隐藏层

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$
$$\mathbf{h}_3 = \sigma(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3)$$
$$\mathbf{o} = \mathbf{W}^o\mathbf{h}_3 + \mathbf{b}^o$$



超参数：取决于模型的**设计**

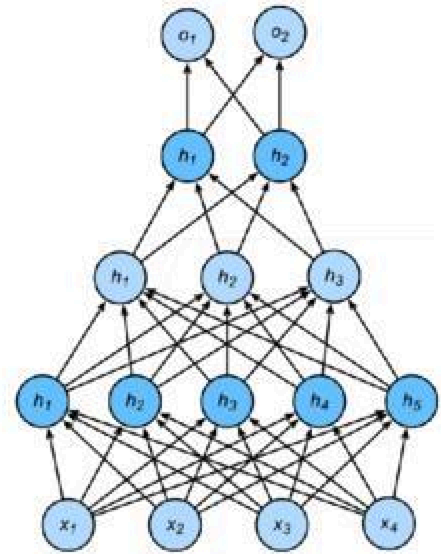
- 隐藏层数
- 每个隐藏层的大小

多层感知机是第一个深度模型

- 注意：不存在“深度线性模型”

多隐藏层

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$
$$\mathbf{h}_3 = \sigma(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3)$$
$$\mathbf{o} = \mathbf{W}^o\mathbf{h}_3 + \mathbf{b}^o$$



超参数：取决于模型的**设计**

- 隐藏层数
- 每个隐藏层的大小

多层感知机是第一个深度模型

- 注意：不存在“深度线性模型”

问题：如何确定超参数？

小结

- 感知机使用隐藏层和激活函数得到非线性模型
- 常用激活函数：`ReLU`, `sigmoid`, `tanh`
- 使用`softmax`处理多分类任务
- 超参数：隐藏层数、每个隐藏层的大小

实验：多层感知机的从零开始实现

实验：多层感知机的简洁实现

Review

本章内容

感知机。多层感知机模型。激活函数。多层感知机的实现。误差分类与验证方法。欠拟合与过拟合。减轻过拟合。

重点：多层感知机模型；激活函数；多层感知机的实现；训练误差、验证误差、泛化误差；欠拟合、过拟合；减轻过拟合：模型容量限制，权重衰减（正则化），丢弃法。

难点：XOR问题；模型泛化性；减轻过拟合。

学习目标

感知机

- 理解感知机的原理、二分类实现方法、理论缺陷。
- 理解多层感知机的原理、分类实现方法。
- 理解三种基本激活函数的特点和用途。
- 掌握多层感知机的实现方法。
- 了解XOR问题的解决历史。

欠拟合、过拟合

- 理解训练误差、验证误差、泛化误差。
- 理解验证集、测试集，以及验证方法。
- 理解欠拟合、过拟合，以及过拟合的主要原因。
- 了解减轻过拟合的两个主要思路：限制参数个数、取值范围。
- 理解减轻过拟合的主要方法：模型容量限制，权重衰减（正则化），丢弃法。

问题

(*) 简述多层感知机解决XOR问题的方法。

简述为什么多层感知机可以逼近非线性函数?

简述三种基本激活函数的特点和用途。

简述训练误差、验证误差、泛化误差的区别。

简述验证集、测试集的区别，并举例说明验证方法。

结合图示简述欠拟合、过拟合的判断方法，以及过拟合的主要原因。

简述减轻过拟合的两个主要思路，及其主要原理。

简述权重正则化、丢弃法的主要方法和原理。