机器学习基础理论 & 为什么选择深度学习

机器学习基础理论

机器学习模型基础三步：

1. function with unknown
2. define loss
3. optimization

训练集的选择

我们将一个问题相关的所有的数据（世界上所有的数据，收集不齐）定义为 D_{all}，所有的可能的参数集为 H，我们在 D_{all} 训练出的参数称为 h^{all}

$$h^{all} = arg\min_{h} L(h, D_{all})$$

我们实际训练用的数据集 D_{train} 是 D_{all} 的子集。但是我们的训练集不一定能很好的“代表”总集。

我们希望：$L(h^{train}, D_{all})$ 和 $L(h^{all}, D_{all})$ 是很接近的。

上式可化为：

$$|L(h^{train}, D_{all}) - L(h^{all}, D_{all})| \leq \delta$$

可进一步化为：

$$\forall h \in H, |L(h, D_{train}) - L(h, D_{all})| \leq \epsilon$$
$$\epsilon = \delta/2$$

也就是说，好的训练集可以满足上式。那么如何让坏的的训练集减少，让我们的训练集是好的训练集的概率更大?

我们可以得出下式，训练集是坏的训练集的概率满足：

$$P(D_{train} is\ bad) \leq |H| \cdot 2exp(-2N\epsilon^2)$$
$$N \text{ is the number of examples in } D_{train}$$

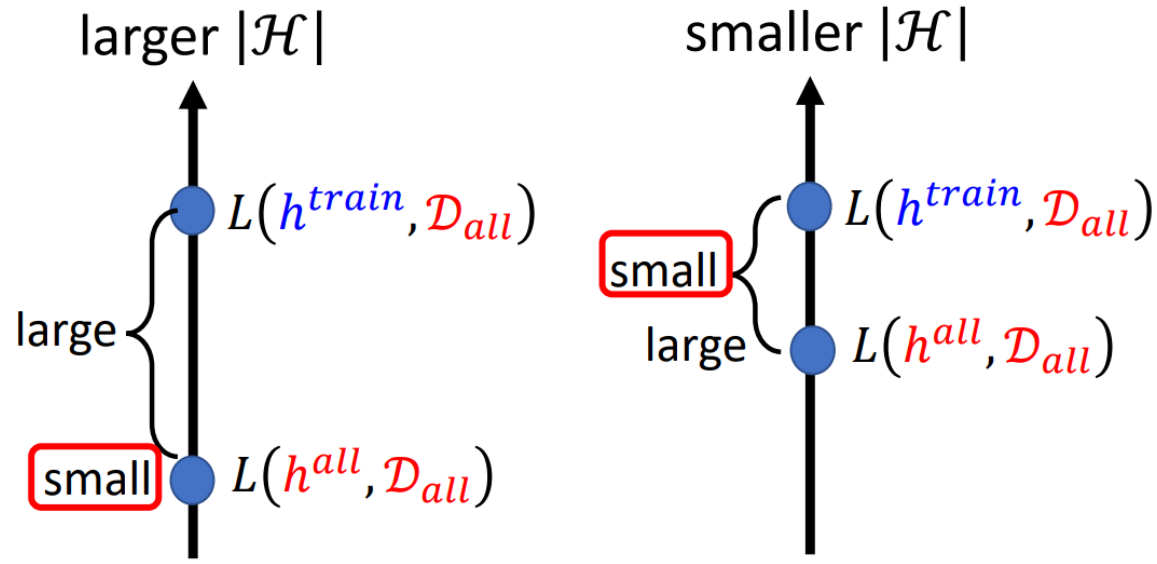
所以为了让训练集更好，为了“现实”“理论”更加接近：

- N 越大越好
- H 越小越好

但是 H 越小，h 可取的值也越少，$L(h^{all}, D_{all})$ 也不会很小

也就是说

- Larger N and smaller $|H| \rightarrow |L(h^{train}, D_{all}) - L(h^{all}, D_{all})| \leq \delta$
- Smaller $|H|$ $\rightarrow$ Larger $L(h^{all}, D_{all})$

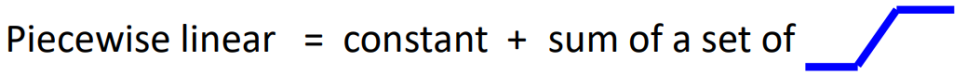


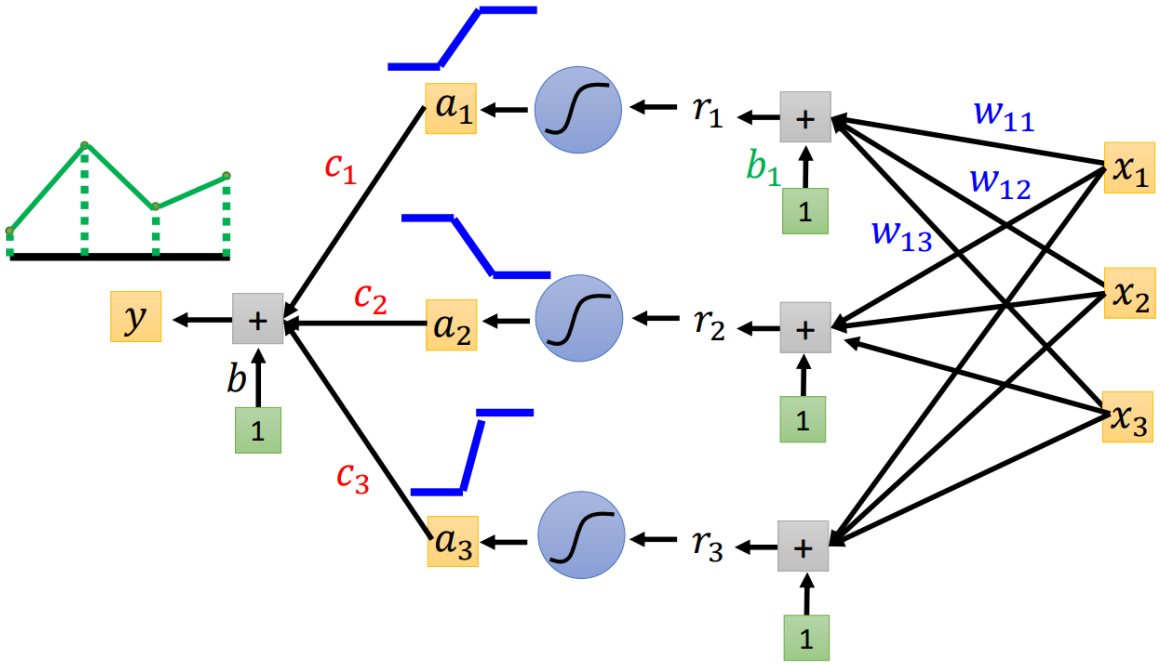
鱼与熊掌是否可以兼得？**YES, DEEP LEARNING**

为什么选择深度学习

> 鱼与熊掌是否可以兼得：让 H 小（理想与现实距离更近）的同时，让 $Loss$ 也尽可能小？

我们知道，机器学习实际上就是拟合一个函数的过程，理论上说只要我们的模型足够“fat”（神经元数量多，参数多），不需要多层，只需一层就可以完成问题。





是这样，但是：

实验结果：

Layer X Size	Word Error Rate (%)	Layer X Size	Word Error Rate (%)
1 X 2k	24.2		
2 X 2k	20.4		
3 X 2k	18.4		
4 X 2k	17.8		
5 X 2k	17.2	1 X 3772	22.5
7 X 2k	17.1	1 X 4634	22.6
		1 X 16k	22.1

Why?

- `Layer` 越多，误差越小
- 在保持参数数量差不多相同的情况下：层数越深越有效率

为什么会这样?

比如我们想生成一个锯齿状函数，其边数为 2^k：

- 深度学习由于有"嵌套"，"传递"性，只需要 $2K$ 个 `neurons`
- 但是 `shallow` 的模型，需要 2^k 个 `neurons`

^ 指数级的差距！

所以，当我们目标函数是一个 **规律 且 复杂** 的函数时，深度学习会效果更好，效率更高（比如音频，影像等）

定义复杂：在函数 $y = x^2$ 上，深度学习都要做的更好！