DW_共读AI圣经：第一章

先来攻坚克难

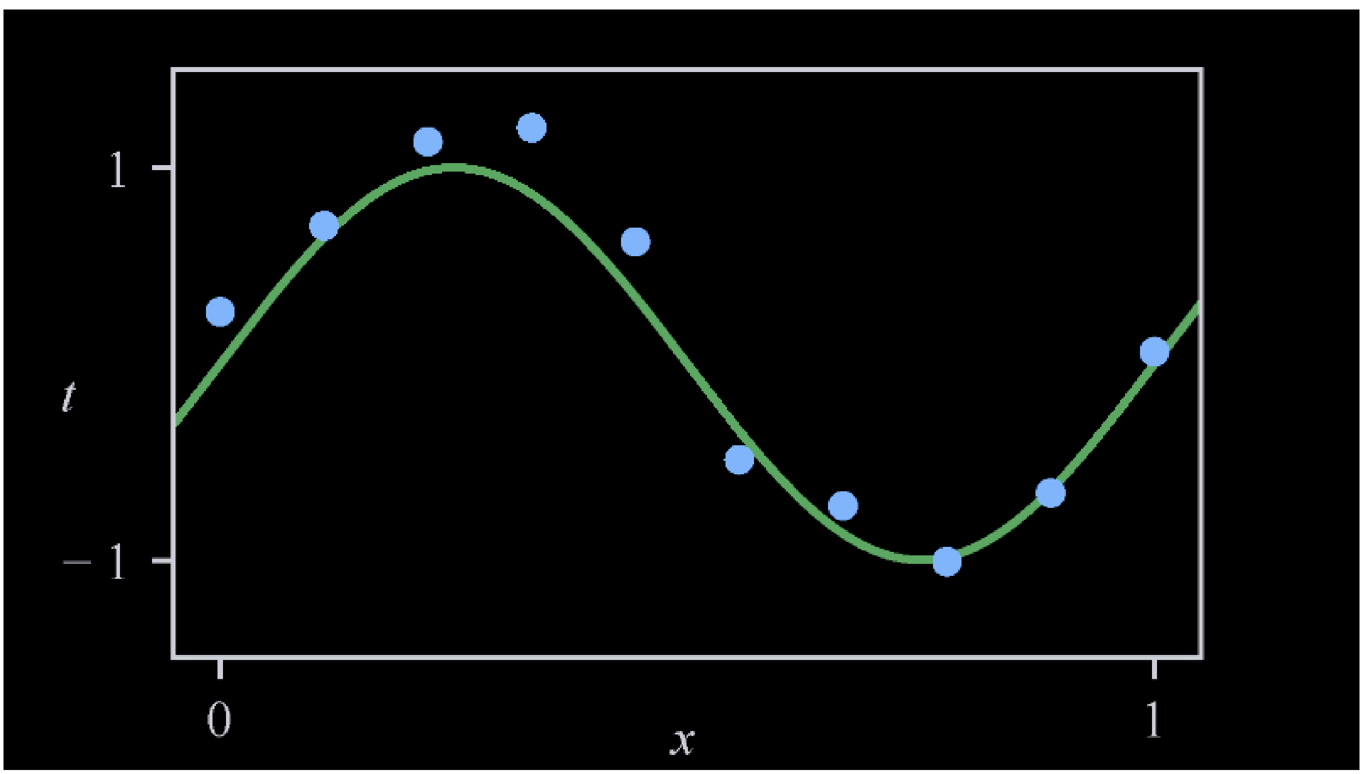
$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2$$

$$\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2} \|w\|^2$$

> 从数据中学习概率是机器学习的核心

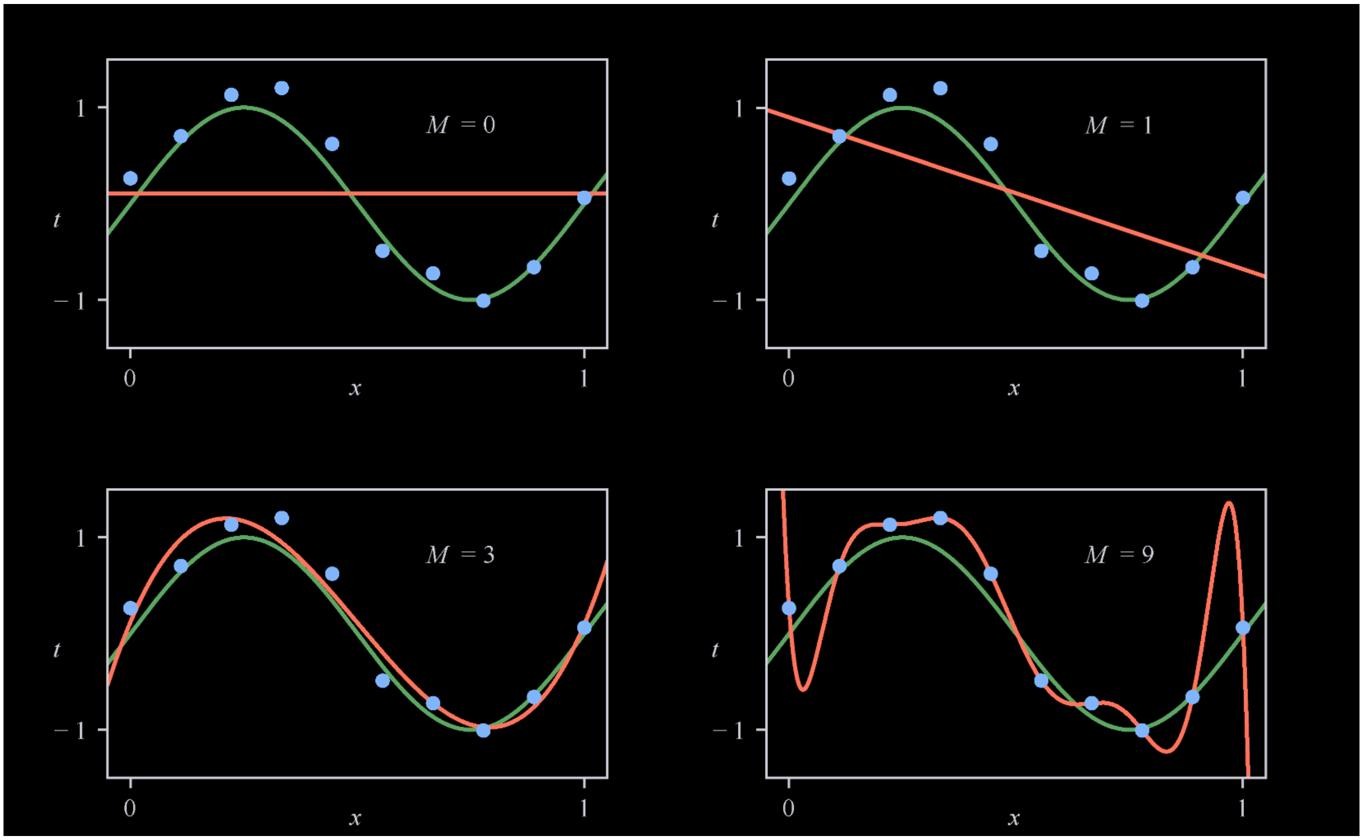
数据：　由$\sin(2\pi x)$上几个离散的点加入随机噪声得到的数据



拟合步骤

1. 考虑用多项式来拟合（重点需要确定超参数M的选择）

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$



由于数据本来主要来自于$\sin(2\pi x)$，考虑到$\sin(2\pi x)$的展开式越高阶越准确，从理论上来讲，M越大拟合效果越好。

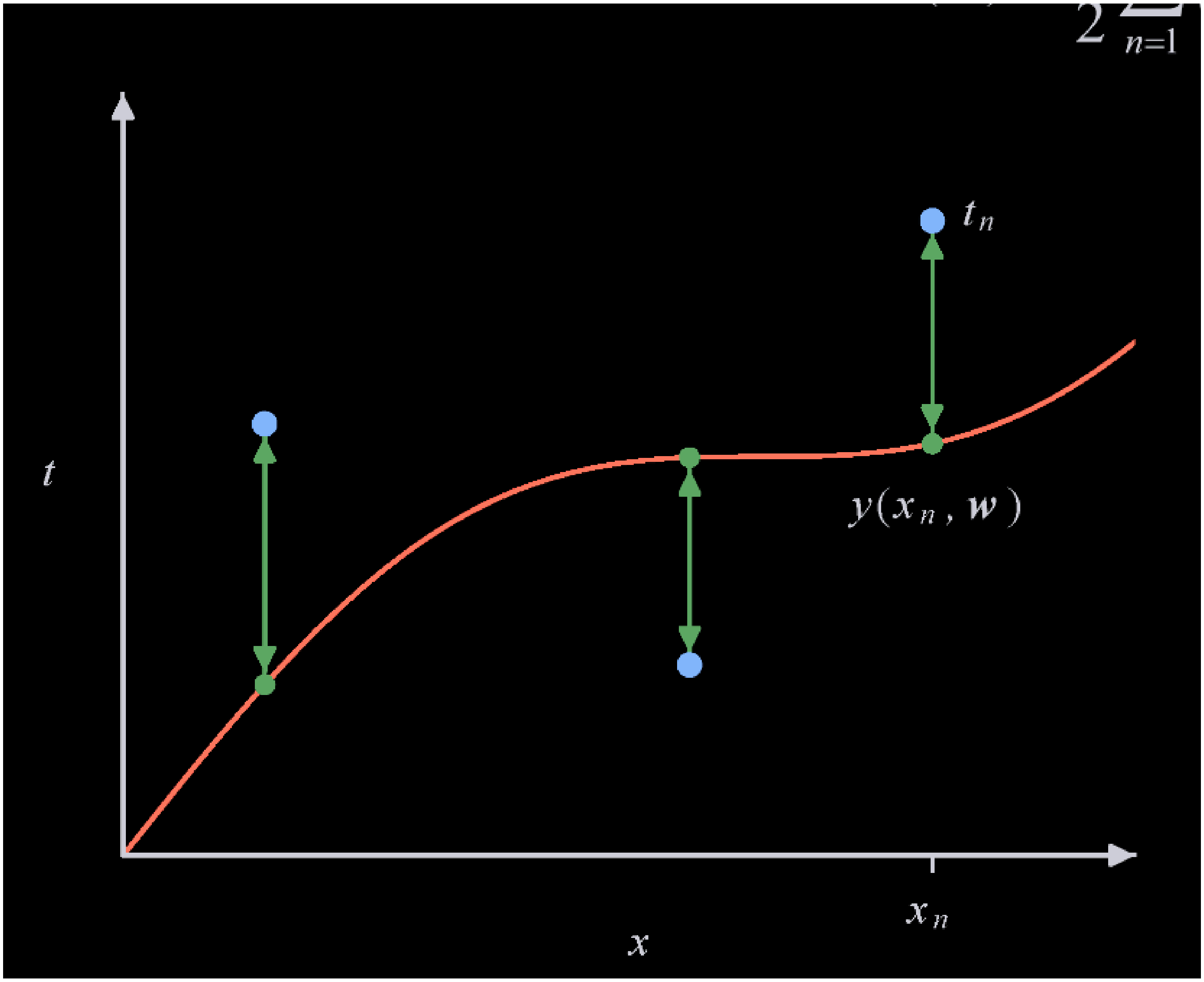
但由于随机噪声的存在（实际中的数据噪声更复杂），随着阶数的增大，拟合效果并不是越来越好。本例中在$M=3$的时候最好，而$M=9$的时候过分依赖训练数据而无法正确预测新数据（即**过拟合现象**）。

因此我们不能只依据阶数来判断拟合效果，那么就需要引入误差函数。

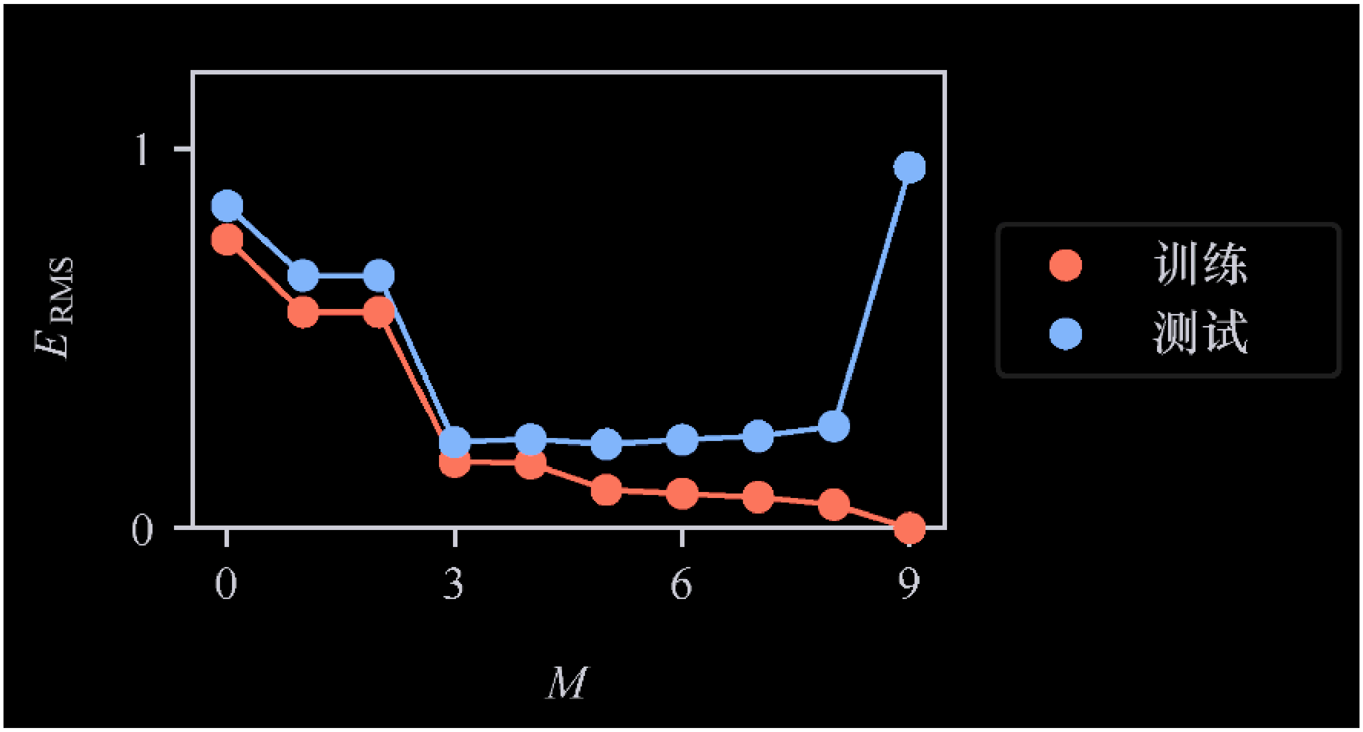
2. 用误差函数来优化拟合效果

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2$$

误差函数的几何意义是拟合函数到某个点的垂直距离（实际会用到不同的误差函数），当拟合函数正好经过每个点时其值为0。



为了衡量拟合函数的泛化性，引入测试集（和训练集经过同样的预处理步骤）。在拟合工程中，北京可以计算训练集的误差函数，还可以计算测试集的误差函数。



当测试集和训练集的误差均比较小的时候，说明拟合效果和泛化性都得到了保障。

> 数据量越大我们就越能用复杂的模型（即阶数更大）去拟合数据

但是阶数越大意味着参数量越大，这会带来更大的训练成本也会导致过拟合。正则化就是用来处理这个问题的。

3. 用正则化实现权重衰减

$$\tilde{E}(w)=\frac{1}{2}\sum_{n=1}^{N}\left\{y\left(x_{n},w\right)-t_{n}\right\}^{2}+\frac{\lambda}{2}\|w\|^{2}$$

正则项中包含拟合函数的所有参数，且皆为正值（λ的符号会影响正负）。拟合的目标是最小化误差函数，如果参数过大误差函数也会增大，直到找到一个平衡点，达到误差值最小但权重也不大的情况。

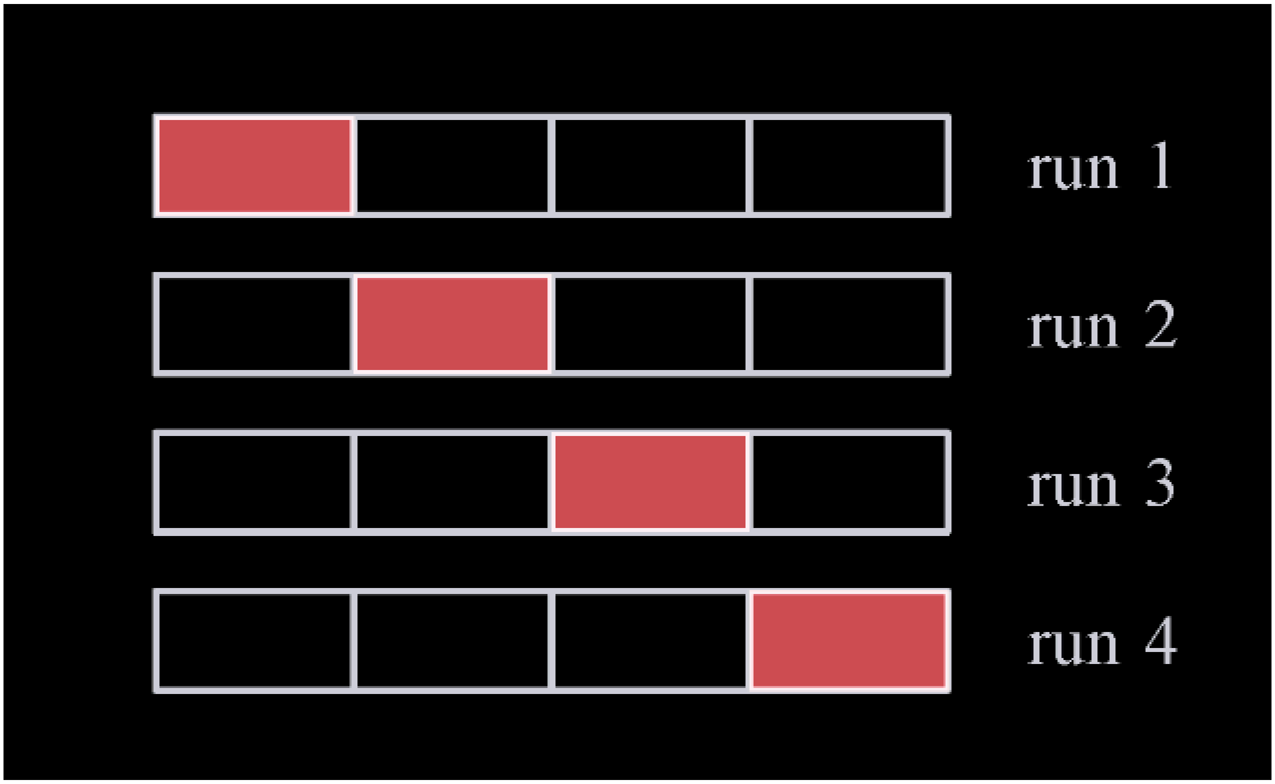
λ 取值较小的时候，可以达到抑制过拟合的效果；λ 取值过大的时候，会出现欠拟合的现象。

那么目前已经引入M,λ 两个超参数，无法通过同时最小化误差函数的方法来确定两个值，否则就会出现一个训练集上误差极小甚至为0的拟合函数。

那么就需要引入验证集来辅助确定超参数。

4. 验证集和交叉验证

训练集用于确定主要的参数，而最终会在测试集上通过最小化误差值来确定所有参数。



当实际数据集有限的情况下，可以采用交叉验证的方法来充分利用数据。

再来点轻松的

机器学习简史

这里主要简单地讲解了神经网络的基础常识，深入分析和数学推导在之后的章节学到。

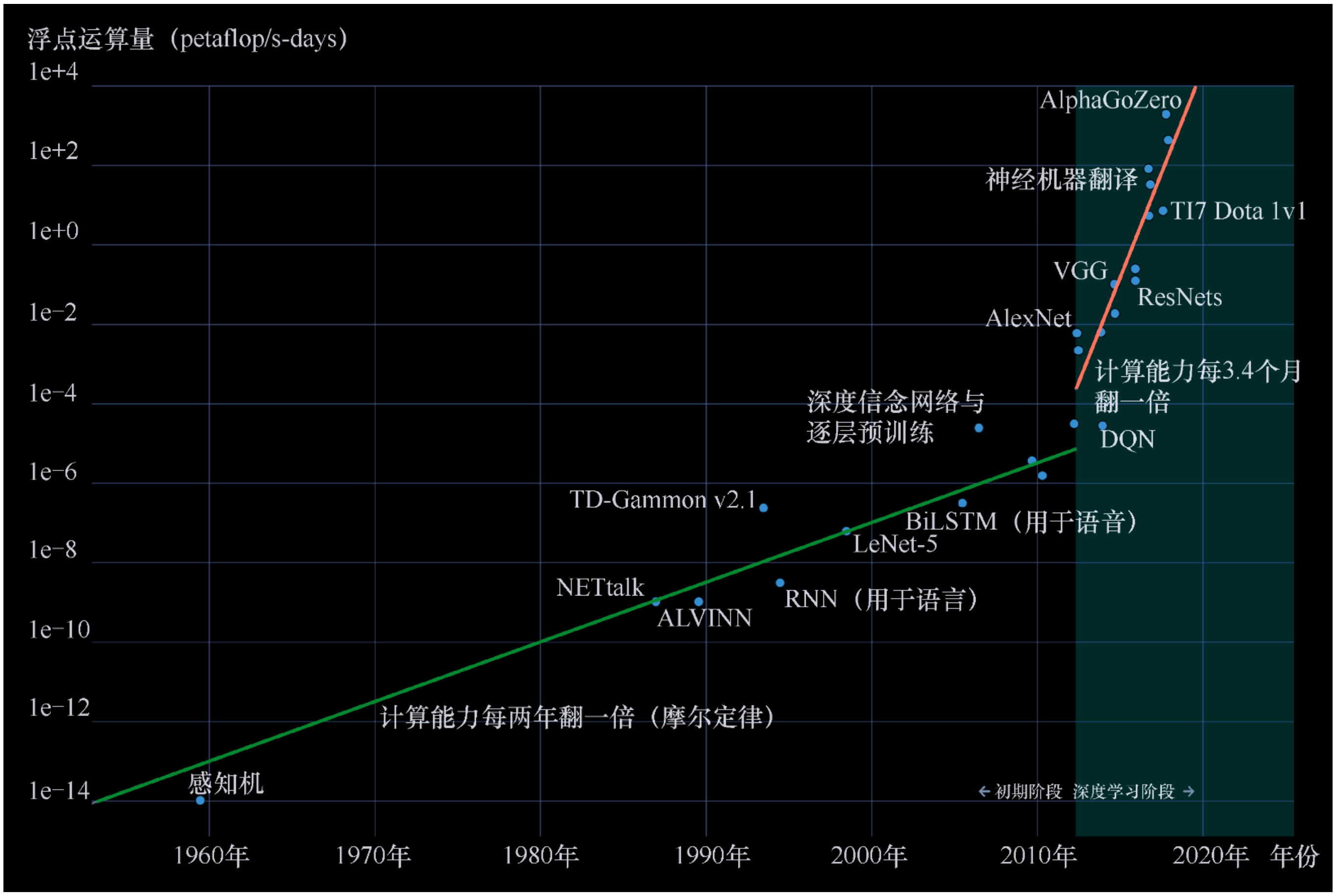
> 一个神经元能否引发另一个神经元的放电取决于突触的连接强度,而这些突触 的连接强度的动态变化正是大脑存储信息和进行经验学习的关键机制。

1. 最早的单层网络感知机模型的激活函数是阶跃函数

2. 通过用具有非零梯度的连续可微激活函数取代 了原有的阶跃函数式并引入可微的误差函数实现了前馈神经网络。实际训练多层的神经网络会采用基于梯度优化的反向传播来进行。（后面慢慢研究）

3. 通过架构创新或引入更复杂形式的归纳偏置而带来的性能改进, 很快就会被简单地通过扩大训练数据量替代,同时伴随着相应的模型规模和用于训练

的相关计算能力的扩展。



机器学习应用

机器学习算法广泛地应用在医疗诊断、蛋白质结构预测、图像合成、大语言模型等领域。

致谢

1. 感谢本书的作者团队，深入浅出，给我带来很多启发。
2. 感谢DW团队的组织和一起学习的小伙伴，进行前进。