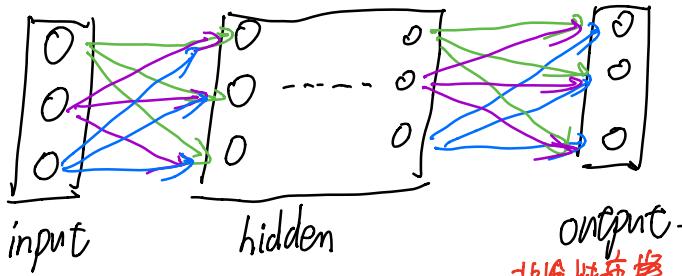


一、DL中的基础.

1. 一个机器学习算法有3个要素：模型、损失函数、优化方法。

2. DL中的模型。(只讨论前馈型)。

一般由三个部分组成，分别是输入层、隐藏层与输出层。



每一个神经元 O 由自己的参数 w, b 与 激活函数 组成。

即当他收到前面邻近层的输出后，它将会输出 $g(w^T x + b)$

常见的激活函数：

名称	表达式	优点	缺点
relu	$\max(0, x)$	优化速度快。	大量占用神经元
sigmoid	$\frac{1}{1 + \exp(-x)}$	某些场合好 (如作为输出层)	易产生梯度弥散，爆炸

针对 relu, sigmoid 的缺点，产生了 leakyrelu 与 tanh。

目前主流仍以 relu 为主。

除以上普遍的简单层以外，还有卷积层等。

没有激活函数时，层数推至无穷为线性。

如 $y = k_2(k_1x + b_1) + b_2 = k_1k_2x + k_2 + k_2b_1$ 无意义。

3. 损失函数。

损失函数主要衡量的是与训练结果相差。

对于回归任务，损失函数一般选择均方误差。 $loss = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$

梯度。

交叉熵 $loss = \frac{1}{n} \sum y_i \log \hat{y}_i$

4. 优化方法。

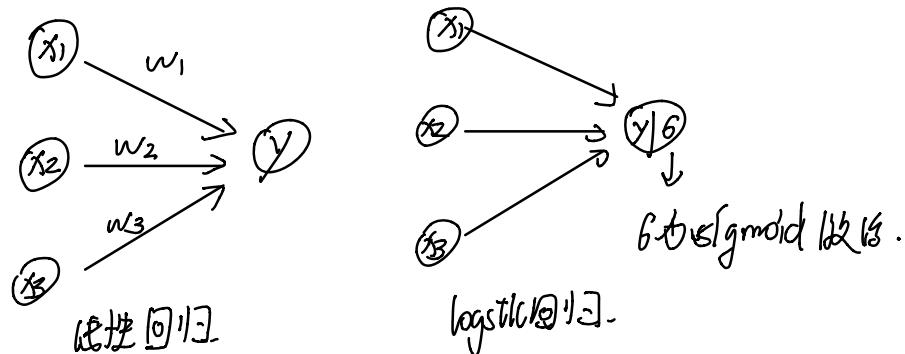
传统DL使用BP算法，近几年产生以BP为基础的一些衍生算法，如动量、RMSprop等。

5. 具体实现。

传统机器学习可以看为特殊的神经网络。

① 线性回归 = 输入 + 阶梯输出、均方损失、梯度下降
无激活函数。

② logistic回归或 softmax = 输入 + 输出(sigmoid)、交叉熵、梯度下降。



二、模型训练中的若干问题。

1. 泛化性能

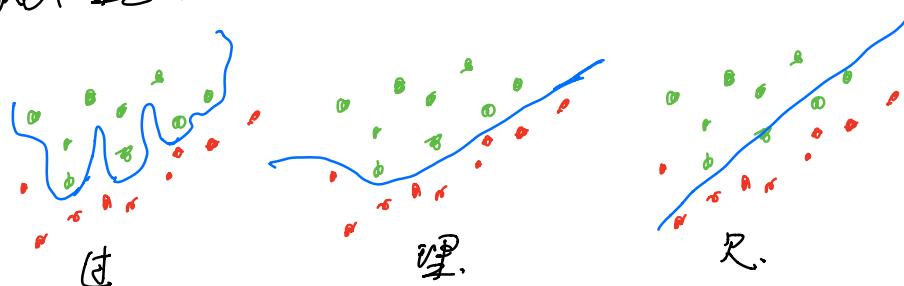
一般我们使用训练集得到的精度并不是在实际应用中的精度。
 t_1 t_2

若 $t_1 > t_2$, 行为过拟合.

$t_1 = t_2$ 理想.

$t_1 < t_2$. 欠拟合.

过拟合、理想、欠拟合的图示.



2. 欠拟合.

针对欠拟合, 一是选择更为复杂的模型(假设已充分训练)

二 适合.

复杂模型, 堆叠更厚的网络, 增加每一层的神经元个数.

(待) ; 如针对 image 更应用卷积层处理

3. 过拟合.

① 权重衰减.

简单可以认为过大、复杂的参数会对应更复杂的模型

在损失函数中加入对权重 W 的惩罚.

L1 正则化: $loss + \sum_i |w_i|$.

L2 正则化: $loss + \frac{1}{2} \sum_i w_i^2$.

② dropout

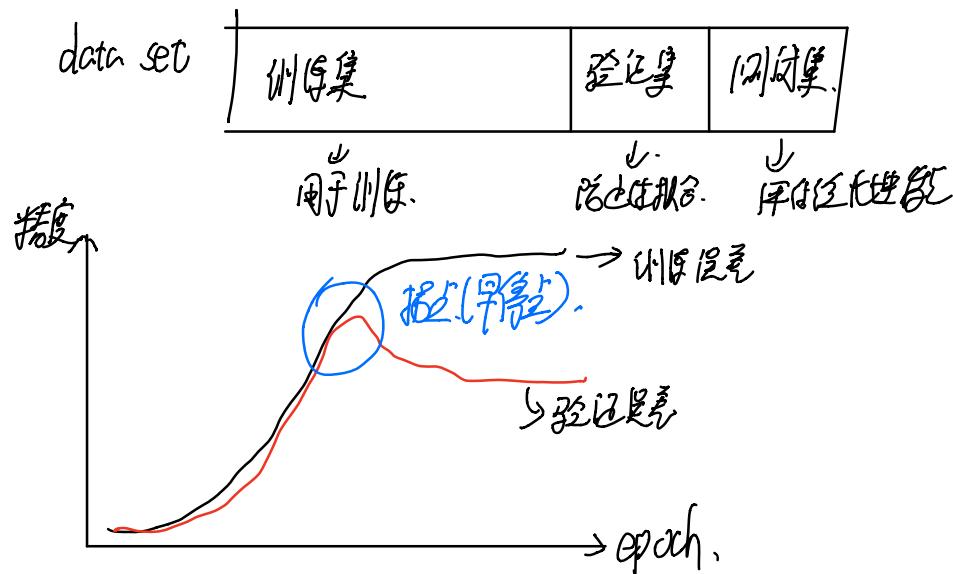
在每一轮训练中, 以一定概率使神经元输出为0.

$$P(A_i) = \begin{cases} p_i & \text{以 } 1-p_i \\ 0 & \text{以 } p_i \end{cases}$$

简单理解可以认为是避免了神经元之间的共度。

③ 减小模型复杂度、更多的数据(数据增强)。

④ 早停。



4. 数值稳定性。

训练过程中容易发生上溢与下溢，需要及时归一化
除此之外，舍弃的规范化参数可以帮助减少落入
极值点的可能性，加快训练速度。