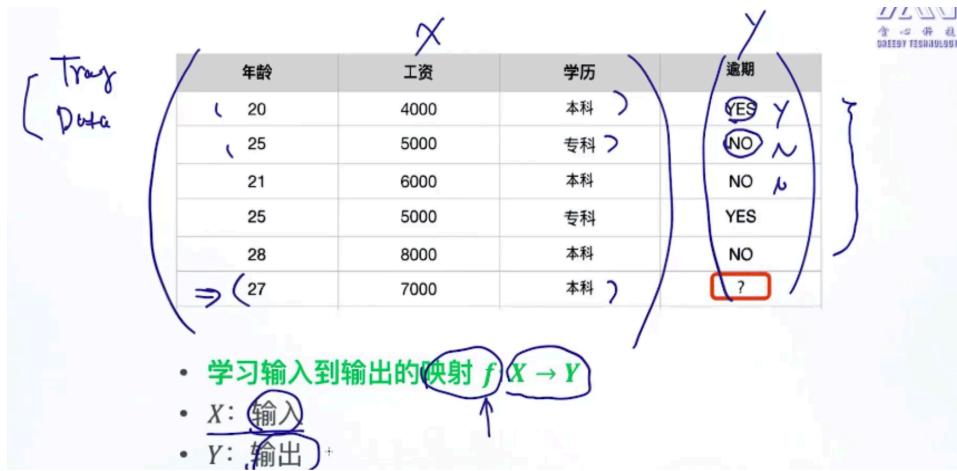


1. Logistic Regression

1.1 Classification tasks:

贷款违约、广告点击、商品推荐、情感分析、疾病诊断, etc

逻辑回归是一个很好的 baseline



1.2 problem

学习输入到输出的映射 $f: X \rightarrow Y$

X : 输入

Y : 输出

- 定义条件概率: $P(Y|X) = ?$ $\Leftrightarrow (20, 4000, \text{本科}) (Y=)$
 $P(Y= | (20, 4000, \text{本科})) \uparrow$
- 假设我们明确知道条件概率 $P(Y|X)$, 怎么做分类?
 $P(Y=|x) \geq ? P(Y=|x) \text{ if } P(Y=|x) > P(Y=|x)$

核心：怎么去设计条件概率

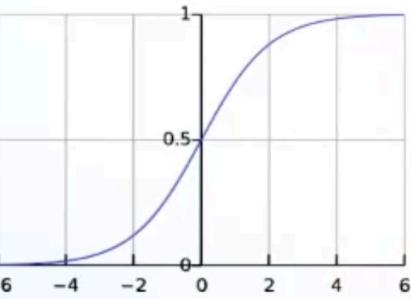
- 可不可以用线性回归来表示 $P(Y|X) = w^T x + b$? 为什么? No!

$$P(y|x) = \frac{1}{1 + e^{-w^T x + b}}$$

① $0 \leq P(y|x) \leq 1$ \otimes

② $\sum_y P(y|x) = 1$

1.3 logistic function



$$y = \frac{1}{1 + e^{-x}}$$

$x: (-\infty, +\infty)$

$y: (0, 1)$

逻辑函数 $y = \frac{1}{1+e^{-x}}$

原始条件概率 : $P(Y|X) = w^T x + b$

新条件概率 $P(Y|X) = \frac{1}{1+e^{-(w^T x + b)}}$

$$\omega = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, b \in \mathbb{R}$$

对于二分类问题：

$$p(y=1|x, w) = \frac{1}{1 + e^{-w^T x + b}}$$

$$p(y=0|x, w) = \frac{e^{-w^T x + b}}{1 + e^{-w^T x + b}} = 1 - p(y=1|x, w)$$

两个式子可以合并成：

$$p(y|x, w) = p(y=1|x, w)^y [1 - p(y=1|x, w)]^{1-y}$$

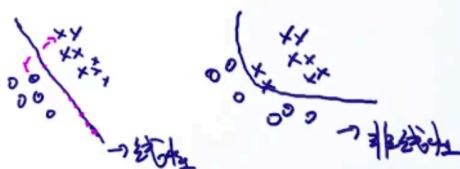
$$\text{if } y=1 \text{ 时. } P(Y=1|x, w) = p(y=1|x, w)$$

1.4 linear classifier

$$p(y=1|x, w) = \frac{(1)}{1 + e^{-w^T x + b}}$$

$$p(y=0|x, w) = \frac{e^{-w^T x + b}}{1 + e^{-w^T x + b}}$$

决策边界 (Decision Boundary)



1.5 objective function

假设我们拥有数据集 $D = \{(x_i, y_i)\}_{i=1}^n \quad x_i \in R^d, \quad y_i \in \{0, 1\}$

而且我们已经定义了：

$$p(y|x, w) = p(y=1|x, w)^y [1 - p(y=1|x, w)]^{1-y}$$

我们需要最大化目标函数：

$$\hat{w}_{MLE}, \quad \hat{b}_{MLE} = \operatorname{argmax}_w \prod_{i=1}^n p(y_i|x_i, w, b)$$

$$\begin{aligned} \hat{w}_{MLE}, \quad \hat{b}_{MLE} &= \operatorname{argmax}_{w,b} \prod_{i=1}^n p(y_i|x_i, w, b) \quad \textcircled{1} \\ &= \operatorname{argmax}_{w,b} \log \left(\prod_{i=1}^n p(y_i|x_i, w, b) \right) \\ &= \operatorname{argmax}_{w,b} \sum_{i=1}^n \log p(y_i|x_i, w, b) \end{aligned}$$

$\log(xyz) = \log x + \log y + \log z$

Objective Function

$$\begin{aligned} \operatorname{argmax}_{w,b} &\sum_{i=1}^n \log p(y_i|x_i, w, b) \\ \operatorname{argmin}_{w,b} &- \sum_{i=1}^n \log p(y_i|x_i, w, b) \\ &= \operatorname{argmin}_{w,b} - \sum_{i=1}^n \log \left[p(y_i|x_i, w, b)^y \cdot [1 - p(y_i|x_i, w, b)]^{1-y} \right] \\ &= \operatorname{argmin}_{w,b} - \sum_{i=1}^n y \cdot \log p(y_i|x_i, w, b) + (1-y) \log [1 - p(y_i|x_i, w, b)] \end{aligned}$$

$\log(a^y \cdot b^x) = \log(a^y) + \log(b^x) = y \log a + x \log b$

2. 梯度下降

2.1 求解函数的最小值

求使得 $f(w)$ 值最小的参数 w

- 是否凸函数
- 最优化算法
 - $f'(w) = 0 \Rightarrow w = ?$
 - Iterative
 - GD (Gradient Descent)
 - SGD (Stochastic GD)

Global Optimal vs Local Optimal

2.2 梯度下降

求使得 $f(w)$ 值最小的参数 w

初始化 w^1
for $t = 1, 2, \dots$:
 $w^{t+1} = w^t - \eta \nabla f(w^t)$

Gradient Descent

求使得 $f(w)$ 值最小的参数 w

初始化 w^1
for $t = 1, 2, \dots$:
 $w^{t+1} = w^t - \eta \nabla f(w^t)$

例子：求解函数 $f(w) = 4w^2 + 5w + 1$ 的最优解

$\eta = -\frac{b}{2a} = -\frac{5}{8}$

$w^1 = 0, f(w) = 8w + 5, \eta = 0.1$
 $w^2 = w^1 - \eta \cdot (8 \cdot 0 + 5) = 0 - 0.5 = -0.5$
 $w^3 = w^2 - \eta \cdot (8 \cdot (-0.5) + 5) = -0.5 - 0.5 = -0.6$
 $w^4 = w^3 - \eta \cdot (8 \cdot (-0.6) + 5) = -0.6 - 0.5 = -0.62$
 $w^5 = w^4 - \eta \cdot (8 \cdot (-0.62) + 5) = -0.62 - 0.5 = -0.625$

0.625 左右 所以我们通过这种计算的方法

2.3 逻辑回归的梯度下降

$$p(y=1|x, w) = \frac{1}{1 + e^{-w^T x + b}}$$

$$\operatorname{argmin}_{w,b} - \sum_{i=1}^n y_i \log p(y_i=1|x_i, w) + (1-y_i) \log(1-p(y_i=1|x_i, w))$$

$$\begin{aligned}
 &= \operatorname{argmin}_{w,b} - \sum_{i=1}^n y_i \cdot \log \sigma(w^T x_i + b) + (1-y_i) \cdot \log [1 - \sigma(w^T x_i + b)] \\
 \boxed{\frac{\partial L(w,b)}{\partial w}} &= - \sum_{i=1}^n y_i \cdot \frac{\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{\sigma(w^T x_i + b)} \cdot x_i + (1-y_i) \cdot \frac{\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{1 - \sigma(w^T x_i + b)} \cdot x_i \\
 &= - \sum_{i=1}^n y_i \cdot (1 - \sigma(w^T x_i + b)) + (1-y_i) \cdot \sigma(w^T x_i + b) \cdot x_i \\
 &\Rightarrow \sum_{i=1}^n [y_i \cdot \sigma(w^T x_i + b)] \cdot x_i = \sum_{i=1}^n [\sigma(w^T x_i + b) - y_i] \cdot x_i
 \end{aligned}$$

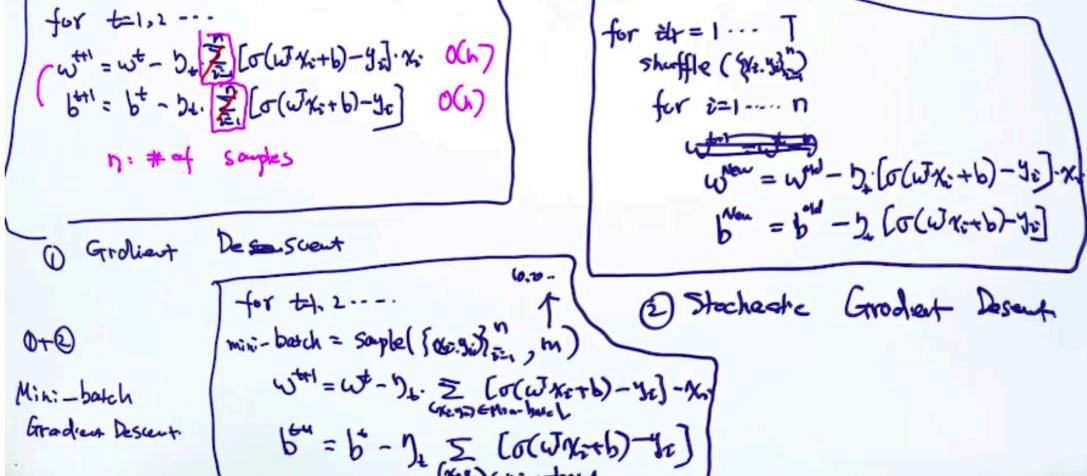
$$\begin{aligned}
 L(w,b) &= - \sum_{i=1}^n y_i \cdot \sigma(w^T x_i + b) + (1-y_i) \log [1 - \sigma(w^T x_i + b)] \\
 \boxed{\frac{\partial L(w,b)}{\partial b}} &= - \sum_{i=1}^n y_i \cdot \frac{\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{\sigma(w^T x_i + b)} + (1-y_i) \cdot \frac{-\sigma(w^T x_i + b) \cdot [1 - \sigma(w^T x_i + b)]}{1 - \sigma(w^T x_i + b)} \\
 &= - \sum_{i=1}^n y_i \cdot (1 - \sigma(w^T x_i + b)) + (1-y_i) \cdot (-\sigma(w^T x_i + b)) \\
 &= \sum_{i=1}^n [\sigma(w^T x_i + b) - y_i]
 \end{aligned}$$

初始化 w^0, b^0
for $t=1, 2, \dots$ learning rate
 $w^{t+1} = w^t - \eta \cdot \sum_{i=1}^n [\sigma(w^T x_i + b^t) - y_i] \cdot x_i$
 $b^{t+1} = b^t - \eta \cdot \sum_{i=1}^n [\sigma(w^T x_i + b^t) - y_i]$

Gradient Descent



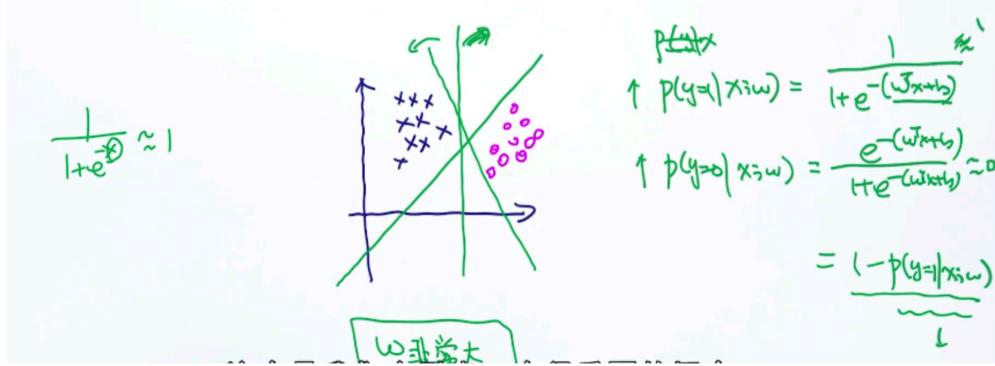
Stochastic Gradient Descent for Logistic Regression



3. 正则化

3.1 问题

如果数据线性可分，那么使用逻辑回归的时候， w 会趋于无穷大。（过拟合）



3.2 L2 正则化

Problem: avoid w become too large

Adding a Term – L2 Norm (正则化)

$|w| \gg 100/1000,$
magnitude
 $\hat{w}_{MLE}, \hat{b}_{MLE} = \underset{\substack{\min \\ w, b}}{\operatorname{argmax}} \prod_{i=1}^n p(y_i | x_i, w) + \lambda \cdot \|w\|_2^2$

目标函数
 $\hat{w}_{MLE}, \hat{b}_{MLE} = \underset{\substack{\min \\ w, b}}{\operatorname{argmax}} \prod_{i=1}^n p(y_i | x_i, w) + \lambda \cdot \|w\|_2^2$

$\lambda: \text{Hyperparameter (超参数)} \Rightarrow \text{weighting factor}$

a: if $\lambda=0$, no restriction
b: λ large, w 变得很小
c: λ 小, w 变得很大.

$\hat{w}_{MLE}, \hat{b}_{MLE} = \underset{\substack{\min \\ w, b}}{\operatorname{argmax}} \prod_{i=1}^n p(y_i | x_i, w, b) + \lambda \cdot \|w\|_2^2$

$\frac{\partial L(w, b)}{\partial w} = \sum_{i=1}^n (\underbrace{\sigma(w^T x_i + b) - y_i}_{\text{prediction}} \cdot x_i) + 2 \cdot \lambda w$

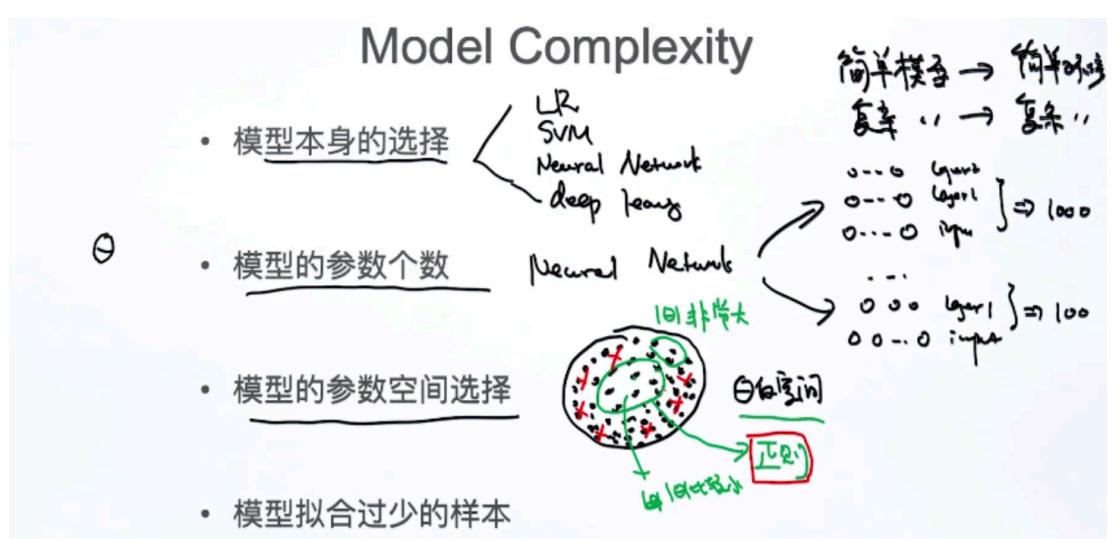
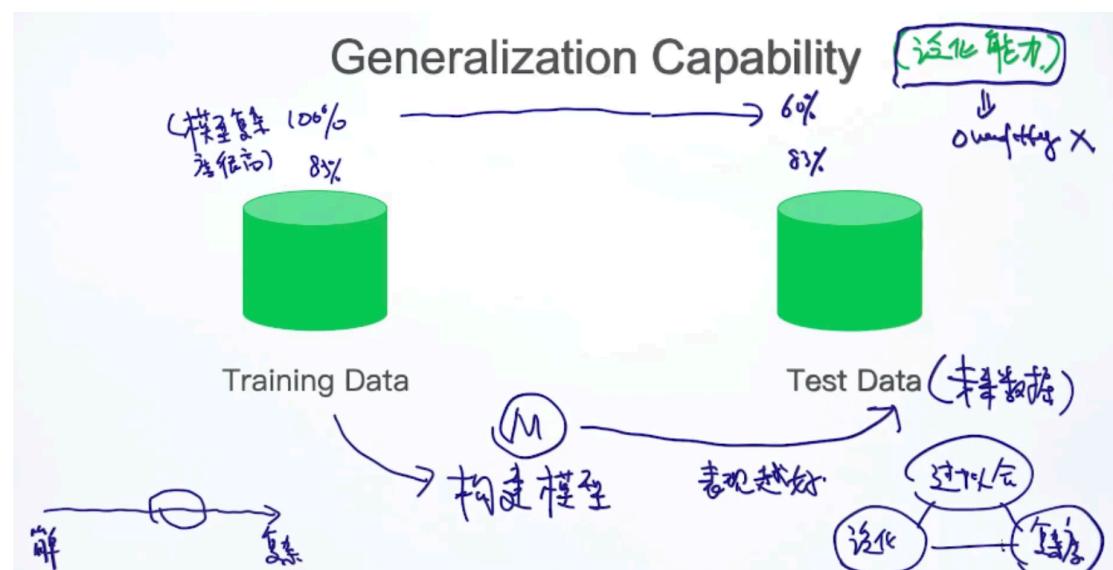
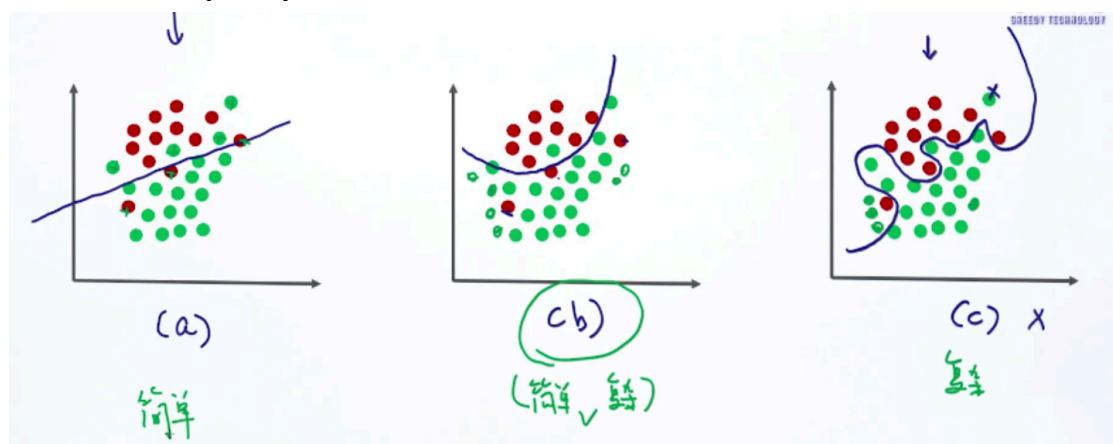
$\frac{\partial L(w, b) + R(w)}{\partial w} = \sum_{i=1}^n (\underbrace{\sigma(w^T x_i + b) - y_i}_{\text{prediction}} \cdot x_i + 2 \cdot \lambda w) \quad (\text{Batch Gradient Descent})$

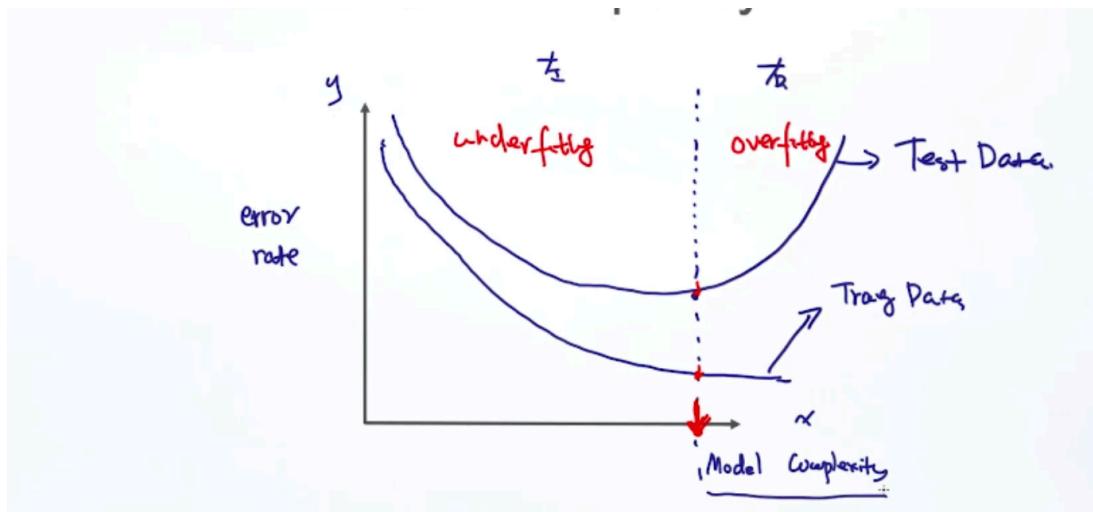
$\frac{\partial L(w, b) + R(w)}{\partial w} = (\underbrace{\sigma(w^T x_i + b) - y_i}_{\text{prediction}} \cdot x_i + 2 \cdot \lambda w) \leftarrow (\text{Stochastic gradient descent})$

我们就采用这种方式去不断地更新咱们的 w 的

4. 过拟合

4.1 model complexity





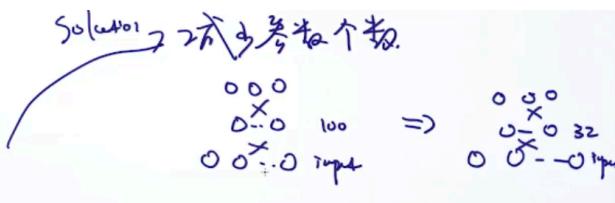
4.2 avoid overfitting

模型复杂度源于

- 模型本身的选择
- 模型的参数个数
- 模型的参数空间选择
- 模型拟合过少的样本

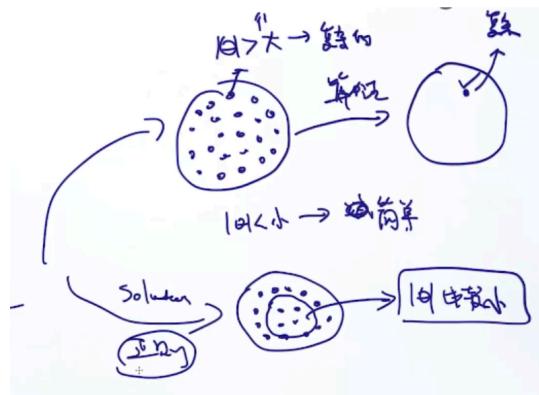
Solution → 选择更简单的模型
(LR, SVM Class)

- 模型本身的选择
- 模型的参数个数
- 模型的参数空间选择
- 模型拟合过少的样本



模型复杂度源于

- 模型本身的选择
- 模型的参数个数
- 模型的参数空间选择
- 模型拟合过少的样本



- 模型本身的选择
- 模型的参数个数
- 模型的参数空间选择
- 模型拟合过少的样本

solver → 手动取更多样本

5. 正则化

5.1 recap

$$\text{L}_2\text{-Norm} \rightarrow \text{LR}$$

$$\arg \min - \underbrace{\sum_{i=1}^n P(y_i | x_i; w)}_{\text{objective}} + \lambda \underbrace{\|w\|_2^2}_{\text{regularization}}$$

Hyperparameter (超参数)

$\lambda \uparrow \rightarrow \|w\|_2 \uparrow$

$\lambda \downarrow \rightarrow \|w\|_2 \downarrow$

$\lambda = 0 \text{ 时 } \text{没有正则限制}$

$$\|w\|_2^2: \text{L}_2\text{-Norm} \quad w_1^2 + w_2^2 + \dots + w_d^2 = \|w\|_2^2$$

5.2 Regularization Terms

The diagram illustrates four types of regularization terms:

- L0-Norm**: $\|w\|_0$
- Nuclear Norm**: $\|A\|_*$, $\|A\|_F$, $\text{rank}(A)$. It shows a matrix A with rank 1.
- L1-Norm**: $\|w\|_1 = |w_1| + |w_2| + |w_3| + \dots + |w_d|$
- L2-Norm**: $\|w\|_2 = \sqrt{\sum_{i=1}^d w_i^2}$
- L_p-Norm**: $\|A\|_p = \left(\sum_{i=1}^M \sum_{j=1}^N |a_{ij}|^p \right)^{1/p}$ (labeled as Sparse)

L1：会把很多参数变为 0. 遇到比较稀疏的问题，可以选择用 L1

L1 和 L2 比较常见

第四种：nuclear norm

限制矩阵的 rank, 把 rank 比较大的矩阵去掉
还有很多种范数。

$$\widehat{w}_{MLE}, \widehat{b}_{MLE} = \underset{\substack{\text{objective} \\ \text{regularization}}}{\operatorname{argmax}}_{w,b} \prod_{i=1}^n p(y_i|x_i, w, b) + \lambda \|w\|_1$$

$$\|w\|_1 \Rightarrow L_1\text{-Norm of } w$$

$$\|w\|_1 = |w_1| + |w_2| + \dots + |w_d|$$

$L_1 \leq L_2$

Magnitude of w 较小

5.3 L1 vs L2

$$\Theta: \underset{\substack{\text{objective}}}{\operatorname{argmin}} f(\Theta) \xrightarrow{L_2} \underset{\substack{\text{objective} \\ \text{regularization}}}{\operatorname{argmin}} f(\Theta) + \lambda \| \Theta \|_2^2$$

$L_1 \leq L_2$

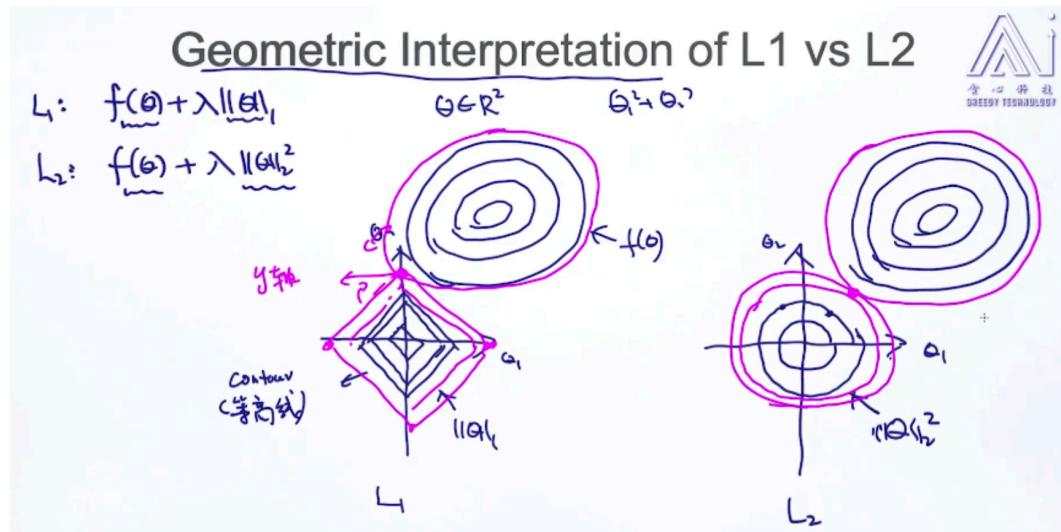
- Make Θ smaller

$L_1 \leq L_2$ (证明)

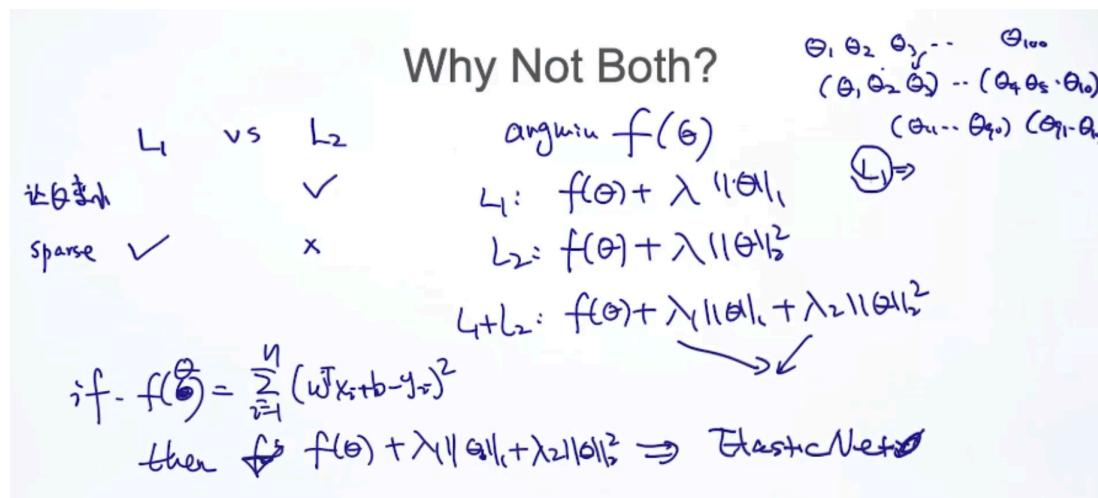
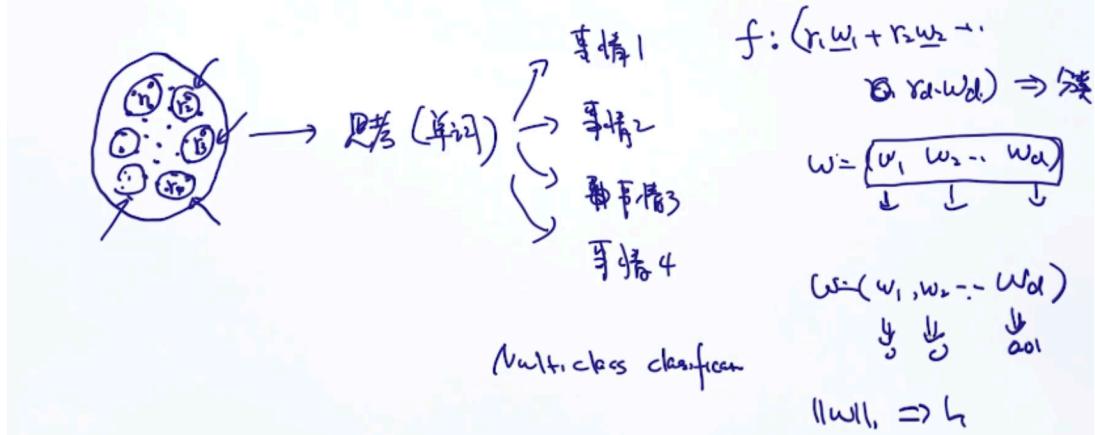
- L_1 induces Sparse solution

$$\begin{aligned} \textcircled{1} \quad \hat{\Theta}_{L_2} &= (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_d) \xrightarrow{\text{Non-Sparse Solution}} \text{Non-Sparse Solution} \\ \textcircled{2} \quad \hat{\Theta}_{L_1} &= (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_d) \xrightarrow{\text{Sparse Solution}} \text{Sparse Solution} \end{aligned}$$

为什么 L_1 稀疏, L_2 不稀疏



Some Applications of L1 Regularization



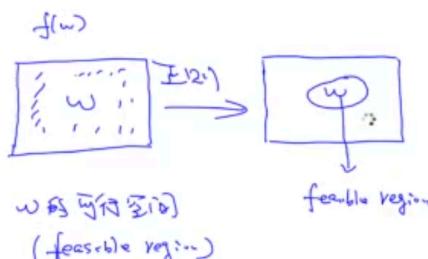
5.4 正则的作用

正则:

$$\begin{aligned} a) \min_w f(w) & \cdots \text{无正则} & \frac{\hat{w}_1}{\hat{w}_2} \\ b) \min_w f(w) + \lambda \|w\|_2^2 & \cdots \text{有正则} & \end{aligned}$$

$f(\hat{w}_1) \leq f(\hat{w}_2)$

$f(\hat{w}_1) \leq f(\hat{w}_2)$



相当于对原来的空间做了过滤。

5.5 正则化的灵活运用

1) neuron science

w_{11}, \dots, w_{1n} r_1 : nr. of neurons in region 1
 w_{21}, \dots, w_{2r_2} r_2 : nr. of neurons in region 2
 w_{p1}, \dots, w_{pr_p} r_p : nr. of neurons in region p

[某一个区域内只有部分 neuron 会被激活 ①
 空间上, 相邻的作用也类似 ② → 将条件加入到目标函数]

old: minimize $f(w)$

New: minimize $f(w) + \sum_{i=1}^p \lambda_i \|w_i\|_F + \sum_{i=1}^p \sum_{j=1}^{r_i} \|w_j - w_{j-1}\|^2$

通过正则化的方式加条件 ① ②

2) time-aware recommendation

Time-Aware Recommendation

Matrix Factorization

user-rating matrix R (sparse) = user matrix U (sparse) \times item matrix V (dense) \times factor count K

$R_{ij} \approx U_i^T \cdot V_j$ $\|U_i^T - U_i^{\text{true}}\|_F^2$

minimize $\sum_{(i,j) \in R} (R_{ij} - U_i^T \cdot V_j)^2 + \frac{\lambda_u}{2} \|U_i^T\|_F^2 + \frac{\lambda_v}{2} \|V_j\|_F^2$

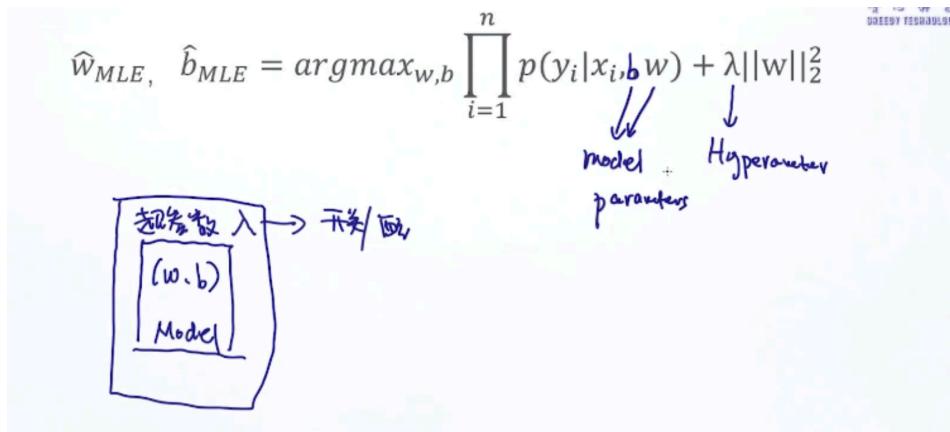
λ_u : # of users λ_v : # of items

通过引入时间信息

minimize $\sum_{t=1}^T \sum_{(i,j) \in R_{it}} (R_{ij}^t - (U_i^t \cdot V_j^t))^2 + \frac{\lambda_u}{2} \sum_{t=1}^T \|U_i^t\|_F^2 + \frac{\lambda_v}{2} \sum_{t=1}^T \|V_j^t\|_F^2 + \frac{\lambda_{ut}}{2} \sum_{t=1}^T \|U_i^t - U_i^{\text{true}}\|_F^2 + \frac{\lambda_{vt}}{2} \sum_{t=1}^T \|V_j^t - V_j^{\text{true}}\|_F^2$

6. cross validation 交叉验证

6.1 如何选择 lambda



Cross-validation

LR minimize $-\sum_{i=1}^n y_i \log P(y_i|x_i; w) + (1-y_i) \log [1 - P(y_i|x_i; w)] + \lambda ||w||_2^2$

$\begin{aligned} \text{minimize } & f(w) + \lambda ||w||_2^2 \\ & \text{objective} \quad \text{regularization} \end{aligned}$

$\lambda = 0 \Rightarrow \text{无正则}$
 $\lambda \text{越大} \Rightarrow \text{正则作用} \uparrow$
 $w \downarrow$

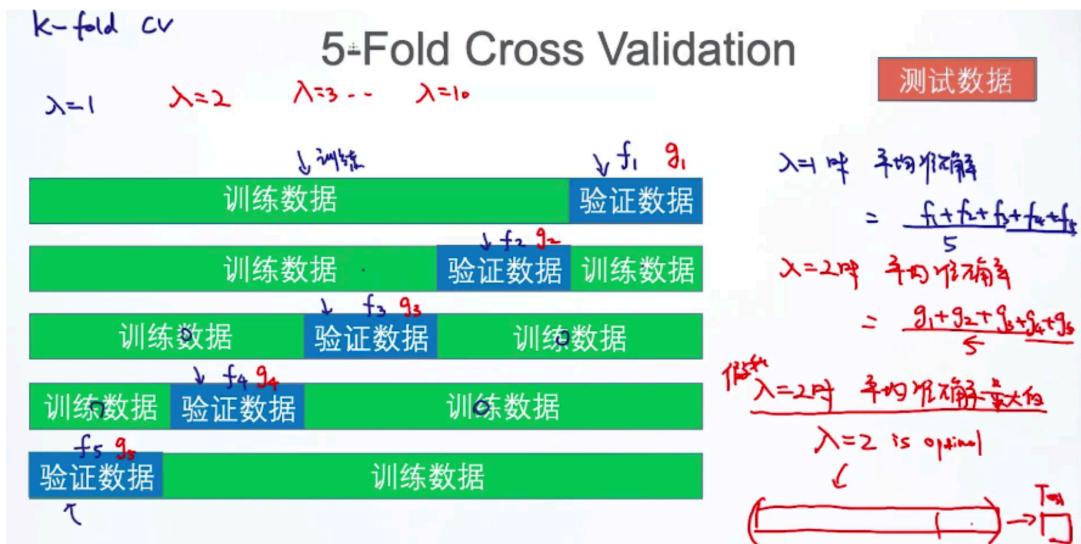
$\lambda = 1, \lambda = 2, \lambda = 3$
 \vdots
 入: 超参数
 W → Model
 模型参数

Intuition

把训练数据进一步分成训练数据 (Training Data) 和验证集 (Validation Data)。选择在验证数据里最好的超参数组合。



6.2 交叉验证



绝对不能用测试数据来引导(guide)模型的训练!

7. MLE & MAP

7.1

MLE 最大似然估计
 通过观测值 samples 来
 估计参数 θ^*

$$HHHTTH : \theta^* = \frac{2}{3}$$

MAP 最后验估计
 不仅依赖于 sample, 同时
 依赖于 prior

$$\left\{ \begin{array}{l} HHHTTH : \theta^* = \frac{2}{3} \\ prior = 80\% \\ \therefore \theta^* : 67\% \sim 80\% \\ prior 随着 sample \uparrow, 重要性 \downarrow \end{array} \right.$$

$$\arg \max P(D|\theta)$$

↓
观测值 参数

$$\begin{aligned} \arg \max P(\theta|D) &\rightarrow \text{post } p \\ &= \arg \max P(D|\theta) P(\theta) \end{aligned}$$

7.2 from Gaussian prior to L2 regularization

$$P(D|w, b) = \prod_{i=1}^n P(y_i|x_i; w, b)$$

MLE: $\operatorname{argmax} P(D|\theta)$

$$= \operatorname{argmax} \sum_{i=1}^n P(y_i|x_i; w)$$

$$= \operatorname{argmax} \sum \log P(y_i|x_i; w)$$

MAP: $\operatorname{argmax} P(D|\theta) \cdot P(\theta)$

$$= \underbrace{\log P(D|\theta)}_{\text{MLE}} + \underbrace{\log P(\theta)}_{?}$$

$$= \underbrace{\sum \log P(y_i|x_i; w)}_{\text{MLE}} + \underbrace{\log P(\theta)}_{\theta = \{w\}}$$

$$\text{Ex } P(\theta) = P(w) \sim N(0, \sigma^2) \quad (\text{假设})$$

$$P(w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{w^2}{2\sigma^2}\right) \rightarrow \text{pdf}$$

$$\begin{aligned} \log P(\theta) &= \log P(w) \\ &= -\log(\sqrt{2\pi}\sigma) - \frac{w^2}{2\sigma^2} \end{aligned}$$

$$= \underbrace{\sum \log P(y_i|x_i; w)}_{\text{MLE}} - \log(\sqrt{2\pi}\sigma) - \frac{w^2}{2\sigma^2}$$

$$= \underbrace{\sum \log P(y_i|x_i; w)}_{\text{L2 正则}} - \frac{1}{2\sigma^2} \|w\|_2^2$$

$$\lambda = \frac{1}{2\sigma^2}$$

7.3 from Laplace prior to L1 regularization

$$P(D|w, b) = \prod P(y_i|x_i, w, b)$$

MLE: arg max $P(D|\theta)$

$$= \arg \max \sum_{i=1}^n \log P(y_i|x_i; w)$$

假设 $p(\theta) = p(w) \sim \text{Laplace } (\mu, b)$

$$p(w) \sim L(0, b)$$

$$= \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$$

MAP: arg max $\log P(D|\theta) + \log p(\theta)$

$$= \dots + \sum \log P(y_i|x_i; w) + \log p(\theta)$$

$$\log p(\theta) = \log p(w)$$

$$= \log\left(\frac{1}{2b}\right) - \frac{|x|}{b}$$

$$= \dots + \sum \log P(y_i|x_i; w) - \frac{|x|}{b}$$

L1 regularization

$$\lambda = -\frac{1}{b}$$

7.4

Adding Prior is Equivalent to Regularization

Logistic Regression

任何模型

Gaussian Prior $\Rightarrow L_2$ Reg

Laplace Prior $\Rightarrow L_1$ Reg

7.5 MAP \rightarrow MLE

当参数非常多时，MAP Solution \rightarrow MLE Solution

$$P \text{ MAP: } \arg \max \log P(D|\theta) + \log p(\theta)$$

$$= \arg \max \underbrace{\log \sum_{i=1}^n \log P(y_i|x_i; w)}_{\text{MLE}} + \underbrace{\log p(\theta)}_{\text{prior.}}$$

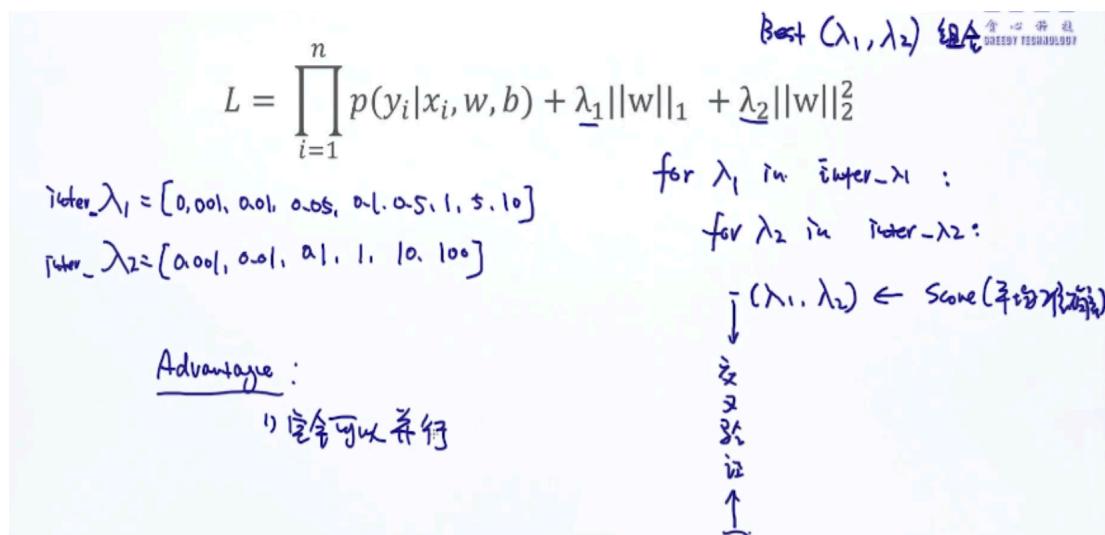
$h \rightarrow +\infty$ 时 MAP 解 = MLE 解

8. 参数搜索策略

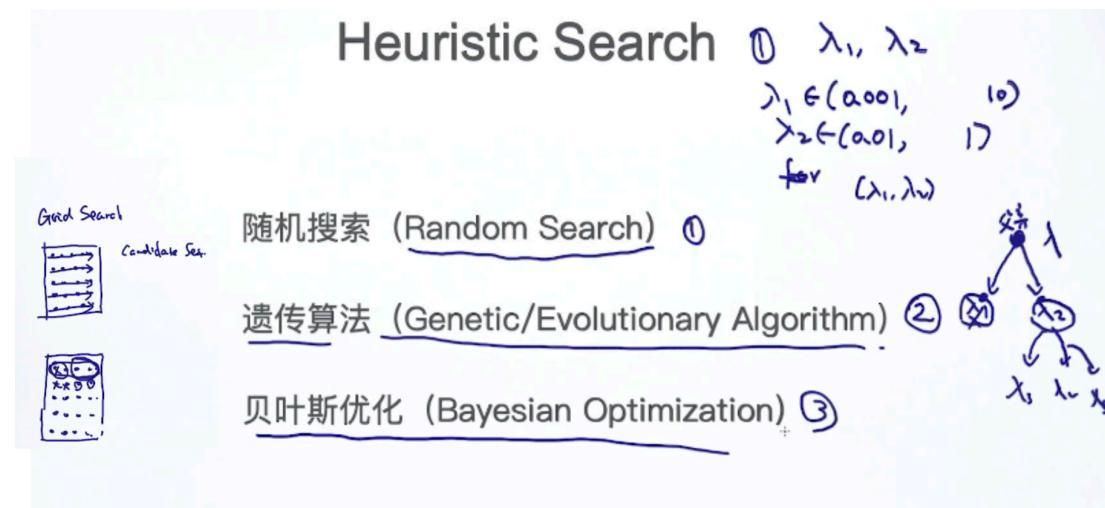
$$L = \prod_{i=1}^n p(y_i|x_i, w, b) + \lambda_1 ||w||_1 + \lambda_2 ||w||_2^2$$

How to search for λ_1 and λ_2 ?

8.1 Grid Search

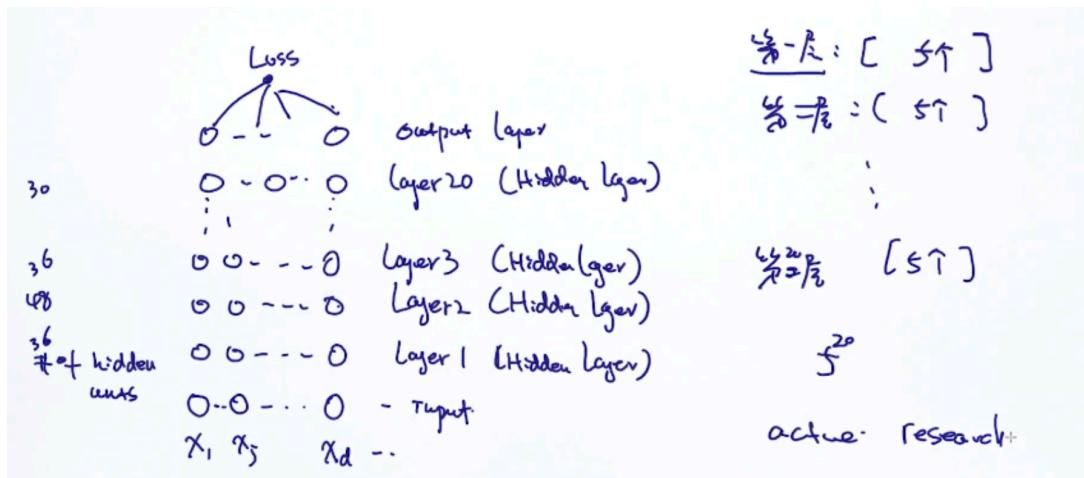


8.2 heuristic search



8.3 deep network

每一层都包含了一个超参数，神经元的个数



9. summary

- 好的模型拥有高的泛化能力
- 越复杂的模型越容易过拟合
- 添加正则项是防止过拟合的一种手段
- L1正则会带来系数特性
- 选择超参数时使用交叉验证
- 参数搜索过程最耗费资源