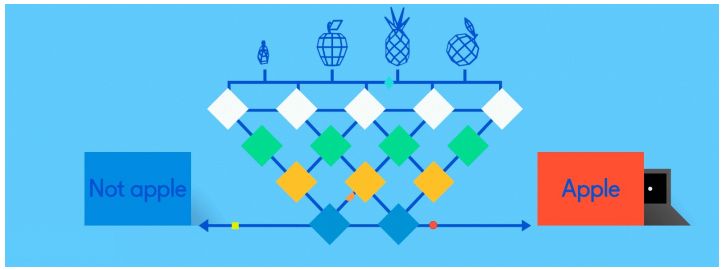
# 第十八讲 预测II

2022.12.30



# 复共线性（collinearity）

**复共线性**

回归分析中如果某些或全部自变量之间存在较强的相关性，表现为设计阵$X$的各列近似是线性相关的，这种线性称为复共线性。复共线性会导致某些LS估计的方差会过大，进而均方误差偏大。

例1. 假设$(y_i, x_{i1}, x_{i2}), i = 1,2,\ldots,n,$满足模型：

$y_i = a + bx_{i1} + cx_{i2} + \varepsilon_i$， $\varepsilon_i$ iid $\sim (0,\sigma^2)$, $\varepsilon_i$ 与 $x_{i1}, x_{i2}$ 独立,

即$\mathbf{y} = \mathbf{1}a + \mathbf{x}_1 b + \mathbf{x}_2 c + \boldsymbol{\varepsilon}$。

记$r_{12}$为$\mathbf{x}_1$与$\mathbf{x}_2$的相关系数，$S_{y1}$为$\mathbf{y}$与$\mathbf{x}_1$的样本协方差，等等。

由第十二讲命题3, $\hat{b}$的方差

$$\mathrm{var}(\hat{b} \mid X) = \frac{\sigma^2}{S_{11\bullet 2}} = \frac{\sigma^2}{S_{11} - S_{12}^2/S_{22}} = \frac{\sigma^2}{S_{11}} \times \frac{1}{1 - r_{12}^2},$$

对比$\mathbf{x}_1$与$\mathbf{x}_2$不相关时的方差$\frac{\sigma^2}{S_{11}}$，方差增大了倍数$\mathrm{VIF} = \frac{1}{1 - r_{12}^2}$，

称为方差扩大因子(VIF，variance inflation factor）。

下面考察一般模型中LS估计$\hat{\beta}_k$的方差大小与复共线性 之间的关系。

模型$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \mathbf{1}\beta_0 + \mathbf{x}_{(1)}\beta_1 + \ldots + \mathbf{x}_{(p-1)}\beta_{p-1} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim (0, \sigma^2 I_n),$

对$k \geq 1$, 若$\mathbf{x}_{(k)}$与其它自变量不相关， 则LS估计$\hat{\beta}_k = \mathbf{x}_{(k)}^{\mathsf{T}}\mathbf{y} / \| \mathbf{x}_{(k)} \|^2$ ，

$$\mathrm{var}_0(\hat{\beta}_k \mid X) = \frac{\sigma^2}{\| \mathbf{x}_{(k)} - \mathbf{1}\bar{x}_{(k)} \|^2} \qquad (1)$$

一般地，由第十二讲命 题3（LS估计的分量）， $\hat{\beta}_k = \mathbf{x}_{(k)}^{\perp\mathsf{T}}\mathbf{y} / \| \mathbf{x}_{(k)}^{\perp} \|^2$,

$$\mathrm{var}(\hat{\beta}_k \mid X) = \sigma^2 / \| \mathbf{x}_{(k)}^{\perp} \|^2 , \qquad (2)$$

其中$\mathbf{x}_{(k)}^{\perp} = \mathbf{x}_{(k)} - P_{X_{(-k)}}\mathbf{x}_{(k)} = \mathbf{x}_{(k)} - \hat{\mathbf{x}}_{(k)}$为$\mathbf{x}_{(k)}$相对于其它列的正交化 。相对于正交情形下的方差（2）， （1）中$\hat{\beta}_k$的方差增大的倍数为

$$VIF = \frac{1}{1 - R_k^2},$$

其中$R_k^2 = \| \hat{\mathbf{x}}_{(k)} - \mathbf{1}\bar{x}_k \|^2 / \| \mathbf{x}_{(k)} - \mathbf{1}\bar{x}_k \|^2$ 为$\mathbf{x}_{(k)}$与其它自变量的决定系 数。

# L2惩罚最小二乘: 岭回归

设计阵$X$存在复共线性时，$X^{\mathsf{T}}X$不可逆或接近于不可逆。

通常以条件数$\lambda_{\max}(X^{\mathsf{T}}X)/\lambda_{\min}(X^{\mathsf{T}}X)$度量复共线的程度。

若条件数过大（比如大于1000），即认为存在严重的复共线性，

此时LS估计的方差$\mathrm{var}(\hat{\boldsymbol{\beta}})=\sigma^2(X^{\mathsf{T}}X)^{-1}$的某些对角元会很大。

岭估计

模型：$\mathbf{y}=X\boldsymbol{\beta}+\boldsymbol{\varepsilon},\boldsymbol{\varepsilon}\sim(0,\sigma^2 I_n)$。定义岭估计(ridge estimator)

$$\tilde{\boldsymbol{\beta}}^{(\mathrm{Ridge})}=(X^{\mathsf{T}}X+\lambda I_p)^{-1}X^{\mathsf{T}}\mathbf{y}\ ,$$

其中$\lambda>0$为常数（Hoerl and Kennard，1970）

什么是岭(ridge)? 当目标函数的二阶导数矩阵不可逆时（LS中二阶导数为$-X^{\mathsf{T}}X$），最大值附近不是一个山峰而是等高的山脊(*ridge*)，最大点不唯一或不稳定。

L2惩罚最小二乘得到岭估计

$\tilde{\boldsymbol{\beta}}^{(\text{Ridge})}$ 是如下惩罚最小二乘 (penalized LS) 估计或规则化最小二乘估计的解 (regularize d LS):

$$\tilde{\boldsymbol{\beta}}^{(\text{Ridge})} = \text{argmin}\left\{ \| \mathbf{y} - X\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|^2 \right\}$$

$$\Leftrightarrow \tilde{\boldsymbol{\beta}}^{(\text{Ridge})} = \text{argmin} \| \mathbf{y} - X\boldsymbol{\beta} \|^2, \quad \text{约束} \| \boldsymbol{\beta} \| < t$$

命题1.

(a) 岭估计是压缩估计: $\| \tilde{\boldsymbol{\beta}}^{(\text{Ridge})} \| < \| \hat{\boldsymbol{\beta}} \|$, $\hat{\boldsymbol{\beta}} = (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}\mathbf{y}$

(b) 存在$\lambda > 0$, 使得$m(\tilde{\boldsymbol{\beta}}^{(\text{Ridge})}) \le m(\hat{\boldsymbol{\beta}})$,

即基于某些岭估计比基于LS估计的预测误差更小。

证明较为复杂，我们只考虑一个简单情形（命题2）

$XX^{\mathsf{T}} = I_p$ 情形

命题2. 假设$X^{\mathsf{T}}X = I_p$,则岭估计 $\tilde{\boldsymbol{\beta}}^{(\text{ridge})} = \hat{\boldsymbol{\beta}}/(1+\lambda)$（压缩估计），且 $m(\tilde{\mathbf{y}}^{(\text{ridge})}) = \dfrac{p\sigma^2}{(1+\lambda)^2} + \dfrac{\lambda^2}{(1+\lambda)^2}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta}$，其最小值在$\lambda_{\text{optimal}} = \dfrac{p\sigma^2}{\|\boldsymbol{\beta}\|^2}$达到，且最小值小于$m(\hat{\mathbf{y}}) = p\sigma^2$.

证明：$\tilde{\boldsymbol{\beta}}^{(\text{ridge})} = (X^{\mathsf{T}}X + \lambda I_p)^{-1}X^{\mathsf{T}}\mathbf{y} = X^{\mathsf{T}}\mathbf{y}/(1+\lambda)$，则

方差：$\text{var}(\tilde{\boldsymbol{\beta}}^{(\text{ridge})}) = \sigma^2 I_p/(1+\lambda)^2 \quad \downarrow \lambda$

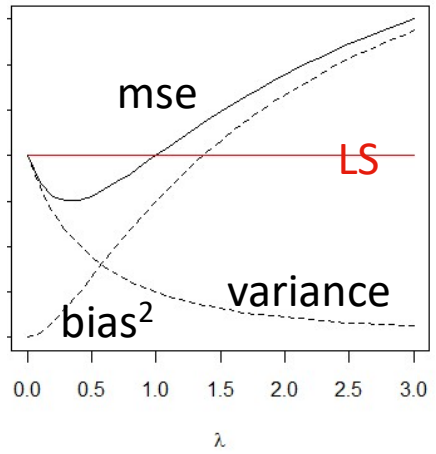
偏差：$\mathbf{b} = bias(\tilde{\boldsymbol{\beta}}^{(\text{ridge})}) = E(\tilde{\boldsymbol{\beta}}^{(\text{ridge})}) - \boldsymbol{\beta} = \lambda\boldsymbol{\beta}/(1+\lambda) \quad \uparrow \lambda$

$$\Rightarrow M(\tilde{\boldsymbol{\beta}}^{(\text{ridge})}) = \text{var}(\tilde{\boldsymbol{\beta}}^{(\text{ridge})}) + \mathbf{b}\mathbf{b}^{\mathsf{T}} = \sigma^2 I_p/(1+\lambda)^2 + \left(\lambda^2/(1+\lambda)^2\right)\boldsymbol{\beta}\boldsymbol{\beta}^{\mathsf{T}},$$

$$M(\tilde{\mathbf{y}}^{(\text{ridge})}) = XM(\tilde{\boldsymbol{\beta}}^{(\text{ridge})})X^{\mathsf{T}}$$

$$\Rightarrow m(\tilde{\mathbf{y}}^{(\text{ridge})}) = trM(\tilde{\mathbf{y}}^{(\text{ridge})}) = trM(\tilde{\boldsymbol{\beta}}^{(\text{ridge})}) = \frac{p\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2}{(1+\lambda)^2}\|\boldsymbol{\beta}\|^2,$$

其余易证。




**岭回归与JS估计**

$\lambda_{\text{optimal}} = \dfrac{p\sigma^2}{\|\boldsymbol{\beta}\|^2}$中含未知参数，需要代入其估计。

因为 $\text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2 I_p, E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$, 所以 $E\|\hat{\boldsymbol{\beta}}\|^2 = \|\boldsymbol{\beta}\|^2 + p\sigma^2$

以 $\|\hat{\boldsymbol{\beta}}\|^2 - p\hat{\sigma}^2$ 估计 $\|\boldsymbol{\beta}\|^2$，得 $\hat{\lambda}_{\text{optimal}} = 1 - \dfrac{p\hat{\sigma}^2}{\|\hat{\boldsymbol{\beta}}\|^2}$，所以“最优”岭估计：

$$\tilde{\boldsymbol{\beta}}^{\text{ridge}}(\hat{\lambda}_{\text{optimal}}) = \left(1 - \frac{p\hat{\sigma}^2}{\|\hat{\boldsymbol{\beta}}\|^2}\right)\hat{\boldsymbol{\beta}},$$

该估计称为（回归问题的）James – Stein估计。

# L1惩罚最小二乘: LASSO

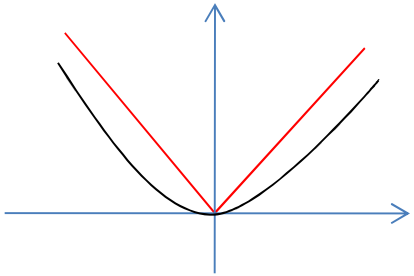
LASSO方法（Least Absolute Shrinkage and Selection Operator，Tibshirani，1996）惩罚回归系数的$L1$模，把接近0的回归系数估计为0，被认为是一种变量选择方法。

LASSO估计：$\tilde{\boldsymbol{\beta}}^{(\text{lasso})} = \text{argmin}\left\{ \frac{1}{2} \| \mathbf{y} - X\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|_1 \right\}$，$\lambda > 0$给定

$$\Leftrightarrow \tilde{\boldsymbol{\beta}}^{(\text{lasso})} = \text{argmin} \| \mathbf{y} - X\boldsymbol{\beta} \|^2, \text{约束} \| \boldsymbol{\beta} \|_1 < t,$$

其中$L_1$模$\| \mathbf{u} \|_1 = \sum | u_i |$，$\lambda$与$t$之间1:1对应。

$L1$与$L2$:



尖锐:二阶导(能量)无穷大

LS的目标函数 $\| \mathbf{y} - X\boldsymbol{\beta} \|^2 = \| \mathbf{y} - X\hat{\boldsymbol{\beta}} \|^2 + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^{\mathsf{T}} X^{\mathsf{T}} X (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}),$

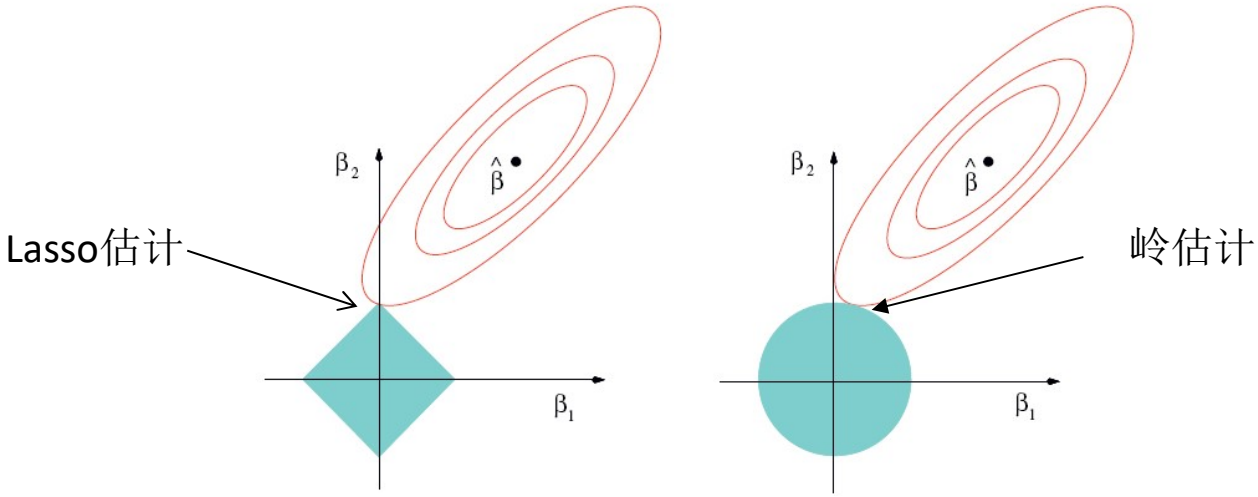
$\min \| \mathbf{y} - X\boldsymbol{\beta} \|^2 \Leftrightarrow \min (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^{\mathsf{T}} X^{\mathsf{T}} X (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})$，$\hat{\boldsymbol{\beta}}$是LS估计

下图蓝色区域分别是$L1$(左图)和$L2$(右图)约束区域。设$\hat{\boldsymbol{\beta}}$不在约束区域。

红色椭圆为目标函数$(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^{\mathsf{T}} X^{\mathsf{T}} X (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})$的等高线。

椭圆与约束区域的切点 满足约束，且使得目标 函数达到最小。

（左图：尖点更容易与 椭球碰到，此时 $\beta_1 = 0$）。



简单回归情况

假设简单回归模型$y_i = x_i\beta + \varepsilon_i$（$y, x$都已中心化），误差平方和

$$f(\beta) = \| \mathbf{y} - \mathbf{x}\beta \|^2 = \mathbf{x}^{\mathsf{T}}\mathbf{x}\beta^2 - 2\mathbf{x}^{\mathsf{T}}\mathbf{y}\beta + \mathbf{y}^{\mathsf{T}}\mathbf{y} \triangleq a\beta^2 - 2b\beta + c$$

L1惩罚的目标函数（$\lambda > 0$给定）：

$$\boxed{Q(\beta) = f(\beta)/2 + \lambda |\beta| = a\beta^2/2 - b\beta + c/2 + \lambda |\beta|,}$$

$\Rightarrow$ 一阶导数：$Q'(\beta) = a\beta - b + \lambda \operatorname{sgn}(\beta)$

$$\Rightarrow \text{最优解} \tilde{\beta}_{\text{lasso}} = \begin{cases} (b-\lambda)/a, & b > \lambda \\ 0, & |b| \le \lambda \\ (b+\lambda)/a, & b < -\lambda \end{cases}$$

当$|b| < \lambda$时，估计值等于0

$$\operatorname{sgn}(b) = \begin{cases} 1 & b > 0 \\ 0 & b = 0, \\ -1 & b < 0 \end{cases}$$

对比LS， 岭回归、LASSO

| | 目标函数 | 一阶导数 | 二阶导数 | 最优解 |
|---|---|---|---|---|
| LS | $f(\beta)/2$ | $a\beta - b = \mathbf{x}^{\mathrm{T}}\mathbf{x}\beta - \mathbf{x}^{\mathrm{T}}\mathbf{y}$ | $a$ | $b/a = \mathbf{x}^{\mathrm{T}}\mathbf{y}/\mathbf{x}^{\mathrm{T}}\mathbf{x}$ |
| 岭回归 | $f(\beta)/2 + \lambda\beta^2$ | $a\beta - b + \lambda\beta$ | $a + \lambda$ | $b/(a+\lambda)$ |
| LASSO | $f(\beta)/2 + \lambda\lvert\beta\rvert$ | $a\beta - b + \lambda sgn(\beta)$ | $a + \lambda\delta(0)$ (0点处无穷大) | $\begin{cases}(b-\lambda)/a & \text{当} b > \lambda \\ (b+\lambda)/a & \text{当} b < -\lambda \\ 0 & \text{当} \lvert b\rvert < \lambda\end{cases}$ |

注1. 一般情形下， LASSO没有显式解，的常用解法有：坐标下降法，least-angle regression (LARS) , 梯度下降法…

注2. 通常只对回归系数（不包括截距）进行约束，故$X, y$首先需要中心化

## 坐标下降法

为了极小化二元函数 $\min_{x,y} f(x,y)$，在给定其中一个变量的条件下，对另一个变量极小化：

初始化$y = y_0$,

$(a)$ 给定$y_0$，对$x$极小化： $x_0 = arg\min_x f(x, y_0)$

$(b)$给定$x_0$，对$y$极小化： $y_0 = arg\min_y f(x_0, y)$

重复$(a)$ - $(b)$.

## 坐标下降法求LASSO估计

不妨设设计阵$X$已经中心化且标准化（ 模为1）。

记设计阵$X$的第$j$列为$\mathbf{x}_j$，其它列组成$X_{(-j)}$。

LASSO方法目标函数 $Q(\boldsymbol{\beta}) = \frac{1}{2}\| \mathbf{y} - X\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|_1$ 中，假设除了$\beta_j$之外，其它 $\boldsymbol{\beta}_{(-j)} = (\beta_1, ..., \beta_{j-1}, \beta_{j+1}, ... \beta_p)^{\mathsf{T}}$给定（假设都已解得），记$\mathbf{y}^* = \mathbf{y} - X_{(-j)}\boldsymbol{\beta}_{(-j)}$已知。目标函数为 $\beta_j$的二次函数：

$$Q(\boldsymbol{\beta}) = \frac{1}{2}\| \mathbf{y} - X_{(-j)}\boldsymbol{\beta}_{(-j)} - \mathbf{x}_j\beta_j \|^2 + \lambda | \boldsymbol{\beta}_{(-j)} |_1 + \lambda | \beta_j |$$

$$= \frac{1}{2}\| \mathbf{y}^* - \mathbf{x}_j\beta_j \|^2 + \lambda | \beta_{(-j)} |_1 + \lambda | \beta_j |$$

$Q(\boldsymbol{\beta})=\frac{1}{2}\beta_j^2-\mathbf{x}_j{}^{\mathsf{T}}\mathbf{y}^*\beta_j+\lambda|\beta_j|+C$，其中 常数$C=\frac{1}{2}\|\mathbf{y}^*\|^2+\lambda|\boldsymbol{\beta}_{(-j)}|_1$.

$Q$作为$\beta_j$的函数，与简单回归情况下LASSO的目标函数一致，

记$b=\mathbf{x}_j{}^{\mathsf{T}}\mathbf{y}^*=(\mathbf{x}_j{}^{\mathsf{T}}\mathbf{x}_j)^{-1}\mathbf{x}_j{}^{\mathsf{T}}\mathbf{y}^*$, 最优解为

$$\beta_j=\begin{cases}b-\lambda & b>\lambda,\\ 0 & |b|\le\lambda,\\ b+\lambda & b<-\lambda,\end{cases}=\operatorname{sgn}(b)(|b|-\lambda)_+$$

对所有$j=1,\ldots,p$进行上述过程，得到所有回归系数的当前解。

给定这些回归系数的当前的解，再次对每个分量依次极小化。

重复直至收敛。

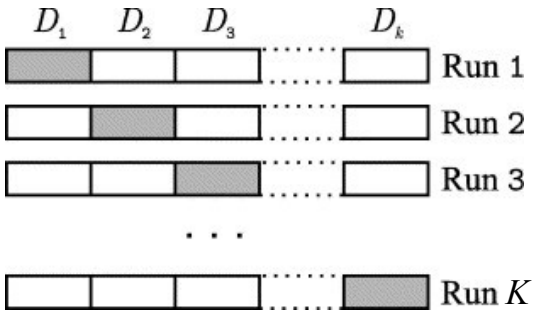
# 交叉验证 (CV, Cross-validation)

$C_p$/AIC/BIC准则都是在一定假设下推导得到的预测精度准则，并不一定适用于所有预测问题。具体实践中，更为合理的方式是将预测方法应用于测试数据集（真实响应已知，但假装未知并对之预测），比较预测值于真实值的差距即预测误差。交叉验证将数据分为两部分：

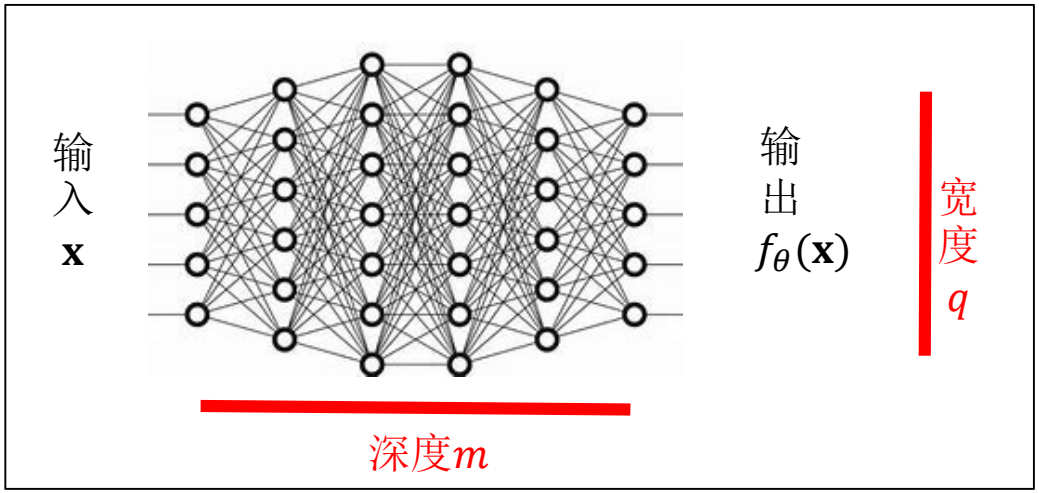
- 训练样本(training sample): 用来建立模型和预测方法
- 检验/测试样本(testing sample): 用来评价训练得到的预测。

K-fold CV

将数据集划分成$K$份，取$K-1$份作为训练集，另外一份作为测试集用于计算预测误差。轮流 $K$ 次。一般$K=5$或10。



# 深度神经网络 (Deep Neural Network, DNN)简介




深度神经网络DNN是深度学习的核心方法，主要用于预测和分类。DNN将输入（自变量、特征）进行若干非线性变换，变换后的特征与响应变量建立通常的回归模型。

若回归函数$E(y|x)$不是$x$的线性函数，一种简单策略是添加高阶项，即多项式回归：

$$y = a + bx + cx^2 + \cdots + dx^q + \varepsilon = a + \boldsymbol{\beta}^{\top}\mathbf{h} + \varepsilon$$

基本思路：将$x$映射到高维：$x \to \mathbf{h} = (x, \cdots, x^q)$,然后在$q$维空间中建立线性回归（注意：关于h线性，关于$x$非线性）。

类似地，神经网络DNN的核心也是估计非线性回归函数。
DNN将自变量（预测变量）向更高维空间映射，多次映射，得到的$\mathbf{h}_m$与响应变量$y$建立线性回归模型（但与原始自变量$x$之间是非线性关系）。

# 模型结构

**数据和问题**

自变量/特征: $\mathbf{x} \in R^p$ ， 响应/类别标号： $y \in R^1$

目标：估计非线性回归函数$f_\theta(\mathbf{x}) = E(y|\mathbf{x})$ ， 预测$y$

**深度神经网络DNN**

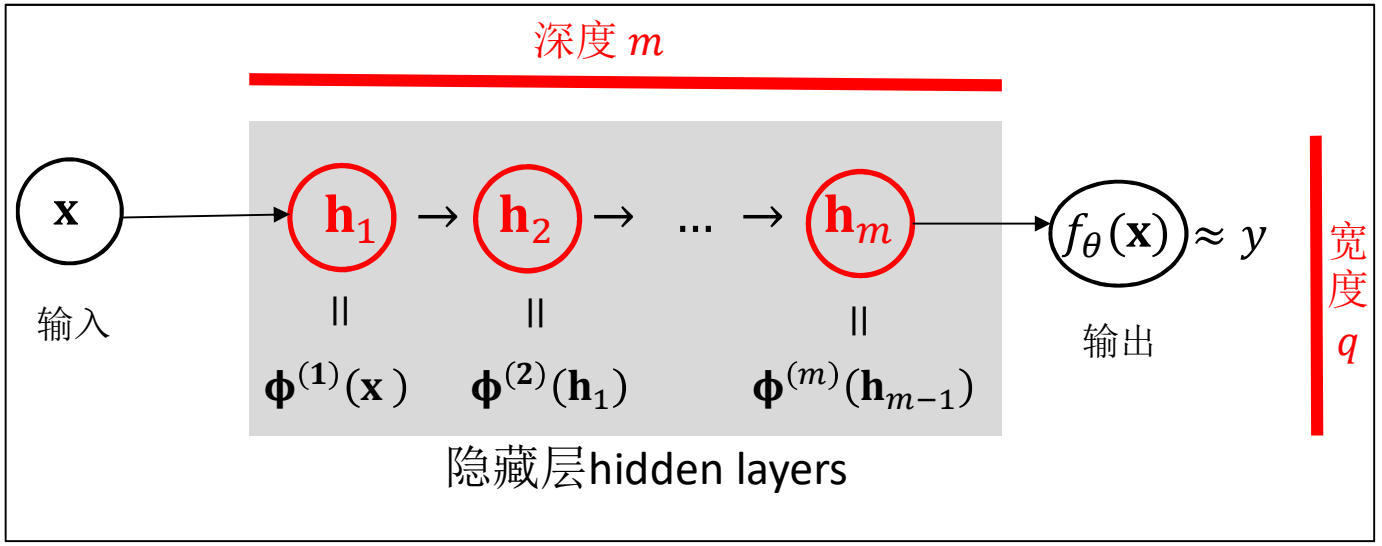
$DNN$将自变量(特征) $\mathbf{x}$进行多次复合变换，最终得到非线性变换

$$\mathbf{h}_m = \boldsymbol{\phi}^{(m)}\left(\boldsymbol{\phi}^{(m-1)}\left(\cdots \boldsymbol{\phi}^{(1)}(\mathbf{x})\right)\right)$$

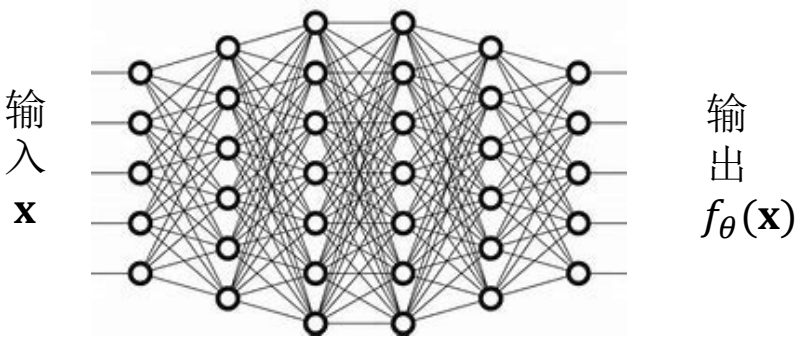
并建立线性回归$E(y|\mathbf{x}) = f_\theta(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{h}_m$ 或其它回归模型

参考：G.Strang (2019) Linear algebra and learning from data (Part VII). /books/LA4.pdf

## DNN图示




DNN通常图示如下，每个节点代表一个变量，称为神经元。



# DNN = 自变量复合变换 + 线性模型（或logistic模型等）

- **复合**：对输入$\mathbf{x}$进行递归非线性变换(隐藏层)：$\mathbf{h}_k = \boldsymbol{\phi}^{(k)}(\mathbf{h}_{k-1}), k = 1,2,\ldots,m,$

$$\mathbf{h}_m = \boldsymbol{\phi}^{(m)}\left(\boldsymbol{\phi}^{(m-1)}\left(\cdots \boldsymbol{\phi}^{(1)}(\mathbf{x})\right)\right)$$

  - 常用的非线性变换：仿射ReLU激活 $\mathbf{h}_k = (A_k\mathbf{h}_{k-1} + \mathbf{b}_k)_+$

ReLU变换：$\mathbf{h}_k = \boldsymbol{\phi}^{(k)}(\mathbf{h}_{k-1}) = (A_k\mathbf{h}_{k-1} + \mathbf{b}_k)_+$

$A_k$:矩阵，$\mathbf{b}_k$:偏置(bias)，$(\bullet)_+$: ReLU非线性激活

- **回归**：$\mathbf{h}_m$与响应变量满足回归模型：$E(y|\mathbf{x}) = \varphi(\mathbf{h}_m) \triangleq f_\theta(\mathbf{x})$，比如
  - 线性回归：$\varphi(\mathbf{h}_m) = \beta_0 + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{h}_m \approx y$
  - Logistic回归：$\varphi(\mathbf{h}_m) = 1/(1+\exp(-\beta_0 - \boldsymbol{\beta}^T\mathbf{h}_m))$

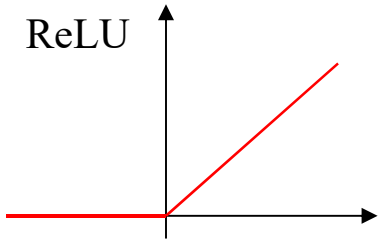
# 宽 度

ReLU函数

正部函数也被称为 ReLU (Rectified Linear Unit)激活函数：

$$\mathrm{ReLU}(u) = u_+ = \max(0,u) = u1_{(u>0)} = \begin{cases} u & u \geq 0 \\ 0 & u < 0 \end{cases}$$

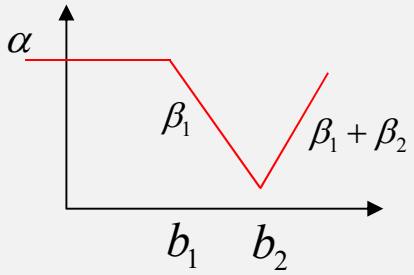
ReLU是神经网络中最基本和 最简单的非线性变换函 数。




例如：$f(x) = \alpha + \beta_1(x-b_1)_+ + \beta_2(x-b_2)_+$

$$= \begin{cases} \alpha & x < b_1 \\ \alpha + \beta_1(x-b_1) & b_1 \leq x < b_2 \\ \alpha + \beta_1(x-b_1) + \beta_2(x-b_2) & x \geq b_2 \end{cases}$$

这是连续的分段函数（三段）。

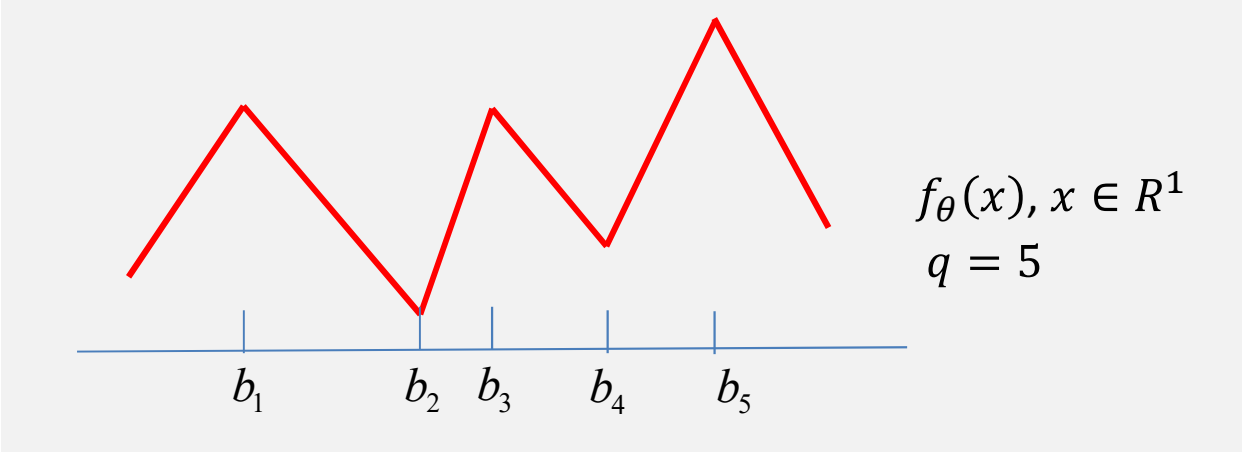


# 偏置：转折点、分割点

一元输入($p=1$)情形：$q$个仿射变换/偏置,ReLU函数激活，
相加生成多段连续线性函数(转折点为偏置$b_1,...,b_q$)

$$f_\theta(x)=\beta_0+\beta_1(x-b_1)_+ +....+\beta_q(x-b_q)_+,\ x\in R^1$$

通用逼近定理证明了只要$q$最够大，$f_\theta(x)$可逼近任何连续函数。

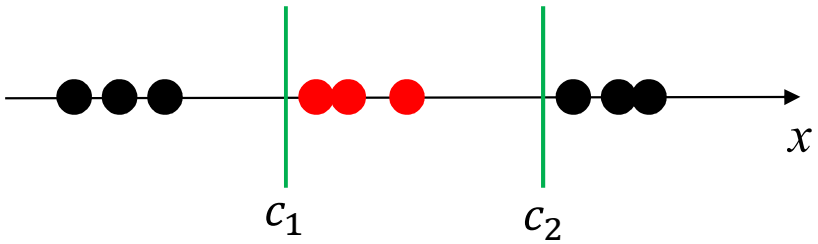


宽度$q$

宽度即神经元的个数，也即偏置的个数$q$。偏置代表分段的位置，个数由数据点的可分性决定，称为VC（ Vapnik-Chervonenkis）维数）

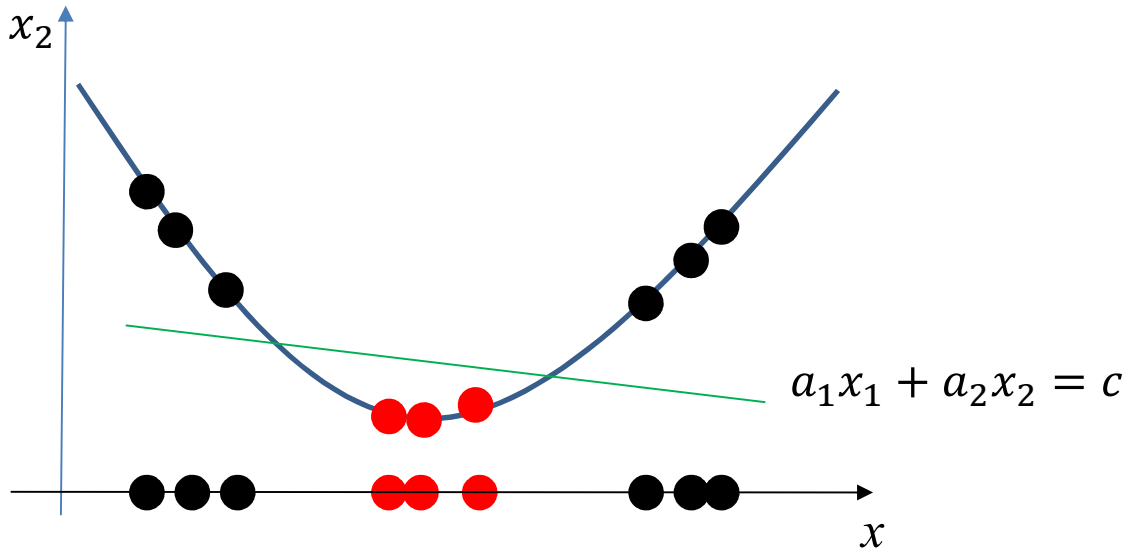
如果数轴上两类数据点（红、黑）可用一个线性函数 $x = b$分开，即由 $x < b$ 和 $x > b$ 判别两类，那么只需要1个神经元（一个偏置或一个阈值），此时是线性可分的。

如果一维数轴上两类数据点（红、黑）不能用一个分割点分开 $(x = c)$，称为是线性不可分的（下图），但下图可用2个分割点分开 $(x = c_1, x = c_2)$。



将 $x$ 映射到二维空间：

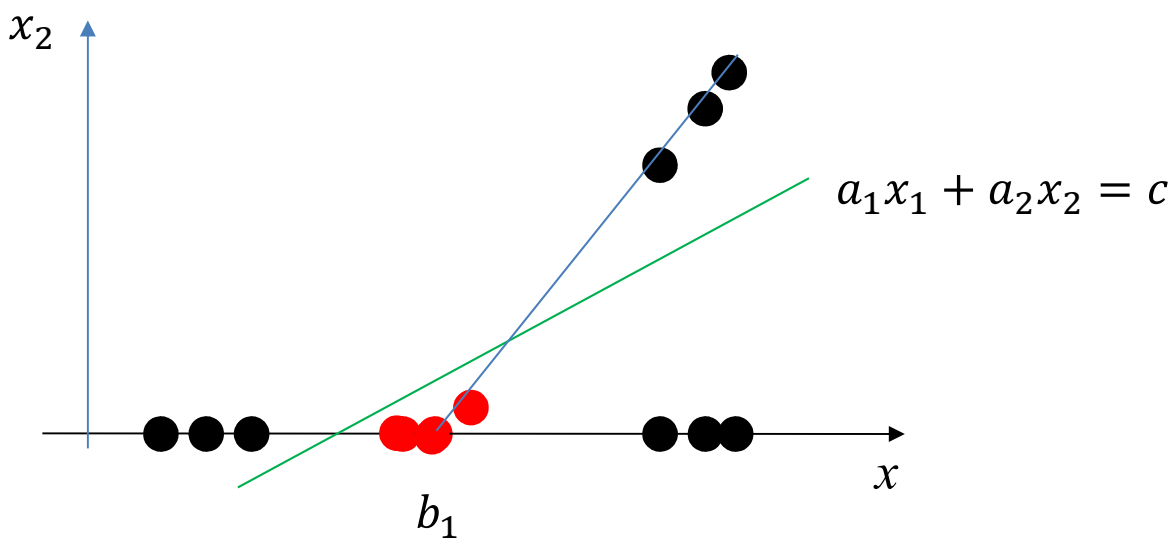
$$x \rightarrow (x_1, x_2) = (x, x^2)$$




在二维空间中直线$a_1x_1 + a_2x_2 = c$可将两类分开（线性可分）

ReLU可达到类似效果，将 $x$ 映射到二维空间：

$$x \to (x_1, x_2) = (x, (x - b_1)_+)$$




在二维空间中直线$a_1x_1 + a_2x_2 = c$可将两类分开（线性可分）

单层神经网络(一元输入)

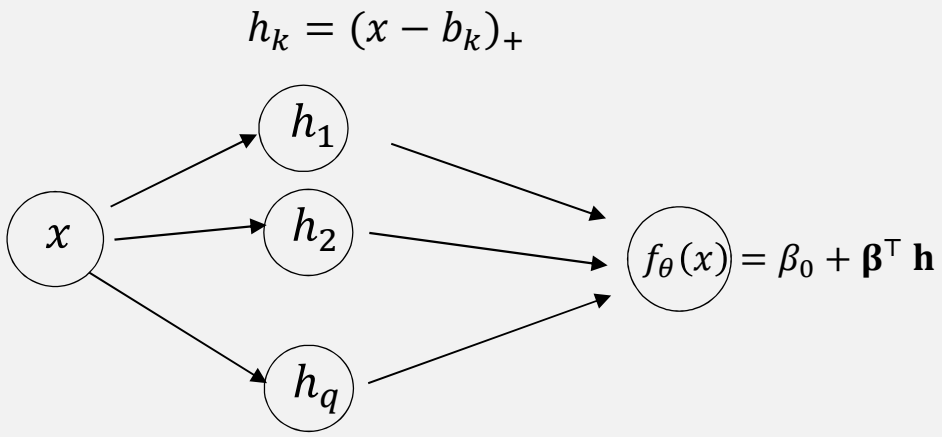
一般地，将一维数据映射到$q$维空间：

$$x \in R^1 \to \mathbf{h} = \left((x-b_1)_+, \ldots, (x-b_q)_+\right)^{\mathsf{T}} \in R^q,$$

假设$\mathbf{h}$的线性回归函数（单层神经网络）

$$f_\theta(x) = \beta_0 + \beta_1(x-b_1)_+ + \cdots + \beta_q(x-b_q)_+ = \beta_0 + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{h}$$

当偏置参数$b's$以及斜率参数$\beta's$足够多时，该函数可逼近任何连续函数。



单层神经网络($p$元输入)

数据:$(\mathbf{x}_i, y_i), i = 1,..., n.$ 原始$\mathbf{x} \in R^p$与$y$没有线性关系，但$\mathbf{x}$变换之后，$\mathbf{h} = (h_1,..., h_q)^{\mathsf{T}}$与$y$具有线性关系(单层神经网络).

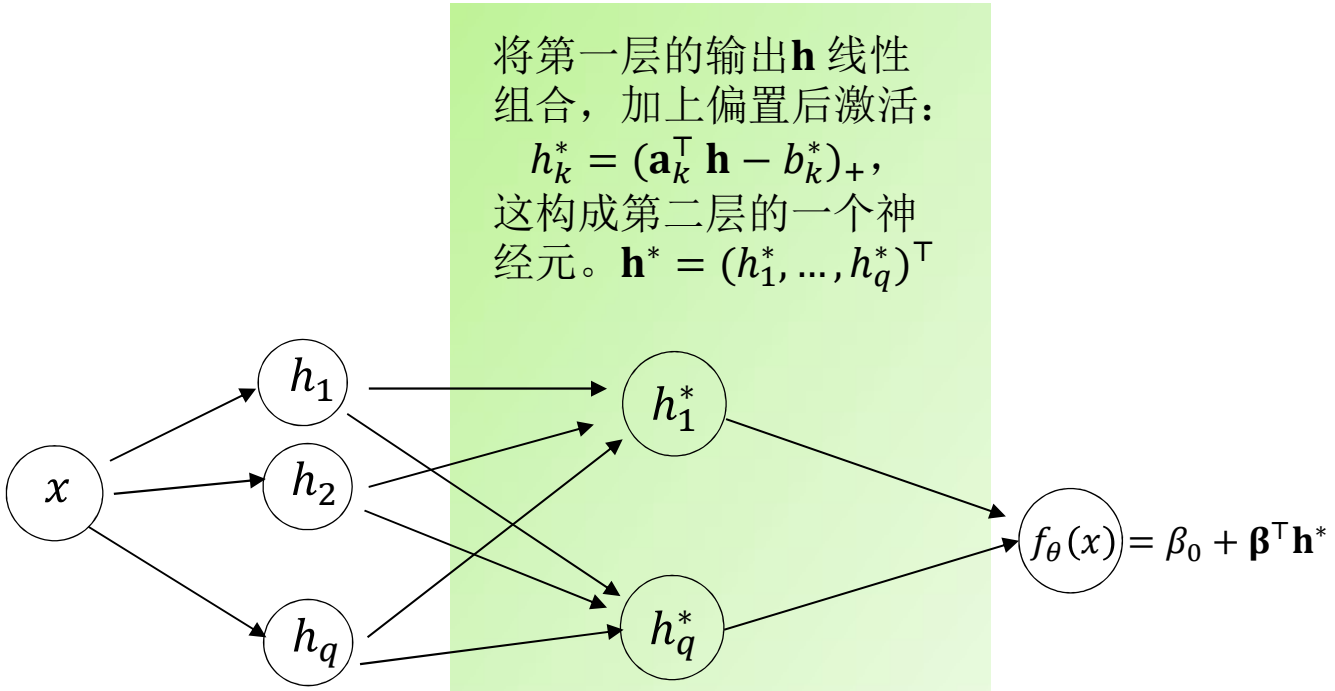
$$E(y \mid \mathbf{x}) = \beta_0 + \sum_{k=1}^{q} \beta_k h_k, \ \mathbf{x} \in R^p$$

其中变换$\phi_k$含有未知参数$\boldsymbol{\theta}_k, h_k = \phi_k(\mathbf{x}, \boldsymbol{\theta}_k), k = 1,..., q$构成隐藏层的神经元。

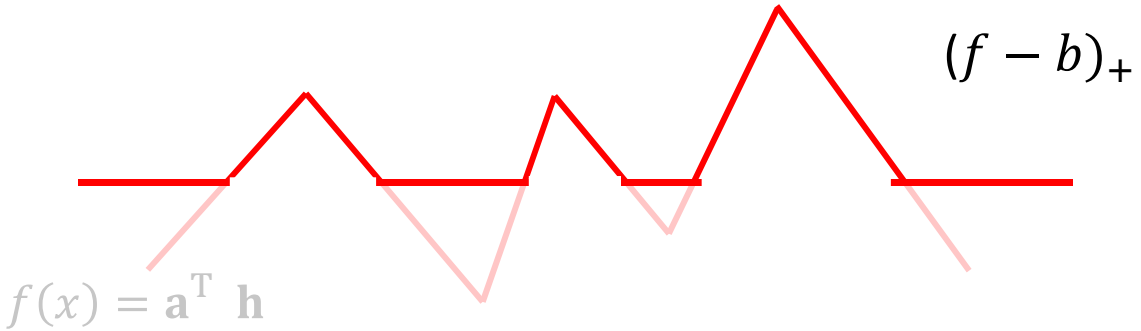
- 若$h_k = \phi_k(\mathbf{x}, \boldsymbol{\theta}_k) = \phi_k(\boldsymbol{\theta}_k^{\mathsf{T}} \mathbf{x})$，称为投影追踪(projection pursuit).
- 若$\phi_k$取为幂次函数$x^k$，则为多项式回归。
- Box - Cox变换对应于$\mathrm{q} = 1, \phi_1 = x^{\lambda}$
- ReLU激活的仿射变换：$h_k = \phi_k(\mathbf{x}, \boldsymbol{\theta}_k) = (A_k \mathbf{x} + \mathbf{b}_k)_+$

# 深度

两层神经网络：对第一层的输出再次应用非线性变换，特别地，仿射+ReLU激活（仍以一元输入为例）

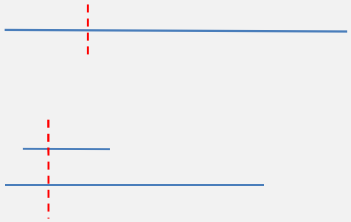


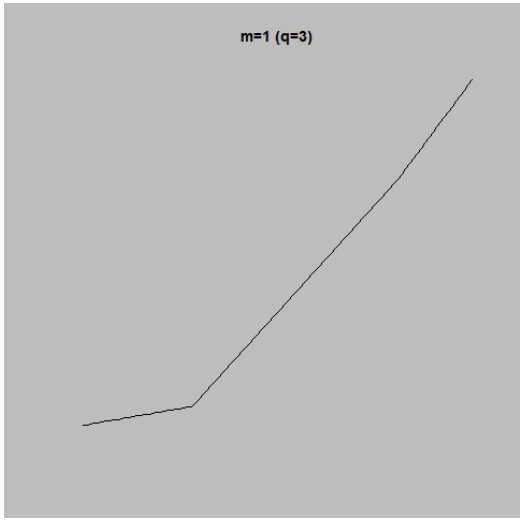
第一层输出$\mathbf{h}$的线性组合$f(x) = \mathbf{a}^{\mathsf{T}}\ \mathbf{h}$是一个$q$个转折点（分割点）的分段线性函数，对其再应用仿射+ ReLU激活，则得到~$q/2$个新的分割点（如下图）



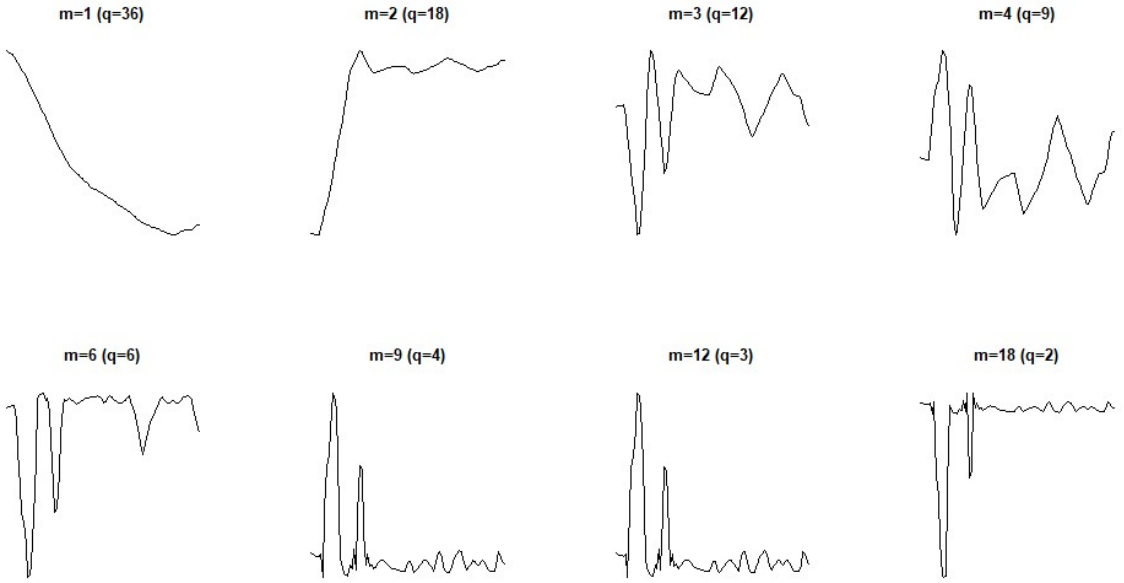

- 两层神经网络：对第一层的输出应用$q$次仿射+ReLU激活，大约得到~$\frac{q}{2} \times q$个分割点，即两层神经网络使用$2q$个偏置得到~$q^2$段的分段线性函数。
- $m$层DNN，每层$q$个神经元，$mq$个偏置可生成$O(q^m)$段的分段线性函数，相比之下，单层网络需要$O(q^m)$个偏置参数。所以多层网络可以以较少的偏置参数能得到足够多段的分段函数。

类比折叠剪绳：
一条绳子先剪$q$次（$q$个偏置），把$q+1$段折叠在一起（线性组合），再剪$q$次，共得到$\sim q^2$个线段 。剪$mq$次可产生$\sim q^m$段。





$mq = 36$（偏置个数固定），随着深度$m$增加分段线性函数倾向于越来越复杂



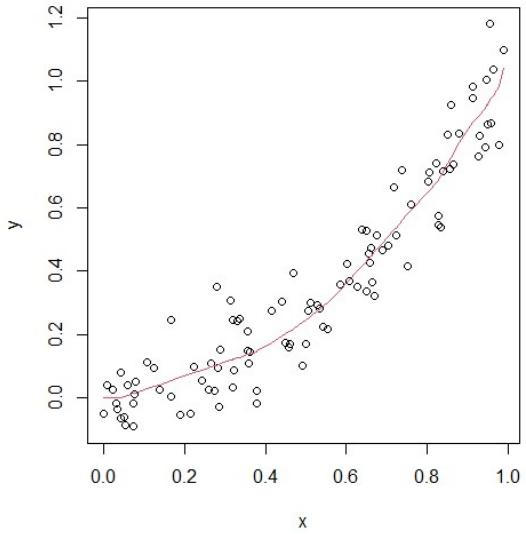
# 梯度下降法阻止过拟合 (收敛到局部最优)

梯度下降法

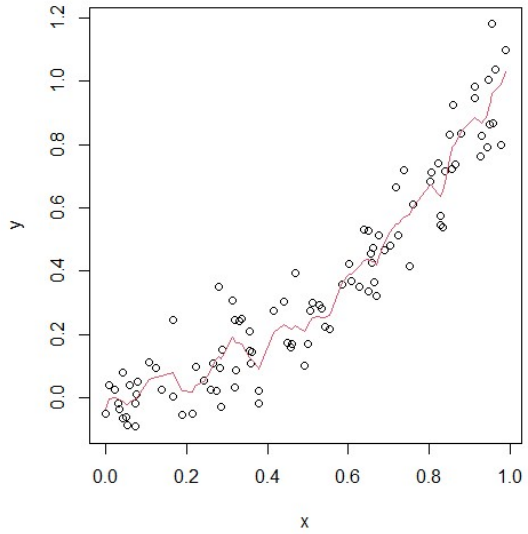
损失函数 $L(\boldsymbol{\theta}) = \sum(y_i - f(\mathbf{x}_i, \boldsymbol{\theta}))^2$

梯度下降解法：$\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{\text{old}} - \eta \nabla f(\boldsymbol{\theta}^{\text{old}})$，$\eta$：学习率

梯度计算：反向传播法（adjoint！）




局部最优
（较大的学习率或者early stopping）




接近全局最优
（较小的学习率或稳定收敛）

# 深度学习

深度学习(deep learning)方法是基于DNN的机器学习方法。深度学习之父G.Hinton （2006）应用反向传播法解决了多层深度网络模型的可计算性问题。此后，深度学习普及应用于多种场景的不同网络结构的学习任务。

**主要模型结构**

基本模型架构：深度神经网络DNN。
各种应用的不同在于输入不同，以及针对特殊场景的模型结构的调整。

**主要应用领域**

图像识别（CV: computer vision）
语言处理（NLP: natural language processing）
语音识别（ASR： automatic speech recognition）
推荐系统
自动翻译
最新成就：alphafold2预测蛋白质空间结构、alphacode

**深度学习框架**

Pytorch, Tensorflow, keras

参考书：G Strang (2019) Linear algebra and learning from data.

# 关于考试

考试时间：下学期1-2周（具体时间未定）
复习网站：http://home.ustc.edu.cn/~zhouwenbin/TA/RAreview.html（谢谢助教）
习题课：2022-1-6 (9:45-12:00) 腾讯
复习范围：

- 课件（除了第8-9-10、18讲，以及课件中虚线框内容和附录）
- 作业（考试题目类型将与作业类似，难度大致相当（除了hw4.2））。

| 课件 | 阅读材料 | 作业 | 上机 |
|---|---|---|---|
| 第一讲：回归分析简介 (9.2) | Freedman:第一章 | hw1 | |
| 第二讲：相关系数与偏相关系数 (9.9) | 《统计学》第1-2章 中文 | hw2 | |
| 第三讲：随机向量及去相关化 (9.16) | 王松桂：第二章 | hw3 | |
| 第四讲：随机向量之间的相关性 (9.23) | 王松桂：第二章 | | lab1 |
| 第五讲：线性回归模型 (9.30) | R in action | hw4 | |
| 第六讲：简单线性回归模型 (10.9) | | hw5 | |
| 第七讲：简单线性回归模型 (续10.14) | | | |
| 第八讲：简单线性模型的应用 (10.21) | Freedman:第2章 | hw6 | lab2 |
| 第九讲：线性代数 (10.28) | Axler 线性代数 | hw7: Axler第1,3章 | |
| 第十讲：线性代数 - 矩阵 (11.4) | 广义逆：《线性模型引论》2.2节 | hw8 | |
| 第十一讲：投影与最小二乘 (11.11) | | hw9 | |
| 第十二讲：最小二乘II (11.18) | | hw10 | |
| 第十三讲：约束最小二乘和F检验 (11.25) | | hw11 | lab3 |
| 第十四讲：广义最小二乘 (12.2) | Freedman: 4,5章 | | lab4: lab3的3-5 |
| 第十五讲：方差分析 (12.9) | | hw12 | |
| 第十六讲：回归诊断 (12.16) 腾讯 | Google Flu Trends | hw13 | lab5 |
| 第十七讲：预测 (12.23) 腾讯 | 特殊时期，hw13，lab5，hw14提交时间可以弹性 | hw14 | |
| 第十八讲：预测 II (12.30) 腾讯 | | | |