regression

eg. 自动驾驶

$f(\cdots)$ = 方向盘角度

eg. 推荐系统

$f$(使用者A, 商品B) = 购买可能性.

① 找一个 model　A set of function

$y = b + w \cdot x_{cp}$ → 进化前CP值

得到几个 function

Linear moder: $y = b + \sum w_i x_i$　$x_i$是不同的属性.

② step 2: 判断好坏

$x^1 \longrightarrow \hat{y}^1$　$x^2 \to \hat{y}^2$　hat 正确的值

收集数据 编号 (真正)

Loss function L　input: function　output: how bad it is

$L(f) = L(w,b) = \sum_{n=1}^{10} (\hat{y}^n - (b + w \cdot x_{cp}^n))^2$ 真值-预测的平方.

③ step 3: Best function

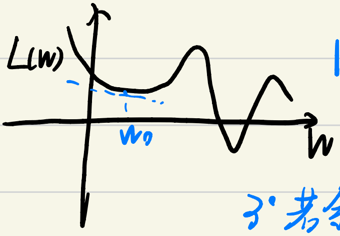
挑选出最好的func 让 $L(f)$ min

$f^* = \arg\min_f L(f)$

$w^*, b^* = \arg\min_{w,b} L(w,b)$

# Gradient Descent 坡度下降

eg · Loss func 只有一个参数 找一个w, 使L(w) min




1° 随机选取 $w_0$

2° $\frac{dL}{dw}\Big|_{w=w^0}$ (微分)

3° 若斜率为负 则增加w.
若为正, 则减少 w.

$w_1 \leftarrow w^0 - \eta \frac{dL}{dw}\Big|_{w=w_0}$ "$\eta$" learning rate 事先定义好

4° $\frac{dL}{dw}\Big|_{w=w^1}$ $w^2 \leftarrow w^1 - \eta \frac{dL}{dw}\Big|_{w=w_1}$ 最后到达斜率为0.

假如有两个参数呢?

$\frac{\partial L}{\partial w}\Big|_{w=w_0,b=b_0}$, $\frac{\partial L}{\partial b}\Big|_{w=w_0,b=b_0}$

$w^1 \leftarrow w_0 - \eta \frac{\partial L}{\partial w}\Big|_{w=w^0,b=b^0}$ $b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b}\Big|_{w=w_0,b=b_0}$ 反复计算.

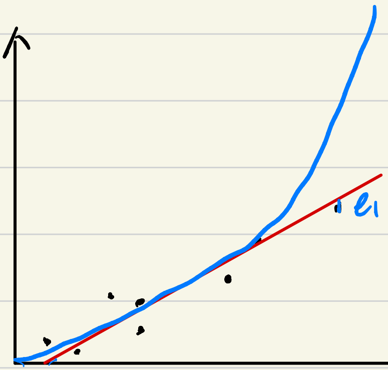
$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w} \\ \frac{\partial L}{\partial b} \end{bmatrix}_{gradient}$ 坡度.

linear 里 func 凸的

$$L(w,b) = \sum_{n=1}^{10} \left(\hat{y}^n - (b + w \cdot x_{cp}^n)\right)^2$$

$$\frac{dL}{dw} = \sum_{n=1}^{10} 2\left(\hat{y}^n - (b + w \cdot x_{cp}^n)\right)(-x_{cp}^n)$$ 偏微分.

$$\frac{\partial L}{\partial b} = 2\sum_{n=1}^{10} \left(\hat{y}^n - (b + w \cdot x_{cp}^n)\right) \cdot (-1)$$




$\sum_{n=1}^{10} e^n = 31.9$ 误差分析

关于 testing data 同样计算 $\sum_{n=1}^{10} e^n$

想要优化，重新设计 model

$y = b + w_1 \cdot x_{cp} + w_2 \cdot x_{cp}^2 \rightarrow$ Best Function $w_1 = \cdots,\ w_2 \cdots\ b = \cdots$

考虑更复杂的model 三次方、

Overfitting 过拟合 → 选择最合适的model

假如数据更多 只考虑一个维度是不够的

$x_s$ : 种类 代入不同的func

$y = \underbrace{b_1 \cdot \delta(x_s = \cdots)}_{} + w_1 \cdot \delta(x_s \cdots) \cdots$

若是则为1，其余全变为0.

Regularization 正则化.

$y = b + \sum w_i x_i$

$L = \sum_n \left(\hat{y}^n - (b + \sum w_i x_i)\right)^2 + \boxed{\lambda \sum (w_i)^2}$ 参数越小越好. 曲线平滑.

对输入不敏感 △

为什么不平滑、b的影响，好的结

λ的值表示考虑smooth的权重，

又不能过于平滑

为什么不考b

re：预期平滑，

b无关

Validation.

关于课后作业的一些思路

采用的是最简单的线性回归模型

$$y=\sum_{i=0}^{8} w_i x_i + b$$

$x_i$是对应数据帧中第i个的PM2.5含量，$w_i$为其对应的权重

b为偏置项，y为预测的结果

损失函数

用预测值与label之间的平均欧氏距离来衡量预测的准确程度

$$loss = \frac{1}{2n}\sum_{i=0}^{n-1}(\hat{y}^n - y^n)^2$$ + 加上正则化系数

# 梯度下降

$\theta^* = \arg\min_{\theta} L(\theta)$

随机选取起始的点 $\theta^0 = \begin{bmatrix}\theta_1^0 \\ \theta_2^0\end{bmatrix}$ 用下标代表

计算偏微分，减去 得新的参数

$$\begin{bmatrix}\theta_1^1 \\ \theta_2^1\end{bmatrix} = \begin{bmatrix}\theta_1^0 \\ \theta_2^0\end{bmatrix} - \eta \begin{bmatrix}\partial L(\theta_1^0)/\partial \theta_1^0 \\ \partial L(\theta_2^0)/\partial \theta_2^0\end{bmatrix}$$ update

反复计算 $\theta^{n+1} = \theta^n - \eta \nabla L(\theta^n)$

learning rate $\eta$ （怎么去调）

通常随着参数的update越来越小

设为 $\eta(t) = \eta/\sqrt{t+1}$ 次数越多越小 因材施教

Adagrad

$$w^{t+1} \leftarrow w^t - \eta^t g^t \Rightarrow w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t$$

$\sigma^t$ 过去所有微分值 root mean square

eg. $w^1 \leftarrow w_0 - \frac{\eta^0}{\sigma^0} g^0$ $\quad \sigma^0 = \sqrt{(g^0)^2}$

$w^2 \leftarrow w_1 - \frac{\eta^1}{\sigma^1} g^1$ $\quad \sigma^1 = \sqrt{\frac{1}{2}[(g^0)^2 + (g^1)^2]}$

$w^3 \leftarrow w_2 - \frac{\eta^2}{\sigma^2} g^2$ $\quad \sigma^2 = \sqrt{\frac{1}{3}[(g^0)^2 + (g^1)^2 + (g^2)^2]}$

... 反复进行

$$\sigma^t = \sqrt{\frac{1}{t+1} \sum_{i=0}^{t} (g^i)^2} \qquad w^{t+1} \leftarrow w^t - \frac{\eta}{\sqrt{\sum_{i=0}^{t} (g^i)^2}} g^t$$

多维

$$y = b + w_1 x_1 + w_2 x_2$$