1 - Linear Regression.

$$\hat{y} = w_1 x_1 + w_2 x_2 + w_d x_d + b$$

$\vec{w}$: weights

$\vec{x}$: features

$\hat{y}$: label predicted.

b : bias (offset)

$\Longrightarrow$ vector form : $\boxed{\hat{y} = \vec{w}^T\vec{x} + b}$

Now we have n examples.

也就是: $\underbrace{n \times d}$ 的维度

$$\begin{array}{c} \\ n \end{array}\begin{bmatrix} \text{---}\ \vec{x}^{(1)}\ \text{---} \\ \text{---}\ \vec{x}^{(2)}\ \text{---} \\ \vdots \\ \text{---}\ \vec{x}^{(n)}\ \text{---} \end{bmatrix}$$
$$\underbrace{\qquad\qquad}_{d}$$

So $\underbrace{X}_{n\times d}\ \underbrace{w}_{d} = \begin{bmatrix} \vec{x}^{(1)}\cdot\vec{w} \\ \vdots \\ \vec{x}^{(n)}\cdot\vec{w} \end{bmatrix}$

1.1 Matrix form

$\Longrightarrow$ 预测结果:

$$\boxed{\vec{\hat{y}} = X\cdot\vec{w} + b}$$

$$\left(\begin{bmatrix} \hat{y}^{(1)} \\ \vdots \\ \hat{y}^{(n)} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ \vdots \\ x_1^{(n)} & x_2^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}\begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} + \begin{bmatrix} b \\ b \\ \vdots \\ b \end{bmatrix}\right)$$

我们用一个 loss function 来 measure 这 n 个 examples (data points) 的 performance:

用 squared error 作为 regression loss (第(i)个)

$$\boxed{l^{(i)}(w,b) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2}$$

1.2 Matrix form of Loss function

$\Longrightarrow$ 整体n个 examples 的performance

$$L(w,b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(w,b)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(\boxed{w^T\vec{x}^{(i)} + b} - y^{(i)})^2$$

注意到这里的 bias b 可以被 absorb 进 weights, 如果给每个 $\vec{x}^{(i)}$ 加上一个恒为 1 的 dummy feature

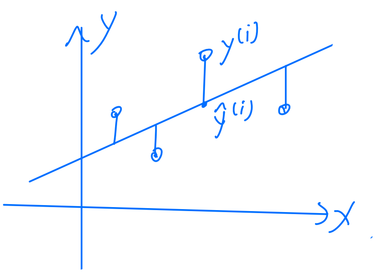
即: $w^T\vec{x}^{(i)} + b = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix}\cdot\begin{bmatrix} x_1 & x_2 & \cdots & x_d & 1 \end{bmatrix}$

于是把这样的每个 $\vec{x}^{(i)} = [x_1\ x_2 \dots x_d\ \ 1]$ 的vector

组成 matrix $X = \begin{bmatrix} \text{---}\ \vec{x}^{(1)}\ \text{---} \\ \text{---}\ \vec{x}^{(2)}\ \text{---} \\ \text{---}\ \vec{x}^{(3)}\ \text{---} \\ \vdots \\ \text{---}\ \vec{x}^{(n)}\ \text{---} \end{bmatrix}$

我们可以得到 $L(w,b)$ 的 matrix form:

$$\boxed{L(\underset{(,b)}{\vec{w}}) = \frac{1}{2}\|\vec{y} - X\vec{w}\|^2}$$



我们希望的参数 w 和 b 是:

Denote ideal (w,b) 为 (w^*, b^*)

$\Longrightarrow$ $\boxed{(\vec{w}^*, b^*) = \underset{w,b}{\arg\min}\ L(\vec{w}, b)}$

如何 minimize the loss 来获得 (w^*, b^*)

1.3 Analytic Solution

虽然没用,但推导值得一看。

即 $\nabla_w L(\vec{w}, \vec{b}) = \vec{0}$,

前已阐述: b 可被 absorb 进 $\vec{w}$, 使 $L(\vec{w}, b)$ 简化为 $L(\vec{w})$ (这里两个 $\vec{w}$ 不一样, 第二个多个 b, 实际就是把 b 放进 $\vec{w}$ 里)

因而我们要求的就是 $L(\vec{w}) = \frac{1}{2}\|y - X\vec{w}\|$ 对于 $\vec{w}$ 的 derivative 为 $\vec{0}$ 的时候的 $\vec{w}$.

$\Longrightarrow$ 即 $\nabla_w L(\vec{w}) = \vec{0}$, 此时的 $\vec{w}$ 为 $\vec{w}^*$

可解出 $\Longrightarrow$ $\boxed{w^* = (X^TX)^{-1}X^Ty}$

如何解得这个解：以下为一个简略的推导.

Let: $G(\vec{w}) = y - X\vec{w}$ $\left(\Rightarrow \nabla_w G(\vec{w}) = -X^T\right)$

$F(\vec{u}) = \frac{1}{2}\|\vec{u}\|$ $\left(\Rightarrow \nabla_u F(\vec{u}) = \vec{u}\right)$

$$\Rightarrow L(\vec{w}) = F(G(\vec{w}))$$

$$\Rightarrow \nabla_w L(\vec{w}) = \nabla_w G(\vec{w}) \cdot \nabla_u F(\overset{\vec{u}}{\uparrow}{G(\vec{w})}) \quad \underline{= \vec{u} = G(\vec{w})}$$

$$= (-X^T) \times (\vec{y} - X\vec{w})$$

$$\boxed{= -X^T\vec{y} + X^T X\vec{w}}$$

而后 set $\nabla_w L(\vec{w}) = \vec{0}$ 得 $\boxed{w^* = (X^T X)^{-1} X^T y}$

而有这样的 analytic solution 的条件是 $(X^T X)$ invertible.

(which is nearly impossible)

*(由于没学过 multi calc，再详细推一遍)

附

$$H(\vec{w}) = \vec{w}^T \cdot \vec{x} \quad \left(= \sum_{i=1}^{d} w_i x_i\right)$$

$$\left(= \underbrace{w_1 x_1 + w_2 x_2 + \dots + w_d x_d}_{\text{doesn't depend on } w_1 !!}\right)$$

$$\Rightarrow \nabla_w H(w) = \begin{bmatrix} \frac{\partial H}{\partial w_1} \leftarrow = x_1 \\ \frac{\partial H}{\partial w_2} \leftarrow = x_2 \\ \vdots \\ \frac{\partial H}{\partial w_d} \leftarrow = x_d \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = x^T$$

$\Rightarrow$ So $\nabla_w G(\vec{w}) = \nabla_w (y - X\vec{w}) = -X^T$ (转X)

$$F(\vec{u}) = \frac{1}{2}\|\vec{u}\|^2 = \frac{1}{2}\sum_{i=1}^{d} w_i^2$$

$$= \frac{1}{2}w_1^2 + \underbrace{\frac{1}{2}w_2^2 + \dots + \frac{1}{2}w_d^2}_{\text{does not depend on } w_1}$$

$$\Rightarrow \frac{\partial F}{\partial w_i} = w_i \Rightarrow \nabla F(w) = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} = \vec{w}$$

一些其他重要的 gradient：

$$H(\vec{w}) = \frac{1}{2}\underset{1\times d}{\vec{w}^T}\underset{d\times d}{A}\underset{d\times 1}{\vec{w}}$$

经过一些 algebra 可得

$$\nabla_w H(\vec{w}) = \frac{1}{2}(A + A^T)\vec{w}.$$

(In particular, when A is symmetric, $\nabla_w H(\vec{w}) = A\vec{w}$.)

现在回归正题：

1.5 Gradient Descent

既然 closed form solution 很少见（并且如果有就和 Deep Learning 没关系了）

那么我们需要一个能用数值方法来优化 loss function 的方法.

$\Rightarrow$ 以 negative gradient 的方法 update the parameters, 从而 iteratively reduce loss.

For any objective function $J(\vec{w})$,

Gradient descent's form:

$$\boxed{w \leftarrow w - \eta \nabla J(w)}$$

Gradient 给出了 J 的 fastest local increase 的方向.

从而快速 minimize J.

这里的 η controls how fast we move, 称为 learning rate.

1.6 如何计算 $\nabla_w L(\vec{w})$

现在我们已推出

$$\nabla_w L(\vec{w}) = \boxed{= -X^T\vec{y} + X^T X\vec{w}}$$

我们可拿掉 matrix 令它更易理解和计算.

在此处（linear regression 下的 least square sum 为 loss func）

$$\nabla_w L(w) = \frac{1}{n}\sum_i \frac{1}{2}\nabla(\vec{w}^T\vec{x}^{(i)} - \vec{y}^{(i)})^2$$

$$= \frac{1}{n}\sum_i \vec{x}^{(i)}(\vec{w}^T\vec{x}^{(i)} - \vec{y}^{(i)})$$

因而 GD update 为 $\boxed{w \leftarrow w - \frac{\eta}{n}\sum_i x^{(i)}(w^T x^{(i)} - y^{(i)})}$

可用matrix推导：

$$X = \begin{bmatrix} — \vec{x}^{(1)} — \\ — \vec{x}^{(2)} — \\ \vdots \\ — \vec{x}^{(n)} — \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \vec{x}_1 & \vec{x}_2 & \cdots & \vec{x}_d \\ | & | & & | \end{bmatrix}$$

(n examples) (d features)

$$X^T = \begin{bmatrix} — \vec{x}_1^T — \\ — \vec{x}_2^T — \\ \vdots \\ — \vec{x}_d^T — \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \vec{x}^{(1)T} & \vec{x}^{(2)T} & \cdots & \vec{x}^{(n)T} \\ | & | & & | \end{bmatrix}$$

这里：$\underset{d\times n \ n\times d}{X^TX}: d\times d$ $\underset{n\times d \ d\times n}{XX^T}: n\times n$

$(X^TX)^T = X^T(X^T)^T = X^TX$ (itself)

$(XX^T)^T = XX^T$ (itself)

我们可以发现 XX^T 和 X^TX 都是symmetric的.

$$X^TX = \begin{bmatrix} — \vec{x}_1^T — \\ — \vec{x}_2^T — \\ \vdots \\ — \vec{x}_d^T — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \vec{x}_1 & \vec{x}_2 & \cdots & \vec{x}_d \\ | & | & & | \end{bmatrix}$$

(symmetric)

$$= \begin{bmatrix} \vec{x}_1^T\cdot\vec{x}_1 & \vec{x}_1^T\cdot\vec{x}_2 & \cdots & \vec{x}_1^T\cdot\vec{x}_d \\ \vec{x}_2^T\cdot\vec{x}_1 & \vec{x}_2^T\cdot\vec{x}_2 & \cdots & \vec{x}_2^T\cdot\vec{x}_d \\ \vdots & & & \vdots \\ \vec{x}_d^T\cdot\vec{x}_1 & \vec{x}_d^T\cdot\vec{x}_2 & \cdots & \vec{x}_d^T\cdot\vec{x}_d \end{bmatrix}$$

$$XX^T = \begin{bmatrix} — \vec{x}^{(1)} — \\ — \vec{x}^{(2)} — \\ \vdots \\ — \vec{x}^{(n)} — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \vec{x}^{(1)T} & \vec{x}^{(2)T} & \cdots & \vec{x}^{(n)T} \\ | & | & & | \end{bmatrix}$$

$$= \begin{bmatrix} \vec{x}^{(1)}\cdot\vec{x}^{(1)T} & \vec{x}^{(1)}\cdot\vec{x}^{(2)T} & \cdots & \vec{x}^{(1)}\cdot\vec{x}^{(n)T} \\ \vec{x}^{(2)}\cdot\vec{x}^{(1)T} & \vec{x}^{(2)}\cdot\vec{x}^{(2)T} & \cdots & \vec{x}^{(2)}\cdot\vec{x}^{(n)T} \\ \vdots & \vdots & & \vdots \\ \vec{x}^{(n)}\cdot\vec{x}^{(1)T} & \vec{x}^{(n)}\cdot\vec{x}^{(2)T} & \cdots & \vec{x}^{(n)}\cdot\vec{x}^{(n)T} \end{bmatrix}$$

因而 $\nabla_w L(\vec{w}) = \boxed{= -X^T\vec{y} + X^TX\vec{w}}$

$$= \begin{bmatrix} \vec{x}_1^T\cdot\vec{x}_1 & \vec{x}_1^T\cdot\vec{x}_2 & \cdots & \vec{x}_1^T\cdot\vec{x}_d \\ \vec{x}_2^T\cdot\vec{x}_1 & \vec{x}_2^T\cdot\vec{x}_2 & \cdots & \vec{x}_2^T\cdot\vec{x}_d \\ \vdots & & & \vdots \\ \vec{x}_d^T\cdot\vec{x}_1 & \vec{x}_d^T\cdot\vec{x}_2 & \cdots & \vec{x}_d^T\cdot\vec{x}_d \end{bmatrix}\cdot\vec{w} - \begin{bmatrix} — \vec{x}_1^T — \\ — \vec{x}_2^T — \\ \vdots \\ — \vec{x}_d^T — \end{bmatrix}\vec{y}$$

$\boxed{= \vec{x}^{(i)}(\vec{w}^T\vec{x}^{(i)} - \vec{y}^{(i)})}$

总结，

$\nabla_w L(w) = \frac{1}{n}\sum_i \vec{x}^{(i)}(\vec{w}^T\vec{x}^{(i)} - \vec{y}^{(i)})$

因而 GD update 为 $\boxed{w \leftarrow w - \frac{\eta}{n}\sum_i x^{(i)}(w^T x^{(i)} - y^{(i)})}$

1.6 Minibatch Stochastic Gradient Descent.

(小批量随机梯度下降)

每次iteration，随机选取一个minibatch B, consisting of a number of training data points.

那么 $\boxed{\vec{w} \leftarrow \vec{w} - \frac{\eta}{|B|}\sum_{i\in B}\vec{x}^{(i)}(\vec{w}^T\vec{x}^{(i)} - \vec{y}^{(i)})}$

此处的 η 和 $|B|$ 被称为 hyperparameters (超参数).

它们在同一次training中为fixed的，
但也可由一个outer loop来optimize
by tracking performance on a validation set.

(下面就不推了)

可得到从上一次的 $\vec{w}$ (用 $\vec{w}^{(t)}$ 来表示)
更新到这一次的 $\vec{w}$

Loss function的值的iteration为：

$$\tilde{L}(w; w^{(t)}) = L(w^{(t)}) + \nabla L(w^{(t)})(w - w^{(t)}) + \frac{\eta}{2}\|w - w^{(t)}\|^2$$

1.7 用 learned model 进行 prediction

我们通过了用 least square sum 的 loss function $L(\vec{w})$

对 linear regression 的 weights $(\vec{w}, b)$

用 gradient descent 求出了使 Loss function $L(\vec{w})$

最小的 weight 记作 $\hat{\vec{w}}, \hat{b}$ 后.

$\boxed{\hat{y} = \hat{\vec{w}}^T \cdot \vec{x} + \hat{b}}$ 就是我们现在的 learned model.

我们可以把由一组 features $x_1, x_2, \ldots, x_d$

组成的一个 testing data point:

$\vec{x}^{(i)} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$ 代入, 来预测这个

data point $\vec{x}^{(i)}$ 的 estimated target $\hat{y}^{(i)}$.

这个过程在 deep learning jargon 中叫做 inference.