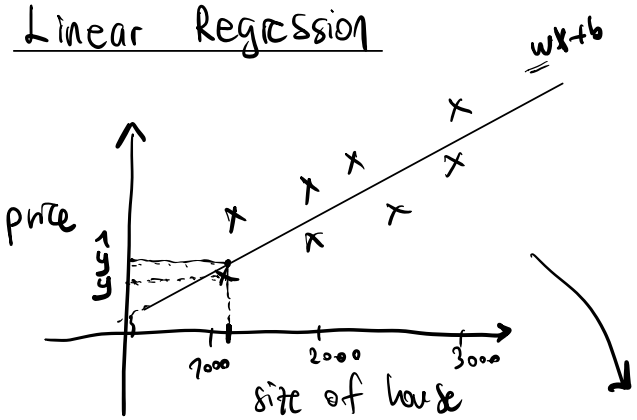
Linear Regression



input var / feature.
output var / target.

$x = (x, y)$

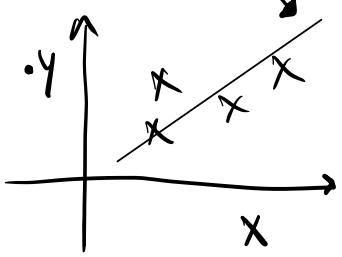
i^{th} traing ex: $(x^{(i)}, y^{(i)})$

size of house	price
2010	400
⋮	⋮

$f_{w,b} = wx + b$

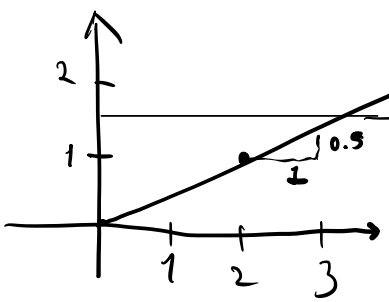
model

parameters (coefficients, weights)



linear regression with one variable.

univarate linear regression.
one variable.



$w = 0$, $b = 1.5$ $\quad f(x) = 0.x + 1.5$

$w = 0.5$, $b = 0$ $\quad f(x) = 0.5x$

slope.

find w, b:

$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

prediction

ground truth

MLE

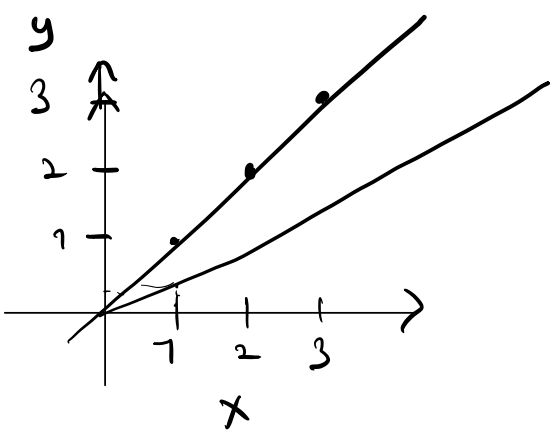
Cost function: Squared error cost function.

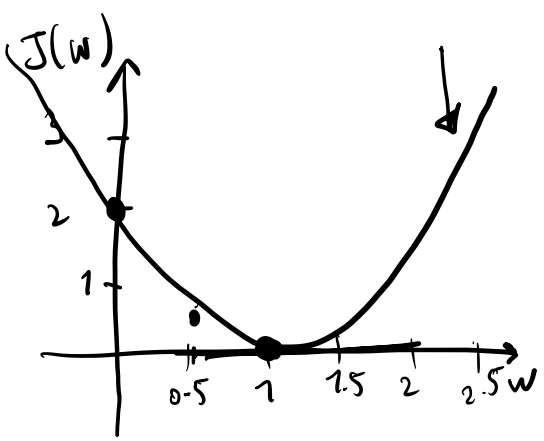
$$J(\underline{w}, \underline{b}) = \frac{1}{2m} \sum_{i=1}^{m} \left(\underline{\hat{y}}^{(i)} - \underline{y}^{(i)} \right)^2 \quad \rightarrow \text{\# of trang ex.}$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Goal: minimize $J(w,b)$ $\rightarrow$ $\min_{w} J(w)$

$f_w(x) = wx$ $\quad b = 0$ $\quad 0.5x$





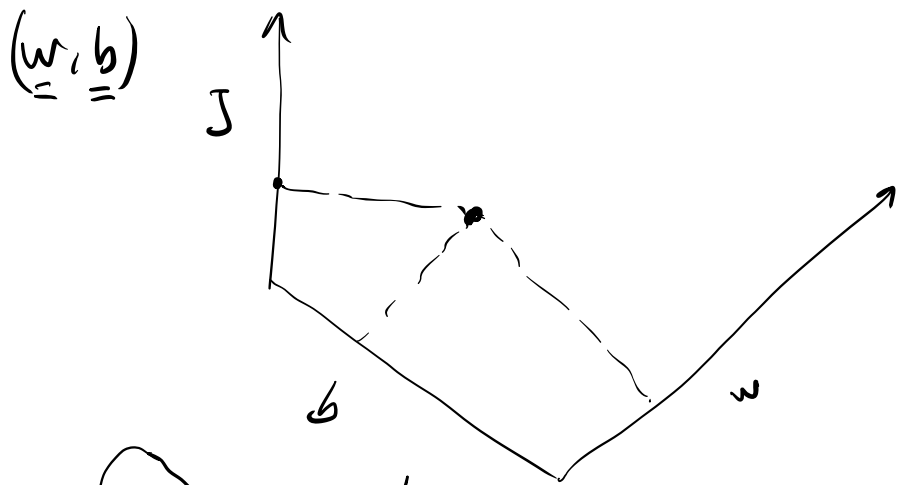
$w = 0.5$

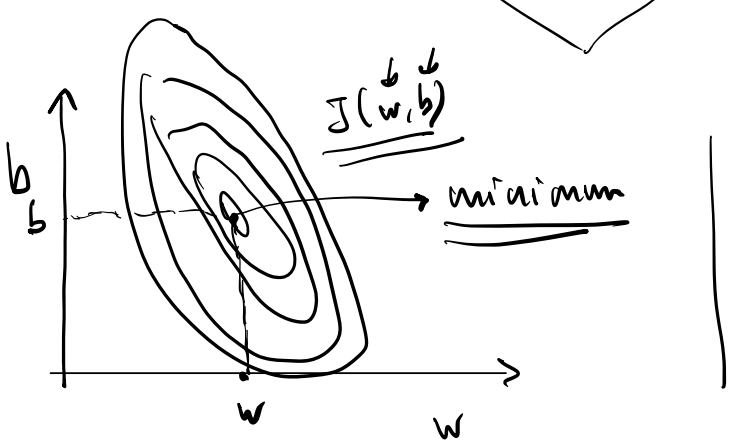
$$J = \frac{1}{2m} \left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right] \approx 0.58$$

$w = 0 \Rightarrow J \approx 2.3$ $\quad w = 1$

→ how to choose w?

⟶ choose w to minimize $J(w)$





$$\min_{w_1, w_2 \ldots w_n, b} J(w_1, w_2 \ldots w_n, b)$$

Gradient Descent Algorithm.

Outline:

Start with some w, b (ex: $w = 0.01$, $b = 0.01$) (0-1)

keep changing w, b to reduce $J(w,b)$

until we settle at or near a minimum.

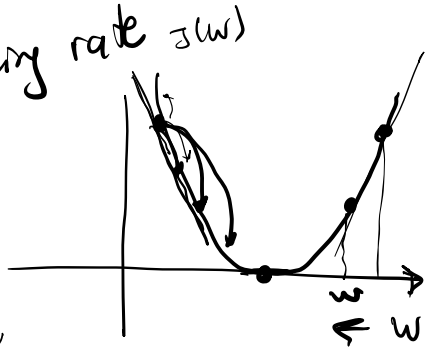
Algorithm.

repeat until convergence {

$$w = w - \alpha \frac{d}{dw} J(w,b) \quad (1)$$

$$b = b - \alpha \frac{d}{db} J(w,b) \quad (2)$$

} simultaneously update w, b



learning rate

$\alpha > 0$

$$\text{temp\_w} = w - \alpha_1 \frac{d}{dw} J(w,b)$$

$$\text{temp\_b} = b - \alpha_2 \frac{d}{db} J(w,b)$$

$w = \text{temp\_w}$

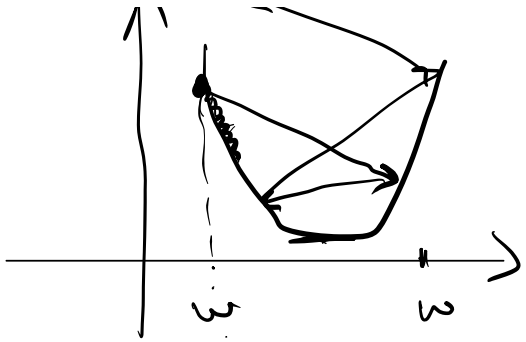
$b = \text{temp\_b}$

fix size $\alpha = 0.1 \rightarrow 1000$

$\wedge \quad \ll$

if α is too small

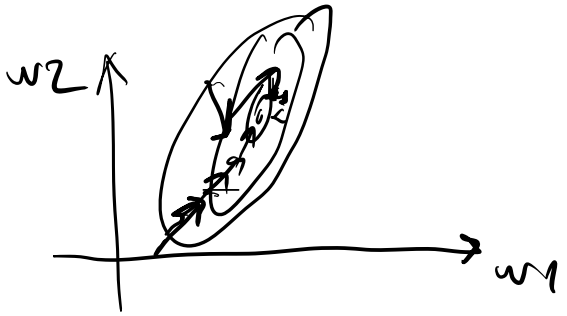
$\alpha = 0.000001$



if α is too large

$\alpha = 100$ / overshoot, never reach minimum

Adam optimization.



$$w_1 = w_1 - \alpha_1 \frac{d}{dw_1} J(\vec{w}, b)$$

$$w_{10} = w_2 = \alpha_{10}$$

if w_j keeps moving in same direction, increase α_j.

if w_j keeps oscillating, reduce α_j.

$$\frac{d}{dw} J(w,b) = \frac{d}{dw} \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

$f_{w,b}(x^{(i)}) = wx^{(i)} + b$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)} \right) 2x^{(i)}$$

$$\frac{d}{dw} J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

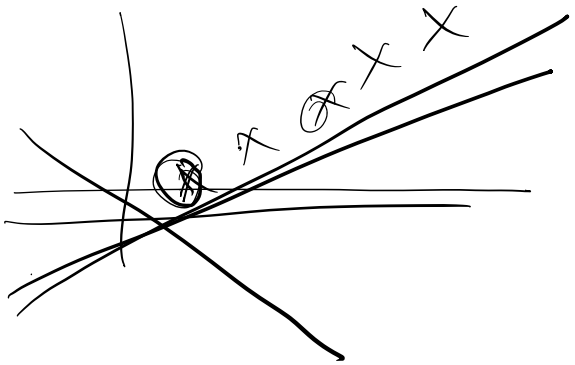
$$\frac{d}{db} J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)$$

Batch gradient

↳ each step of gradient descent.
we use all the traing ex.

→ Stochastic Gradat Descent (X)

↳ 1 sample



→ Mini-match.

1 calculation

1M data → m batch gradent → 1 iteration

→ Stechastic → 1M in 1 iteration

→ (mini-batch (700)) → 1000 in 1 iteration

Multiple features

Site in feet	# of bedrooms	# of flors	Agein lone	price.

(x1) | (x2) | (x3) | (x4) |

$f_{w,b}(x) = wx + b$.

$f_{w,b} = w_1x_1 + w_2x_2 \dots \dots + b$.

$\vec{w} = [w_1 \dots w_n]$ $\vec{x} = [x_1 \dots x_n]$

b = is a number.

$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$ } multiple linear regression.

↙ dot product.

Vectorization

$f_{\vec{w},b}(\vec{x}) = \sum_{j=1}^{n} w_j x_j + b$.

$\vec{w} = [w_1\ w_2\ w_3]$

b is a number

$\vec{x} = [x_1\ x_2\ x_3]$

```
w = [1, 2, 3], x = [2, 3, 4]
b = 10
total = 0
for j in range(0, n):
```

sequential.

```
total = total + w[j] * x[j]
total = total + b
```

Vectorization.

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b.$$

```
f = np.dot(w, x) + b
```

→ Shorter
→ much faster
→ use parallel comp. CPU GPU.

6.1 for Multiple LR

repeat {

$$w_1 = w_1 - \alpha \frac{\partial}{\partial w_1} J(w,b)$$

$$\vdots$$

$$w_{10} = w_{10} - \alpha \frac{\partial}{\partial w_{10}} J(w,b)$$

} Simultaneously update

Normal Equation.

if you < 100

solve w, b without iterations.

$$w = (X^T X)^{-1} (X^T \cdot y)$$

Feature Scaling.

max. (1)

$x_1 \rightarrow \frac{300}{2000} \leq x_1 \leq \frac{2000}{2000}$

$x_{1,\text{scaled}} = \frac{x_1}{2000}$

(2) mean normalization. $= \left[\frac{\overbrace{\quad\quad}^{100}}{M} \right]$

$x_1 = \frac{x_1 - M}{\max - \min}$ $\quad 300$

$\frac{2000 - 300}{}$

$300 \leq x_1 \leq 2000$

$= 0.18 \leq x_1 \leq 0.8$

(3) Z-score

$$x_1 = \frac{x_1 - \mu_1}{\sigma_1} \quad -0.67 \leq x_1 \leq 3.1$$

Ex/ $\hat{price} = w_1 x_1 + w_2 x_2 + b$

x_1: size, x_2: # bedrooms.

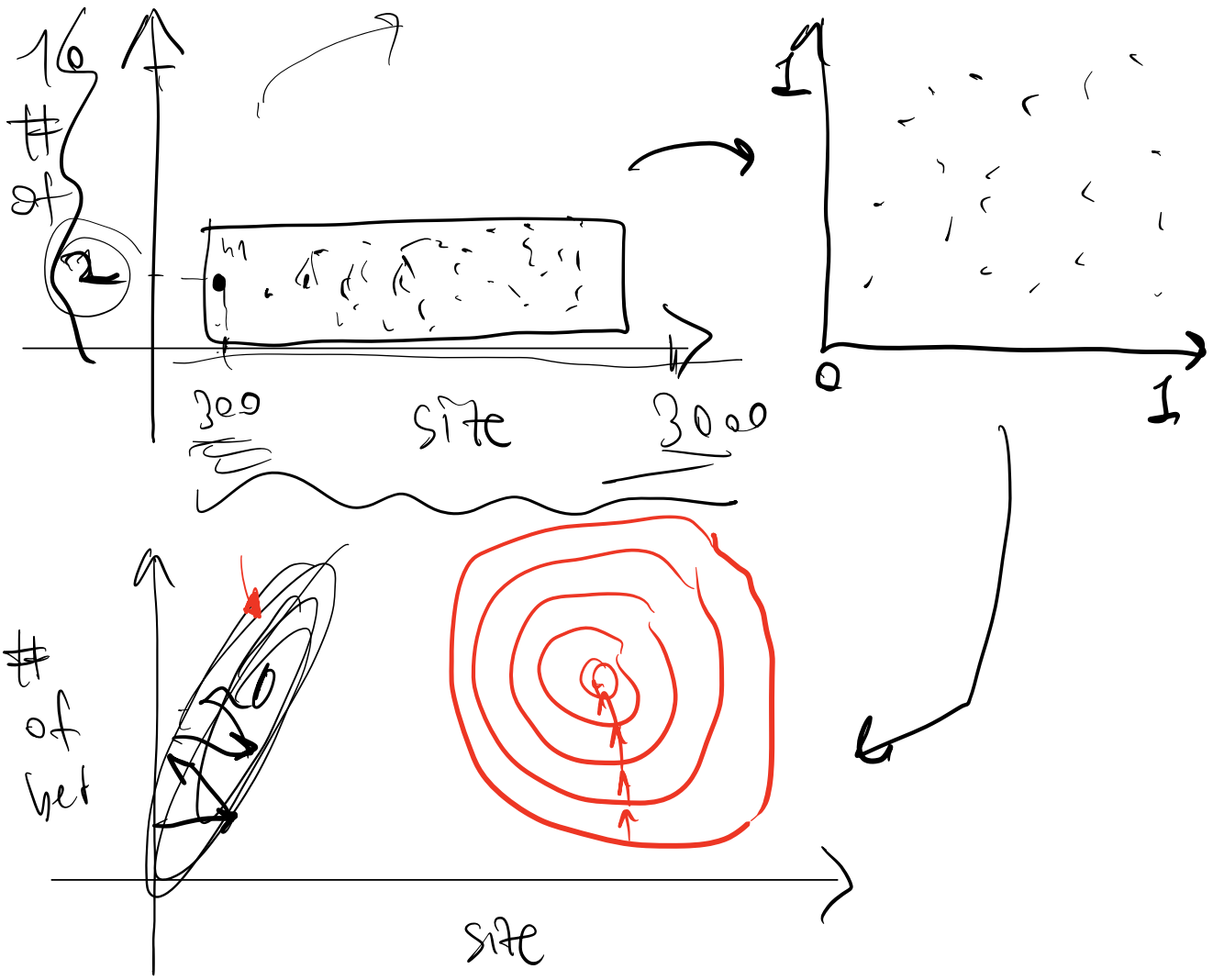
House 1: $x_1 = 2000$, $x_2 = 5$, price = \$500K

$w_1 = 0.1$, $w_2 = 50$, $b = 50$

$\hat{price} = 0.1 * 2000 + 50 \times 5 + 50$

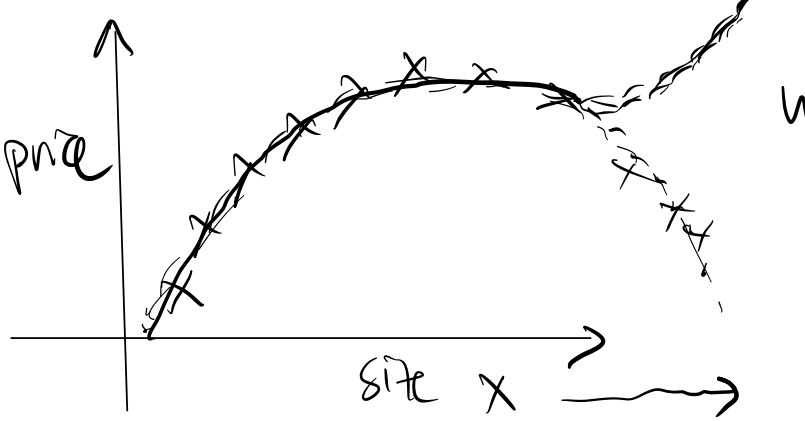
2000K 250K 50K

$= 500\text{K}\$$



polynomial regression.

$$w_1 x + w_2 x^2 + w_3 x^3$$



$$w_1 x + w_2 x^2 + b$$

feature engineering

$w_1 x_1 + w_2 x_2 + b$

↓ frontage · ↓ depth

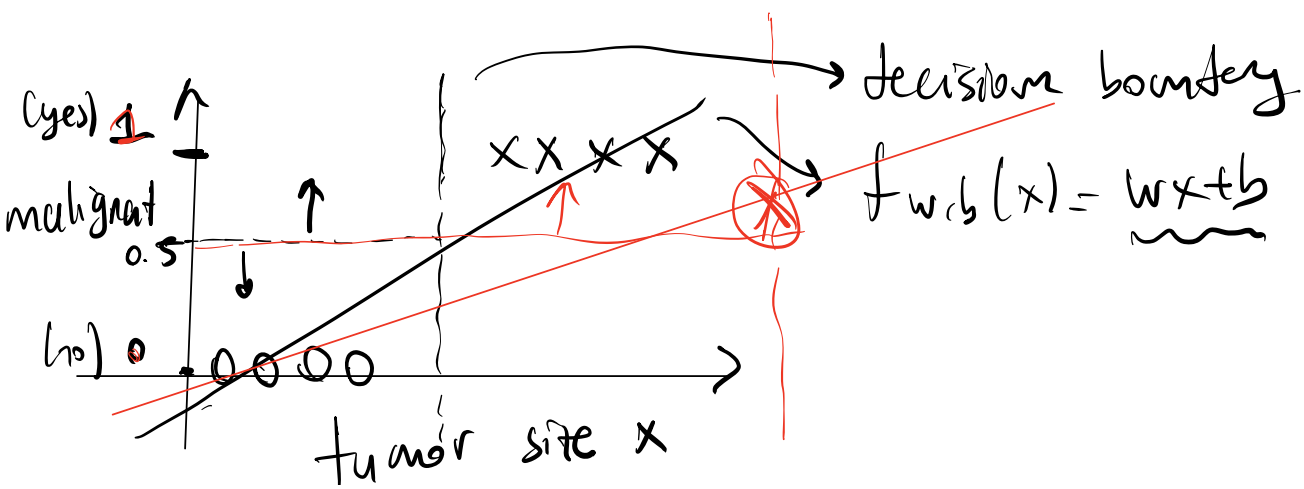
$x_3 = \underbrace{\text{frontage} \times \text{depth}}_{\text{area}}$

$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3$

Classification

↳ binary → y → 0, 1 (false)

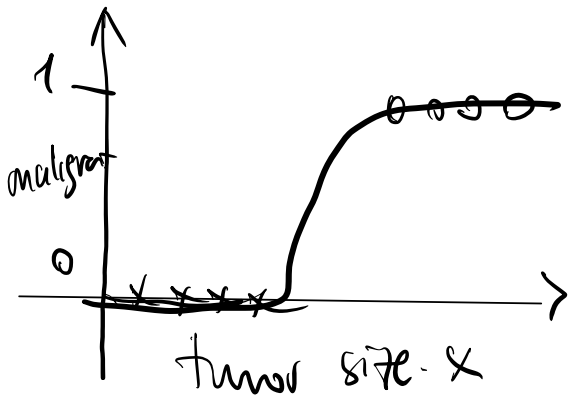
↳ multiclass → y → 0 ... 9



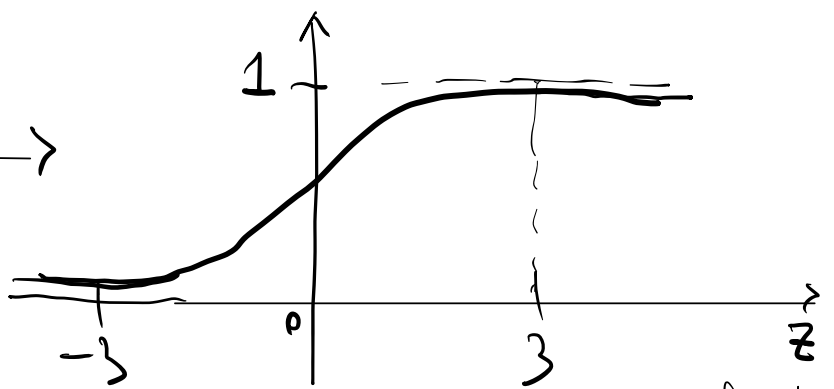
if $f_{w,b}(x) < 0.5 \quad \hat{y} = 0$

if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$

Logistic Regression.



want outputs btw 0-1



sigmoid (logistic) function.

outputs 0 - 1

sigmoid

$$g(z) = \frac{1}{1+e^{-z}}$$

$$0 < g(z) < 1$$

$$f_{\vec{w},b}(\vec{x}) = wx + b$$

$$z = \vec{w}\vec{x} + b \longrightarrow g(z) = \frac{1}{1+e^{-z}}$$

$$0 - 1$$

$$= \boxed{\frac{1}{1 + e^{-(\vec{w}\cdot\vec{x} + b)}}}$$

$$f_{\vec{w},b}(\vec{x}) = 0.7$$ 70% chance that y is 1

30% chance that y is 0

$$P(y=0) + P(y=1) = 1$$

$$f_{\vec{w},b}(\vec{x}) = P(y=1 \mid \vec{x}; \vec{w}, b)$$

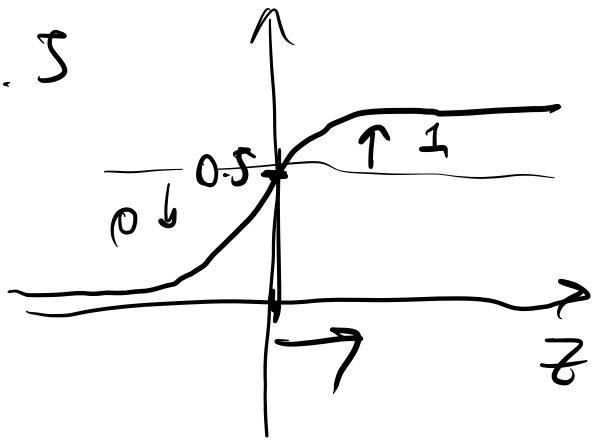
Decision Boundary.

$$\frac{1}{1+e^{-(\vec{w}\vec{x}+b)}} = P(y=1 \mid \vec{x}; \vec{w}, b) = 0.7$$

is $f_{\vec{w},b}(\vec{x}) \geq 0.5$ → threshold.

yes $\Rightarrow \hat{y} = 1$, $\hat{y} = 0$

when is $f_{\vec{w},b} \geq 0.5$

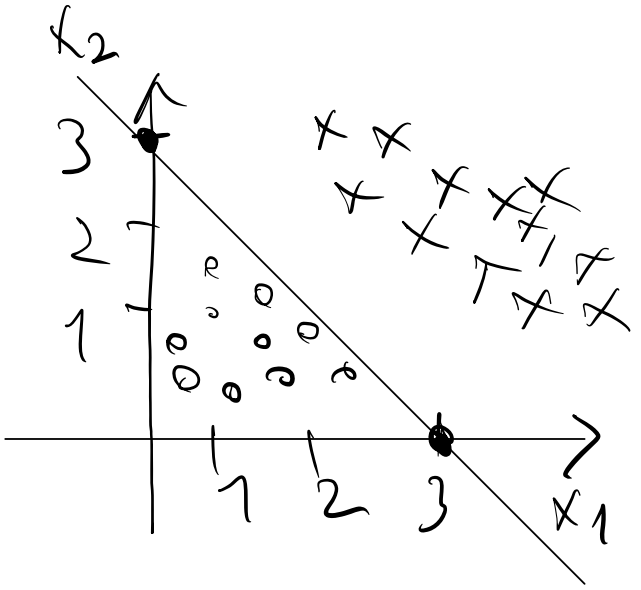


$g(z) \geq 0.5$

$z \geq 0$

$\vec{w}\vec{x}+b \geq 0 \Rightarrow \hat{y} = 1$

$\vec{w}\vec{x}+b < 0 \Rightarrow \hat{y} = 0$



$$f_{\vec{w},b} = g(z)$$

$$g(z) = g(w_1 x_1 + w_2 x_2 + b)$$

w_1 → 1, w_2 → 1, b → −3

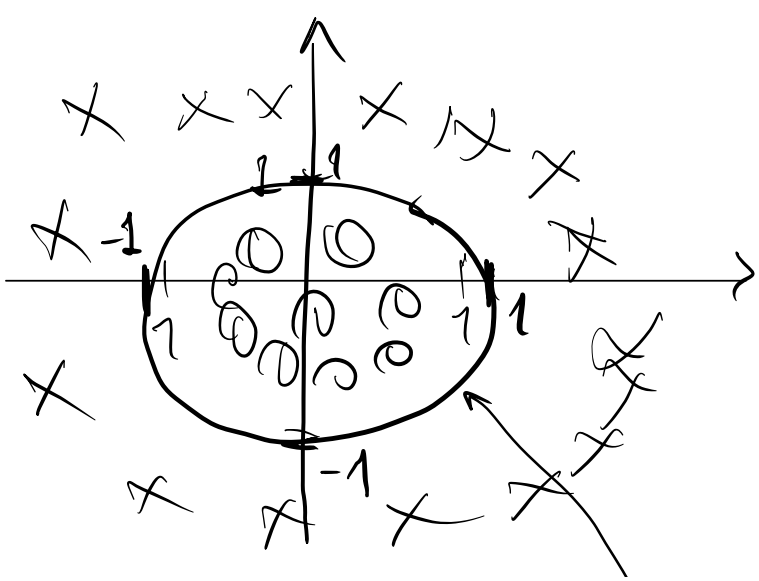
decision boundary.

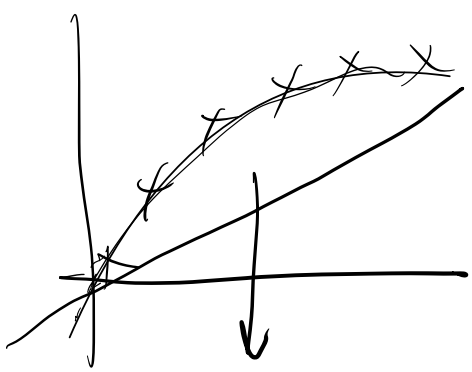
$$z = \vec{w}\,\vec{x} + b = 0$$

$$z = x_1 + x_2 - 3 = 0$$

$$x_1 + x_2 = 3$$

Non-linear decision boundaries.





→ $wx + b$

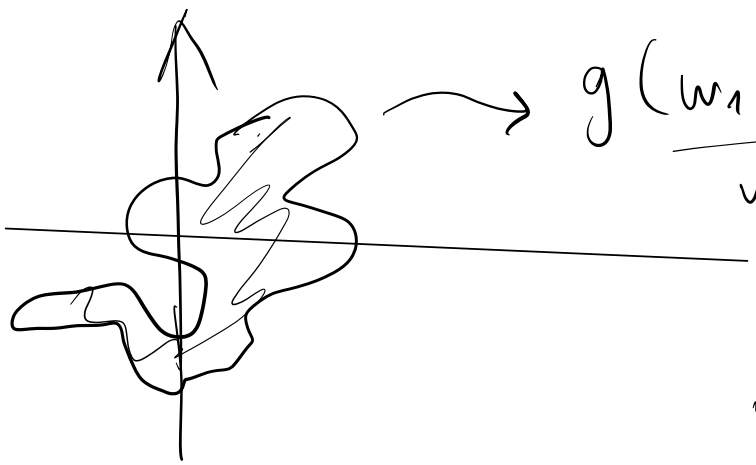
→ $wx + wx^2$

$$g(w_1 x_1^2 + w_2 x_2^2 + b)$$
$$1 \quad 1 \quad -1$$

$$x_1^2 + x_2^2 < 1$$
$$y = 0$$

$$z = x_1^2 + x_2^2 - 1 = 0$$

$$\boxed{x_1^2 + x_2^2 = 1}$$



$$\rightarrow g(w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + \dots + w_6 x_1^3 + b)$$

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w}\vec{x} + b)}}$$

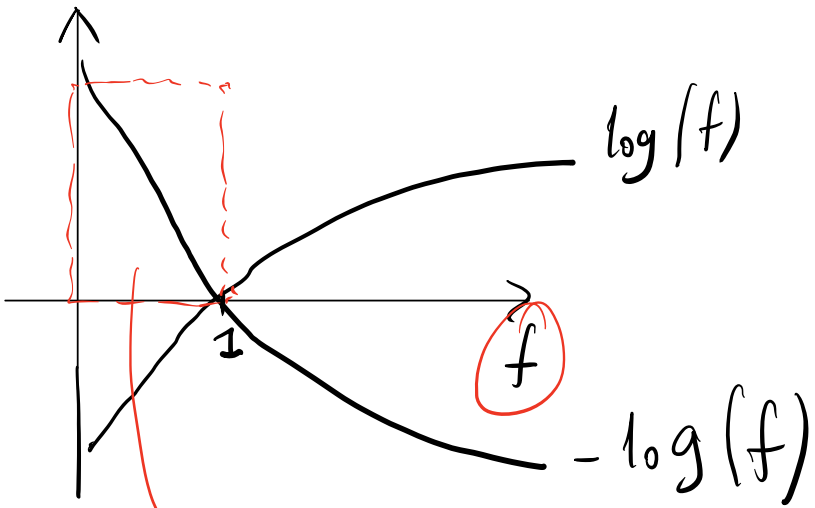
$$\boxed{\vec{w} = [w_1 \dots w_n] \text{ and } b \in ?}$$

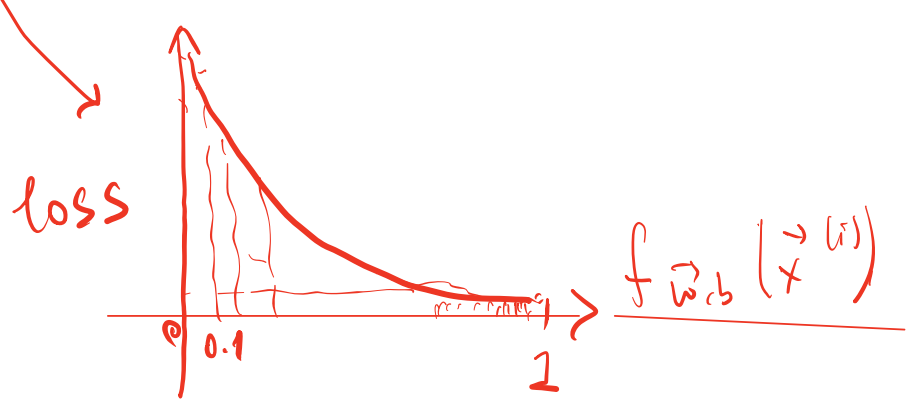
predict weight.

Linear Reg → $\boxed{MSE} = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(x^{(i)}) - y^{(i)} \right)^2$

Logistic loss function. for only one tr. sample-

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)}=1 \quad (1) \\ -\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)}=0 \quad (2) \end{cases}$$

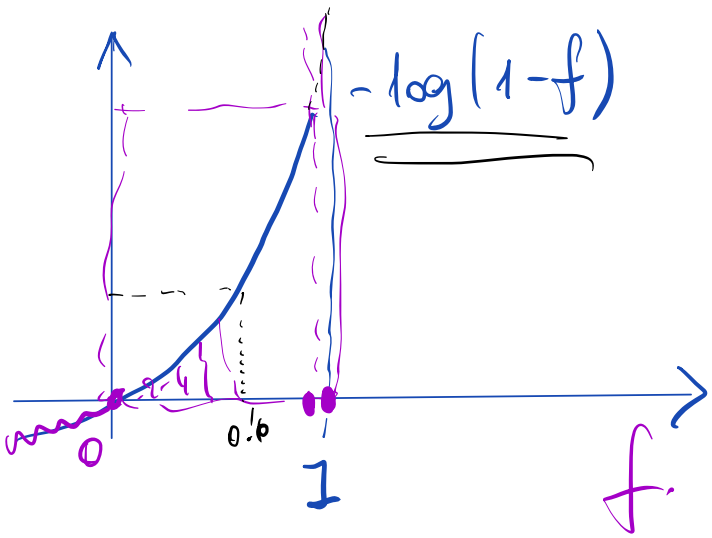




$x^{(i)}$, $y^{(i)}$

email spam (1)

- $f_{\vec{w},b}(\vec{x}) \to 1$ then loss $\to 0$
- $f_{\vec{w},b}(\vec{x}) \to 0$ the loss $\to \infty$



$y=0$

$0.51 \geq 0.5 \rightarrow 1$

$0.9999 \geq 0.5 \rightarrow 1$

$f_{\vec{w},b}(\vec{x}) = 1$ then loss $\rightarrow \infty$

$f_{\vec{w},b}(\vec{x}) = 0$ then loss $\rightarrow 0$, ...

↗ loss for one sample

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -y^{(i)} \log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) - (1-y^{(i)}) \log\left(1 - f_{\vec{w},b}(\vec{x})\right)$$

if $y^{(i)} = 1$: $-1 \log(f(\vec{x}))$

if $y^{(i)} = 0$: $-1 \log[1 - f(\vec{x})]$

# of ex : (m)

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^{m} [\quad\quad]$$

Repeat {

$$w_j = w_j - \alpha \frac{d}{dw_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{d}{db} J(\vec{w}, b)$$

} Simultaneous update.

$$J(\vec{w}) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(w^T x_i) + (1-y_i) \log(1-\sigma(w^T x_i))]$$

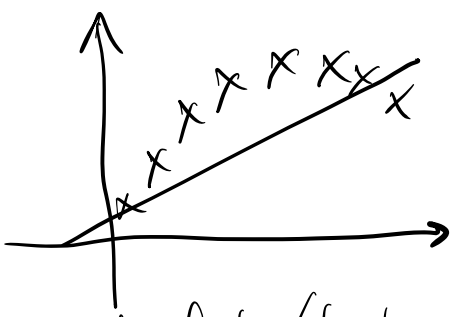
$$\sigma = \frac{1}{1+e^{-z}}, \quad z = w^T x_i$$

① $\sigma'(z) = \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) = \sigma'(z) = \sigma(z)(1-\sigma(z))$

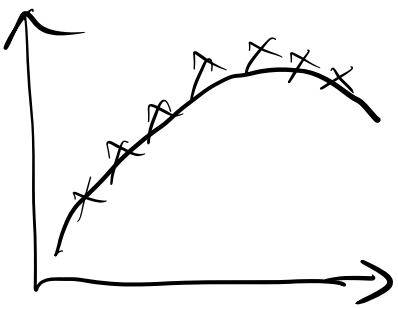
(2)

(4) $$\frac{\partial J(w)}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} \left[y_i \left(1 - \sigma(w^T x_i)\right) - (1 - y_i)\, \sigma(w^T x_i) \right] x_{ij}$$

Overfitting



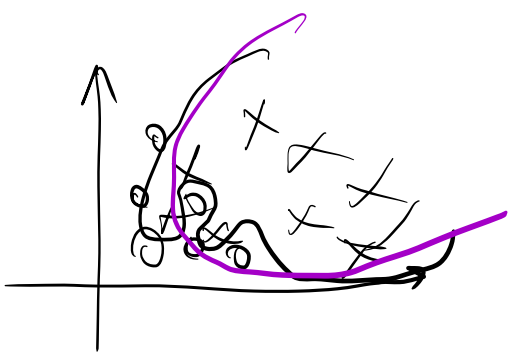
underfit / high bias.



generalize.

$w_1 x + \dots w_3 x^3$

overfit.
high variance



Address overfitting.

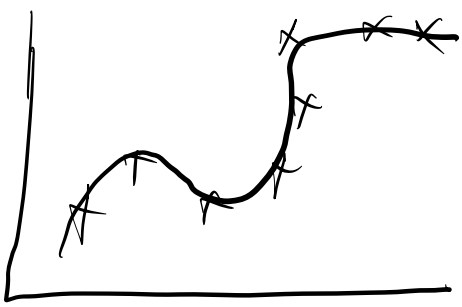
(1) collect more training data

(2) select features to include/exclude.

feature selection.

1 ——— 100

(3) Regularization.



large values for w_j

$$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$$

$$f(x) = 13x - 0.23x^2 + 0.00000014x^3 - 0.0001x^4 + 10$$

# of features

w_1 ——— w_{100} ($n = 100$)

min

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2$$

$\hat{y} - y$

① fit data

② keep w_j small

lambda regularization term > 0

balance goal.

$\lambda \downarrow \rightarrow$ overfit

$\lambda \uparrow \rightarrow$ underfit

Ridge (L2) = cost func = OLS loss + $\lambda \sum^{n} w_j^2$

Lasso (L1) = cost func = OLS loss + $\lambda \sum_{i=1}^{n} |w_i|$

→ feature selection, make zero.

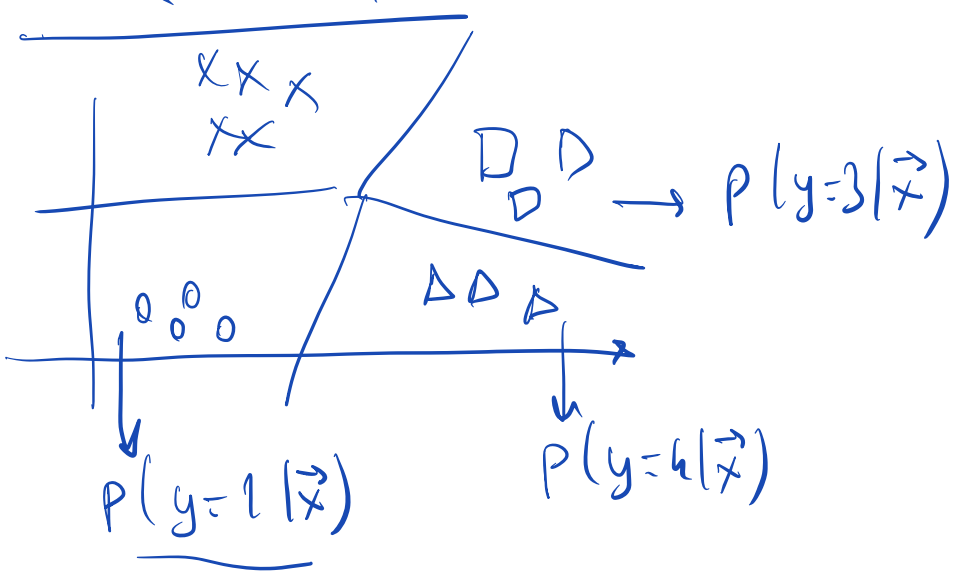
$w_1 x_1 + \cancel{w_2 x_2}$ ($w_2 = 0$)

Elastic-Net Regularization.

$$\text{OLS Loss} + \underbrace{p\,\lambda \sum_{i=1}^{n} |w_i|}_{\text{lasso}} + \underbrace{\frac{1-p}{2} \times \sum_{i=1}^{n} w_i^2}_{\text{ridge.}}$$

(p: $p=1$)

multiclass.



$$z = \vec{w}\vec{x} + b$$

$$a_1 = g(z) = \frac{1}{1+e^{-z}} = p(y=1|\vec{x}) \rightarrow 0.70$$

$$p(y=0|\vec{x}) = 0.3$$

4 possible outputs.

Softmax

$$z_1 = \vec{w}_1 x + b_1 = \quad a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} = P(y=1|\vec{x})$$

$z_2 =$ $\qquad = P(y=2|\vec{x})$

$z_3 =$ $\qquad = \cdots$

$$z_4 = w_4 x + b_4 = \quad a_4 = \frac{e^{z_4}}{e^{z_4} + e^{z_2} + e^{z_1} + e^{z_4}} = \cdots$$

logistic Reg 2output

$a_1 = \frac{e^{z_1}}{e^{z_1} \cdots + e^{z_N}}$

$a(1) = P(y=1 \mid \vec{x})$

$a(2) \quad P(y=0 \mid \vec{x})$

$loss = \underbrace{-y \log a_1}_{\text{if } y=1} \underbrace{- (1-y) \log(1-a_1)}_{\text{if } y=0}$

$a_N = \frac{e^{z_N}}{e^{z_1} \cdots e^{z_N}}$

Crossentropy loss

$= -\sum_{i=1}^{N} y_i \log(y_i)$

$\begin{cases} -\log a_1 & \text{if } y=1 \end{cases}$

$$loss(a_1 \dots a_n) = \begin{cases} -\log a_2 & \text{if } y = 2 \\ \vdots & \\ -\log_N & \text{if } y = N \end{cases}$$