AI vs ML

Data is knowledge. AI is the representation & manipulation of that knowledges.

Deep Learning is a branch of Neural Network.

"How much does a rhino weight?"

Descriptive Statistic : mean / median / mode.

Is the data we have valid enough to answer?

Did not miss a rhino.

PDF. x has pdf f. $\int_a^b f(x)dx : P(a \le X \le b)$

cdf

$$F(x) = \int_{-\infty}^{\infty} f(\epsilon)\, d\epsilon = P(X \le x)$$

Average weight of rhino.

True answer.

$$\mu = E[X] = \int_{-\infty}^{\infty} x f(x)\, dx$$

Procedue :

$$\bar{X}_n = \frac{1}{n}(X_1 + \cdots + X_n)$$

How close is $\bar{X}_n$ to μ?

$$E[\bar{X}_n] = E\left[\frac{1}{n}(X_1 + \cdots + X_n)\right]$$
$$= \frac{1}{n} E[X_1 + \cdots + X_n]$$
$$= \frac{1}{n}(E[X_1] + \cdots + E[X_n])$$
$$= \frac{1}{n}(\mu + \cdots \mu) = \mu$$

$\mu \rightarrow$ Expectation

valid way

Variance.

$$Var(X) = E[(X - \mu)^2] \Rightarrow \text{expect to be zero.}$$

$E[X] = \mu$

$Var(X) = \sigma^2$

Variance $(\bar{X}_n)$

$$= Var\left(\frac{1}{n}(X_1 + \cdots + X_n)\right)$$
$$= \frac{1}{n^2} Var(X_1 + \cdots + X_n)$$
$$= \frac{1}{n^2}[Var(X_1) + \cdots + Var(X_n)]$$
$$= \frac{1}{n^2}[\sigma^2 + \cdots + \sigma^2]$$
$$= \frac{\sigma^2}{n} \Rightarrow Var(\bar{X}_n) = E[(\bar{X}_n - \mu)^2] = \frac{\sigma^2}{n}$$

$n \uparrow$ $Var(\bar{X}_n) \downarrow$ more accurate.

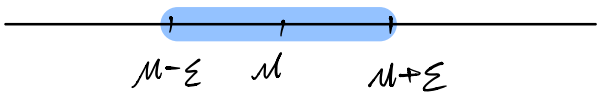
"How confident can we be in our estimate?"

$$P(|\bar{X}_n - \mu| \geq \varepsilon)$$
$$\leq \frac{Var(\bar{X}_n)}{\varepsilon^2} = \frac{\sigma^2}{\varepsilon^2} = O\left(\frac{1}{n}\right)$$

Chebyshev.

$$P(|X - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2}$$

The more we measure, the more likly it falls in this range.



$X_i \in \{0, 1\}$ $\quad E[X_i] = P(X_i = 0) \cdot 0 + P(X_i = 1) \cdot 1$

$= P(X_i = 1) = \mu$

Hoeffding Inequality.

$P(|\bar{X_n} - \mu| \geq \varepsilon) \leq 2e^{-2\varepsilon^2 n}$ (only hold for binary measuring)

For binary data, we need less sample.

$P(|\bar{X_n} - \mu| \geq \varepsilon) \leq \delta$

$2e^{-2\varepsilon^2 n} \leq \delta$

$e^{-2\varepsilon^2 n} \leq \frac{1}{2}\delta$

$-2\varepsilon^2 n \leq \ln(\frac{1}{2}\delta)$

$n \geq -\frac{1}{2\varepsilon^2}\ln(\frac{1}{2}\delta)$

$n \geq \frac{1}{2}\frac{1}{\varepsilon^2}\ln(\frac{2}{\delta})$

$n \geq O(\frac{1}{\varepsilon^2}\ln(\frac{1}{\delta}))$

$\varepsilon \downarrow$ $\delta \downarrow$ and $n \uparrow$ (more sample)

Maximum Likelihood Estimation.

what parameter make your data as likely as possible?

Data : $x_1 . x_2 \cdots x_n$

$$\begin{aligned} lik_{\mu\sigma^2}(data) &= f(x_1)\cdot f(x_2)\cdot \cdots f(x_n) \\ &= \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x_i-\mu)^2\right) \\ &= \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2\right) \\ &= \ln\left(\frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2\right)\right) \\ &= -N\ln(\sigma) - \frac{N}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 \end{aligned}$$

$$L = \ln lik_{\mu\sigma^2}(data) = -N\ln(\sigma) - \frac{N}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2$$

$$\frac{dL}{d\mu} = 0 - 0 - \frac{1}{2\sigma^2}\sum_{i=1}^{N} -2(x_i-\mu) = 0$$

$$= \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i-\mu) = 0$$

$$\frac{dL}{d\sigma} = -\frac{N}{\sigma} - \frac{(-2)}{2\sigma^3}\sum_{i=1}^{N}(x_i-\mu)^2 = 0$$

find a extremum of a function

take derivative and set to zero.

$$\sum_{i=1}^{N}(x_i-\mu) = 0$$

$$\left(\sum_{i=1}^{N} x_i\right) - \left(\sum_{i=1}^{N}\mu\right) = 0$$

$$\sum_{i=1}^{N} x_i - N\mu = 0$$

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \Leftarrow \text{estimate of true value}$$

$$-\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\hat{\mu})^2 = 0$$

$$\frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\hat{\mu})^2 = \frac{N}{\sigma}$$

$$\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{\mu})^2 = \sigma^2$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{\mu})^2$$

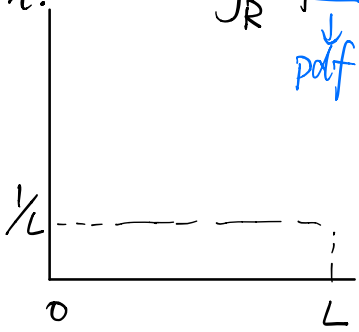
Uniform Distribution.

$X \sim Unif[0, L]$

$X_1, X_2, \ldots, X_N \sim Unif[0, L]$

From data, estimate L.

$E[X] = \int_R x f(x)\, dx$ ($f(x)$: pdf)



$\int_0^L x \left(\frac{1}{L}\right) dx$ → pdf of uniform Distribution

$= \frac{1}{L} \cdot \frac{1}{2} x^2 \Big|_0^L$

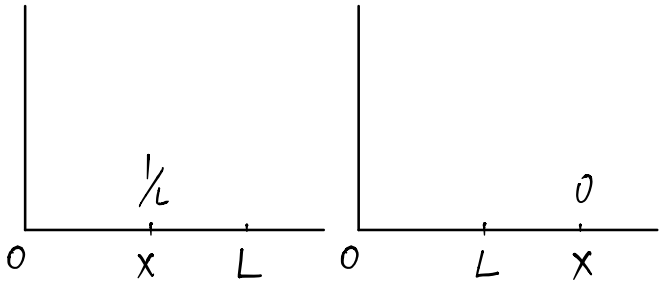
$= \frac{1}{L} \cdot \frac{1}{2} \cdot L^2 = \frac{1}{2} L$

M.O.M

$E[X] = \frac{1}{2} L$

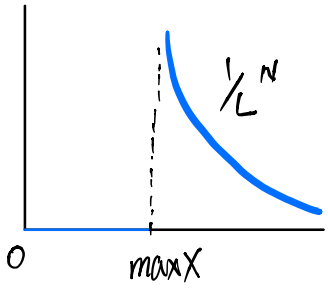
$\hat{L} = 2\bar{X}_n$

$likelihood_L(x) = \begin{cases} 1/L & L \geq x \\ 0 & else \end{cases}$



$lik_L(data) = \begin{cases} (1/L)^N & L \geq \max X \\ 0 & else \end{cases}$

Find L to maximise lik_L.

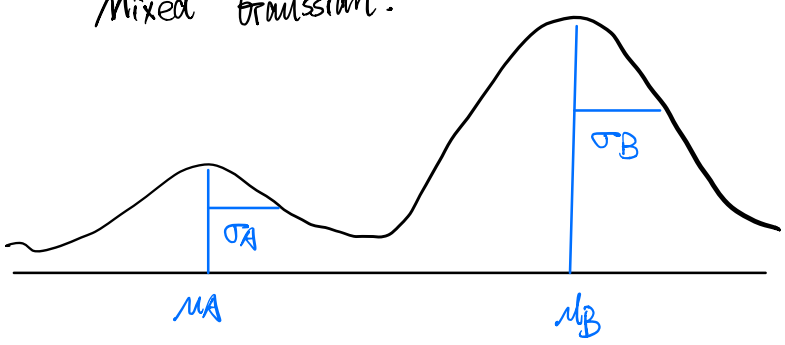


$\hat{L}_{MLE} = \max \bar{X}_i$

$P(|\hat{L}_{MOM} - L| \geq \varepsilon)$

$P(|\hat{L}_{MLE} - L| \geq \varepsilon)$

Mixed Gaussian.



Typicality of Data

last 10 days. lost 7 times.

H_0: hypothesis: the coin is fair

H_1: hypothesis: the coin is not fair (The coins is biased toward tails)

$P(7 \text{ tails in 10 flips} \mid H_0)$

$P(7 \text{ or more tails in 10 flips} \mid H_0)$

$$= \sum_{k=7}^{10} \binom{10}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{10-k} \doteq 0.17$$

≤ 0.05

exceedingly unlikly

$P(70 \text{ tails in 100 flips} \mid H_0)$

$$= \sum_{k=70}^{100} \binom{100}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{100-k} \doteq 0.00003?$$

pill A / 50%

pill B / 70%

Bayesian Analysis

H_0: the pills are pill A

H_1: the pills are pill B

Data.

$$\frac{P(\text{Data}\mid H_0)}{P(\text{Data}\mid H_1)} \Rightarrow \frac{P(H_0\mid \text{Data})}{P(H_1\mid \text{Data})} = \frac{P(H_0)\,P(\text{Data}\mid H_0)}{P(\text{Data})}$$

How much should you believe the hypothesis given the data.

$$\frac{P(H_0)}{P(H_1)} \cdot \frac{P(\text{Data}\mid H_0)}{P(\text{Data}\mid H_1)} = \frac{P(H_0)\,P(\text{Data}\mid H_0)}{P(H_0)P(\text{Data}\mid H_0) + P(H_1)\cdot P(\text{Data}\mid H_1)}$$

$$P(H_1\mid \text{Data}) = \frac{P(H_1)\,P(\text{Data}\mid H_1)}{P(H_0)P(\text{Data}\mid H_0) + P(H_1)\cdot P(\text{Data}\mid H_1)}$$

if $P(H_0) = P(H_1) = \frac{1}{2}$

$$P(H_0\mid \text{Data}) = \frac{P_0}{P_0+P_1}$$

$$P(H_1\mid \text{Data}) = \frac{P_1}{P_0+P_1}$$

Hypothesis. X. Y are independent.

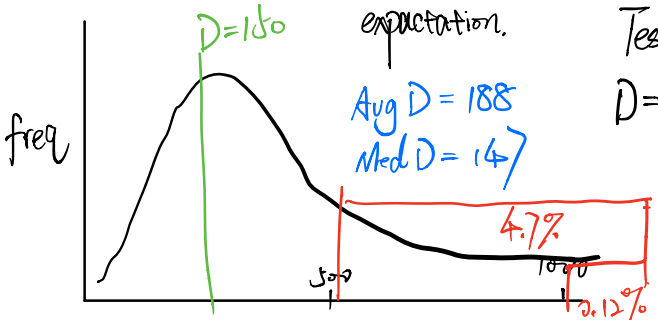
flip coin X, coin Y 1000 times together.

$O_{HH} = 236$ (250)

$O_{HT} = 265$ (250) $\quad P(X=H) = \frac{1}{2}$

$O_{TH} = 260$ (250) $\quad P(Y=H) = \frac{1}{2}$

$O_{TT} = 239$ (250)



Test statistic $\quad$ Deviation

$$D = \frac{1}{4}\left((O_{HH}-\bar{E}_{HH})^2 + (O_{HT}-\bar{E}_{HT})^2 + (O_{TH}-\bar{E}_{TH})^2 + (O_{TT}-\bar{E}_{TT})^2\right)$$

$$E_{HH} = P(X=H)\,P(Y=H)\cdot N = N/4$$

($P(X=H)$ and $P(Y=H)$ each $\frac{1}{2}$)

$$P = \frac{O_{HT} + O_{HH}}{O_{HH} + O_{HT} + O_{TH} + O_{TT}}$$

$$Q = \frac{O_{HH} + O_{TH}}{O_{HH} + O_{HT} + O_{TH} + O_{TT}}$$

$$E_{HH} = PQN \qquad T = \sum_x \sum_y \frac{(O_{xy} - E_{xy})^2}{E_{xy}}$$

If x, y are independent. then $T \sim \chi^2_1$, $P(T \geq T_0)$

$$= \int_{T_0}^{\infty} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\epsilon}} e^{-\epsilon/2} d\epsilon$$

$$P(T \geq 3.841) = 5\%$$

X has r values. Y has c values

$$T \sim \chi^2_{(r-1)(c-1)}$$

Data Input $x = (x_1, x_2, \cdots x_d)$

output $y = 0/1$

$\{(x_i, y_i)\}_{i=1}^{m}$ i.i.d from some distribution.

Given $X = x$, find $P(Y=y \mid X=x)$.

Given $\{(x_i, y_i)\}_{i=1}^m$. Assume $y = f(x)$ for some f.
How can we find f to model our data?

Hypothesis space $f: x \to y$. set of boolean function

find hypothesis f

X_1	X_2	Y
1	1	0
1	0	1
0	1	1
0	0	0

Truth table.

$$err(f) = \mathbb{P}(f(x) \neq y)$$

Goal: find f to minimize $err(f)$.

we don't know, so we can not compute $err(f)$.

↓ This goal doesn't stand.

$$err_{train}(f) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{f(x_i) \neq y_i\}$$

if $f(x_i) \neq y_i$, $= 1$
else $= 0$.

$$= \text{avg misclassification on the data}$$

Solve: $\min_{f \in H} err_{train}(f) \Rightarrow f^*$

f is overfit if $err_{train}(f)$ small, $err(f)$ large.

if $f^* \to$ true f. $\hat{f} \to$ estimate f.

$$\hat{f}(x) = \begin{cases} y_i & \text{if } x = x_i \\ 1 - f^*(x) & \text{else} \end{cases}$$

estimate $\hat{f}$ is zero.
but f^* could be quite large.

If H is finite, if we want $|err(f) - err_{train}(f)| < \varepsilon$.

$m \geq O\left(\frac{1}{\varepsilon^2} \ln|H|\right)$ the more hypothesis, the more data we need.

↳ number of functions in H.

H: boolean functions on D variables.

Truth table

X_1	X_2	X_3	...	X_D	$f(X)$
2^D ↕ 0					
				1	

$|H| = 2^{(2^D)}$

How many data points do we need to prevent overfitting?

$$\frac{1}{\varepsilon^2} \ln|H| = \frac{1}{\varepsilon^2} \ln(2^{2^D}) = O(2^D)$$

$$= 2^D \frac{1}{\varepsilon^2} \ln(2)$$

Regularization: Restricting H.

- Occams Razor.
- If H is smaller, we need less data to prevent overfitting.
- Simplier models have less room for spurious relationships. 假的

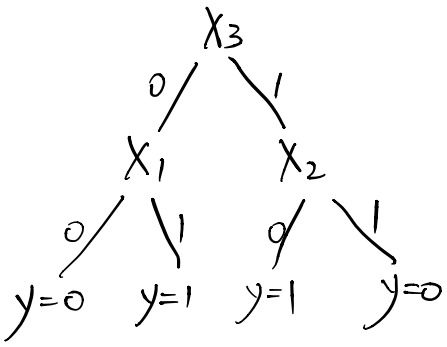
Problem of machine learning.

$\min_{f \in H} err_{train}(f)$ → algorithms we trying to solve this problem.

To estimate $err_{train}(f) \approx err(f)$

- read a lot of data
- need few hypothesis.

Decision Tree. Graphical models of restricted if/else statements.



$f(X_1, X_2, X_3) = Y$

Any boolean functions can be write into decision tree.

1) How to fit decision trees to data?
2) How can we regularize?
   - Learn towards shorter trees.
3) Given a tree, how can we improve it?

Algorithm.

1) Pick a decision variable X_i.
2) Partition data : $X_i = 0/1$
3) Recursively split on each part.
4) Continue until all data in one branch has the same y value

ID3.

X_1	X_2	X_3	Y	
1	1	0	1	
1	0	1	1	
1	0	0	0	we don't know.
0	0	0	?	No data about X_1.

entropy

$$H(Y) = \sum_y -P(Y=y)\log_2 P(Y=y)$$

$y = C$. $H(Y) = 0$.

$y = \{1, 2, 3, \ldots, 1024\}$. $H(Y) = \sum_{i=1}^{1024} -\left(\frac{1}{1024}\right)\log_2 1024$

$$= 10 \times \frac{1}{1024} \times 1024 = 10$$

$$H(Y|X=x) = -\sum_y P(Y=y|X=x)\log_2 P(Y=y|X=x)$$

Suppose x, y are independent. $P(Y=y|X=x)$

$= P(Y=y)$ X tell us nothing.

Avg information → $H(y|x) = \sum_x P(x=x) H(Y|X=x)$

x dependency over y

$$IG(x) = H(y) - H(Y|x)$$

information game.

if $x = Y$. $IG(x) = H(Y) - H(Y|y) = H(Y)$

$H(y|y) = \sum_y P(y=y) H(Y|y=y) = 0$

x, y are independent. $H(Y|x) = H(Y)$

$IG(x) = H(Y) - H(Y|x) = 0$.

ID3 – Iterative Dichotomiser 3

1) For each X_i, estimate $IG(x_i)$.
2) Find X_i with maximum IG and split on it.
3) Recurse on each part.
4) Stop when the remaining data having same Y.

Estimation of ID3.

Data Set

X_1	X_2	X_3	...	X_D	Y

$$IG(X_3) = H(Y) - H(Y|X_3)$$

$$H(Y) = -P(Y=0)\log_2(P(Y=0)) - P(Y=1)\log_2(P(Y=1))$$

$$P(Y=0) = \frac{\#\{y=0\}}{\text{total } \#\text{samples}}.$$

⇓ estimation.

$$\hat{IG}(X_3) = \hat{H}(Y) - H(Y|X_3).$$

$$H(Y|X_3) = P(X_3=0)H(Y|X_3=0) + P(X_3=1)H(Y|X_3=1)$$

$$H(y|X_3=0) = -P(Y=0|X_3=0)\log_2 P(Y=0|X_3=0) - P(Y=1|X_3=0)\log_2 P(Y=1|X_3=0)$$

$$P(Y=0|X_3=0) = \frac{P(Y=0, X_3=0)}{P(X_3=0)} = \frac{\#\{y=0, X_3=0\}/N}{\#\{X_3=0\}/N}$$

$$\hat{IG}(X_3) = \hat{H}(Y) - \hat{H}(Y|X_3)$$

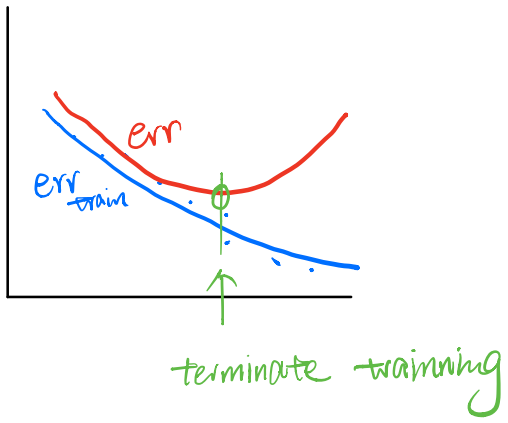
How to prevent overfitting.

- Stop when we hit a certain depth.
- Assign by majority
- Assess probability.
- Stop when you hit certain sample size.

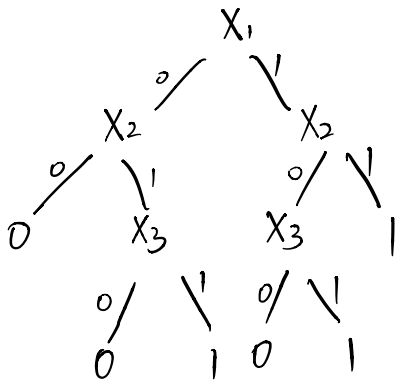
$err_{train}(f)$

$err(f)$

Data: 80/20



$f(x_1, x_2, x_3) = \text{majority}(x_1, x_2, x_3)$



Perceptron

if = 0, output +1.

$$f(\underline{x}) = \text{sgn}(w_1 x_1 + w_2 x_2 + \cdots + w_D x_D + b)$$

$f(x_1, x_2, x_3) = \text{majority}(x_1, x_2, x_3)$, $x_1, x_2, x_3 \in \{+1, -1\}$

1-D data, $x \in \mathbb{R}$, $y \in \{+1, -1\}$

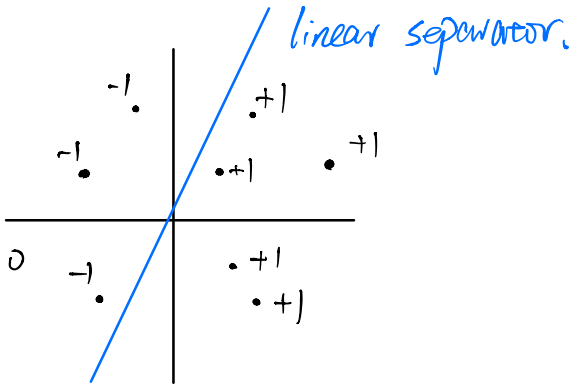
$$f(x) = \begin{cases} +1 & x \geq -b/w \\ -1 & \text{else.} \end{cases}$$

threshold

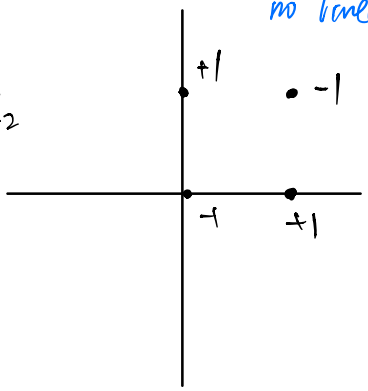
$X = (x_1, x_2) \in \mathbb{R}^2$

$y \in \{-1, +1\}$

$$f(x_1, x_2) = \begin{cases} +1 & w_1 x_1 + w_2 x_2 + b \geq 0 \\ -1 & \text{else} \end{cases}$$



$f(x_1, x_2) = x_1 \text{ xor } x_2$



$$f(\underline{x}) = \begin{cases} +1 & \underline{w}\underline{x} + b \geq 0 \\ -1 & \text{else} \end{cases}$$

Perceptron.

- $\underline{w}$ (vector) $= \underline{0}$, $b = 0$
- Find a misclassified point $(\underline{x}, y)$ — if no linear separator, gonna run forever.
- $\underline{w} \leftarrow \underline{w} + y\,\underline{x}$ $\quad f_{\underline{w}}(\underline{x}) \neq y$
  $b \leftarrow b + y$
- Repeat

current model $\underline{w}, b$

$\underline{w}' = \underline{w} + y\underline{x}$

$b' = b + y$

$f(\underline{x}) = +1, \; y = -1$

$f(\underline{x}) = -1, \; y = +1$

$(\underline{w} \cdot \underline{x} + b)\, y < 0$

$\underline{w}' \cdot \underline{x} + b'$

$= (\underline{w} + y\underline{x}) \cdot \underline{x} + (b + y)$

$= \underline{w} \cdot \underline{x} + b + y(\|\underline{x}\|^2 + 1)$

$(\underline{w}' \cdot \underline{x} + b')y = (\underline{w} \cdot \underline{x} + b)y + y^2(\|\underline{x}\|^2 + 1)$

mis classified.

$(\underline{w} \cdot \underline{x} + b) \cdot y < 0$

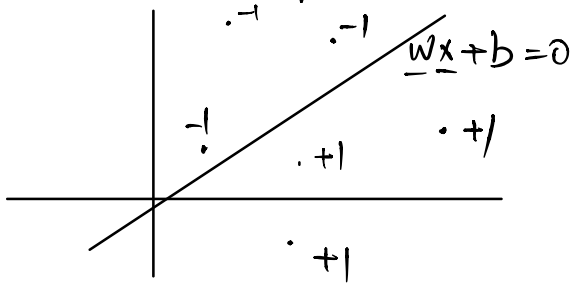
cur classified.

$(\underline{w} \cdot \underline{x} + b) \cdot y > 0$

$\gamma(\underline{w}, b)$ = margin of the separator $\underline{w}\underline{x} + b$

= the distance of the closest data point to $\underline{w}\underline{x} + b$

$= \min_i \frac{|\underline{w}\underline{x}_i + b|}{\|\underline{w}\|}$



If a linear separator exist ($\underline{w}\underline{x} + b = 0$).

the PLA terminates in at most $T \leq \left(\frac{2}{\gamma(\underline{w}, b)}\right)^2$ steps.

Run the algo certain times. if it output a linear separator, Great. If not, then either there's no linear separator for this data or the data is not reasonably separatable. (too close, separating this is basically noise and not important).

Why perceptron?

Vs. decision tree. different model but giving same output is closer geometrically.

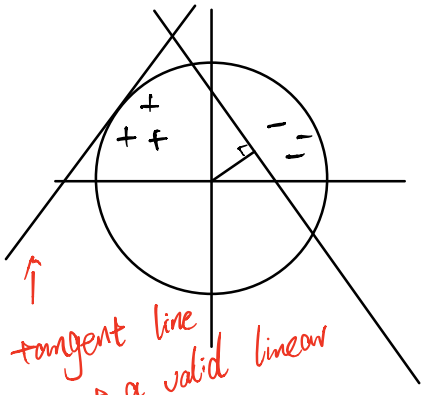
PLA.

- $\underline{w} \cdot b = 0$
- find misclassified data $\underline{x}, y$
- update $\underline{w} \leftarrow \underline{w} + y\underline{x}$

  $b \leftarrow b + y$
- Repeat

$$\gamma(\underline{w}, b) = \min_i \frac{|\underline{w} \cdot \underline{x}_i + b|}{\|\underline{w}\|_2}$$

Assume $\|\underline{x}_i\| \le 1$. if $\underline{w} \cdot \underline{x} + b = 0$ cuts into the unit circle, that, $b^2 / \|\underline{w}\|^2 \le 1$. $b^2 \le \|\underline{w}\|^2$



tangent line is not a valid linear separator.

H: sets of all possible perceptrons on a data set (from the PLA) (size m).

$|H| \leq m^T \leq m^{\frac{4}{(\gamma^*)^2}}$

Perceptrons.

Given $\{(\underline{X}^i, y^i)\}_{i=1}^m$ $\qquad T \leq (\frac{2}{\gamma^*})^2$

$\underline{W} = \underline{X}_1 y_1 + \underline{X}_2 y_2 + \cdots + \underline{X}_T y_T$ (iterations)

$|H| \leq m^T \leq m^{(\frac{2}{\gamma^*})^2}$

γ^* maximum separation between two data sets.

$|err_{train}(f) - err(f)| < \varepsilon \longrightarrow m \geq \frac{1}{\varepsilon^2} \log|H|$

$m \geq \frac{1}{\varepsilon^2}(\frac{2}{\gamma^*})^2 \log(m) \Rightarrow m \geq \frac{4}{\varepsilon^2(\gamma^*)^2} \log^2(\frac{4}{\varepsilon^2(\gamma^*)^2})$

Size of data

$\gamma(\underline{w}, b) = \min_i \frac{|\underline{w}\underline{x}^i + b|}{\|\underline{w}\|_2}$ $\qquad \gamma^* = \max_{\substack{\underline{w}, b \\ \text{valid}}} \gamma(\underline{w}, b)$

→ valid linear separator to separate two data sets

Constructed Satisfactory

Find $\underline{w}, b$ s.t. $\forall \operatorname{sgn}(\underline{w}\underline{x}^i + b) = y^i$

↓

Find $\underline{w}, b$ s.t. $\forall y_i(\underline{w}\underline{x}^i + b) > 0$.

$w_1(x_1^i y^i) + w_2(x_2^i y^i) + \cdots + w_b(x_b^i y^i) + \underline{b}y > 0$

Linear Programming

$$\max_{x_1 \cdots x_n} C_1 x_1 + \cdots + C_n x_n \quad \text{s.t.} \quad A\underline{x} \geq \underline{b}$$

trying to map this problem into linear programming problem.

To find a linear separator solve

$$\max_{\underline{w}, b} b, \quad \forall i \cdot y_i (\underline{w} \underline{x}_i + b) > 0$$

b: vector

There must be a δ s.t. $\square > 0$ and $\square \geq \delta$.

$$\downarrow$$

$$y_i (\underline{w} \underline{x}_i + b) \geq \delta \quad \text{strictly positive}$$

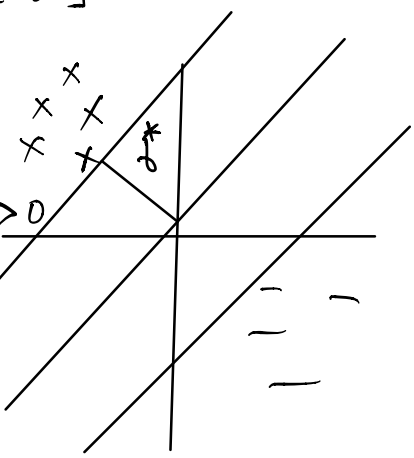
$$\downarrow \text{scale.}$$

$$y_i (\underline{w} \underline{x}_i + b) \geq 1$$

$m \geq \frac{1}{\varepsilon^2} d$ to present overfitting.

$$m \geq \min \left[\frac{4}{\varepsilon^2 (\gamma^*)^2} \log^2 \left(\frac{4}{\varepsilon^2 (\gamma^*)^2} \right), \frac{1}{\varepsilon^2} d \right]$$

Support Vector Machines



$$\max_{\underline{w}, b} \gamma(\underline{w}, b) \quad \text{s.t.} \quad \forall i : y_i (\underline{w} \underline{x}_i + b) \geq 0$$

$$\updownarrow$$

$$\max_{\underline{w}, b} \frac{1}{\|\underline{w}\|_2} \quad \text{s.t.} \quad \forall i : y_i (\underline{w} \underline{x}_i + b) \geq 1$$

$\|\cdot\|_2$: L_2 Norm

$$\max_{\underline{w}, b, \gamma} \left[\min_i \frac{|\underline{w} \underline{x}_i + b|}{\|\underline{w}\|} \right]$$

$$\forall i : \frac{|\underline{w} \underline{x}_i + b|}{\|\underline{w}\|} \geq \gamma \cdot \gamma \geq 0$$

$$y_i (\underline{w} \underline{x}_i + b) = |\underline{w} \underline{x}_i + b|$$

$$\forall i: \frac{y_i(\underline{w}\underline{x}_i + b)}{\|\underline{w}\|} \geq \gamma \quad \rightarrow \quad \forall i: y_i(\underline{w}\underline{x}_i + b) \geq \gamma \|\underline{w}\|$$

$\|\underline{w}\|$: norm of $\underline{w}$

scale $\|\underline{w}\|$: $\|\underline{w}\| \rightarrow 1/\gamma$

$$\max_{\underline{w}, b} \frac{1}{\|\underline{w}\|} \text{ s.t. } \forall i: y_i(\underline{w}\underline{x}_i + b) \geq 1.$$

find w, b

primal SVM.

$$\star \begin{cases} \min_{\underline{w}, b} \frac{1}{2}\|\underline{w}\|^2 \text{ s.t. } \forall i: y_i(\underline{w}\underline{x}_i + b) \geq 1 \\ m \geq \frac{1}{\varepsilon^2} \min\left[\frac{4}{(\gamma^*)^2} \log^2\left(\frac{4}{\varepsilon^2(\gamma^*)^2}\right) d\right] \end{cases}$$

solve this and we got support vectors $y_i(\underline{w}\underline{x}_i + b) = 1$

number of data size.

$$\underline{w}^* = \sum_{i=1}^{m} \alpha_i y_i x_i \quad . \quad \alpha_i \geq 0$$

$\alpha_i = 0$. iff (x_i, y_i) not a support vector.

$$f(x) = \text{sgn}(\underline{w} \cdot \underline{x} + b)$$
$$= \text{sgn}\left(\left(\sum_{i=1}^{m} \alpha_i y_i \underline{x}_i\right) \cdot \underline{x} + b\right)$$
$$= \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i (\underline{x}_i \cdot \underline{x}) + b\right)$$
$$= \text{sgn}\left(\sum_{\text{s.v.}} \alpha_i y_i (\underline{x}_i \cdot \underline{x}) + b\right)$$

we only need to consider the support vectors.

Dual SVM Problem

$$\max_{\underline{\alpha}} \left[\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{m} \alpha_i y_i (\underline{x}_i \cdot \underline{x}_j) y_j \alpha_j\right]$$
$$\text{s.t. } \sum_{i=1}^{m} \alpha_i y_i = 0 \qquad \forall i: \alpha_i \geq 0.$$

- solve for the dual α_i's
- Generate $\underline{w}^* = \sum_i \alpha_i y_i \underline{x}_i$
- Pick some i s.t. $\alpha_i > 0$

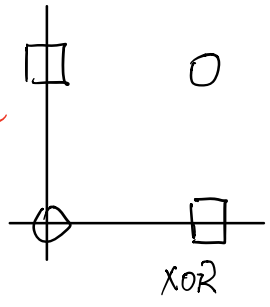
$$y_i(\underline{w}^{*T}\underline{x}_i + b) = 1$$

$$y_i y_i(\underline{w}^{*T}\underline{x}_i + b) = y_i$$

$$\underline{w}^{*T}\underline{x}_i + b = y_i$$

$$b = y_i - \underline{w}^{*T}\underline{x}_i$$

Non-separable Data



C-SVM. easier to solve

$$\min_{\underline{w},b,\xi} \frac{1}{2}\|w\|^2 + C\left(\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$$

s.t. $\forall i.\ y_i(\underline{w}\underline{x}_i + b) \geq 1 - \xi_i$ $\quad \xi$ = efa.

$\forall i.\ \xi_i \geq 0$ $\quad$ how much error we allow.

ν-SVM. better result.

$$\min_{\underline{w},b,\xi,\rho} \frac{1}{2}\|w\|^2 - \rho\nu + \frac{1}{m}\sum_{i=1}^{m}\xi_i \qquad err_{train}(f) \leq \nu$$

s.t. $\forall i.\ y_i(\underline{w}\underline{x}_i + b) \geq \rho - \xi_i$

$\forall i.\ \xi_i \geq 0$

$\rho \geq 0$

$$\min \frac{1}{2}\|\underline{w}\|^2 \qquad \text{s.t. } 0 \geq 1 - y_i(\underline{w}\cdot\underline{x}_i + b)$$

$$\nabla f(\underline{w}) + \sum_{i=1}^{m}\lambda_i \nabla g_i(\underline{w}, b) = 0$$

/

$$\underline{w} + \sum_{i=1}^{m} \lambda_i (-y_i \underline{x}_i) = 0$$

$$0 + \sum_{i=1}^{m} \lambda_i (-y_i) = 0$$

$$\underline{w} = \sum_{i=1}^{m} \lambda_i y_i \underline{x}_i$$

$$\sum_{i=1}^{m} \lambda_i y_i = 0$$

Lagrangian Duality

$$\min_{\underline{w}, b} f(\underline{w}, b)$$

s.t. $\forall i: 0 \geq g_i(\underline{w}, b)$

$$\max_{\underline{\lambda}} \min_{\underline{w}, b} \left[f(\underline{w}, b) + \sum_{i=1}^{m} \lambda_i g_i(\underline{w}, b) \right]$$

$$\lambda_i \geq 0$$

Non-linear SVMs

Feature Embedding

$$\phi(x_1, x_2) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

$$x_1 \mid x_2$$

$$\downarrow$$

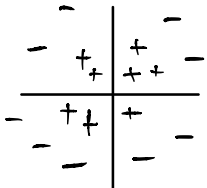
$$x_1 \mid x_2 \mid x_1 \cdot x_2 \mid x_1^2 \mid x_2^2$$

XOR

$-_B$	$+_A$
$+_C$	$-_D$

A: (1, 1)
B: (-1, 1)
C: (-1, -1)
D: (1, -1)

$sgn(-x_1 \cdot x_2)$



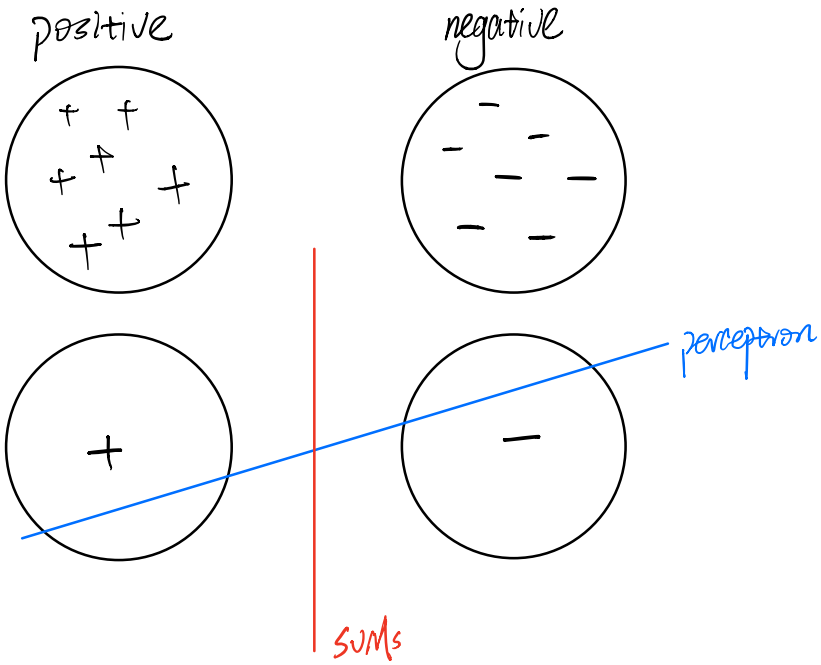
$x_1^2 + x_2^2 \leq 1$

$0 \leq 1 - x_1^2 - x_2^2$

$sgn(1 - x_1^2 - x_2^2)$

Sample Complexity

s.c. for perceptrons $\geq$ s.c. for SVMs



Optimization

$\min_{x \in C} f(x)$ $\longrightarrow$ Given f, C. find $\min_{x \in C} f(x)$.

Focus on d.

$C = \mathbb{R}$

f: differentiable.

Neccessary Condition:

Any minimizer must satisfy $\nabla f(x^*) = 0$

$$f(x) = \|x - a\|^2 + \|x - b\|^2$$

$$= (x - a)^T (x - a) + (x - b)^T (x - b)$$

$$= 2x^T x^T - 2x^T a - 2x^T b + \|b\|^2 + \|a\|^2$$

$$\nabla F(\underline{x}) = 2(\underline{x}-\underline{a}) + 2(\underline{x}-\underline{b}) = 0$$
$$\underline{x} = \frac{1}{2}(\underline{a}+\underline{b})$$

Guess $\underline{x}$, try to improve it to $\underline{x}'$ where $F(\underline{x}') < F(\underline{x})$

$\underline{x}' = \underline{x} + \underline{d}$

What must be true about $\underline{d}$ so that $F(\underline{x}') < F(\underline{x})$

$$x_1 = x_0 + d_0$$
$$f(x_1) \doteq f(x_0) + f'(x_0)(d_0)$$

Optimization

$$\min_{\underline{x} \in \mathbb{R}^d} F(\underline{x})$$

$\underline{x}_0$ = Some guess

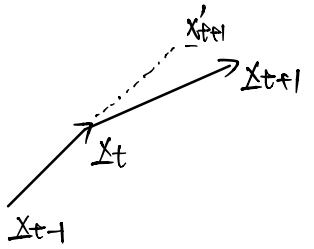
$\underline{x}_{t+1} = \underline{x}_t - \alpha_t \nabla F(\underline{x}_t)$ gradiant direction

$\underline{x}_t \to \underline{x}^*$

$\nabla F(\underline{x}^*) = 0$

Coordinate Descent : one coordinate at a time.

momentum method



$$\underline{x}_{t+1} = \underline{x}_t - \alpha_t \nabla F(\underline{x}_t) + \beta_t(\underline{x}_t - \underline{x}_{t-1})$$

$$F(\underline{x}) = \sum_{i=1}^{m} F_i(\underline{x})$$

Stochastic Gradient Descent

$$F(\underline{x}) \approx \mathbb{E}[F_i(\underline{x})]$$

$$F(\underline{x}) = \sum_{i=1}^{m} \frac{1}{m} F_i(\underline{x}) \qquad \nabla F(\underline{x}) \approx \mathbb{E}[\nabla F_i(\underline{x})]$$

SGD.

x_0 = Some Guess

At time t, pick i at random

$$\underline{x}_{t+1} = \underline{x}_t - \alpha_t \nabla F_i(\underline{x}_t)$$

Total Error

$$\sum_{i=1}^{m} (\underline{w} \cdot \underline{x}_i + b) - y_i)^2$$

Constrained Optimization

$$\min_{\underline{x} \in C} F(x)$$

$$\min \frac{1}{2} \|\underline{w}\|^2$$
$$\text{s.t. } \forall i: y_i(\underline{w}\underline{x}_i + b) \geq 1$$

→

Computing SVMs

$$\max_{\alpha_1 \cdots \alpha_m} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i (\underline{x}_i \cdot \underline{x}_j) y_j \alpha_j$$
$$\text{s.t. } \sum_{i=1}^{m} \alpha_i y_i = 0. \quad \forall i: \alpha_i \geq 0$$

↓

$$\max_{\alpha_1 \cdots \alpha_m} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i (\underline{x}_i \cdot \underline{x}_j) y_j \alpha_j$$
$$\text{s.t. } \sum_{\text{positive class } i} \alpha_i = \sum_{\text{negative class } j} \alpha_j \qquad \forall i: \alpha_i \geq 0$$

SMO

- α
- Pick α_i, α_j
- Modify together to maintain positive equality
- improve objection function

$$\max_{\alpha_i, \alpha_j} [A + \alpha_i + \alpha_j]$$

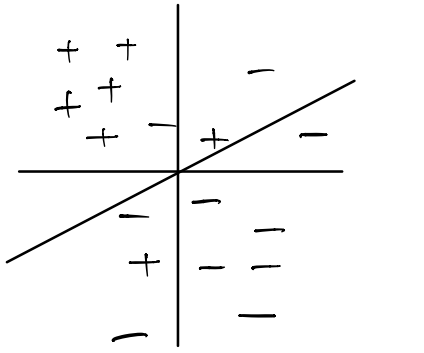
Logistic Regression

$F(x)$ = Probability that $\underline{x}$ belongs to class +1

$$F(\underline{x}) = \frac{1}{1 + e^{-(\underline{w}\underline{x} + b)}}$$

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

$\{(\underline{x}_i, y_i)\}$ $y_i \in \{0, 1\}$.



$$\text{Error}_S(\underline{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (y_i - F(\underline{x}_i))^2$$

$$lik(y_1 \cdots y_N | x_1 \cdots x_n) = \prod_{i=1}^{N} F(\underline{x}_i)^{y_i} (1 - F(x_i))^{1-y_i}$$

$$\downarrow$$

$$\ln lik(y_1 \cdots y_N | x_1 \cdots x_N)) = \sum_{i=1}^{N} \ln F(\underline{x}_i)^{y_i} (1 - F(\underline{x}_i))^{1-y_i}$$

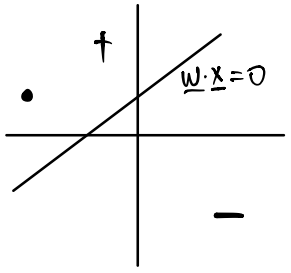
$$\min_{\underline{w}, b} \frac{1}{N} \sum_{i=1}^{N} -[y_i \ln F(\underline{x}_i) + (1-y_i) \ln (1 - F(\underline{x}_i))]$$

Logistic Error $-y_i \ln F(x_i) - (1-y_i)\ln(1-F(x_i))$

Logistic Regression

$F_w(x) = \sigma(w \cdot x)$

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



$$Err(w) = \frac{1}{m}\sum_{i=1}^{m}\left[-y_i \ln F_w(x_i) - (1-y_i)\ln(1-F_w(x_i))\right]$$

$\min_w Err(w)$ $\quad w_0 =$ Some guess

At time t, pick x_i, y_i

$w_{t+1} = w_t - \alpha_t \nabla_w Err_i(w_t)$

$$\frac{\partial Err_i}{\partial w_j} = -y_i \cdot \frac{\frac{\partial F(x_i)}{\partial w_j}}{F(x_i)} - (1-y_i)\left[\frac{-\frac{\partial F(x_i)}{\partial w_j}}{1-F(x_i)}\right]$$

$$= \left[-y_i \frac{1}{F(x_i)} + (1-y_i)\frac{1}{1-F(x_i)}\right]\frac{\partial F(x_i)}{\partial w_j}$$

$$\frac{\partial F(x)}{\partial w_j} = \sigma'(w,x)\frac{\partial}{\partial w_j}[w,x]$$

$$= \sigma'(w,x)\cdot x_j$$

$\sigma'(z) = \sigma(z)(1-\sigma(z))$

$\sigma'(w,x) = \sigma(w,x)(1-\sigma(w,x))$

$= F(x)(1-F(x))$

$\nabla Err_i = [F(x_i) - y_i]x_i$

SGD on Logistic Reg

$\underline{w}_0$: Some guess

At the time t. pick (x_i, y_i)

$$\underline{w}_{t+1} = \underline{w}_t - \alpha_t [F(\underline{x}_i) - y_i] \underline{x}_i$$

$$F(\underline{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2 + w_{11} x_1^2 + w_{12} x_1 x_2 + w_{22} x_2^2)}}$$

Multiclass Problems

$\{(x_i, y_i)\}$ $y \in \{1, 2, \cdots, C\}$

probability of F in class 1.

$y = 3$

$\bar{F}(\underline{x}) = (\bar{F}_1(\underline{x}), \bar{F}_2(\underline{x}) \cdots, \bar{F}_C(\underline{x}))$

$y = (0, 0, 1, 0, 0, 0, 0)$

cross entropy loss

$$Err(\bar{F}) = \frac{1}{m} \sum_{i=1}^{m} [-\sum_{c=1}^{C} y_i \ln \bar{F}_c(\underline{x}_i)]$$

$$-y_i \ln \bar{F}(\underline{x}_i) - (1 - y_i) \ln (1 - \bar{F}(\underline{x}_i))$$

$\{\underline{w}_1, \underline{w}_2, \cdots, \underline{w}_C\}$ $\quad \bar{F}_c(\underline{x}) \geq 0$

$$\bar{F}_c(\underline{x}) = \frac{e^{\underline{w}_c \cdot \underline{x}}}{\sum_{c'=1}^{C} e^{\underline{w}_{c'} \cdot \underline{x}}} \qquad \sum_c \bar{F}_c(\underline{x}) = 1$$

$\{0.1, 0.7, 0.2\}$

$\{0.1, 0.7, 0.93$ $\bar{F}_c(\underline{x}) = \frac{1}{1 + e^{-\underline{w}_c \cdot \underline{x}}}$ ~~$\sum_c \bar{F}_c(\underline{x}) = 1$~~

can in multiple class.

$C = 2$. $\bar{F}(x) = (e^{\underline{w}_1 \underline{x}} \quad e^{\underline{w}_2 \underline{x}})$

$$\left(\frac{e^{\underline{w}_1 \underline{x}}}{e^{\underline{w}_1 \underline{x}} + e^{\underline{w}_2 \underline{x}}}, \frac{e^{\underline{w}_1 \underline{x}}}{e^{\underline{w}_1 \underline{x}} + e^{\underline{w}_2 \underline{x}}} \right)$$

$$= \left(\frac{1}{1 + e^{(\underline{w}_2 - \underline{w}_1)\underline{x}}}, 1 - \frac{1}{1 + e^{(\underline{w}_2 - \underline{w}_1)\underline{x}}} \right)$$

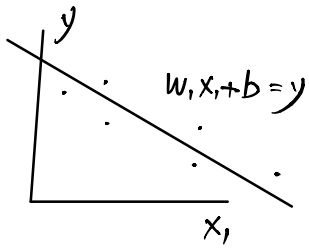
Linear Regression.

$\{(\underline{x}_i, y_i)\}_{i=1}^m \quad y_i \in \mathbb{R}$. Find $\underline{w}$.

s.t. $\underline{w} \cdot \underline{x}_i$ matches y_i.

$$err(\underline{w}) = \frac{1}{m} \sum_{i=1}^m (\underline{w} \cdot \underline{x}_i - y_i)^2$$

$$\min_{\underline{w}} \frac{1}{m} \sum_{i=1}^m (\underline{w} \cdot \underline{x}_i - y_i)^2$$



linear regression → regression problem

logistic regression → classification problem

Linear Regression

$\{(\underline{x}_i, y_i)\}_{i=1}^m \qquad f(\underline{x}) = \underline{w} \cdot \underline{x}$

$$err_{train}(\underline{w}) = \sum_{i=1}^m (\underline{w} \cdot \underline{x}_i - y_i)^2 \rightarrow \min_{\underline{w}} \|X\underline{w} - \underline{y}\|^2$$
$$= \|X\underline{w} - \underline{y}\|^2$$

$$F(\underline{w}) = \|X\underline{w} - \underline{y}\|^2 = (X\underline{w} - \underline{y})^T (X\underline{w} - \underline{y})$$

$$\nabla F(\underline{w}) = 0 \quad = (\underline{w}^T X^T - \underline{y}^T)(X\underline{w} - \underline{y})$$

$$= \underline{w}^T X^T X \underline{w} - \underline{w}^T X^T \underline{y} - \underline{y}^T X \underline{w} + \|\underline{y}\|^2$$

$$= \underline{w}^T X^T X \underline{w} - 2\underline{w}^T X^T \underline{y} + \|\underline{y}\|^2$$

$$\nabla f = 2X^T X \underline{w} - 2X^T \underline{y} = 0$$

$$X^T X \underline{w} = X^T \underline{y}$$

$$\underline{w}^* = [X^T X]^{-1} X^T \underline{y}$$

small noise vector

$$\underline{y} = X\underline{w} + \underline{\varepsilon}$$

$$\underline{w}^* = [X^T X]^{-1} X^T (X\underline{w} + \varepsilon)$$

$$= [X^T X]^{-1} (X^T X \underline{w} + X^T \underline{\varepsilon})$$

final model with noise.

$$= \underline{w} + [X^T X]^{-1} X^T \underline{\varepsilon}$$

small noise in model.

if $\underline{y} = X\underline{w} + \underline{\varepsilon}$

$$\underline{w}^* = \underline{w} + [X^T X]^{-1} X^T \underline{\varepsilon} \qquad \underline{\varepsilon} \sim N(\underline{0}, \sigma^2 I)$$

$$M\underline{\varepsilon} \sim N(0, MM^T \sigma^2 I)$$

$$\varepsilon \sim N(0, 1)$$

$$m\varepsilon \sim N(0, m^2)$$

$$w^* \sim \underline{w} + [X^T X]^{-1} X^T N(0, \sigma^2 I)$$

$$\sim \underline{w} + N(\underline{0}, \sigma^2 [X^T X]^{-1} X^T X [X^T X]^{-1})$$

$$w^* \sim \underline{w} + N(\underline{0}, [X^T X]^{-1} \sigma^2)$$

$$Cov(x, y) = E[(x - \mu_x)(y - \mu_y)]$$

$(x_1, \ldots, x_N)$

C: Cov matrix

$C_{ij} = Cov(x_i, x_j)$

$$\underline{\varepsilon} \sim N(\underline{0}, \sigma^2 I)$$

$\Sigma = X^T X$ symmetric matrix $\rightarrow \Sigma = Q D Q^T$

$w^* \sim \underline{w} + N(\underline{0}, \Sigma^{-1}\sigma^2)$

Q: eigenvectors of Σ

D: diagonal matrix of eigenvalues of Σ.

$\Sigma^{-1} = Q D^{-1} Q^T$

$w^* \sim \underline{w} + N(\underline{0}, Q D^{-1} Q^T \sigma^2)$

$w^* \sim \underline{w} + \sigma Q N(\underline{0}, D^{-1})$ $\rightarrow$ $z \sim N(0,1)$, $\sigma z \sim N(0, \sigma^2)$

$$D^{-1} = \begin{bmatrix} 1/\lambda_1 & & 0 \\ & 1/\lambda_2 & \\ 0 & & \ddots \; 1/\lambda_k \end{bmatrix}$$

$1/\lambda_1$ → variance in first component.

$\Sigma = Q D Q^T$

$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$

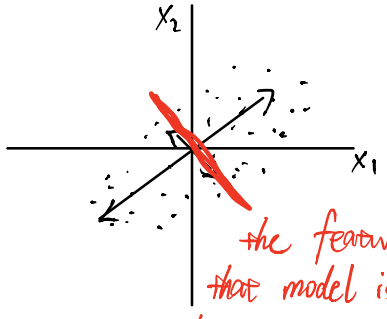
The smaller the eigenvalue of Σ.
the larger the noise in $\underline{w}^*$.

$1/\lambda_1$ very small, $1/\lambda_k$ very large.

$$K(\Sigma) = \frac{\text{largest eigenvalue}}{\text{smallest eigenvalue}}.$$

if any λ_i is very small, huge Error on $\underline{w}^*$.

What does it mean if eigenvalues of $\Sigma = X^T X$ are small?

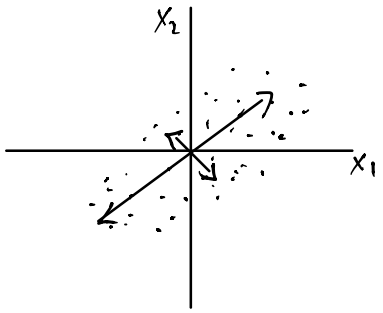


the feature that model is least accurate.

largest eigenvalue capture the most spread direction of the data.

smallest eigenvalue capture the lease spread direction of the data

The geometry of data is directly related to the quality of model.



find the largest projection direction of the data.

Find direction $\underline{v}$, to maximize

$$\sum_{i=1}^{m} (\underline{v} \cdot \underline{x}_i)^2$$

$$\|X\underline{v}\|^2$$

$$\underline{v}^T(X^TX)\underline{v}$$

$$\max_{\underline{v}} \ \underline{v}^T X^T X \underline{v}$$

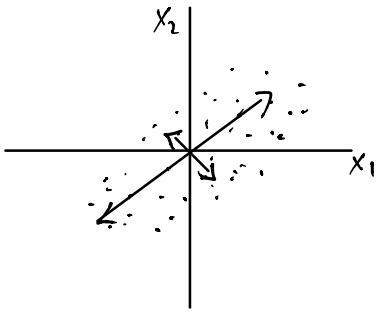
$$\text{s.t. } \|\underline{v}\| = 1$$

Linear Regression

$$\min_{\underline{w}} \sum_{i=1}^{m} (\underline{w} \cdot \underline{x}_i - y_i)^2$$

$$\min_{\underline{w}} \|X\underline{w} - \underline{y}\|^2$$

$$w^* = [X^TX]^{-1}X^Ty$$



$$y = X\underline{w} + \underline{\varepsilon} \qquad \underline{\varepsilon} \sim N(\underline{0}, \sigma^2 I)$$

$$w^* \sim \underline{w} + \sigma Q N(\underline{0}, D^{-1})$$

$\underline{w}$: true model. $\sigma Q N(\underline{0}, D^{-1})$: noise.

$$\Sigma = X^TX$$

Q: eigenvector of Σ.

D: eigenvalues of Σ.

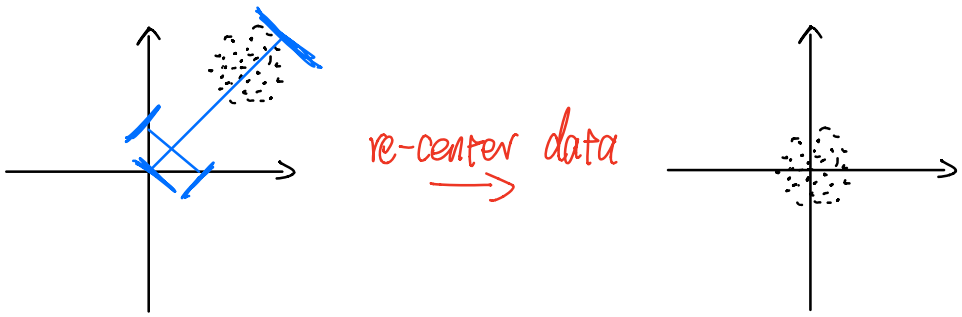
$$D^{-1} = \begin{bmatrix} 1/\lambda_1 & & 0 \\ & 1/\lambda_2 & \\ 0 & & \ddots \ 1/\lambda_k \end{bmatrix}$$

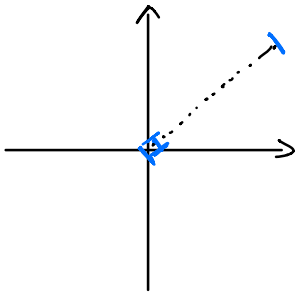
$$\frac{\text{Largest Error}}{\text{Smallest Error}} = \frac{1/\lambda_k}{1/\lambda_1} = \frac{\lambda_1}{\lambda_k}$$

$$K(\Sigma) = \frac{\lambda_1}{\lambda_k}$$

Sample complexity

$m \geq O(k \frac{H(\varepsilon)}{\varepsilon^2}) \Rightarrow err(\underline{w}^*) \leq \varepsilon^2$



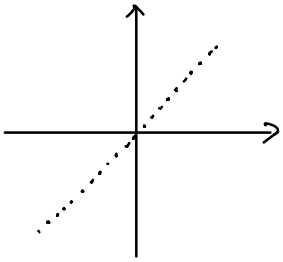


Error is massive.

Dimensionally Reduction

$\Sigma = X^T X$ ← invertable.

$\Sigma \underline{w} = X^T y$



$x_2 = \alpha x_1$

$\min_w \sum_{i=1}^{m} (\underline{w} x + y_i)^2 + \lambda \|\underline{w}$ ~~without~~ the bias $\|_2^2$

$\underline{w}_\lambda^* = [X^T X + \lambda I]^{-1} X^T y$

Linear Regression

$$\min_{\underline{w}} \|X\underline{w} - \underline{y}\|_2^2$$

$$\underline{w}^* = [X^T X]^{-1} X^T \underline{y}$$

Ridge: $\min_{\underline{w}} \|X\underline{w} - \underline{y}\|_2^2 + \underline{\lambda}\|\underline{w}\|_2^2$

$\underline{\lambda}$: regulation param

$X \rightarrow$ mean centered

$\underline{y} \rightarrow$ mean centered

Feature Selection

$y = \sim + \underline{10^{-24} X_5} + \sim$

X_5 is not important for our model.

Pruning - Cut out data.

$S \subseteq \{1 \dots K\}$.

$\min_{\underline{w}_S} \|X_S \underline{w}_S - \underline{y}\|_2^2$ Find S to minimize this.

subsets are too much!

Lasso: $\min_{\underline{w}} \|X\underline{w} - \underline{y}\|_2^2 + \underline{\lambda}\|\underline{w}\|_1$ Automatically pruned.

$$\|\underline{w}\|_1 = \sum |w_i|$$

$$\min_w (w - \alpha)^2 + \lambda|w|$$

$$w^* = \begin{cases} \alpha - \lambda/2 & \alpha > \lambda/2 \\ \alpha + \lambda/2 & \alpha < -\lambda/2 \\ 0 & \alpha \in [-\frac{\lambda}{2}, \frac{\lambda}{2}] \end{cases}$$

$\underline{w} = (1, 0, 0, 0, \cdots, 0)$ ← sparse vector

$\underline{v} = \left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \cdots \frac{1}{\sqrt{N}}\right)$ ← even vector.

$\|w\|_2^2 = 1 \qquad \|\underline{w}\|_1 = 1$

$\|v\|_2^2 = 1 \qquad \|v\|_1 = \sqrt{N}$

Lasso: $\min_{\underline{w}} \|X\underline{w} - \underline{y}\|^2 + \lambda\|\underline{w}\|_1$

support: X_i that actually matter to y.

Conditions for support recovery

$y = \underline{X}_S \cdot \underline{w}_S + \underline{I}$

↑ random noise

- Mutual Incoherence

- Minimum Spread: $\lambda_{min}(X_S^T X_S)$ sufficient large.

- Minimum signal: For any $j \in S^*$ $|w_j|$ can't be too small.

Boosting

$f_1(x), f_2(x) \cdots f_k(x)$ ← why multiple model.

$f(x) = \text{majority}(f_1(x), f_2(x) \cdots f_k(x))$ | different hypothesis space

Ada boost → mega model. train new model from old ones. adjustment over old model.

$F_1(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t f_t(x))$ $\quad Q_T(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$

sign $\alpha_1, \cdots, \alpha_T$, $f_1 \cdots f_T$

find α_{T+1}, f_{T+1}

$F_{T+1}(x) = \text{sign}(\sum_{t=1}^{T+1} \alpha_t f_t(x))$ ↓

$F_T(x) = \text{sign}(Q_T(x))$

$F_{T+1}(x) = \text{sign}(Q_T(x) + \alpha_{T+1} f_{T+1}(x))$

Error func for binary classification → is discrete.

$$err_{train}(f) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{f(x_i) \neq y_i\}.$$

$$E_T = \sum_{(x,y)} e^{-yQ_T(x)}$$ → correct: $E_T \geq 1$, incorrect: $E_T \leq 1$

$$E_{T+1} = \sum_{(x,y)} e^{-y(Q_T(x) + \alpha_{T+1} f_{T+1}(x))}$$

$$= \sum_{(x,y)} e^{-yQ_T(x)} e^{-\alpha_{T+1} f_{T+1}(x) y}$$

$$= \left(\sum_{f_{T+1}(x) = y} e^{-yQ_T(x)}\right) e^{-\alpha_{T+1}} + \left(\sum_{f_{T+1}(x) \neq y} e^{-yQ_T(x)}\right) e^{\alpha_{T+1}}$$

what α_{T+1} minimize E_{T+1}

$$G_{T+1} = \sum_{f_{T+1}(x)=y} e^{-yQ_T(x)}$$

$$B_{T+1} = \sum_{f_{T+1}(x)\neq y} e^{-yQ_T(x)}$$

$$\bar{E}_{T+1} = G_{T+1} e^{-\alpha_{T+1}} + B_{T+1} e^{\alpha_{T+1}}$$

set derivative = 0 and solve

$$\alpha_{T+1} = \frac{1}{2} \ln\left(\frac{G_{T+1}}{B_{T+1}}\right)$$

$$\bar{E}_{T+1} = 2\sqrt{G_{T+1} B_{T+1}}$$

$$\bar{E}_{T+1} = \sum_{(x,y)} e^{-y(Q_T(x) + \alpha_{T+1} f_{T+1}(x))}$$

$$\bar{E}_T = \sum_{(x,y)} e^{-yQ_T(x)}$$

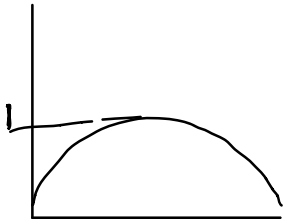
$$= G_{T+1} + B_{T+1}$$

$$\bar{E}_{T+1} = 2\sqrt{(\bar{E}_T - B_{T+1}) B_{T+1}}$$

$$\bar{E}_{T+1} = 2\bar{E}_T \sqrt{\left(1 - \frac{B_{T+1}}{\bar{E}_T}\right) \frac{B_{T+1}}{\bar{E}_T}}$$

$$Q_T(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$$

$$2\sqrt{(1-p)p}$$



To minimize $\bar{E}_{T+1}$. find f_{T+1} to minimize $\frac{B_{T+1}}{\bar{E}_T}$.

$$\frac{B_{T+1}}{E_T} = \frac{\sum_{(x,y)} e^{-yQ_T(x)} \cdot \mathbb{1}\{f_{T+1}(x) \neq y\}}{\sum_{(x,y)} e^{-yQ_T(x)}}$$

weighted error function.

$$= \frac{\sum_{(x,y)} w(x,y)\, \mathbb{1}\{f_{T+1}(x) \neq y\}}{\sum_{(x,y)} w(x,y)} = err_w^{Train}(f_{T+1})$$

minimize $w(x,y) \to e^{-yQ_T(x)}$

Ada Boost

- Initiate $w_1(x,y) = 1$. for all $(x,y) \in$ Data
- At time t, find f_t to minimize $err_{w_t}^{Train}(f)$
- Set $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - err_{w_t}^{Train}(f_t)}{err_{w_t}^{Train}(f_t)}\right)$
- $w_{t+1}(x,y) = w_t(x,y) e^{-\alpha_t y f_t(x)}$ $err_w^{Train}(f) = \frac{\sum_{(x,y)} w(x,y)\, \mathbb{1}\{f(x) \neq y\}}{\sum_{(x,y)} w(x,y)}$
- $G(f_1 \cdots f_T)$
- $F(x) = sign\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right)$

As long as $err_{w_t}^{Train}(f_t) < 1/2$. we have improvement.

If $err_{w_t}^{Train}(f_t) \leq \frac{1}{2} - \gamma$. for $t = 1 \cdots T$.

$err_{train}(F) \leq e^{-2T\gamma^2}$

$f_1 \cdots f_T$ — different hypothesis space no matter what hypothesis space for new model. random pick?

Boosting

$f_1, f_2 \dots f_T$

$F(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right)$

Gradient Boosting

$$Err = \sum_{i=1}^{m} L(y^i, F(x_i))$$

$x = 4$

$$F(4)_{raw} = F(4) - \lambda \frac{\partial Err}{\partial F(4)}$$

Residuals.

$$r_1 = \frac{\partial Err}{\partial F(x_1)}$$

$$r_2 = \frac{\partial Err}{\partial F(x_2)}$$

$$\vdots$$

$$r_m = \frac{\partial Err}{\partial F(x_m)}$$

Fit a new model

$x_i \rightarrow r_i$

$\tilde{F}(x_i) = r_i$

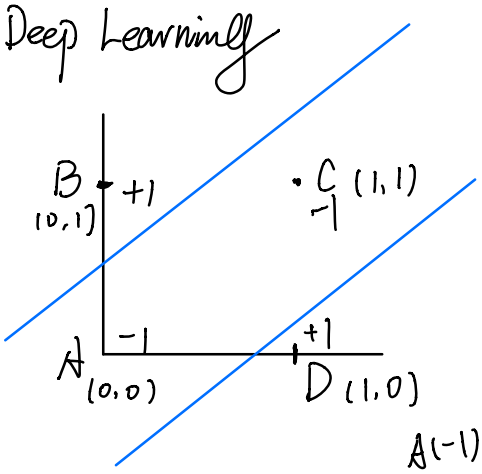
$F_{meta}(x) = F(x) - \lambda \tilde{F}(x)$

- Train f to minimize $\sum_i L(y_i, f(x_i))$
- Set $\lambda_1 = 1$
- At time $t \geq 1$, $F_t = \lambda_1 f_1 + \lambda_2 f_2 + \cdots + \lambda_t f_t$
  - Define residuals $r_t^i = \frac{\partial}{\partial F_t(x_i)} [L(y_i, F_t(x))]$
  - Fit f_{t+1} to $\{(x_i, r_t^i)\}$

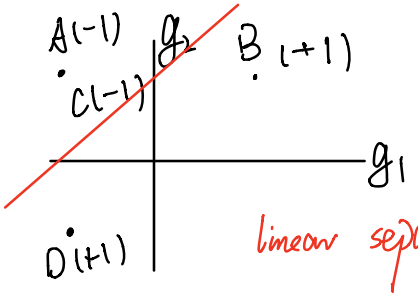
- Find λ_{t+1} to minimize $\sum_i L(y_i, F_t(x_i) - \lambda f_{t+1}(x_i))$
- $F_{t+1} = F_t - \lambda_{t+1} f_{t+1}$

XGBoost ↙ apply on decision tree.

Deep Learning



	g_1	g_2
A	−	+
B	+	+
C	−	+
D	−	−



↙ new feature space.

linear separable.

$g_1(x_1, x_2) = \text{sign}(ax_1 + bx_2 + c)$

$g_2(x_1, x_2) = \text{sign}(dx_1 + ex_2 + f)$

compute features over features

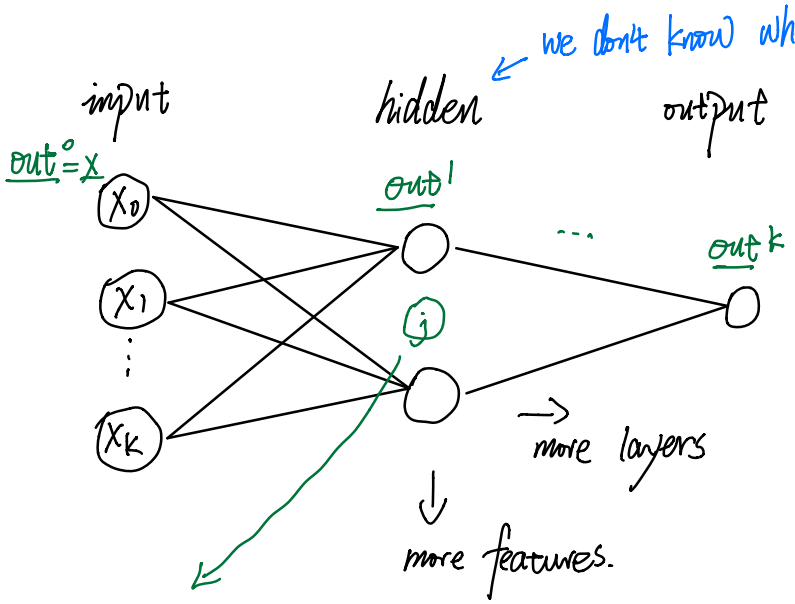
$F(x_1, x_2) = \text{sign}(hg_1(x_1, x_2) + lg_2(x_1, x_2) + p)$ make computer pick the features.

Find $a, b, c, d, e, f, h, l, p$ → fit data NOT find features ourselves.

Neural Networks

$$f(x_1, x_2) = \sigma(\alpha_1 + \alpha_2 \sigma(\alpha_4 + \alpha_5 x_1 + \alpha_6 x_2) + \alpha_3 \sigma(\alpha_7 + \alpha_8 x_1 + \alpha_9 x_2))$$

$$\sigma(z) = \text{sign}(z)$$



highly non-linear function.

$$out_j^t = \sigma(w^{t-1}_{ij} \cdot out^{t-1})$$

dot product.

$out^0 = x$

out^K

Activation Functions.

$$\sigma(z) = \text{sign}(z)$$

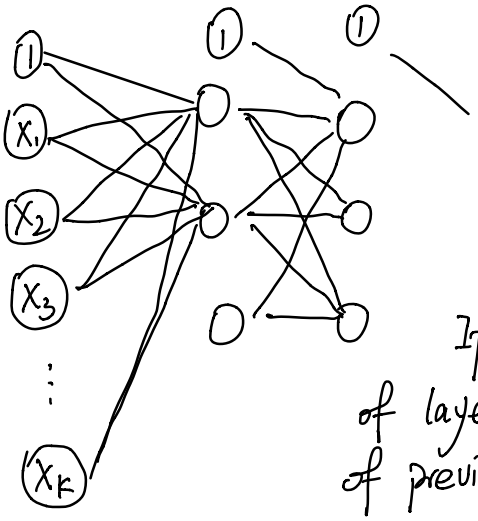
$$\sigma(z) = \text{sigmoid} = \frac{1}{e^z + 1}$$

$$\sigma(z) = \tanh(z)$$

ReLU

$$\sigma(z) = \max(0, z)$$

computational quite good.



If no activation function, backwards of layers are just linear combination of previous layers.

A deeper network allow us compute less features in single layer. This pratically give good result.

Regression

$$\text{Error} = \sum_{i=1}^{m} \| \underline{out}^K(x_i) - y_i \|_2^2$$

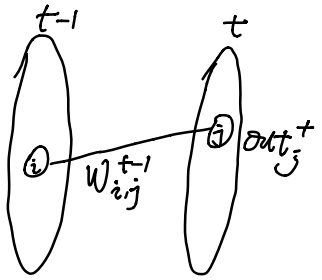
Classification.

$$out_c^K = \frac{e^{out_c^{K-1}}}{\sum_{c_1} e^{out^{K-1}_{c_1}}}$$

$$L = \sum_{i=1}^{m} - \left[\sum_c y_c^i \ln out_c^K(x_i) \right]$$

$$W_{new} = W_{old} - \alpha \nabla_{W_{old}} L$$

$$\text{Goal}: \frac{\partial L}{\partial W_{i,j}^{t-1}} = \frac{\partial L}{\partial out_j^t} \cdot \frac{\partial out_j^t}{\partial W_{i,j}^{t-1}}$$



$$out_j^t = \sigma(\underline{w}^{t-1}(j) \cdot \underline{out}^{t-1})$$

$$\frac{\partial out_j^t}{\partial W_{i,j}^{t-1}} = \frac{\partial}{\partial W_{i,j}^{t-1}} \left[w^{t-1}(j) \cdot out^{t-1} \right] \cdot \sigma(\underline{w}^{t-1}(j) \cdot \underline{out}^{t-1})$$

$$\frac{\partial out_j^t}{\partial W_{i,j}^{t-1}} = out_i^{t-1} \cdot \sigma(\underline{w}^{t-1}(j) \cdot out^{t-1})$$

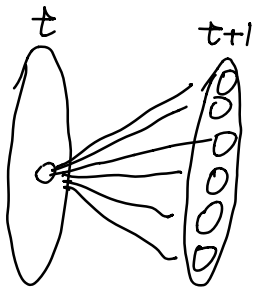
$$\frac{\partial L}{\partial W_{i,j}^{t-1}} = \left(\frac{\partial L}{\partial out_j^t} \right) \sigma(w^{t-1}(j) \cdot out^{t-1}) \, out_j^{t-1}$$

$t = K$: compute directly from L

$t < K$.

$$\frac{dL}{dout_j^t} = \sum_{\text{nodes } k \text{ in the next layer}} \left(\frac{dL}{dout_k^{t+1}}\right)\left(\frac{dout_k^{t+1}}{dout_j^t}\right)$$

back-propagation



$$out_k^{t+1} = \sigma(\underline{w}^t(k) \cdot out^t) \qquad \frac{dout_k^{t+1}}{dout_j^t} = w_{j,k}^t \sigma(\sim)$$

back-propagation → gradient descent. → minimize Loss.