- Supervised Learning
  - Regression

Machine learning (Defn)

H0 : Null hypothesis (applying some tests)
H1 : Alternate hypothesis

→ LOOCV, hold-out, k-fold.

29 Jan 2024 LAB

+ uciml repo
x wget
x RFE
x Random Over sampling , SMOTE

→ normalizing and not normalizing
→ oversampling and ~ oversampling

30 Jan 2024

feature pool → reduced pool
(features matrix)

Classification

→ pre-determined classes exist.
→ we need to find a function f (hypothesis) which maps

Regression

whenever we try to fit a model and we fix some kind of condition (like using only depth = 5 for decision trees) this results in some kind of bias.

linear regression → residual error

hypothesis : given some θ

$h_\theta(\hat{x})$ ← label
↑ hypothesis
↑ parameter

slope
$y = mx + c$ ← intercept
$\beta_1 x + \beta_0$
weight → $wx + b$ ← bias

weights and bias are parameters of the model and we are interested in learning these parameters.

Inductive Bias

absolute error $\Rightarrow |h_0(\hat{x}) - y|$

$\Rightarrow |\hat{y} - y|$

E (squared error) $= (\hat{y} - y)^2$ (Because $|\hat{y} - y|$ is not differentiable at 0)

Learning is acquiring general concepts from specific training examples.

Concept learning if we have a set of examples we want to guess the label of an object using some attributes.

Task: Task is to learn to predict the value of boolean valued attribute (family car). based on the values of its other attributes.

Training set X.

Book: Tom mitchell.

$$X = \{x^t, r^t\}_{t=1}^{N}$$

$$r = \begin{cases} 1 & \text{if } x \text{ is +ve} \\ 0 & \text{if } x \text{ is -ve} \end{cases}$$

$H \rightarrow$ set of all legal hypothesis.

$h \in H$ ↑ some hypothesis.

$h(x) = c(x)$ ↑ hypothesis (approximate) ↑ concept.

hypothesis is consistent.

Concept learning. Inferring a boolean-valued function from training examples of its input and output.

$|I| = |X| = 2^N$ (size of instance space)

x_1	x_2	h_0	h_1	h_2	h_3
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	1	1
1	1	0	1	0	1

No. of legal hypothesis $\rightarrow 2^{|I|}$.

? → any value is acceptable. / don't care
specify required value for the attribute.
"ϕ" → no value is acceptable. [any ϕ leads to -ve example].

most general hypothesis → ⟨ ?, ?, ?, ? ⟩
most specific hypothesis → ⟨ ϕ, ϕ, ϕ, ϕ ⟩

1st feb '24

Pg - 27

As long as we assume that the hypothesis space H contains a hypothesis that describes the true target concept c and that the training

Disadvantage of Finds → finds only 1 consistent → returns true (label) everytime hypothesis.
→ there can be some records having noise or errors.
inconsistent sets of training examples

conjuction of monomials → {Sky = sunny} ∧ {AirTemp = warm} ∧ . . . .

syntactically distinct hypothesis.

Size of hypothesis space.

distinct instances → 3 · 2 · 2 · 2 · 2 · 2 = 96
5 · 4 · 4 · 4 · 4 · 4 = 5120

syntactically distinct hypothesis

F1	F2
A	X
B	Y

⇒ 2 X 2

3 ⟶ 3 + (2)

because of ? and ϕ.

# of syntactically distinct hypothesis.

(?, X), (?, Y), (ϕ, X), (ϕ, Y), (A, ?), (B, ?), (?, ?), (ϕ, ϕ),
(A, ϕ), (B, ϕ), (A, X), (A, Y), (B, X), (B, Y), (?, ϕ), (ϕ, ϕ)

semantically distinct hypothesis ⇒ 1 + (4 X 3 X 3 X 3 X 3 X 3)

Most specific remains.

subtracting 1 from all because of ϕ.

because this ⟨ϕ, ϕ⟩ represent all the -ve instances.

3 X 3 X 2 X 2 X 2 ⇒
(syntactic) 5 X 5 X 4 X 4 X 4 ⇒
(semantic) ⇒ 1 + (4 X 4 X 3 X 3 X 3) ⇒

consistent : A hypothesis h is consistent with a set of training examples D if and only if $h(x) = c(x)$ for each example $\langle x, c(x) \rangle$ in D.

consistent

BOOK example.

since ? matches with A and x matches with X, ∴ the hypothesis predicts this instance is tve and the given class is also tve.

since ? matches with B and x does not matches with Y, ∴ the hypothesis predicts this instance as -ve but the given class is +ve. ∴ inconsistent.

5th Feb '24

candidate elimination algorithm → version space.

Ques check wether $H1 = \langle ?, ?, No, ?, Many \rangle$ is consistent or not? → consistent

Example	citations	size	in Library	Price	editors	
1	some	small	No	Affordable	one	No
2	many	big	No	expensive	many	Yes.

$\langle ?, ?, no, ?, ? \rangle$

↳ not consistent

S = most specific boundary / maximally specific hypothesis.
G = most general boundary. / maximally general hypothesis

+ve class $\xrightarrow{\text{action}}$ G
-ve class $\xrightarrow{\text{action}}$ S

candidate elimination

Example	sky	AirTemp	humidity	Wind	water	Forecast	enjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	yes
2	Sunny	Warm	High	Strong	Warm	same	Yes
3	Rainy	cold	High	strong	Warm	Change	No
4	Sunny	Warm	High	strong	cool	change	Yes

S_0 : ⟨ϕ, ϕ, ϕ, ϕ, ϕ, ϕ⟩

S_1 : ⟨Sunny, warm, Normal, Strong, warm, same⟩

$S_3 = S_2$: ⟨'', '', ?, '', '', ''⟩

S_4 : ⟨Sunny, warm, ?, Strong, ?, ?⟩

⟨Sunny, ?, ?, Strong, ?, ?⟩

⟨Sunny, warm, ?, ?, ?, ?⟩

⟨?, warm, ?, Strong, ?, ?⟩

here we list- all of the possibilities → G_4 : ⟨Sunny, ?, ?, ?, ?, ?⟩, ⟨?, warm, ?, ?, ?, ?⟩

G_3 : ⟨Sunny, ?, ?, ?, ?, ?⟩, ⟨?, warm, ?, ?, ?, ?⟩ ← taking attributes which will result in a negative class.

$G_2 = G_1$: ⟨?, ?, ?, ?, ?, ?⟩

G_0 : ⟨?, ?, ?, ?, ?, ?⟩

for +ve classes we were checking here first

Ques.

Example	Size	Color	Shape	Class
1	Big	Red	Circle	No
2	Small	Red	Triangle	No
3	Small	Red	Circle	Yes
4	Big	Blue	Circle	No
5	Small	Blue	Circle	Yes

06 Feb' 24

Supervised machine learning - andrews ng

linear regression in one variable.

training set (features, targets)

↓

learning algo

↓

$x \rightarrow [f] \rightarrow \hat{y}$

f : function or hypothesis

$\hat{y}$: predicted values estimated

error / residual → $\hat{y} - y$ → $|\hat{y} - y|$ → $(\hat{y} - y)^2$

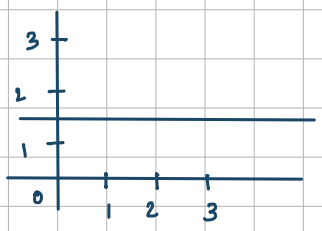
$\hat{y} - y$ can be misleading

$|\hat{y} - y|$ differentiation not defined.

How to represent 'f' in linear regression.

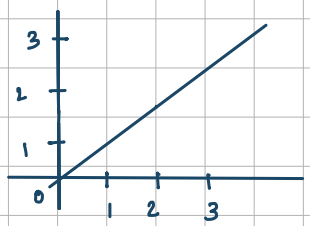
$f_{w,b} = wb + x$ or $f = wb + x$ $\rightarrow wx + b$ (w, b = parameters)

Reason to use linear regression: we want a simple function, complex functions are difficult to analyse.



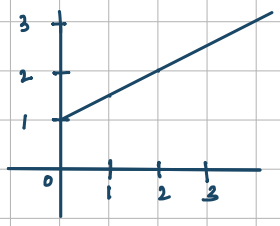
$f(x) = 0x + 1.5$

Case I:
$w = 0$
$b = 1.5$



$f(x) = \frac{1}{2}x + 0$

Case II:
$w = 0.5$
$b = 0$



Case III
$w = 0.5$
$b = 1$
$f(x) = \frac{1}{2}x + 1$

we want to find w, b so that $\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

$$\hat{y}^{(i)} = \underbrace{f_{w,b}(x^{(i)}) = wx^{(i)} + b}_{\text{model}}$$

Cost Function

it tells us how well the model is doing so that we can adjust the parameters to improve our model (if required).

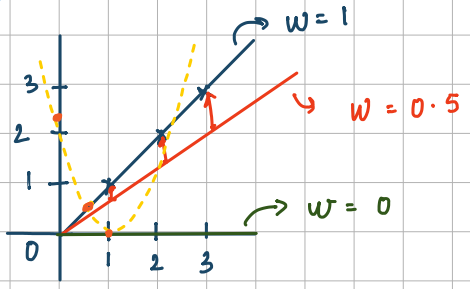
for m examples:

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

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

or

Our goal is to minimize this cost. w.r.t w, b.



$f_w(x) = x$

$w = 1$
$b = 0$

interactive plot.

$f(x) = y$ $\quad f(1) = 1$, $f(2) = 2$, $f(3) = 3$

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

$$J(1) = \frac{1}{2 \times 3} \left((1-1)^2 + (2-2)^2 + (3-3)^2 \right) = 0$$

$$J(0.5) = \frac{1}{2 \times 3} \left((0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right)$$

$$\approx 0.58$$

$$J(0) = \frac{1}{2 \times 3} \left((0-1)^2 + (0-2)^2 + (0-3)^2 \right)$$

$$\Rightarrow \frac{1}{6} \times 14 \approx 2.3.$$

when slope = 0, cost function minimum.

Example	Size	Color	Shape	Class
1		Red	Circle	No
2	Small	Red	Triangle	No
3	Small	Red	Circle	Yes
4	Big	Blue	Circle	No
5	Small	Blue	Circle	Yes

$$\langle \phi, \phi, \phi \rangle$$

S_0 : $\langle \phi, \phi, \phi \rangle$
S_1 : $\langle \phi, \phi, \phi \rangle$
S_2 : $\langle \phi, \phi, \phi \rangle$
S_3 : $\langle$ Small, Red, Circle $\rangle$

S_4 : < small, Red, Circle >

S_5 : < small, ?, Circle >

G_3 : < Big, ?, Triangle > , < ?, Blue, Triangle >, < small, ?, Circle >

< Big, ?, Triangle >, < ?, Blue, Triangle > ×

G_2 : < small, Blue, ? > × , < small, ?, Circle > < ?, Blue, ? > ×

G_1 : < small, ?, ? > , < ?, Blue, ? > , < ?, ?, Triangle >

G_0 : < ?, ?, ? >

07 Feb '24

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

$$\frac{\partial J(w,b)}{\partial w} = 0 \qquad \frac{\partial J(w,b)}{\partial b} = 0$$

$$\frac{\partial J}{\partial w} = \frac{\partial}{\partial w} \frac{1}{2m} \sum_{i=1}^{m} (wx^{(i)} + b - y^{(i)}) = 0$$

$$= \frac{1}{m} (wx^{(i)} + b - y^{(i)})(x^{(i)}) = 0 \quad \text{—— (1)}$$

$$\frac{\partial J}{\partial b} \frac{1}{2m} \sum_{i=1}^{m} (wx^{(i)} + b - y^{(i)})^2 = 0$$

$$\frac{1}{m} (wx^{(i)} + b - y^{(i)}) = 0 \quad \text{—— (2)}$$

By (2)

$$w(x^{(1)} + x^{(2)} + \dots + x^{(m)}) + b - (y^{(1)} + y^{(2)} + \dots + y^{(m)}) = 0$$

$$m w \bar{x} + \sum_{i=1}^{m} b \; m \bar{y} = 0$$

$$mb = m\bar{y} - mw\bar{x}$$

$$b = \bar{y} - w\bar{x} \quad \text{—— (3)}$$

By (1)

$$\sum_{i=1}^{m} w (x^{(i)})^2 + \sum_{i=1}^{m} b x^{(i)} - \sum_{i=1}^{m} y^{(i)} x^{(i)} = 0$$

$$\sum_{i=1}^{m} w (x^{(i)})^2 + b m \bar{x} - \sum_{i=1}^{m} y^{(i)} x^{(i)} = 0$$

Substitute (3).

$$\sum_{i=1}^{m} w (x^{(i)})^2 + (\bar{y} - w\bar{x}) m \bar{x} - \sum_{i=1}^{m} y^{(i)} x^{(i)} = 0$$

$$w \left(\sum_{i=1}^{m} (x^{(i)})^2 - m\bar{x}^2 \right) - \sum_{i=1}^{m} y^{(i)} x^{(i)} + m \bar{y} \bar{x} = 0 .$$

$$w = \frac{\sum_{i=1}^{m} y^{(i)} x^{(i)} - m\bar{y}\bar{x}}{\sum_{i=1}^{m} (x^{(i)})^2 - m\bar{x}^2}$$

$$= \frac{m\overline{xy} - m\bar{x}\bar{y}}{m\overline{x^2} - m(\bar{x})^2}$$

$$= \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - (\bar{x})^2} \quad \text{— (5)}$$

$$= \frac{\text{cov}(x, y)}{\text{var}(x)}.$$

Gradient Descent

12 Feb '24

Multi-variable Regression

$$y = f(x_1, x_2, x_3, \ldots x_n)$$

x_1 = size in feet^2
x_2 = no. of bedrooms
x_3 = age of the house
x_4 = no. of floors.
x_5 = price (\$)

x_1	x_2	x_3	x_4	x_5
2104	5	45	1	460
1416	3	40	2	232

$j = 1, 2, 3, 4$
$= 1 \ldots 4$
m = no. of training samples.
n = # of features

$x_2^{(2)} = 3$

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

for multiple variable: $\hat{y} = f_{\vec{w},b}(\vec{x}^{(i)}) = \vec{w}\,\vec{x}^{(i)} + b$

$= w_1x_1 + w_2x_2 + w_3x_3 + \ldots + b$

$\vec{x}^{(1)} = [2104 \quad 5 \quad 45 \quad 1]$.

$f_{(w,b)}(x) = 0.1x_1 + 4x_2 + (-2)x_3 + 10x_4 + \boxed{80}$.

for any house which doesn't even consists of any feature this value represents base value of the house (in this case land).

$0.1x_1$ = for every additional feet^2 the price ↑ by 0.1×1000. i.e. \$100.

cost function: $J(\vec{w}, b)$ or $J(w_1, w_2, \ldots w_n, b)$.

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

for features:

$n \geq 2$

repeat until convergence {

$$j=1 \quad w_1 = w_1 - \alpha \frac{\partial}{\partial w_1} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 \right)$$

$$= w_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_1^{(i)}$$

$$j=2, \quad w_2 = w_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_2^{(i)}$$

$$\vdots$$

$$j=n \quad w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_n^{(i)}$$

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

Why feature scaling in multiple-variable regression is important?

size in feet^2 x_1 [200 - 2000]

no. of bedrooms x_2 [0 - 5]

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

$y = \$500,000$ (given).

Case I.

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

$x_1 = 2000$, $x_2 = 5$

$\Rightarrow 50 \times 2000 + (0.1) \times 5 + 50$

$\Rightarrow 100,000 + 0.5 + 50$

$\Rightarrow 100,050.5 \times 1000$

large deviation.

Case II

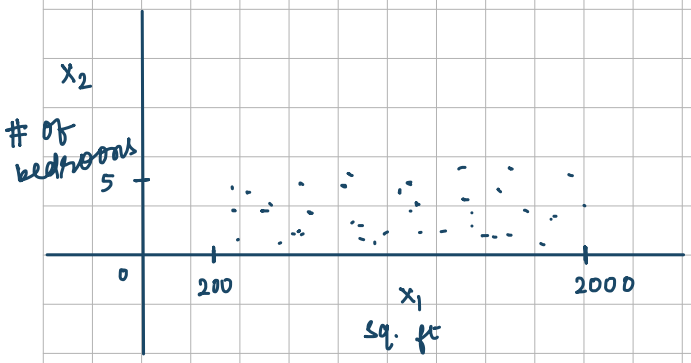
$w_1 = 0.1$, $w_2 = 50$, $b = 50$

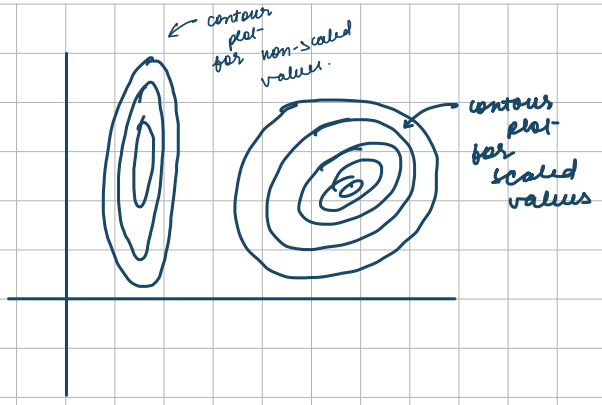
$\Rightarrow 0.1 \times 2000 + 50 \times 5 + 50$

$\Rightarrow 500 \times 1000$

more precise.

w_1 was having ↑ values and assigning $w_1 = 50$ for it resulted in even higher values and the contribution of x_2 was negligible in 1st case.





13 Feb '24

A very small change in w_1 can have a very large impact on the estimated which creates a large impact on the cost.

Because the contours are too tall and somewhat thin, the gradient descent may end up bouncing ~~on~~ back and forth for a long time before it reaches the global minimum.

min - max normalization : $\dfrac{x_i - \min}{\max - \min}$

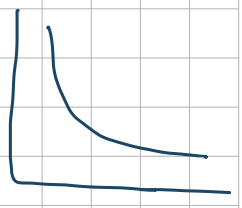
$300 \leq x_1 \leq 2000$

$0.15 \leq x_{1,\text{scaled}} \leq 1$

mean normalization : $\dfrac{x_i - \mu}{\max - \min}$

z - score : $\dfrac{x_i - \mu}{\sigma}$

if the learning curve shows any ↑ in J if the learning rate is chosen or high



for early stopping use epsilon $\varepsilon = 10^{-3}$

$J(w,b)$ ↓ by $\leq \varepsilon$ in one iteration ≤ 0.82

* Feature Engineering.

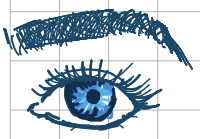
height and width ⇒ area
2 attributes 1 attribute.

created a new feature using existing features.

R^2 metric. (coefficient of determination).

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}$$

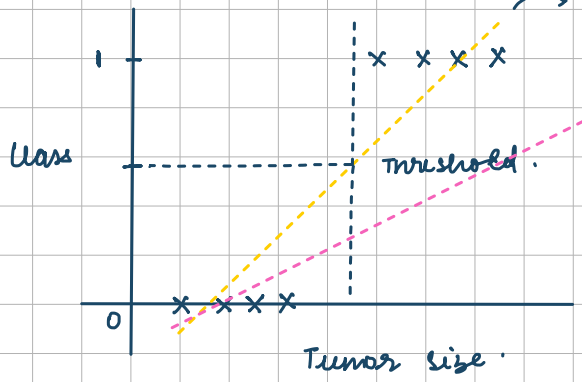
$$= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}$$



14 Feb '24

* Logistic Regression (classification)

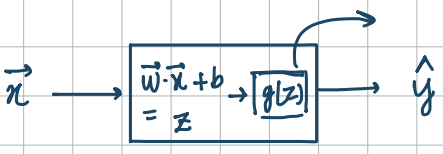
- Binary classification



→ linear regression.

→ because of these samples our line shifts as such that many samples gets misclassified.

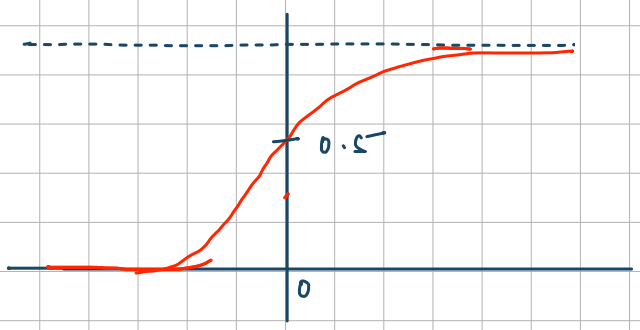
To solve this issue we will introduce sigmoid function



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

$$\hat{y} = f_{\vec{w},\vec{b}}$$

logistic / sigmoid function : $g(z) = \frac{1}{1+e^{-z}}$



$\hat{y} \rightarrow p(y=1 \mid X=x)$.

$p(y=1 \mid X=x) + p(y=0 \mid X=x) = 1$

Case I : $z = 10000$ (high value).

$$g(z) = \frac{1}{1 + e^{-10000}}$$ → this will be a ↓ value ≈ 0.

for higher values of z, $g(z)$ will approach (1).

Case II : $z = 0$

$$g(z) = \frac{1}{1+e^{-0}} = \frac{1}{1+1} = \frac{1}{2} = 0.5$$

Case III : $z = -10000$ (large -ve no.)

$$g(z) = \frac{1}{1 + e^{(+10000)}} \approx \frac{1}{\infty} = 0$$

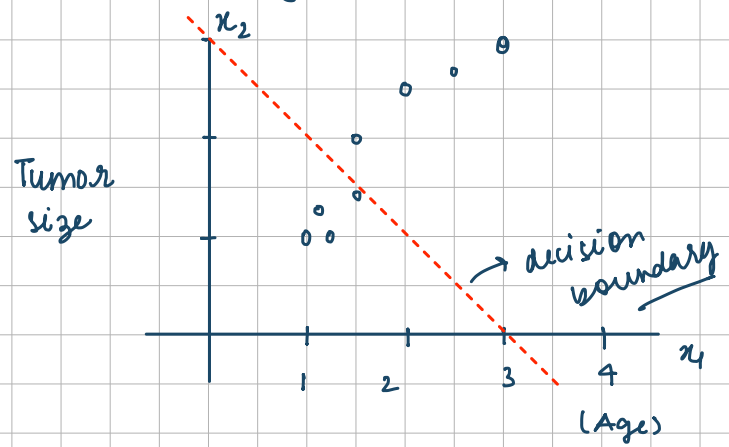
$f_{\vec{w},b} \geq 0.5 \quad \hat{y} \to 1$

$< 0.5 \quad \hat{y} \to 0$

Problem : multiple local minima in SSE.

Solution :

Decision Boundary.



$w_1 x_1 + w_2 x_2 + b$

$w_1 = 1, \quad w_2 = 1, \quad b = -3$

when $w_1 x_1 + w_2 x_2 + b = 0$

↓

decision boundary.

when z = -ve

class ⇒ -ve

when z = +ve

class → +ve.

patient (10)	age (x_1)	tumor size (x_2) (in cm)	y
	1	1	0
	1	1·2	0
	1·2	1·1	0
	1·3	1·5	0
	2	1·5	1
	2·5	2	1
	2·6	2·5	1
	3·0	3	1

19 Feb '24

$y^{(i)} = 1$

$y^{(i)} = 0$

for one example

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

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

cost function :

$$\text{cost } J(\vec{w},b) = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log f_{\vec{w},b}(\vec{x}^{(i)}) + (1-y^{(i)}) \cdot \log(1-f_{\vec{w},b}[\vec{x}^{(i)}])$$

↓

convex cost f(n).

gradient descent

repeat until convergence .

{

$j = 1, 2, \ldots n$ ↷ no. of features.

$$w_j = w_j - \alpha \left(\frac{\partial}{\partial w_j} J(\vec{w},b)\right) \frac{1}{m}\sum_{i=1}^{m}(f_{\vec{w},b}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$b = b - \alpha \left(\frac{\partial}{\partial b} J(\vec{w},b)\right) \rightarrow \frac{1}{m}\sum_{i=1}^{m}(f_{\vec{w},b}(x^{(i)}) - y^{(i)})$$

}

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

$$J(\vec{w},b) = \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log\left(\frac{1}{1+e^{-z}}\right) + (1-y^{(i)}) \log\left(\frac{e^{-z}}{1+e^{-z}}\right)\right]$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} \log(1+e^{-z}) + (1-y^{(i)})(-z\log e - \log(1+e^{-z}))\right]$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} \log(1+e^{-z}) - z\log e - \log(1+e^{-z}) + y^{(i)} z \log e + y^{(i)} \log(1+e^{-z})\right].$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[-\log(1+e^{-z}) - z + y^{(i)} z\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} z - \left[\log\left(1+\frac{1}{e^z}\right) + \log(e^z) \right]\right.$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} z - \log\left(e^z\left(1+\frac{1}{e^z}\right)\right)\right)$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} z - \log(1+e^z)\right)$$

$$\frac{\partial J}{\partial w_j} = \frac{\partial}{\partial w}\left(\frac{-1}{m}\sum_{i=1}^{m}\left(y^{(i)}(\vec{w}\cdot\vec{x}+b) - \log\left(1+e^{(\vec{w}\cdot\vec{x}+b)}\right)\right)\right.$$

$$=$$

Mudita

21 Feb '24

* ↑ data (not always possible).
* hold-out method (not effective with less data).
* K-fold cross validation
* leave one out (less data).

} validation method

mean accuracy $\pm \frac{1}{\sqrt{n}} \times 1.96$ variance

confidence interval

SMOTE (over-sampling) (used for imbalanced data).

↳ **Regularization** (way to overcome overfitting)

n = no. of attributes

Regularization → hyper parameter

- Lasso regression (L1)
- Ridge " (L2)

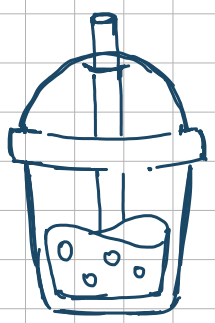
↳ $\lambda \sum_{i=1}^{n} w_i^2$

$\lambda \sum_{i=1}^{n} |w_i|$

(penalty term)

↓

→ penalty to w corresponding to higher values



we will do L2 regularisation

$$J(\vec{w}, b) = \underbrace{\frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) + y^{(i)} \right)^2}_{\text{degree of fitment of data}} + \underbrace{\frac{\lambda}{2m} \sum_{i=1}^{m} w_i^2}_{\text{penalty term. tries to keep } w_j \text{ small.}}$$

regularization parameter

how to choose λ.?

- $\lambda > 0$

$\lambda = 10^8$

$w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

$\approx 0 \quad \approx 0 \quad \approx 0 \quad \approx 0$

$\simeq b$

(simple model ⇒ underfitting)

$\lambda = 0.0001$

gradient descent

$$\min_{\vec{w}, b} J(\vec{w}, b) + \text{regularization term}$$

shrink / shrink ⇒ higher value rapidly decreases.

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j \right]$$

note: no summation

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

26 Feb '24

parameters of layer l, neuron j

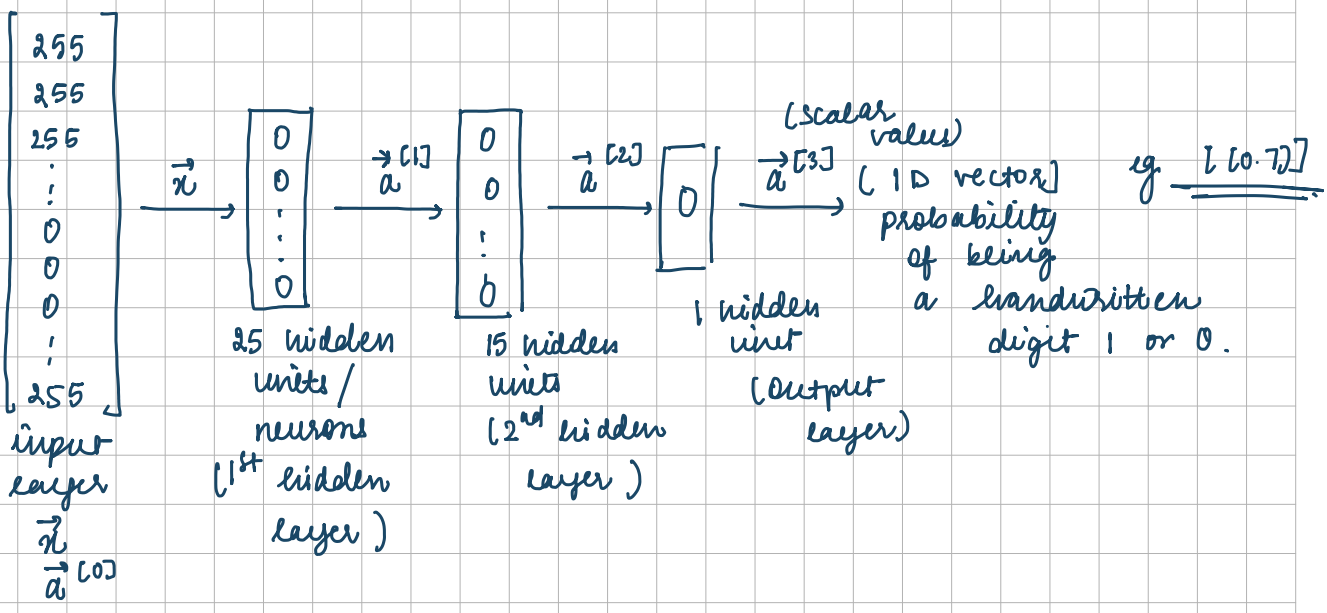
$$a_j^{[l]} = g\left(w_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]}\right)$$

- $a_j^{[l]}$: activation value of layer l, unit (neuron) j
- g: activation function (here sigmoid)
- $\vec{a}^{[l-1]}$: output of layer $l-1$ (i.e. previous layer)

FORWARD PROPOGATION

propogate the information to the next layer.

Handwritten digit recognition → MNIST 0 or 1 (binary images) 8 X 8



in order to get probability we use sigmoid function. (ranges from 0 to 1) (ranges from 0 to 1)

$$\vec{a}^{[1]} = \begin{bmatrix} g\left(w_1^{[1]} \cdot \vec{a}^{[0]} + b_1^{[1]}\right) \rightarrow a_1^{[1]} \\ \vdots \\ g\left(w_{25}^{[1]} \cdot \vec{a}^{[0]} + b_1^{[25]}\right) \end{bmatrix}$$

$$\vec{a}^{[2]} = \begin{bmatrix} g(w_1^{[2]} \vec{a}^{[1]} + b_1^{[2]}) \\ \vdots \\ g(w_{15}^{[2]} \vec{a}^{[1]} + b_{15}^{[2]}) \end{bmatrix}$$

$$\vec{a}^{[3]} = [\, g(w_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]}) \,]$$

$\vec{a}^{[3]} \geq 0.5$

yes: $\hat{y} = 1$; no: $\hat{y} = 0$

Model Training Steps are:

- **create a model.**

$$z = w \cdot x + b$$
$$f_x = \frac{1}{1 + np.exp(-z)}$$

```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense.
```

(one after one)

```
model = Sequential([ Dense(units=25, activation='sigmoid')
                     Dense(units=15, activation='sigmoid')
                     Dense(units=1, activation='sigmoid') ])
```

- **specify loss and cost function**

(training example; avg. loss for all examples.)

$$L(f(\vec{x}), y) = -y \log(f(\vec{x})) - (1-y) \log(1 - f(\vec{x}))$$
$$\text{cost} = \frac{1}{m} \sum_{i=1}^{m} (\ \downarrow\ )$$

[cross entropy]

for binary classification: binary cross entropy.

code:
```
from tensorflow.keras.losses import BinaryCrossentropy.
model.compile( loss = BinaryCrossentropy() )
```
↳ loss function for all the layers.

- **Gradient descent / Back propogation**
(minimizing the cost)

repeat {

$$w_j^{[l]} = w_j^{[l]} - \alpha \frac{\partial}{\partial w_j^{[l]}} J(\vec{w}^{[l]}, b^{[l]})$$

(learning rate)

$$b_j^{[l]} = b_j^{[l]} - \alpha \frac{\partial}{\partial w_j^{[l]}} J(\vec{w}^{[l]}, b^{[l]})$$

}

100 forward 100 backward.

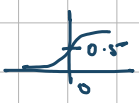
code: model.fit(X, Y, epochs = 100)

Linear activation function → also known as no activation function.

27 feb '24
Tuesday

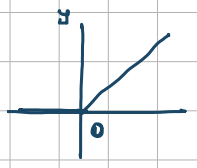
Choice of activation function

Case I: Binary answer
→ sigmoid



$\frac{1}{1+e^{-z}}$

Case II: Non-negative number indicating the degree of activation
→ Rectified linear unit (ReLU)



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

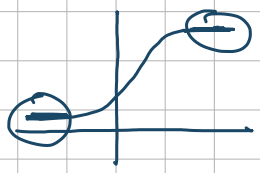
Regression.
(Linear Activation function)

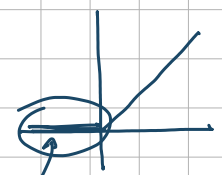
$y = +ve / -ve / 0$.

Case III: Regression → (ReLU)

$y = 0$ to $+\infty$.

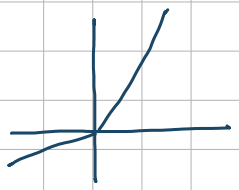
- Most common choice of the activation function at hidden layer is ReLU.





problematic for gradient-descent.

(ReLU converges faster).

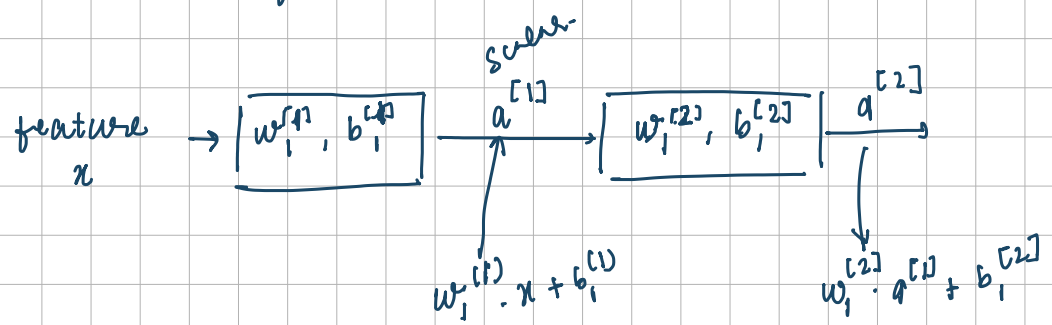


Leaky ReLU.

sequential([Dense(units = 25, activation = 'relu')
Dense(units = 15, activation = 'relu')
Dense(units = 1, activation = 'sigmoid')])

^^^^

* Why do we need activation functions?
  - non-linearity.
  -

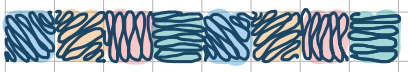


$$a^{[1]} = w_1^{[1]} \cdot x + b_1^{[1]} \quad \text{— (1)}$$
$$a^{[2]} = w_1^{[2]} \cdot a^{[1]} + b_1^{[2]} \quad \text{— (2)}$$

substitute (1) in (2)

$$a^{[2]} = w_1^{[2]} (w_1^{[1]} \cdot x + b_1^{[1]}) + b_1^{[2]}$$

$$a^{[2]} = w_1^{[2]} \cdot w_1^{[1]} \cdot x + w_1^{[2]} \cdot b_1^{[1]} + b_1^{[2]}$$



28 Feb '24

Multiclass Classification

softmax: Regression

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

restricts the value from 0 to 1

↳ Total probability / Relative probability

The estimated chance of $y=1$ given x. (x ↳ feature set.)

softmax activation function transforms the raw output of a neural network into a vector of probabilities.

$$a_j = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}} = P(y=j \mid \vec{x}) \quad \text{[Generalized form]}$$

Note: Softmax regression with two classes is same as Logistic regression.

Softmax Regression is a generalization of logistic regression.

TRANSFER LEARNING

Cost Function:

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

$$a_1 = g(z) = \frac{1}{1+e^{-z}} = P(y=1 \mid \vec{x})$$

$$a_2 = 1 - a_1 = P(y=0 \mid \vec{x})$$

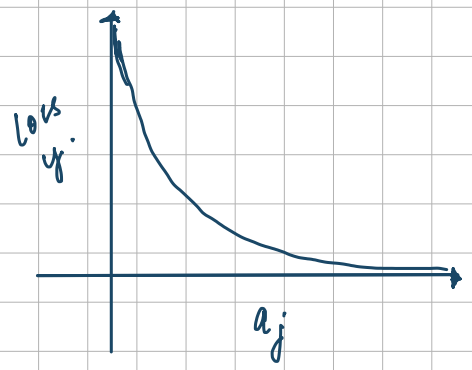
Softmax:

$$a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + \cdots + e^{z_N}}$$

$$\vdots$$

$$a_N = \frac{e^{z_N}}{e^{z_1} + e^{z_2} + \cdots + e^{z_N}}$$

$$\text{loss}(a_1, a_2 \cdots a_N, y) = \begin{cases} -\log a_1 & \text{if } y=1 \\ -\log a_2 & \text{if } y=2 \\ \vdots & \\ -\log a_N & \text{if } y=N. \end{cases}$$



sparse categorical cross entropy.

29 Feb '24

multilabel classification
↳ medical data ?

Specify the model

```
import tensorflow as tf
from tensorflow import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([ Dense(units = 25, activation = 'relu'),
                     Dense(units = 15, activation = 'relu'),
                     Dense(units = 10, activation = 'softmax') ])

from tensorflow.keras.losses import SparseCategoricalCrossentropy
model.compile(loss = SparseCategoricalCrossentropy())
```

- Train the model to minimize $J(\vec{w}, b)$

```
model.fit(X, Y, epochs = 100)
```

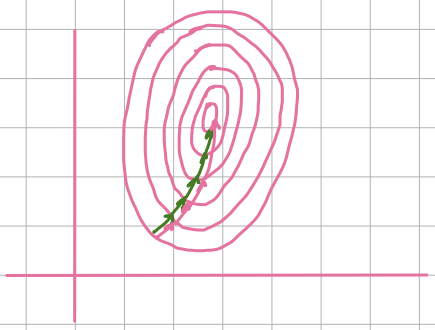
10

ADAM OPTIMIZER (Adaptive Movement Estimation)

some optimization to 'α'.

11

if the gradient descent is going in the right direction we ↑ alpha.



if the learning rate is relatively high than

* if the parameter value keeps on moving in roughly the same direction, let us ↑ the learning rate for that parameter.

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

$$w_2 = w_2 - \alpha_2 \frac{\partial}{\partial w_2} J(\vec{w}, b)$$

$$\vdots$$

$$b = b - \alpha_{11} \frac{\partial}{\partial b} J(\vec{w}, b)$$

4%

conversely if the parameter value keeps oscillating back and forth, we will reduce α_j for that parameter j.

Recommender Systems

- content based filtering
- collaborative filtering (many or more like minded people)

→ hybrid

2006
netflix challenge.

480 000 users
18 000 movies
Rate ?
1984 Animal Farm

1 April '24

why vertical partitioning ? | train test ? I
" horizontal " ? | unbiased estimate of how well the model is doing ?

hyper-parameters.

Development set / validation set →

Traditionally the dataset is divided into training and test-set and generally when the data is scarce, we often use 70/30 or 80/20 split.

In the era of big data we generally have 98/2 or 90/10 split. The work flow in this case would be train your algo on training set. → hyper parameter tuning
2nd use your dev set or cross validation set. to see which of many diff. models perform best on your dev set.

Pycaret
↳ project

gridsearch CV
↳ cross validation.

3rd → take the best model you have found and evaluate it on your test set in order to get an unbiased estimate of how well your algo is doing.

JPEG
JPEG 2000

Mismatched train - test distribution

Explainable AI.
Post hoc

training set → high resolution, professional images.
test set → user uploaded blurred images.

HSDLSS

09 April '24

Eigenvalues and eigenvectors of a 3x3 matrix

$$\det(A - \lambda I) = 0$$

$$A v_1 = \lambda_1 v_1$$

$$A v_2 = \lambda_2 v_2$$

$$A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Basis → These vectors are linearly independent and other vectors depend on these.

feature extraction

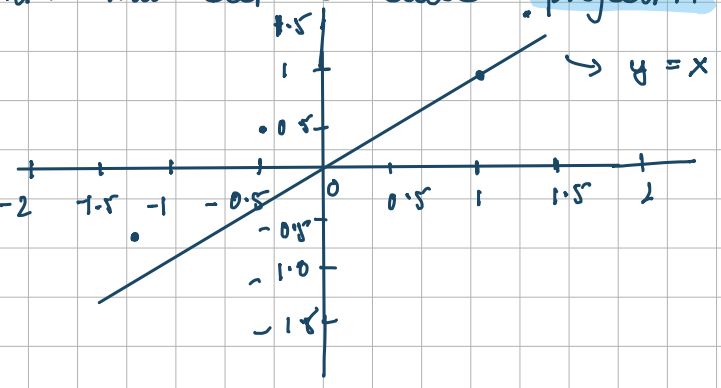
Use of Dimensionality Reduction !

- it leads to smaller datasets.
- it would be easier to visualize.

PCA allows us to reduce the dimension of our data. but it also preserves much of the info which we might loose if we simply started deleting columns.

Dim is to move your data points into a vector space with fewer dimensions. This step is called projection

	X	Y
A	1.0	1.0
B	1.2	1.6
C	-0.5	0.2
D	-1.3	-0.6



Projecting a data to a line.

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \times \frac{1}{\|\begin{bmatrix} 1 \\ 1 \end{bmatrix}\|} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times \frac{1}{\sqrt{2}}$$

$\sqrt{1^2+1^2} = \sqrt{2}$ (norm).

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \frac{1}{\sqrt{2}} = \frac{1+1}{\sqrt{2}} = \frac{2}{\sqrt{2}} = \sqrt{2} = 1.4142$$

$$\begin{bmatrix} 1.2 & 1.6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} = \frac{(1.2 + 1.6)}{\sqrt{2}} = \frac{2.8}{\sqrt{2}} = 1.9799$$

$$\begin{bmatrix} -0.5 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} = \frac{-0.3}{\sqrt{2}} = -0.2121$$

$$\begin{bmatrix} -1.3 & -0.6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} = \frac{-1.9}{\sqrt{2}} = -1.344.$$

In general any matrix A can be projected onto the direction given by a vector V.

$$A_{p_{r \times 2}} = A_{r \times c} \begin{bmatrix} \frac{V_1}{\|V_1\|_2} & \frac{V_2}{\|V_2\|_2} \end{bmatrix}_{c \times 2}$$

↓ Projected space.

r =

c = no. of dimension

Mean, variance and co-variance.

mean (x_i, y_i) $\quad \forall\, i = 1 \dots n$

$$\mu_x = \bar{x} = \text{mean}(x) = \frac{1}{n} \sum_{i=1}^{n} x_i$$ (The avg. of the data).

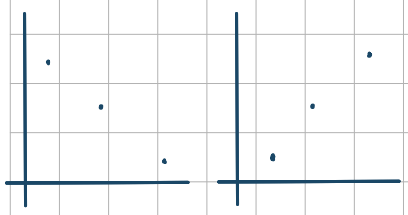
$$\mu_y = \bar{y} = \text{mean}(y) = \frac{1}{n} \sum_{i=1}^{n} y_i$$

$(\bar{x}, \bar{y})$ define the mean

Variance

How much spread your data has?

$$\text{var}(x) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu_x)^2$$



$$\text{var}(y) = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \mu_y)^2$$

$x \uparrow \; y \downarrow$ $\qquad$ $x \uparrow \; y \uparrow$.

This relation is captured by covariance.

$$\text{cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)$$

$$A = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) \\ \text{cov}(y, x) & \text{cov}(y, y) \end{bmatrix}$$

$$\mu = \begin{bmatrix} \mu_x & \mu_y \\ \vdots & \\ \mu_x & \mu_y \end{bmatrix}$$

$$C = \frac{1}{n-1}(A-\mu)^T(A-\mu)$$

→ A ke dimension jitna he μ ka dimension bnaa rahe hai

↓ matrix to be projected.

A		A − μ		$(A-\mu)^T$
x	y	x	y	
10	5	2	−1	
12	3	4	−3	
6	9	−2	3	
6	4	−2	−2	
5	11	−3	5	
14	2	6	−4	
8	1	0	−5	
3	13	−5	7	
$\mu_x = 8$	$\mu_y = 6$			

cov ?

10 April '24

$$C = \begin{bmatrix} 9 & 4 \\ 4 & 3 \end{bmatrix}$$

$$\det(A - \lambda I) = 0$$
$$(9-\lambda)(3-\lambda) - 16 = 0$$
$$\lambda^2 - 12\lambda + 11 = 0$$
$$\lambda = 11, 1$$

eigen value = 11, 1

eigen vector = $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ $\begin{bmatrix} -1 \\ 2 \end{bmatrix}$

① create a matrix

$$X = \begin{bmatrix} x_{11} & \cdots & x_{15} \\ x_{n1} & \cdots & x_{n5} \end{bmatrix}_{n \times 5}$$

② centre the data
"find: $(X - \mu)$"

$$(X - \mu) = \begin{bmatrix} x_{11} - \mu_1 & \cdots & x_{15} - \mu_5 \\ \vdots & & \vdots \\ x_{n1} - \mu_1 & & x_{n5} - \mu_5 \end{bmatrix}$$

③ calculate the covariance matrix

$$C = \frac{1}{n-1} (X - \mu)^T (X - \mu)$$

$$= \begin{bmatrix} var(x_1) & cov(x_1, x_2) & cov(x_1, x_3) & cov(x_1, x_4) & cov(x_1, x_5) \\ & var(x_2) & & & \\ & & var(x_3) & & \\ & & & var(x_4) & \\ & & & & var(x_5) \end{bmatrix}_{5 \times 5}$$

④ find the eigenvalues and eigen vectors
sort eigen values in decreasing order.

⑤ create a projection matrix.

$$V = \begin{bmatrix} \frac{v_1}{\|v_1\|_2} & \frac{v_2}{\|v_2\|_2} \end{bmatrix}$$

⑥ Project the centered data.

$$X_{PCA} = (X - \mu) \cdot V$$

(corresponding 2 principal components)

Ques (2,1) (3,5) (4,3) (5,6) (6,7) (7,8)

10

$$X = \begin{bmatrix} x & y \\ 2 & 1 \\ 3 & 5 \\ 4 & 3 \\ 5 & 6 \\ 6 & 7 \\ 7 & 8 \end{bmatrix}$$

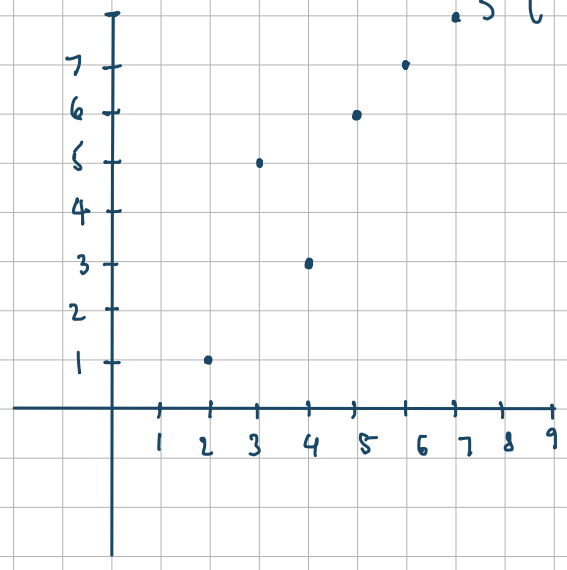
$$\bar{x} = \frac{27}{6} = 4.5$$

$$\bar{y} = \frac{30}{6} = 5$$

$$X - \mu = \begin{bmatrix} -2.5 & -4 \\ -1.5 & 0 \\ -0.5 & -2 \\ 0.5 & 1 \\ 1.5 & 2 \\ 2.5 & 3 \end{bmatrix}$$

$$C = \frac{1}{5} \begin{bmatrix} -2.5 & -1.5 & -0.5 & 0.5 & 1.5 & 2.5 \\ -4 & 0 & -2 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} -2.5 & -4 \\ -1.5 & 0 \\ -0.5 & -2 \\ 0.5 & 1 \\ 1.5 & 2 \\ 2.5 & 3 \end{bmatrix}$$

$$\frac{1}{5} \begin{bmatrix} 17.5 & 22 \\ 22 & 34 \end{bmatrix} \Rightarrow \begin{bmatrix} 3.5 & 4.4 \\ 4.4 & 6.8 \end{bmatrix}$$



$$|C - \lambda I| = 0$$

$$\begin{vmatrix} 3.5-\lambda & 4.4 \\ 4.4 & 6.8-\lambda \end{vmatrix} = 0$$

$$(3.5-\lambda)(6.8-\lambda) - (4.4)^2 = 0$$

$$23.8 - \lambda^2 + 10.3\lambda - 19.36 = 0$$

$\lambda_1 = 9.849 \Rightarrow \begin{bmatrix} 0.693 \\ 1 \end{bmatrix} \Rightarrow$

$\lambda_2 = 0.45 \Rightarrow$

Variance captured by λ_1

$\rightarrow \frac{9.84}{9.84 + 0.45} \rightarrow \frac{9.84}{10.29} = 0.9562$

$$\begin{array}{r} 9.84 \\ 0.45 \\ \hline 10.29 \end{array}$$

$$\begin{bmatrix} -2.5 & -4 \end{bmatrix} \begin{bmatrix} 0.57 \\ 0.82 \end{bmatrix} \Rightarrow -4.73$$

$$\begin{bmatrix} -1.5 & 0 \end{bmatrix} \begin{bmatrix} 0.57 \\ 0.82 \end{bmatrix}$$

15 April 2024

Bayesian Classifiers

class conditional independence.

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$$

X be a data tuple.

$$\frac{|C_{i,D}|}{|D|}$$

class conditional independence.

$$P(X|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_d|C_i).$$

— why naive bayes works even with such high assumption?
if there exist many attributes which are correlated
then are also have some attributes " " not ".

showed follow the normal distribution

should class

a b c d e f g h i j k

l m n o p q r s t

u v w x y z

18 April 2024

End-to-End CNN
Transfer Learning

23 April 2024

Padding

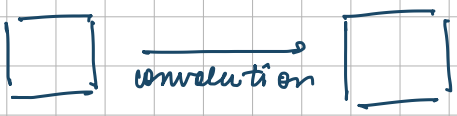
intuition 1: if you don't wish to shrink the image.
intuition 2: if the corner pixel consists of more information, we need padding.

padded convolution

output size without padding: $(n-f+1) \times (n-f+1)$ ⇒ with padding: $(n+2p-f+1) \times (n+2p-f+1)$

n = image, p = padding, f = filter

what should be the filter size?



$n + 2p - f + 1 = n$

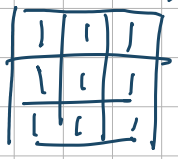
can find padding / filter size using this if we wish to have the same size of output and input.

same padding → output ka size same ho [padding = on]
valid padding → no padding ($p = 0$).

Strided convolution

stride (step): by default stride is 1 but we can have 2, 3 or any number.

but why?



→ if we have the same information then by ↑ the size of stride we can capture the information and also compress.

⇒ $\left\lfloor \frac{n+2p-f+1}{s} \right\rfloor \times \left\lfloor \frac{n+2p-f+1}{s} \right\rfloor$

s = stride. [steps for column as well as row.]

→ to use corner pixels we need this.

ex:

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

$*$ $\begin{matrix} 3 & 4 & 4 \\ 1 & 0 & 2 \\ -1 & 0 & 3 \end{matrix}$

$s = 2$ valid

$9 + 16 + 32 + 7 + 6 - 4 + 3 \Rightarrow$

Output:

91	100	88
69	91	127
44	72	74

$70 - 3 + 67 + 24 = 46 + 6 + 18 + 70$

$12 + 6 + 18 + 28 + 46$

$6 + 12 + 28 + 6 + 18 + (-8) + 8 \times 3$

$\Rightarrow$

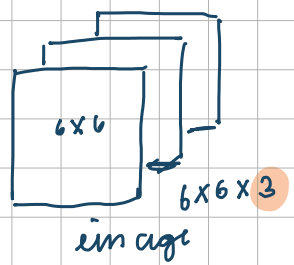
6×3

$18 + 8 + 36 + 7 + 6 - (8) + 21$

24 April 2024

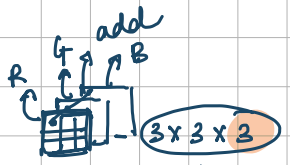
Convolution over Volumes

$n_H \times n_w$ (may not be a square image)



image

$*$



1 filter
3 channel.

results in → 4x4 (RGB).

parameters to learn ⇒ 27.

in image

no. of channels should match the no. of channels in filter

with even filters → aliasing. ← results in

images can't be padded in case of even filter

output image:

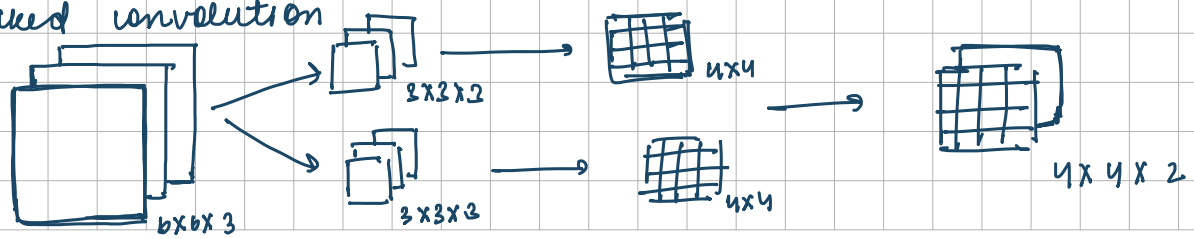
$(n_H - f + 1) \times (n_w - f + 1)$.

$n_H \times n_w \times n_c$

↳ channels

Take each of the 27 no. of the filter and multiply them with corresponding no. from Red, Green and Blue channel.

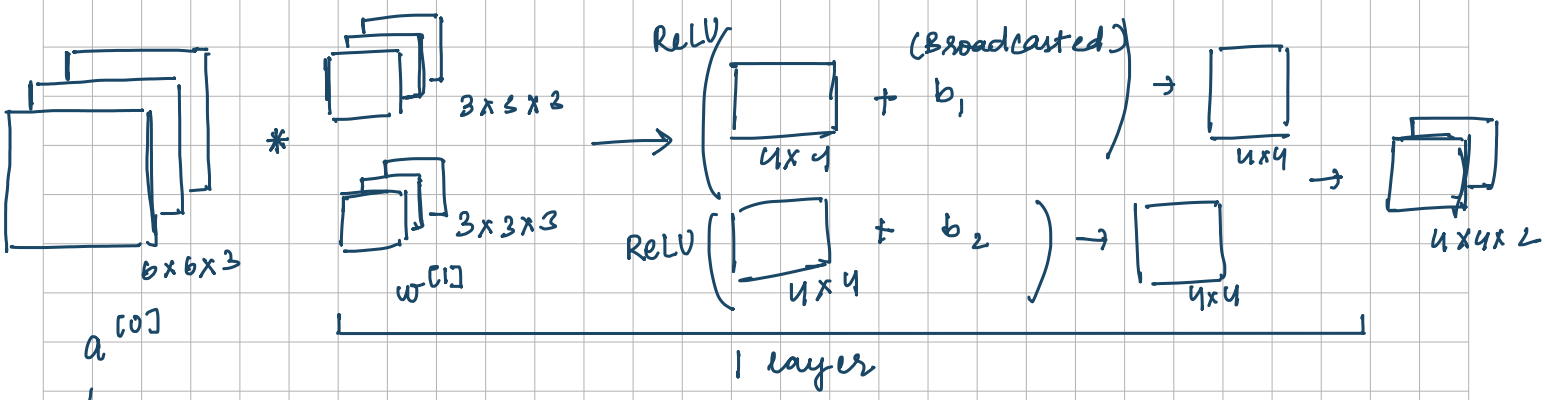
Stacked convolution



$(n_H - f + 1) \times (n_W - f + 1) \times n_c'$

↳ # of filters.

How to represent a layer of convolution neural network?



pooling layer (no parameter)

- if you have 10 filters that are 3x3x3 in size of 1 layer of neural network?

  ⇒ 3x3x3x10 + 10 (bias)

If we are at layer l of CNN, then:

$f^{[l]}$ = filter_size at layer l

$p^{[l]}$ = padding at layer l

$s^{[l]}$ = stride at layer l

$n_c^{[l]}$ = no. of filters at layer l.

Input size: $n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}$

Output: $n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

$$n_H^{[l]} = \left\lfloor \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$

$$n_W^{[l]} = \left\lfloor \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$

25 April 2024

input : $n_H^{[l-1]} \times n_w^{[l-1]} \times n_c^{[l-1]}$

output : $n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

filter size : $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

Activations :

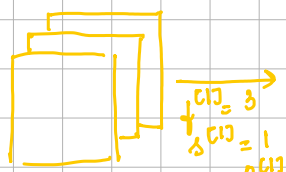
$a^{[l]}$: $n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

weights : $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$ ↳ # of filters

→ detecting multiple edge detection. [multiple filters].

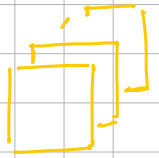
bias : $n_c^{[l]}$ $(1, 1, 1, n_c^{[l]})$ ↳ for broadcast.

A simple convolution neural network.



$f^{[1]} = 3$
$s^{[1]} = 1$
$p^{[1]} = 0$
10 filters.

$f^{[2]} = 5$
$s^{[2]} = 2$
$p^{[2]} = 0$
20 filters



$f^{[3]} = 5$
$s^{[2]} = 2$
$p^{[3]} = 0$
40 filters

$n_H^{[0]} = 39$

$n_w^{[0]} = 39$

$n_c^{[0]} = 3$

$39 + (-3) + 1$

$36 + 1 = 37$

$\frac{37 - 5}{2} + 1$

$\Rightarrow 17 \times 17 \times 20$

no. of filters.

1960 → vector.

if all these can be mapped to a single neuron.

CONV

↳ hyperparameters

(

pooling layer
fully connected layer (FC)

- max pooling
- average pooling

1	3	2	1
2	9	1	1
1	4	2	3
5	6	1	2

pooling layer

$s = 2$

$f = 2$

output of avg. pooling layer

3.75	1.25
4	2

output of pooling layer

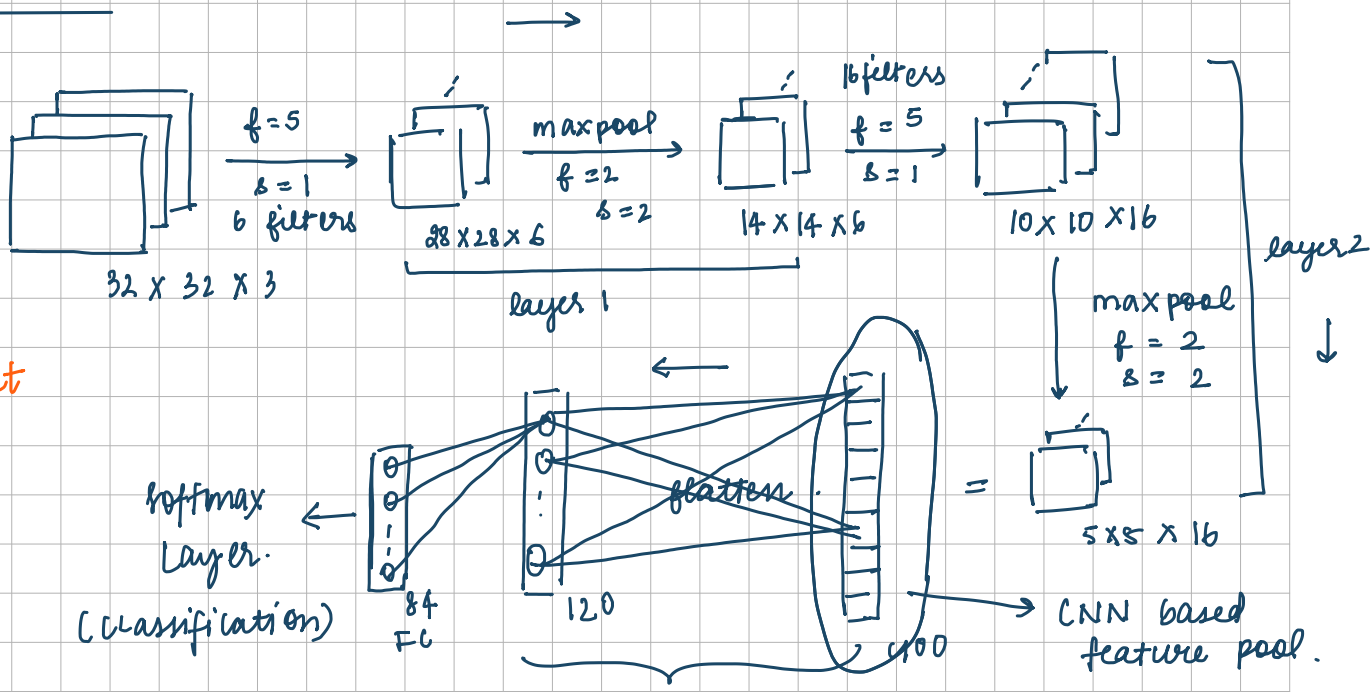
9	2
6	3

disadvantage : loss of information

disadvantage :

→ in pooling layer there is no parameter to learn.

29 April '24



LeNet

why we want this fully connected layer?

No. of trainable parameters ?

	Activation shape.	Activation size	# of parameters
Input	(32, 32, 3)	3072	0
CONV1 (f=5, s=1)	(28, 28, 8)	6272	608
POOL1 (f=2, s=2)	(14, 14, 8)	1568	0
CONV2 (f=5, s=1)	(10, 10, 16)	1600	3216
POOL2 (f=2, s=2)	(5, 5, 16)	400	0
FC1	(120, 1)	120	400 x 120 + 120
FC2	(84, 1)	84	120 x 84 + 84
Softmax	(10, 1)	10	84 x 10 + 10

no. of filters (size of filter x no. of channel (prev layer) + 1)

8 x (5x5 x 3 + 1)

16 x (5 x 5 x 8 + 1)

- Why convolution ?

1000 x 1000 x 3

- reduced no. of parameters (parameter sharing)
- sparsity of connection

feature detector (say a vertical edge detector) that is useful in 1 part of the image is probably useful for the other part of the image.

in each layer, each output value depends only on a smaller no. of inputs (prevents overfitting)

* CNN are translation invariance.