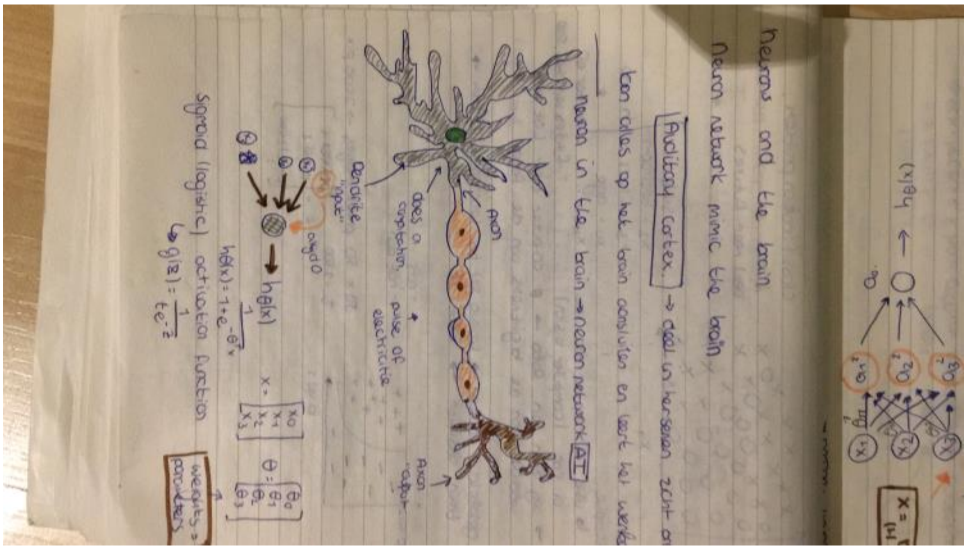
Aantekeningen nn + colleges

Saturday, 5 January 2019 13:04



x_1
θ_{11}
a_1^2
x_2
a_2^2
x_3
a_3^2
a_0
$h_\theta(x)$
$a^{(1)} = x$
Axon
does a computation.
pulse of electricitie
Axon "output"
Dendrite "input"
output
$h_\theta(x)$

neurons and the brain

Neuron network mimic the brain

$\boxed{\text{Auditory cortex.}}$ → deel in hersenen zicht on

kan alles op het brain aansluiten en leert het werk

neuron in the brain → neuron network $\boxed{\text{AI}}$

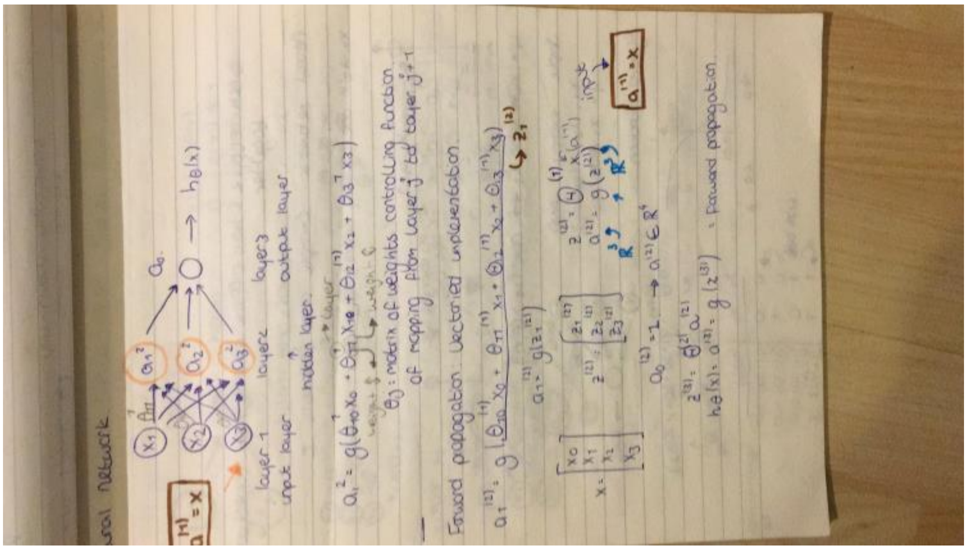
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$\boxed{\text{weights = parameters}}$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

Sigmoid (logistic) activation function

$$\hookrightarrow g(z) = \frac{1}{te^{-z}}$$



x_1
θ_{11}
a_1^2
x_2
a_2^2
x_3
a_3^2
a_0
$h_\theta(x)$
layer 1
input layer
layer2
hidden layer.
layer3
output layer

ural network

$\boxed{a^{(1)} = x}$

$$a_1^2 = g(\theta_{10}^1 x_0 + \theta_{11}^1 x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^1 x_3)$$

→ layer

weight $ ← ↳ weight €

θ_j : matrix of weights controlling function of mapping from layer j to layer j+1

Forward propagation: Vectoried unplerentation

$$a_1^{(2)} = g\big(\underbrace{\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3}_{}\big) \hookrightarrow z_1^{(2)}$$

$$a_1^{(2)} = g(z_1^{(2)})$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$z^{(2)} = \Theta^{(1)} x \, (a^{(1)})$$

$$a^{(2)} = g(z^{(2)})$$

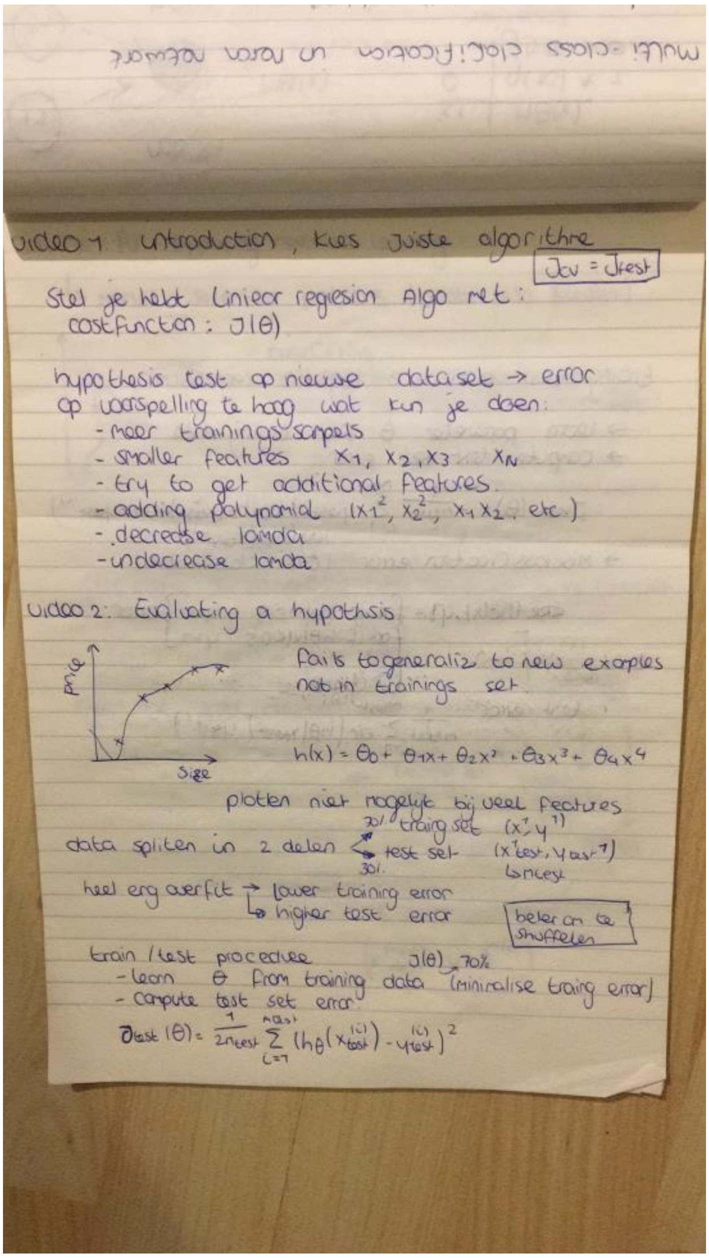
$\mathbb{R}^3$ ↑ $\mathbb{R}^3$

input → $\boxed{a^{(1)} = x}$

$$a_0^{(2)} = 1 \rightarrow a^{(2)} \in \mathbb{R}^4$$

$$z^{(3)} = \Theta^{(2)} a^{(2)}$$

$$h_\theta(x) = a^{(3)} = g(z^{(3)})$$: Forward propagation



multi-class clasification in neural network

video 1 introduction, kies juiste algorithme

$\boxed{J_{cv} = J_{test}}$

Stel je hebt Linieor regresion Algo met:
costfunction: $J(\theta)$

hypothesis test op nieuwe dataset → error
op voorspelling te hoog wat kun je doen:
- meer trainings sampels
- smaller features $x_1, x_2, x_3 \; x_N$
- try to get additional features.
- adding polynomial $(x_1^2, x_2^2, x_1 x_2, \text{etc.})$
- decrease lamda
- undecrease lamda

video 2: Evaluating a hypothsis

fails togeneraliz to new exaples not in trainings set.

$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

plotten niet mogelijk bij veel features

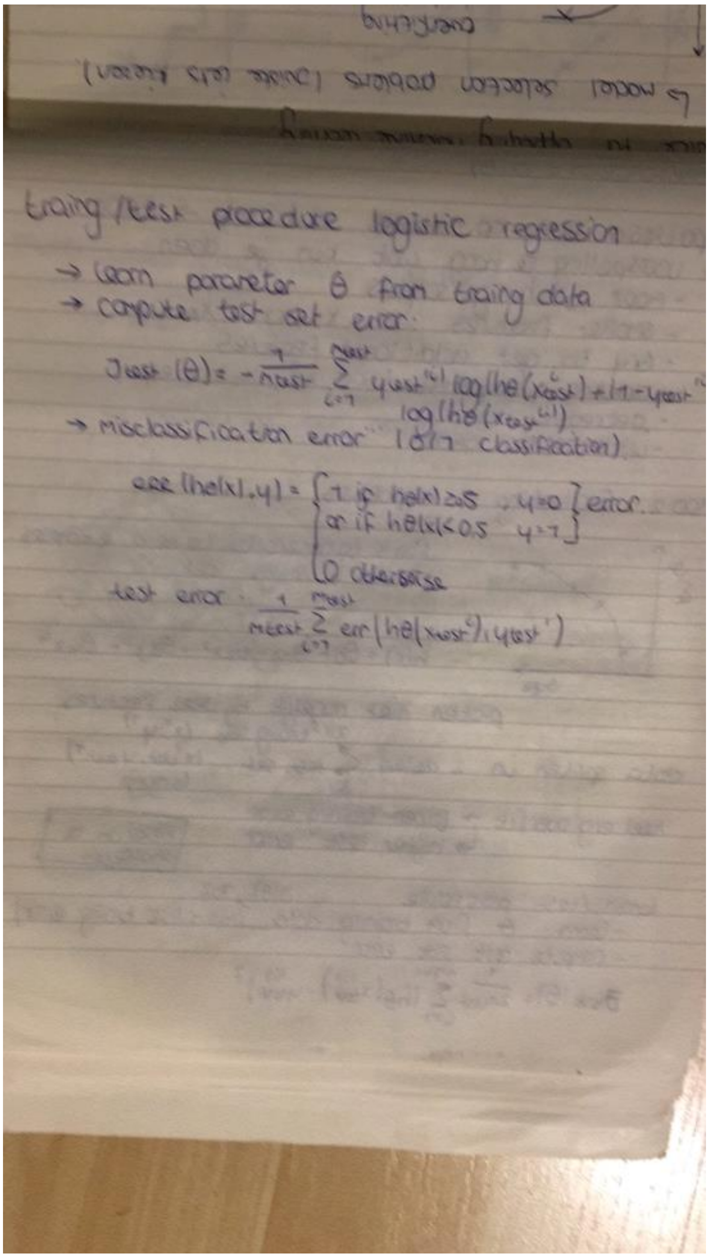
data spliten in 2 delen
- 70% traing set (x^1, y^1)
- 30% test set (x^1_{test}, y^1_{test}) ↳ n_{test}

heel erg overfit → lower training error
↳ higher test error

beter om te shuffelen

train / test procedure $J(\theta)$ → 70%
- learn θ from training data (minimalise traing error)
- compute test set error

$$J_{test}(\theta) = \frac{1}{2n_{test}} \sum_{i=1}^{n_{test}} \left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)}\right)^2$$



traing/test procedure logistic regression

→ learn parameter θ from traing data

→ compute test set error

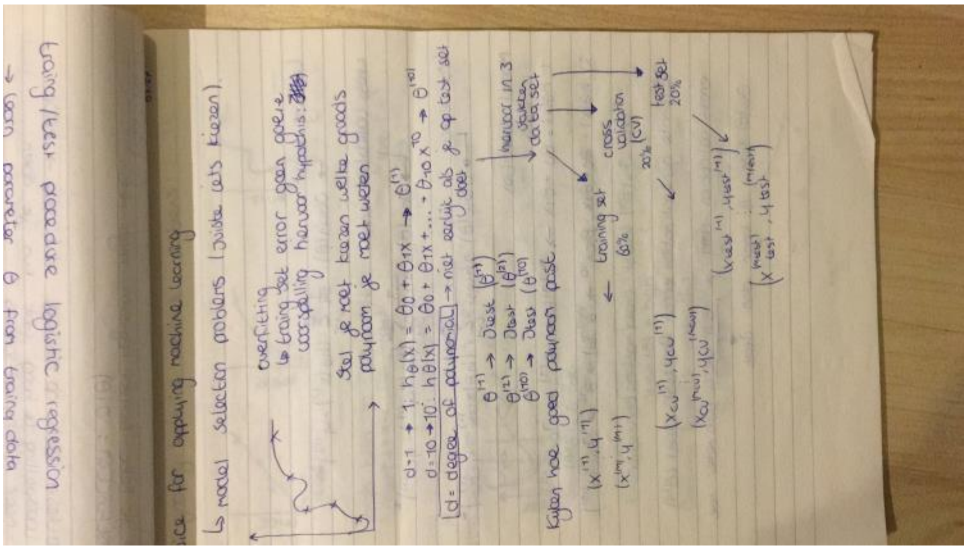
$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log(h_\theta(x_{test}^{(i)}) + (1-y_{test}^{(i)}) \log(h_\theta(x_{test}^{(i)}))$$

→ misclassification error (0/1 classification)

$$err(h_\theta(x), y) = \begin{cases} 1 \text{ if } h_\theta(x) \geq 0.5,\ y=0 \\ \text{or if } h_\theta(x) < 0.5\ \ y=1 \\ 0 \text{ otherwise} \end{cases}$$

error

test error $\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\theta(x_{test}^{(i)}), y_{test}^{(i)})$



...ice for applying machine learning

↳ model selection problems (juiste iets kiezen)

overfitting

↳ traing set error geen goede voorspelling hiervoor hypothesis:

Stel je moet kiezen welke graads polynoom je moet weten

$d=1 \rightarrow 1: h_\theta(x) = \theta_0 + \theta_1 x \rightarrow \theta^{(1)}$

$d=10 \rightarrow 10: h_\theta(x) = \theta_0 + \theta_1 x + \ldots + \theta_{10} x^{10} \rightarrow \theta^{(10)}$

$\boxed{d = \text{degree of polynomial}}$ → niet eerlijk als je op test set doet

$\theta^{(1)} \rightarrow J_{test}(\theta^{(1)})$

$\theta^{(2)} \rightarrow J_{test}(\theta^{(2)})$

$\theta^{(10)} \rightarrow J_{test}(\theta^{(10)})$

Kijken hoe goed polynoom past

hiervoor in 3 stukken data set

$(x^{(1)}, y^{(1)})$ ← training set 60%

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

cross validation (CV) 20%

$(x_{cv}^{(1)}, y_{cv}^{(1)})$

$(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$

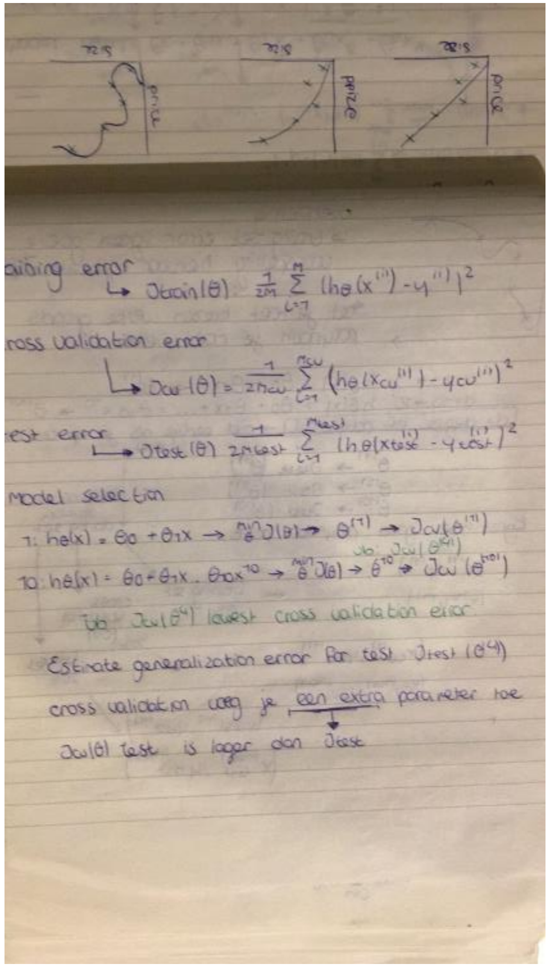
test set 20%

$(x_{test}^{(1)}, y_{test}^{(1)})$

$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

traing/test procedure logistic regression

→ learn parameter θ from traing data



size
price
size
prize
size
price

aining error

$\hookrightarrow J_{train}(\theta)\ \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$

ross validation error

$\hookrightarrow J_{cv}(\theta)=\frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)})-y_{cv}^{(i)})^2$

est error

$\hookrightarrow J_{test}(\theta)\ \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}-y_{test}^{(i)})^2$

Model selection

1: $h_\theta(x)=\theta_0+\theta_1x \rightarrow \min_\theta J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$

ub: $J_{cv}(\theta^{(4)})$

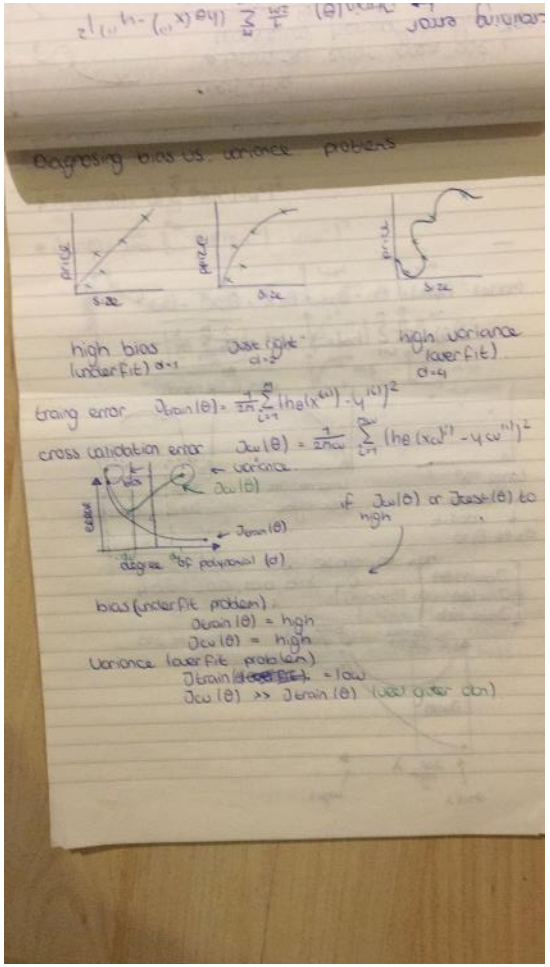
10: $h_\theta(x)=\theta_0+\theta_1x\ldots\theta_{10}x^{10} \rightarrow \min_\theta J(\theta) \rightarrow \theta^{10} \rightarrow J_{cv}(\theta^{(10)})$

ub $J_{cv}(\theta^4)$ lowest cross validation error

Estimate generalization error for test $J_{test}(\theta^{(4)})$

cross validation voeg je een extra parameter toe

$J_{cv}(\theta)$ test is lager dan J_{test}



price
size
prize
size
price
size
error
$J_{cv}(\theta)$
$J_{train}(\theta)$
variance
bias
degree of polynomial (d)

Diagnosing bias vs. variance problems

high bias (underfit) d=1

Just right d=2

high variance (overfit) d=4

traing error $J_{train}(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$

cross validation error $J_{cv}(\theta)=\frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)})-y_{cv}^{(i)})^2$

if $J_{cv}(\theta)$ or $J_{test}(\theta)$ to high

bias (underfit problem)

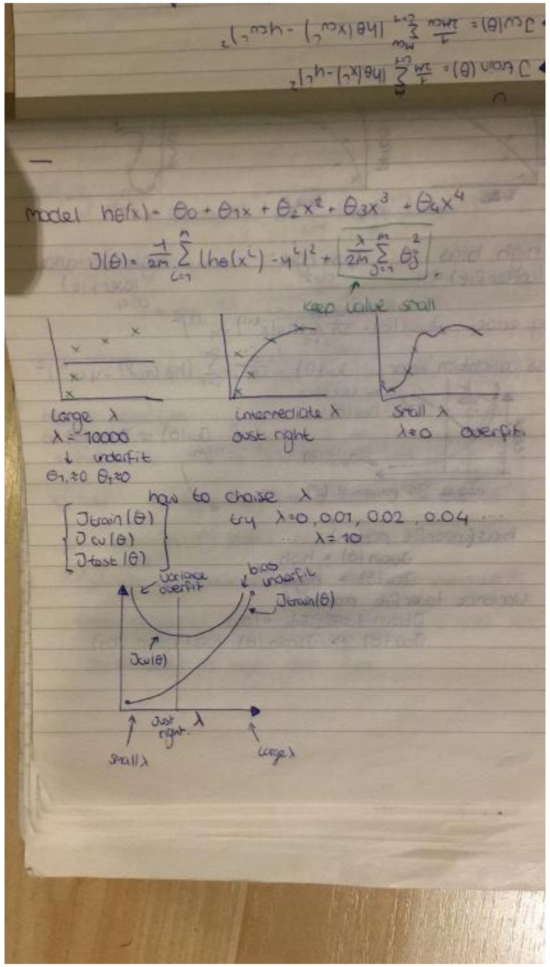
$J_{train}(\theta)$ = high

$J_{cv}(\theta)$ = high

Variance (overfit problem)

J_{train} ~~[illegible]~~ = low

$J_{cv}(\theta) >> J_{train}(\theta)$ (veel groter dan)



Variance overfit
bias underfit
$J_{train}(\theta)$
$J_{cv}(\theta)$
λ
Just right.
Small λ
Large λ

Model $h_\theta(x)=\theta_0+\theta_1x+\theta_2x^2+\theta_3x^3+\theta_4x^4$

$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{i})-y^{i})^2+\boxed{\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2}$

keep value small

Large λ
λ = 10000
↓ underfit
$\theta_1 \approx 0$ $\theta_2 \approx 0$

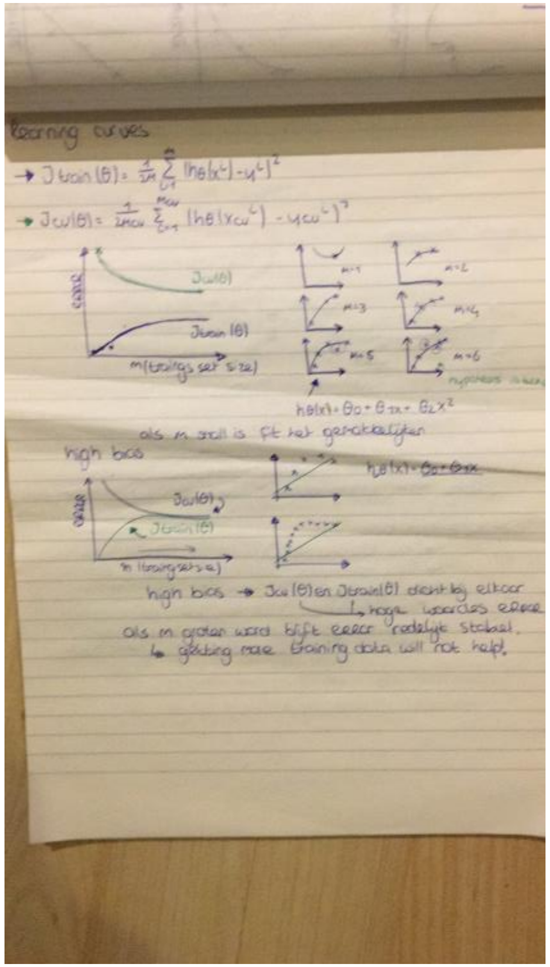
intermediate λ
Just right

Small λ
$\lambda \approx 0$ overfit

how to choise λ

$\left[\begin{array}{l} J_{train}(\theta)\\ J_{cv}(\theta)\\ J_{test}(\theta)\end{array}\right]$ try λ=0, 0.01, 0.02, 0.04 …

… λ = 10


error
Jcv(θ)
Jtrain(θ)
m(trainigs set size)
m=1
m=2
m=3
m=4
m=5
m=6
hypothesis is ...
error
Jcv(θ)
Jtrain(θ)
m (trainingset size)

learning curves

→ $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$

→ $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)})-y_{cv}^{(i)})^2$

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$

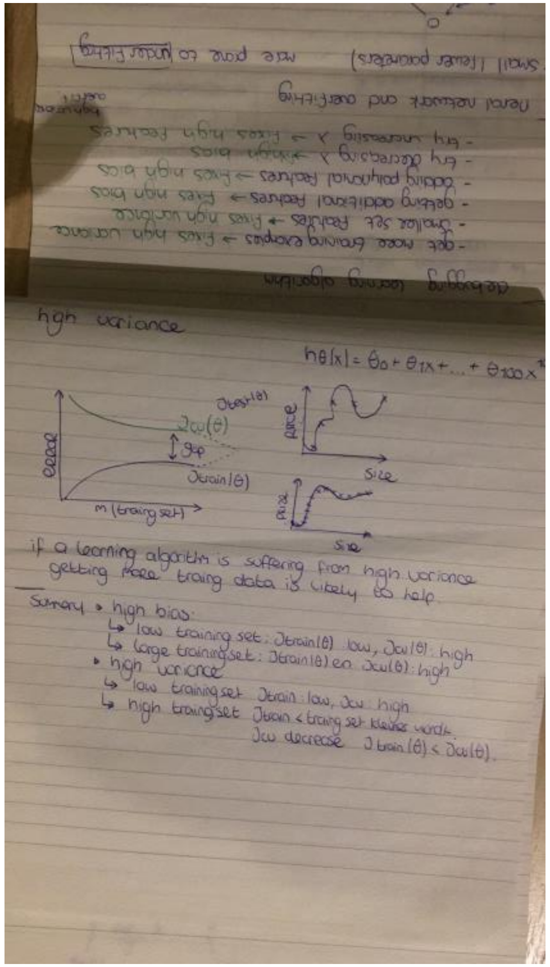
als m small is fit het gemakkelijker

high bias

$h_\theta(x) = \theta_0 + \theta_1 x$

high bias → Jcv(θ) en Jtrain(θ) dicht bij elkaar
↳ hoge waardes error

als m groter word blijft error redelijk stabiel.
↳ getting more training data will not help.


Jtest(θ)
Jcv(θ)
gap
Jtrain(θ)
error
m (traing set)
price
size
price
size

small (fewer parameters) more prone to underfitting

neral network and overfitting

- get more training examples → fixes high variance
- smaller set features → fixes high variance
- getting additional features → fixes high bias
- adding polynomial features → fixes high bias
- try decreasing λ → fixes high bias
- try increasing λ → fixes high features

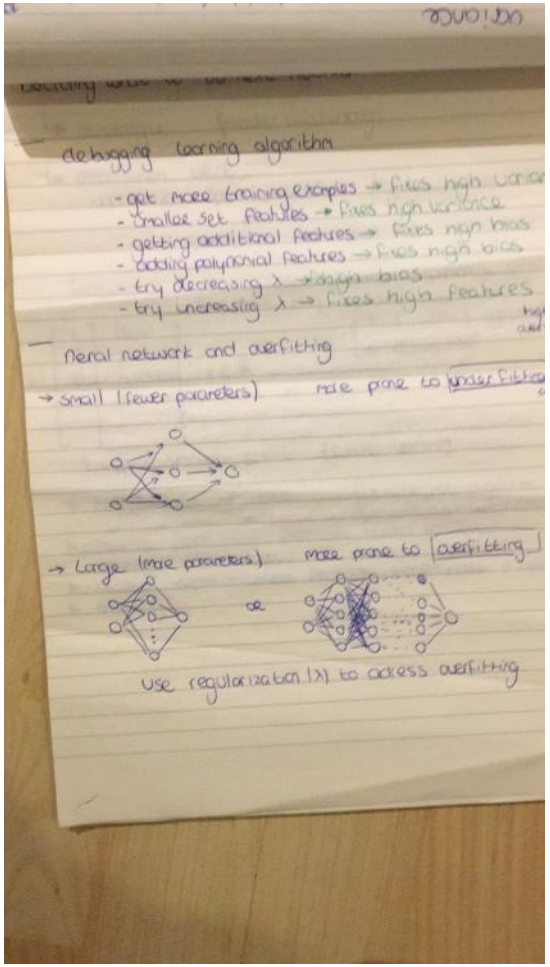
debugging learning algorithm

high variance

$h_\theta(x) = \theta_0 + \theta_1 x + \ldots + \theta_{100} x^{100}$

if a learning algorithm is suffering from high variance getting more traing data is likely to help.

Summary • high bias:
↳ low training set: Jtrain(θ) low, Jcv(θ): high
↳ large trainingset: Jtrain(θ) en Jcv(θ): high
• high variance
↳ low trainingset Jtrain low, Jcv high
↳ high traingset Jtrain < traing set kleiner wordt
Jcv decrease Jtrain(θ) < Jcv(θ).



variance

debugging learning algorithm

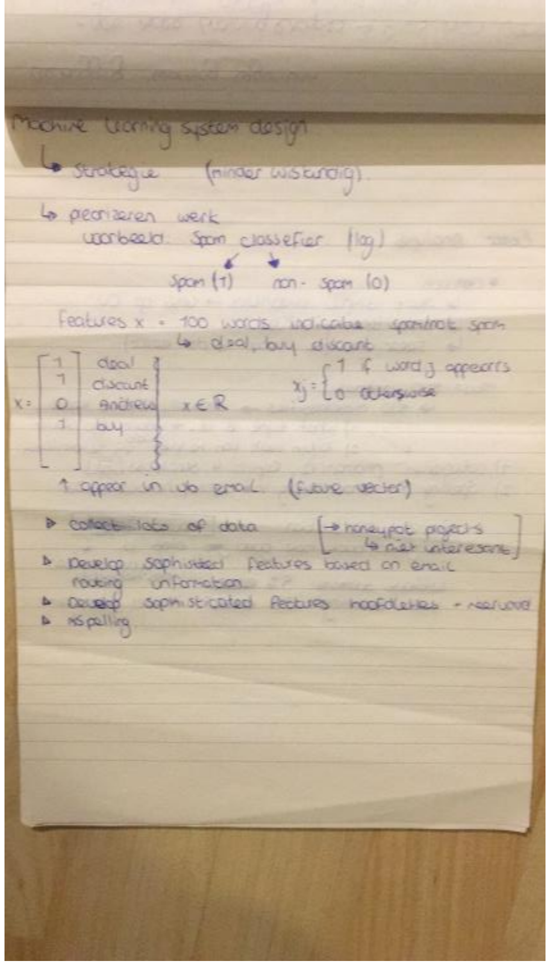
- get more training examples → fixes high varian
- smaller set features → fixes high variance
- getting additional features → fixes high bias
- adding polynomial features → fixes high bias
- try decreasing λ → fixes high bias
- try increasing λ → fixes high features

Neral network and overfitting

→ Small (fewer parameters) more prone to $\boxed{\text{underfitting}}$

→ Large (more parameters) more prone to $\boxed{\text{overfitting}}$

Use regularization (λ) to adress overfitting



Machine learning system design

↳ Strategie (minder wiskundig).

↳ prioriseren werk

voorbeeld: Spam classefier (log)

Spam (1) non- spam (0)

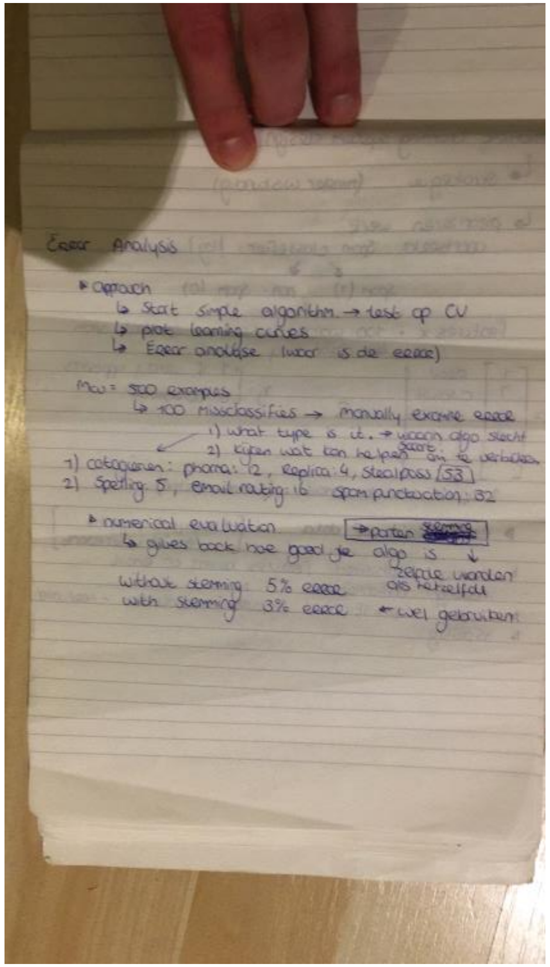
Features x = 100 words indicative spam/not spam

↳ deal, buy discount

$$x = \begin{array}{c l} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ \\ \end{bmatrix} & \begin{matrix} \text{deal} \\ \text{discount} \\ \text{andrew} \\ \text{buy} \\ \\ \end{matrix} \end{array} \quad x \in R \qquad x_j = \begin{cases} 1 & \text{if word j appears} \\ 0 & \text{otherwise} \end{cases}$$

↑ appear in de email (feature vector)

- collect lots of data [→ honeypot projects ↳ niet interessant]
- Develop sophisticated features based on email routing information.
- Develop sophisticated features hoofdletters - keerword
- misspelling



Error Analysis

- Approach
  - ↳ Start simple algorithm → test op CV
  - ↳ plot learning curves
  - ↳ Error analyse (waar is de error)

M_{cv} = 500 examples

↳ 100 missclassifies → manually examine error

1) what type is it. → waarom algo slecht scoort.

2) kijken wat kan helpen om te verbeteren

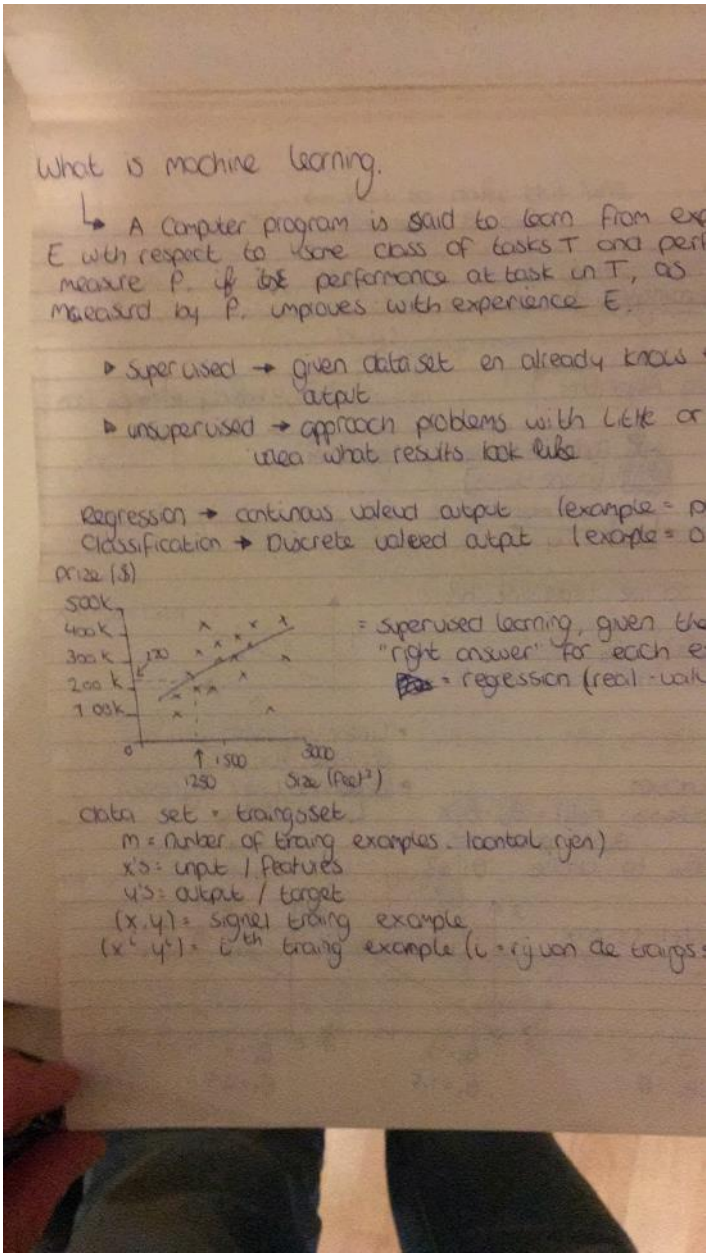
1) categorieën: pharma: 12, Replica: 4, Steal pass $\boxed{53}$

2) Spelling: 5, Email routing: 16 spam punctuation: 32

- numerical evaluation $\boxed{\text{→ porter stemming}}$
  - ↳ gives back hoe goed je algo is ↓ zelfde worden als hetzelfde

without stemming 5% error

with stemming 3% error ← wel gebruiken



What is machine learning.

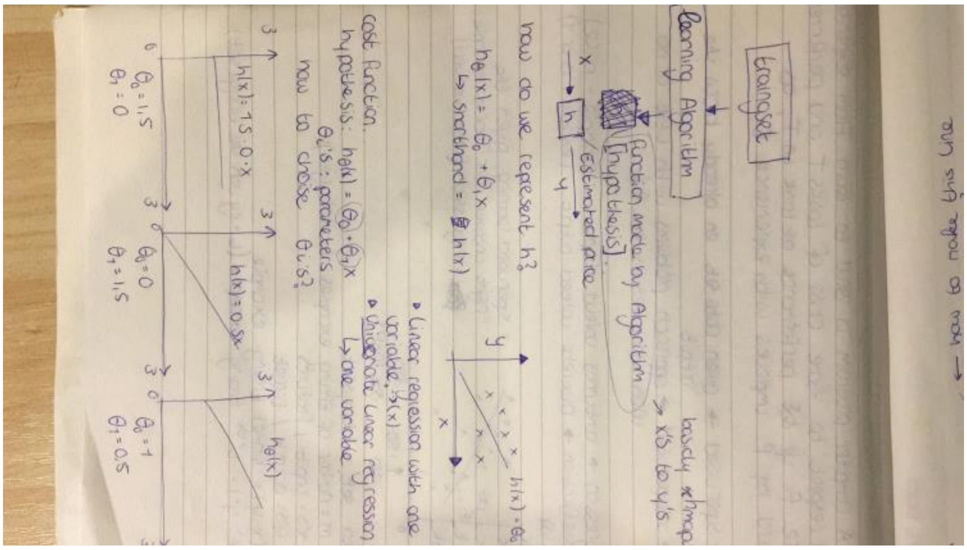
↳ A Computer program is said to learn from ex
E with respect to some class of tasks T and perf
measure P. if ~~its~~ performance at task in T, as
Measurd by P. improves with experience E.

- ▸ Supervised → given dataset en already know
  output
- ▸ unsupervised → approach problems with Little or
  idea what results look like

Regression → continous valued output (example = p
Classification → Discrete valeed output (example = o

= Supervised learning, given the
"right answer" for each e
~~[illegible]~~ = regession (real - valu

data set = traingset
m = number of traing examples . (aantal rijen)
x's = input / features
y's = output / target
(x, y) = signel traing example
(x^i y^i) = i^{th} traing example (i = rij van de traings



← how to make this line

traingset
learning Algorithm
x
h
y

basicly ~~s~~ map
→ x's to y's

function made by Algorithm
[hypothesis].

Estimated price

how do we represent h?

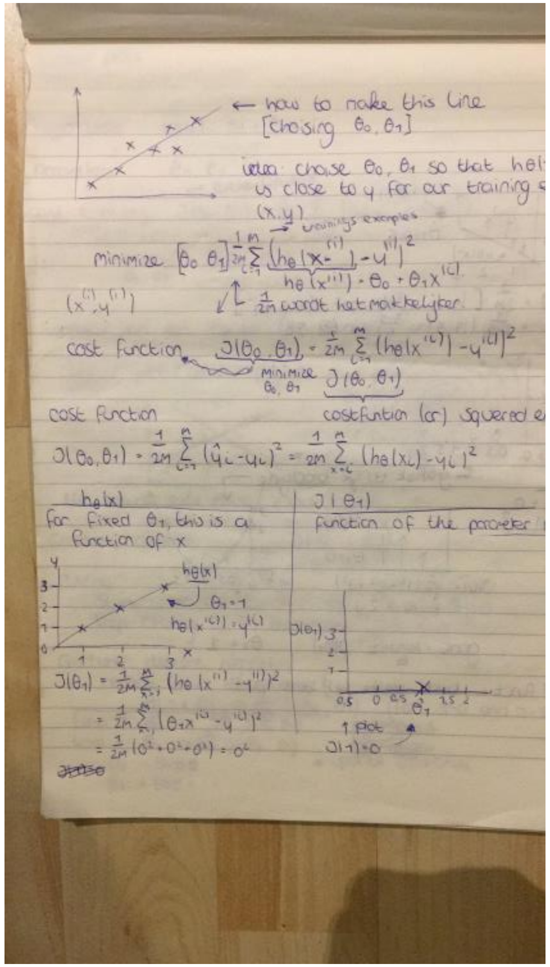
$h_\theta(x) = \theta_0 + \theta_1 x$
↳ shorthand = h(x)

y
x
h(x) = θ₀

- ▸ Linear regression with one variable. →(x)
- ▸ Univariate Linear regression
  ↳ one variable

cost function.
hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
θ_i's : parameters
how to choise θ_i's?

3
h(x) = 1.5 · 0·x
0
3
θ₀ = 1,5
θ₁ = 0
3
h(x) = 0·5x
0
3
θ₀ = 0
θ₁ = 1,5
3
h_θ(x)
0
θ₀ = 1
θ₁ = 0,5



← how to make this line [chosing θ_0, θ_1]

idea: chose θ_0, θ_1 so that h_θ is close to y for our training (x, y) → trainings exemples

minimize $[\theta_0\ \theta_1] \frac{1}{2m} \sum_{i=1}^{m} \underbrace{(h_\theta(x^{(i)}) - y^{(i)})^2}_{h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}}$

$(x^{(i)}, y^{(i)})$ ↑ $\frac{1}{2m}$ wordt het makkelijker

cost function $\underbrace{J(\theta_0, \theta_1)}$ $= \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$\underset{\theta_0, \theta_1}{\text{minimize}}\ \underbrace{J(\theta_0, \theta_1)}$

cost function — costfuntion (or) squered e

$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$

$h_\theta(x)$	$J(\theta_1)$
for fixed θ_1, this is a function of x	function of the parameter

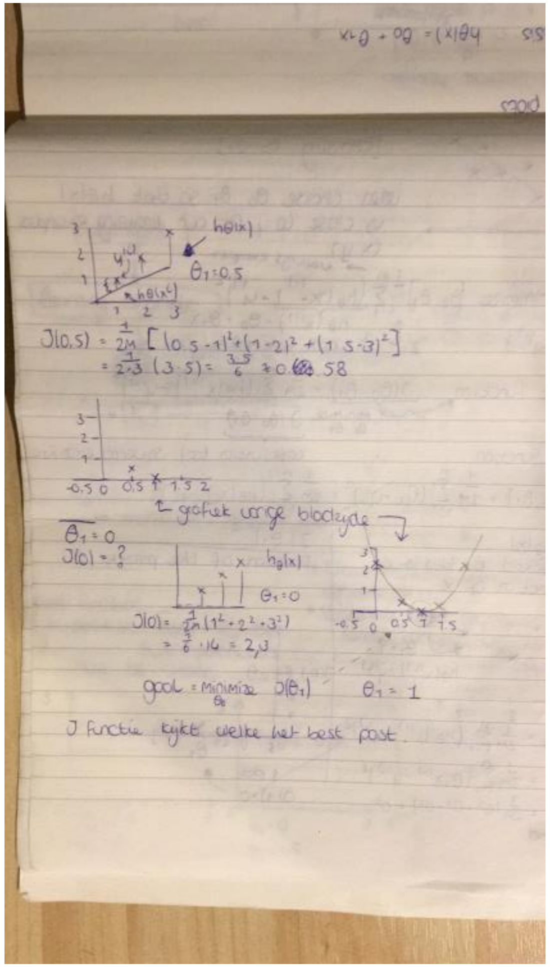
$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$= \frac{1}{2m} \sum_{i=1}^{m} (\theta_1 x^{(i)} - y^{(i)})^2$

$= \frac{1}{2m} (0^2 + 0^2 + 0^2) = 0^2$

↑ plot

$J(1) = 0$



$h\theta(x) = \theta_0 + \theta_1 x$

3
2
1
$y^{(i)}$
x^{i}
$h\theta(x^{i})$
1
2
3
$h\theta(x)$
$\theta_1 = 0.5$

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

$= \frac{1}{2 \cdot 3} (3.5) = \frac{3.5}{6} \approx 0.58$

3
2
1
-0.5
0
0.5
1
1.5
2

↑ grafiek vorige bladzyde →

$\theta_1 = 0$

$J(0) = ?$

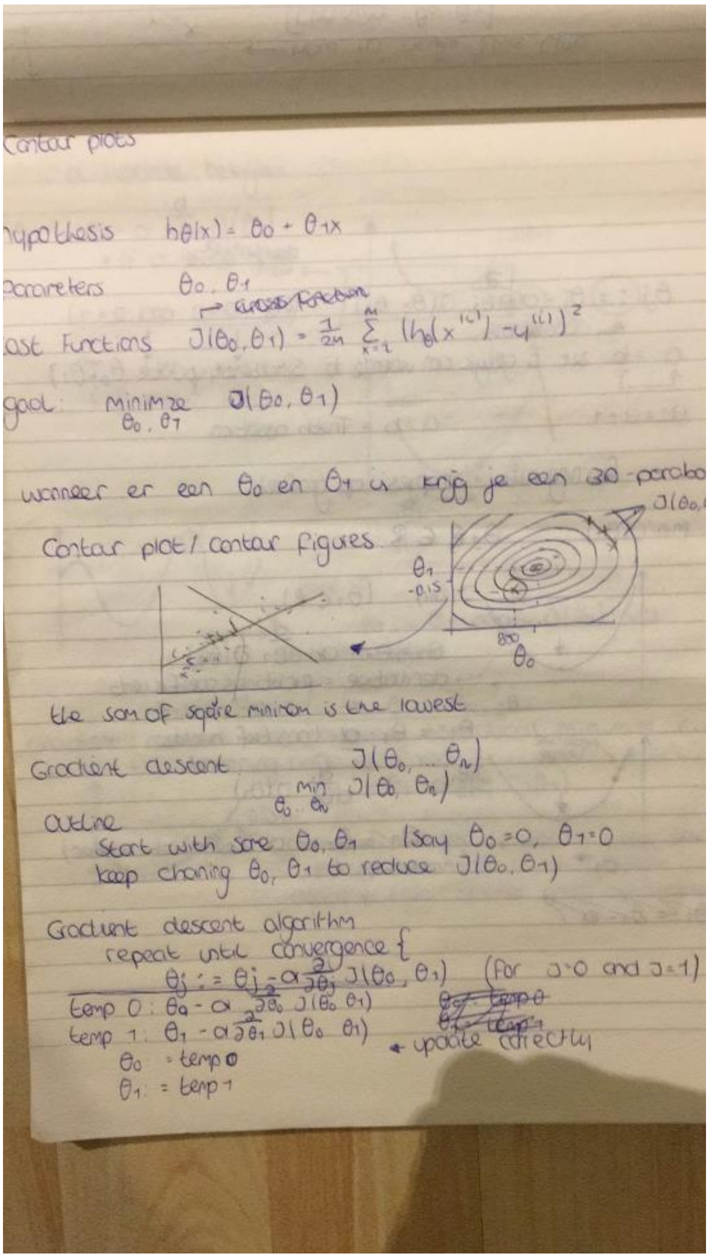
$h_\theta(x)$
$\theta_1 = 0$
3
2
1
-0.5
0
0.5
1
1.5

$J(0) = \frac{1}{2m} (1^2 + 2^2 + 3^2)$

$= \frac{1}{6} \cdot 14 = 2,3$

goal = $\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$ $\theta_1 = 1$

J functie kijkt welke het best past



Contour plots

ypothesis $h_\theta(x) = \theta_0 + \theta_1 x$

arameters θ_0, θ_1

→ cross function

ost Functions $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

goal: $\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$

wanneer er een θ_0 en θ_1 is krijg je een 3D-parabo

Contour plot / contour figures

J(θ0,
θ1
-0.15
800
θ0

the som of sqare minimum is the lowest

Gradient descent $J(\theta_0, \dots \theta_n)$

$\underset{\theta_0 \dots \theta_n}{\min} J(\theta_0 \; \theta_n)$

outline

Start with some θ_0, θ_1 (say $\theta_0 = 0$, $\theta_1 = 0$

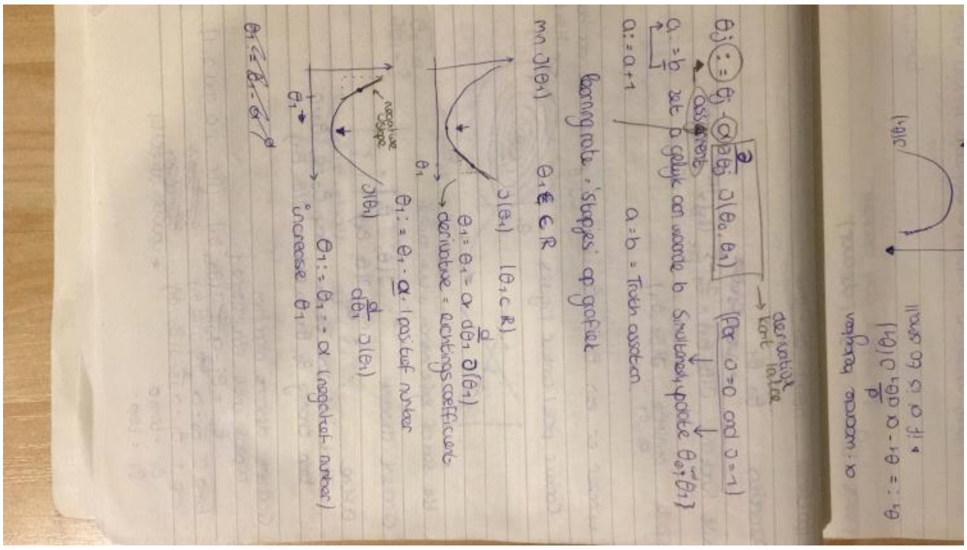
keep chaning θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

Gradient descent algorithm

```
repeat until convergence {
    $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$   (for $j = 0$ and $j = 1$)
temp 0: $\theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0 \; \theta_1)$
temp 1: $\theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0 \; \theta_1)$
    $\theta_0$ = temp 0
    $\theta_1$ = temp 1
```

~~$\theta_0 :=$ temp0~~
~~$\theta_1 :=$ temp1~~

← update correctly



α: waarde bepalen

$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

• if α is to small

$\boxed{\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}$ → derivative kort later

(for $j = 0$ and $j = 1$)

assignment

a := b zet a gelijk aan waarde b

Simultaneously update θ_0; θ_1}

a := a+1

a = b = Truth assotion

learning rate = 'stapjes' op grafiek

min $J(\theta_1)$ $\theta_1 \in R$

$J(\theta_1)$ ($\theta_1 \in R$)

$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

→ derivative = richtingscoefficient

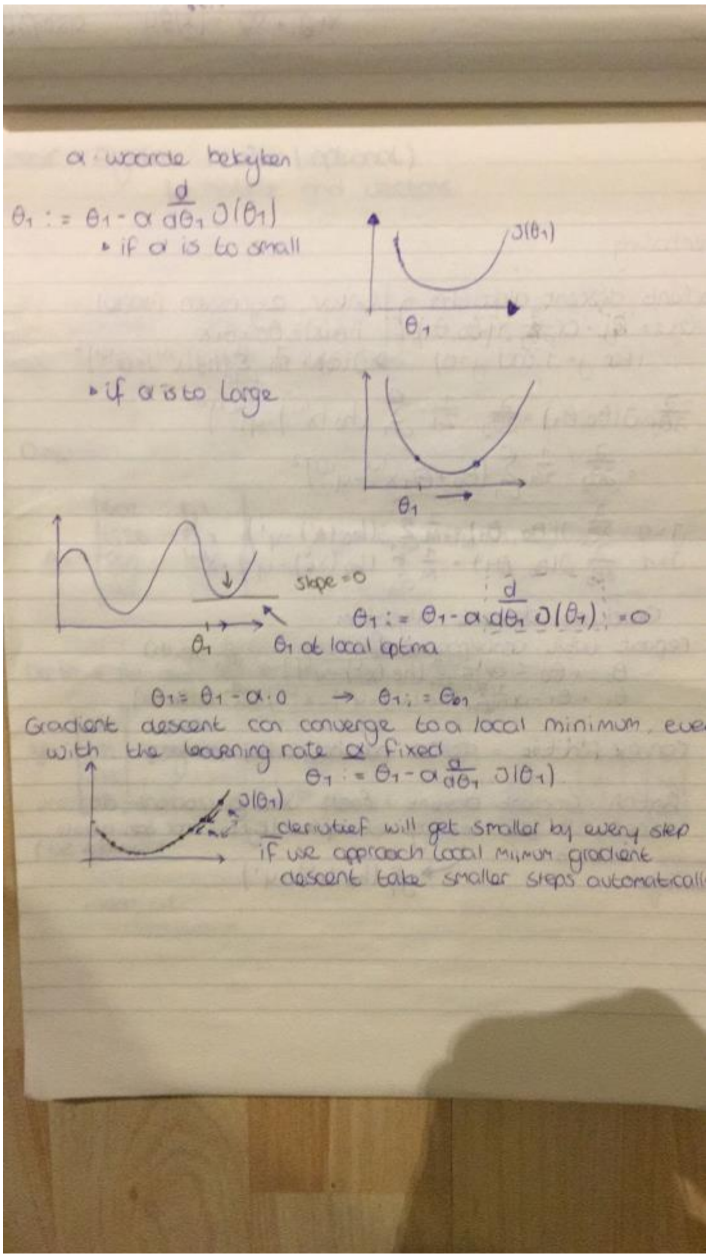
$\theta_1 := \theta_1 - \underline{\alpha} \cdot$ (positief number)

$\frac{d}{d\theta_1} J(\theta_1)$

$\theta_1 := \theta_1 - \alpha$ (negatief number)

increase θ_1

~~$\theta_1 := \theta_1 - \alpha \, d$~~



α woorde bekyken

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

- if α is to small

$J(\theta_1)$
θ_1

- if α is to Large

θ_1

slope = 0
θ_1
θ_1 at local optma

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1) = 0$$

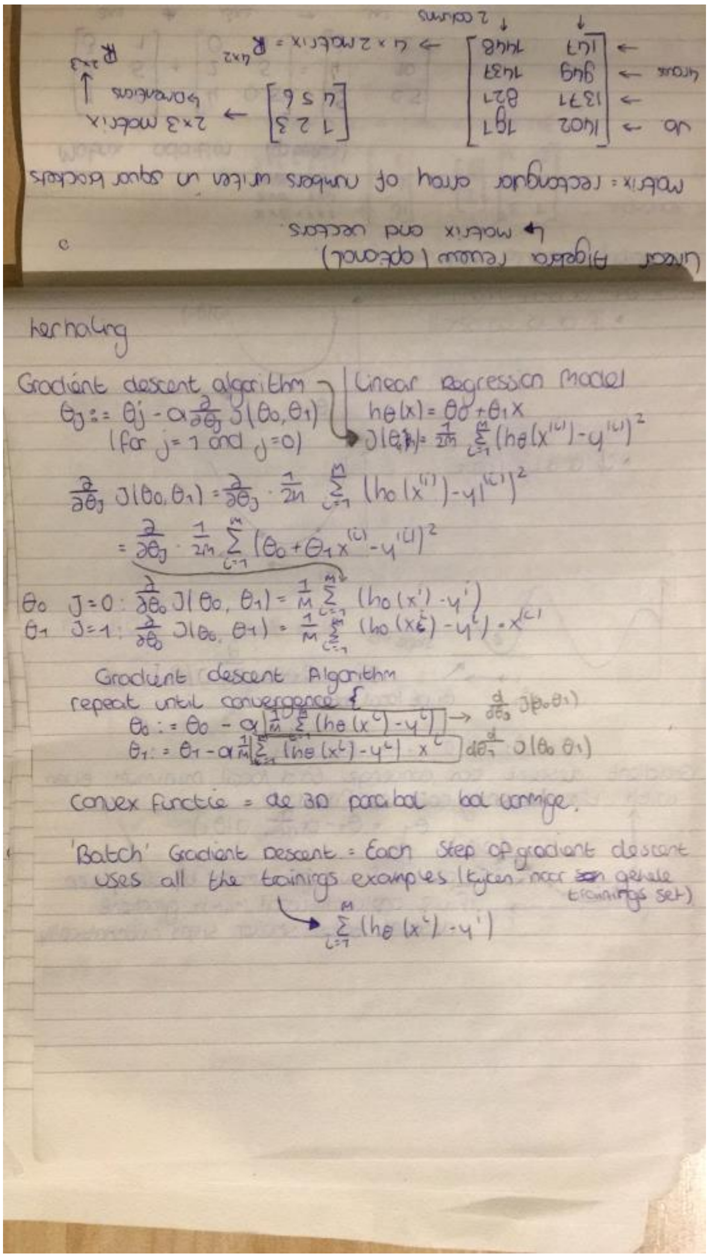
$\theta_1 := \theta_1 - \alpha \cdot 0$ → $\theta_1 := \theta_1$

Gradient descent can converge to a local minimum, eve
with the learning rate $\underline{\alpha}$ fixed

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

$J(\theta_1)$

derivatief will get smaller by every step
if we approach local minum gradient
descent take smaller steps automaticall



Linear Algebra review (optional)
↳ matrix and vectors.

Matrix = rectangular array of numbers writen in squar brackets

$$\begin{array}{ll} \text{uo.} \rightarrow & \\ \rightarrow & \\ \text{4rows} \rightarrow & \\ \rightarrow & \end{array} \begin{bmatrix} 1402 & 191 \\ 1371 & 821 \\ 949 & 1437 \\ 147 & 1448 \end{bmatrix} \rightarrow 4 \times 2 \text{ matrix} = \mathbb{R}^{4\times2}$$

↑ ↑ 2 columns

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow 2\times3 \text{ matrix}$$

↳ dimentions ↓ $\mathbb{R}^{2\times3}$

herhaling

Gradiënt descent algorithm

$\theta_j := \theta j - \alpha \frac{\partial}{\partial\theta_j} J(\theta_0, \theta_1)$

(for $j = 1$ and $j = 0$)

Linear Regression Model

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$$\frac{\partial}{\partial\theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial\theta_j} \cdot \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{\partial}{\partial\theta_j} \cdot \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

θ_0 $J = 0$: $\frac{\partial}{\partial\theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)$

θ_1 $J = 1$: $\frac{\partial}{\partial\theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i) \cdot x^{(i)}$

Gradient descent Algorithm

repeat until convergence {

$\theta_0 := \theta_0 - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)} \rightarrow \frac{d}{d\theta_0} J(\theta_0, \theta_1)$

$\theta_1 := \theta_1 - \alpha \frac{1}{m} \boxed{\sum_{i=1}^{m} (h_\theta(x^i) - y^i) \cdot x^i} \; \frac{d}{d\theta_1} J(\theta_0, \theta_1)$

convex functie = de 3D parabol bal vormige.

'Batch' Gradient Descent = Each Step of gradient descent uses all the trainings examples (kijken naar ~~een~~ gehele trainings set)

$\rightarrow \sum_{i=1}^{m} (h_\theta(x^i) - y^i)$