Gradient Descent

Michael Staniek

University of Heidelberg

November 8, 2017

Inhalt

Heutige Sitzung

- Wir werden die Grundlagen des Maschinellen Lernens durchgehen

Was braucht man dafür?

- Wir wollen eine Lösung für ein Problem
- Dieses muss sich Mathematisch ausdrücken lassen
- Wir müssen IMMER wissen, wann wir eine bessere Lösung haben als vorher => Fehlerfunktion
- Weitere Anforderung an die Fehlerfunktion => Ableitbar

Nochmal?

- Zum einen zeigt die Fehlerfunktion uns, **wie falsch wir liegen**
- Zum anderen zeigt die Ableitung der Fehlerfunktion uns, **wie wir uns verbessern können!**

Und wie geht das?

Nehmen wir an, wir wollen ein x finden, was folgende Funktion minimiert: $f(x) = (x-2)^2$
Dies ist unsere Fehlerfunktion, und in diesem Beispiel einfach nur eine Parabel, die 2 nach Rechts verschoben wurde Als schlaue Menschen können wir direkt die Lösung sagen, aber wir sind grade Maschinen. Probieren wir mal ein paar Zahlen aus.

Und wie geht das? (Part 2)

x	$f(x) = (x-2)^2$
-2	16
0	4
4	4
6	16

Durch ausprobieren können wir also sehen, mit welchen Punkten wir näher am Ergebnis (zwischen 0 und 4) sind als mit anderen. Wie hat man in der Schule gelernt, Minima (und Maxima) herauszufinden?

Und wie geht das? (Part 3)

x	$f(x) = (x-2)^2$	$f'(x) = 2 * (x-2)$
-2	16	-8
0	4	-4
4	4	4
6	16	8

Gradient Descent

Learningrate 0.1

x	$f(x) = (x-2)^2$	$f'(x) = 2 * (x-2)$	Update
0	4	-4	

Gradient Descent

Learningrate 0.1

x	$f(x) = (x-2)^2$	$f'(x) = 2 * (x-2)$	Update
0	4	-4	0.4
0.4	2.56	-3.2	

Gradient Descent

Learningrate 0.1

x	$f(x) = (x-2)^2$	$f'(x) = 2 * (x-2)$	Update
0	4	-4	0.4
0.4	2.56	-3.2	0.32

Gradient Descent

Learningrate 0.1

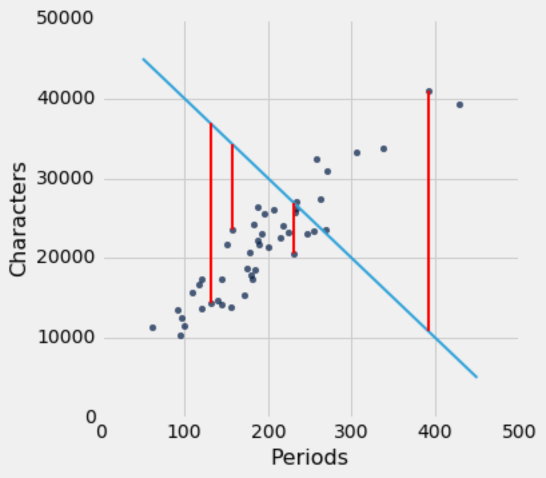
x	$f(x) = (x-2)^2$	$f'(x) = 2 * (x-2)$	Update
0	4	-4	0.4
0.4	2.56	-3.2	0.32
0.72	1.6384	-2.56	0.256

Gradient Descent

Learningrate 0.1

x	$f(x) = (x-2)^2$	$f'(x) = 2 * (x-2)$	Update
0	4	-4	0.4
0.4	2.56	-3.2	0.32
0.72	1.6384	-2.56	0.256
1.9	0.01	-0.2	0.02
2.1	0.01	0.2	-0.02

Squared Error



Neues Problem

Learningrate 0.1
Formel: $f(x) = w * x$
Wir wollen w optimieren, x und y haben wir gegeben Wir wollen den Squared Error minimieren: $L(x) = (w * x - y)^2$

x	y
0	0
1	0.621371

Wir brauchen die Ableitung des Fehlers nach w.

Gradient Descent

Learningrate 0.1

x	y	w	$(w * x - y)^2$	2*(w*x-y)*x
1	0.621371	0.5	0.014	-0.24

Gradient Descent

Learningrate 0.1

x	y	w	$(w * x - y)^2$	2*(w*x-y)*x
1	0.621371	0.5	0.014	-0.24
1	0.621371	0.524	0.009	-0.19

Gradient Descent

Learningrate 0.1

x	y	w	$(w * x - y)^2$	2*(w*x-y)*x
1	0.621371	0.5	0.014	-0.24
1	0.621371	0.524	0.009	-0.19
1	0.621371	0.543	0.006	-0.15

Lernrate ist sehr wichtig!

Learningrate 1

x	y	w	$(w * x - y)^2$	2*(w*x-y)*x
1	0.621371	0.6	0.0004	-0.04
1	0.621371	0.64	0.0003	0.037
1	0.621371	0.603	0.0003	-0.036

Klassifikation

- Bisher haben wir Zahlen vorhergesagt
- Wie sagen wir Kategorien Voraus?

x	y
(0.9, 0.8)	1
(0.9, -0.8)	-1

- Loss: max(0, -y*w*x)
- Intuition: Wenn w*x einen Wert über 0 und damit das Vorzeichen + hat, y aber den Wert -1 hat, kriegt man wenn man sie miteinander multipliziert eine negative Zahl.
- Von dieser nimmt man die Negation, da der Fehler ein Positiver Wert sein muss.
- Kein Fehler wenn Vorzeichen von (w*x) mit y übereinstimmt!
- Wenn Fehler größer als 0, Update. Ableitung?

Gradient Descent

Learningrate 0.1

x	y	w	$\max(0, -y * w * x)$	-y*x if y*w*x $>$ 0
(0.9,0.8)	1	(1,0.1)	0	(0, 0)
(0.9,-0.8)	-1	(1,0.1)	0.82	(0.9, -0.8)
(0.9,0.8)	1	(0.91,0.18)	0	(0, 0)
(0.9,-0.8)	-1	(0.91,0.18)	0.675	(0.9, -0.8)
(0.9,0.8)	1	(0.82,0.26)	0	(0, 0)
(0.9,-0.8)	-1	(0.82,0.26)	0.52	(0.9, -0.8)
(0.9,0.8)	1	(0.73,0.34)	0	(0, 0)
(0.9,-0.8)	-1	(0.73,0.34)	0.385	(0.9, -0.8)