KAPITEL 1

Wie neuronale Netze arbeiten

»Lass dich von all den kleinen Dingen um dich herum inspirieren.«

Leicht für mich – schwer für dich

Computer sind im Grunde nichts weiter als Rechenmaschinen. Arithmetische Aufgaben können sie äußerst schnell ausführen.

Damit sind sie prädestiniert für Aufgaben, die vor allem mit Rechnen zu tun haben – Zahlen addieren, um den Umsatz zu ermitteln, Prozentwerte bilden, um die Umsatzsteuer zu berechnen, Diagramme vorhandener Daten zeichnen usw.

Selbst beim Ansehen von Catch-up-TV oder beim Streamen von Musik hat der Computer nicht viel mehr zu tun, als immer und immer wieder einfache arithmetische Anweisungen auszuführen. Es mag Sie überraschen, doch auch die über das Internet übertragenen Videos, die aus Einsen und Nullen bestehen, werden mit arithmetischen Operationen rekonstruiert, die nicht komplexer sind als die Grundrechenarten, die wir in der Schule gelernt haben.

Zahlen wirklich schnell zu addieren – Tausende oder sogar Millionen pro Sekunde –, ist sicherlich eindrucksvoll, doch das hat nichts mit künstlicher Intelligenz zu tun. Einem Menschen erscheint es vielleicht schwer, schnell große Summen zu bilden, doch ist hierzu kaum Intelligenz erforderlich. Es genügt vollauf, die einfachsten Anweisungen zu befolgen, und genau das ist es, was die Elektronik in einem Computer realisiert.

Drehen wir nun den Spieß um und tauschen wir die Rolle mit dem Computer!

Sehen Sie sich die folgenden Bilder an und versuchen Sie, zu erkennen, was sie enthalten:



Abbildung 1-1: Bilderkennung – einfacher für den Computer oder für den Menschen?

Wir können ein Bild mit menschlichen Gesichtern, einer Katze und einem Baum sehen und erkennen. Praktisch sind wir dazu sehr schnell in der Lage und noch dazu mit einer ziemlich hohen Genauigkeit. Nur in wenigen Fällen liegen wir falsch.

Die recht großen Informationsmengen, die die Bilder enthalten, können wir sehr erfolgreich verarbeiten, um den Bildinhalt zu erfassen. Derartige Aufgaben sind für Computer nicht so einfach lösbar – es ist sogar unglaublich schwierig.

Tabelle 1-1: Wer kann was besonders gut verarbeiten?

Problem	Computer	Mensch
Tausende großer Zahlen schnell multiplizieren	Leicht	Schwer
Gesichter auf einem Foto mit einer Menschenmenge heraussuchen	Schwer	Leicht

Wir ahnen, dass für die Bilderkennung menschliche Intelligenz erforderlich ist – etwas, das Maschinen fehlt, egal wie komplex und leistungsfähig wir sie gebaut haben, weil es eben keine Menschen sind.

Doch es sind genau solche Probleme, die wir dem Computer übertragen möchten – denn Computer arbeiten schnell und werden nicht müde. Um derartige Probleme geht es bei der künstlichen Intelligenz.

Da Computer immer auf Elektronik basieren, besteht die Aufgabe der künstlichen Intelligenz darin, neue Rezepte bzw. *Algorithmen* zu finden, die auf neuartige Weise versuchen, derart schwierigere Probleme zu lösen. Selbst wenn das nicht perfekt gelingt, dann immerhin noch gut genug, um einen Eindruck von einer menschenähnlichen Intelligenz in der Praxis zu geben.

Kernideen

- Manche Aufgaben sind für herkömmliche Computer leicht, für Menschen aber schwer, beispielsweise das Multiplizieren von Millionen Zahlenpaaren.
- Andererseits sind manche Aufgaben für herkömmliche Computer schwer, für Menschen jedoch leicht, beispielsweise das Erkennen von Gesichtern auf einem Foto einer Menschenmenge.

Eine einfache Vorhersagemaschine

Wir beginnen supereinfach und bauen Schritt für Schritt darauf auf.

Stellen Sie sich eine simple Maschine vor, die eine Frage entgegennimmt, etwas »nachdenkt« und eine Antwort ausgibt. Das läuft genau wie im obigen Beispiel ab, in dem wir selbst die Eingaben über die Augen aufnehmen, mit unserem Gehirn die Szene analysieren und daraus ableiten, was die Objekte in dieser Szene bedeuten. Abbildung 1-2 stellt dies schematisch dar.

Abbildung 1-2: Schema einer einfachen Vorhersagemaschine

Computer denken nicht wirklich, sie sind lediglich bessere Taschenrechner. Deshalb wollen wir die Vorgänge mit treffenderen Worten beschreiben (siehe Abbildung 1-3).

Abbildung 1-3: Alternative Beschreibung der Vorhersagemaschine

Ein Computer nimmt eine Eingabe entgegen, führt bestimmte Berechnungen aus und liefert dann eine Ausgabe. Das folgende Beispiel soll das veranschaulichen. Es wird eine Eingabe von »3 x 4« verarbeitet. Das geschieht möglicherweise dadurch, dass die Multiplikation in einen einfacheren Satz von Additionen überführt wird. Die ausgegebene Antwort lautet »12«.

Abbildung 1-4: Beispiel für die Verarbeitung einer Multiplikation

Vielleicht denken Sie jetzt: »Was soll daran beeindruckend sein?« Das stimmt schon. Wir verwenden hier einfache und vertraute Beispiele. Damit veranschaulichen wir die Konzepte, die auf die interessanteren neuronalen Netze angewendet werden, die wir uns später ansehen.

Fahren wir die Komplexität jetzt eine winzige Stufe höher.

Stellen Sie sich eine Maschine vor, die Kilometer in Meilen umrechnet (siehe Abbildung 1-5).

Abbildung 1-5: Umrechnung von Kilometern in Meilen

Nun nehmen wir an, dass wir die Formel für die Umrechnung zwischen Kilometern und Meilen nicht kennen. Wir wissen lediglich, dass die Beziehung zwischen beiden *linear* ist. Wenn man also die Anzahl der Meilen verdoppelt, wird die gleiche Entfernung in Kilometern ebenfalls verdoppelt. Das ist intuitiv verständlich. Das Universum wäre ein seltsamer Ort, sollte dies nicht gelten!

Diese lineare Beziehung zwischen Kilometern und Meilen liefert uns einen Anhaltspunkt über diese geheimnisvolle Berechnung – sie muss die Form haben: `Meilen` = `Kilometer` × `c`, wobei `c` eine Konstante ist. Den Wert dieser Konstanten `c` kennen wir aber noch nicht.

Die einzigen anderen Anhaltspunkte liefern einige Beispiele, die Kilometer und Meilen paarweise angeben. Diese sind wie Beobachtungen der Wirklichkeit, mit denen man wissenschaftliche Theorien überprüft – sie sind Beispiele für die Wahrheit der echten Welt.

Tabelle 1-2: Wertepaare für die Umrechnung zwischen Kilometern und Meilen

Wahrheitsbeispiel	Kilometer	Meilen
1	0	0
2	100	62,137

Was sollten wir tun, um die fehlende Konstante `c` zu ermitteln? Setzen wir einfach einmal einen zufälligen Wert ein und probieren wir es aus! Versuchen wir es mit `c` = 0,5 und schauen wir, was passiert.

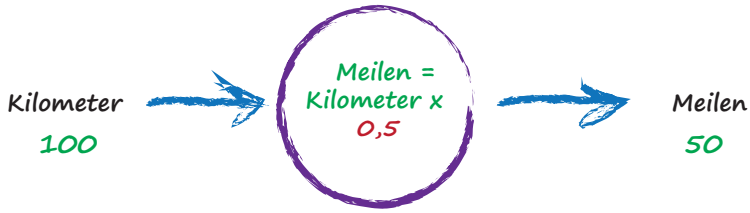

Abbildung 1-6: Zufällig gewählte Konstante c

Hier haben wir `Meilen = Kilometer × c`, wobei `Kilometer` gleich 100 und `c` unsere derzeitige Schätzung 0,5 sind. Damit erhalten wir 50 Meilen.

Nun gut. Das ist gar nicht mal so schlecht unter dem Aspekt, dass wir `c` = 0,5 zufällig ausgewählt haben! Doch wir wissen, dass das Ergebnis nicht genau ist, weil das Wahrheitsbeispiel Nummer 2 uns sagt, dass die Antwort 62,137 sein sollte.

Wir liegen um 12,137 daneben. Das ist der *Fehler*, die Differenz zwischen unserer berechneten Antwort und der tatsächlichen Wahrheit aus unserer Beispielliste. Das heißt,

```
Fehler = wahr - berechnet
= 62,137 - 50
= 12,137
```

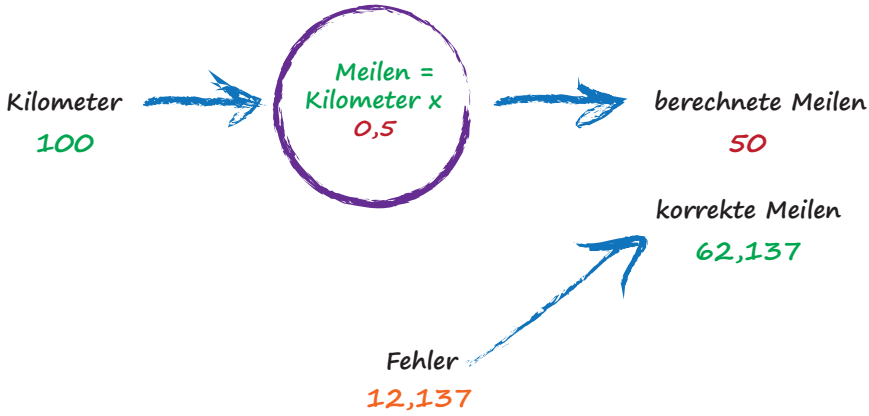

Abbildung 1-7: Der Fehler bei unserer ersten Schätzung

Was kommt als Nächstes? Wir wissen, dass wir falsch liegen und wie groß die Abweichung ist. Anstatt nun aufgrund dieses Fehlers zu verzweifeln, nutzen wir ihn, um zu einer zweiten, besseren Schätzung für `c` zu gelangen.

Sehen Sie sich diesen Fehler noch einmal an. Wir haben 12,137 zu wenig geschätzt. Da die Formel für die Umrechnung von Kilometern in Meilen eine lineare

Beziehung darstellt (`Meilen = Kilometer × c`), wissen wir, dass bei einer Erhöhung von `c` auch der Ausgabewert größer wird.

Wir erhöhen `c` von 0,5 auf 0,6 und sehen uns das neue Ergebnis an. Wenn `c` also auf 0,6 gesetzt ist, erhalten wir `Meilen = Kilometer × c = 100 × 0,6 = 60`. Das ist besser als die vorherige Antwort 50. Zweifellos haben wir einen Fortschritt gemacht!

Der Fehler ist nun mit 2,137 viel kleiner. Es könnte sogar ein Fehler sein, mit dem wir durchaus leben können.

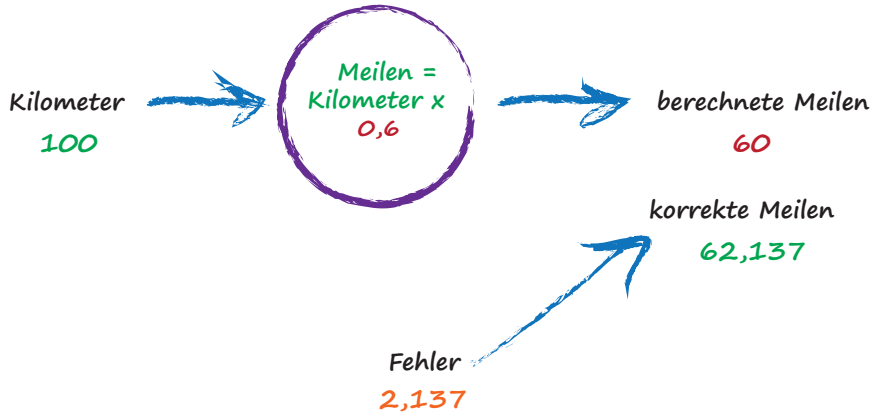

Abbildung 1-8: Die zweite Schätzung ergibt einen kleineren Fehler.

Wichtig ist hier, dass wir uns bei der Entscheidung, um wie viel der Wert von `c` angehoben werden soll, am Fehler orientiert haben. Wir wollten die Ausgabe 50 vergrößern, also haben wir `c` ein wenig erhöht.

Anstatt zu versuchen, den genauen Wert zu ermitteln, um den `c` sich ändern muss, fahren wir mit diesem Verfahren der Verfeinerung von `c` fort. Wenn Sie davon nicht überzeugt sind und meinen, es sei doch einfach genug, die genaue Antwort zu ermitteln, sollten Sie daran denken, dass vielen der interessanteren Probleme keine so einfache Formel zugrunde liegt, die Ausgabe und Eingabe in Beziehung bringt. Deshalb brauchen wir raffiniertere Methoden – wie zum Beispiel neuronale Netze.

Fahren wir also fort. Die Ausgabe von 60 ist immer noch zu klein. Wir schrauben den Wert von `c` ein weiteres Mal nach oben, und zwar von 0,6 auf 0,7.

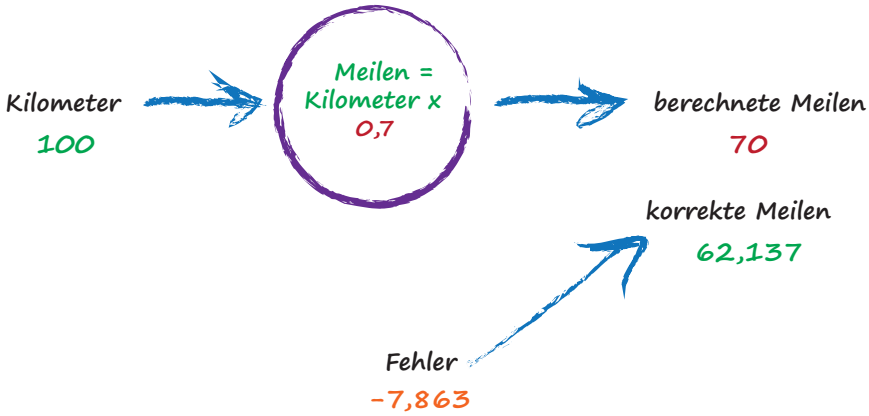

Abbildung 1-9: Die nächste Schätzung liefert einen negativen Fehler.

Oh, nein! Wir sind zu weit gegangen und über die korrekte Antwort hinausgeschossen. Der vorherige Fehler war 2,137, nun liegt er aber bei –7,863. Das Minuszeichen besagt lediglich, dass wir den Zielwert überschritten statt unterschritten haben. Denn der Fehler ergibt sich als `korrekter Wert - berechneter Wert`.

Da nun c = 0,6 besser als c = 0,7 war, könnten wir uns mit dem kleinen Fehler von c = 0,6 zufriedengeben und diese Übung jetzt beenden. Doch wir wollen noch ein wenig weitergehen. Wir könnten c doch auch in kleineren Schritten verändern, beispielsweise von 0,6 auf 0,61.

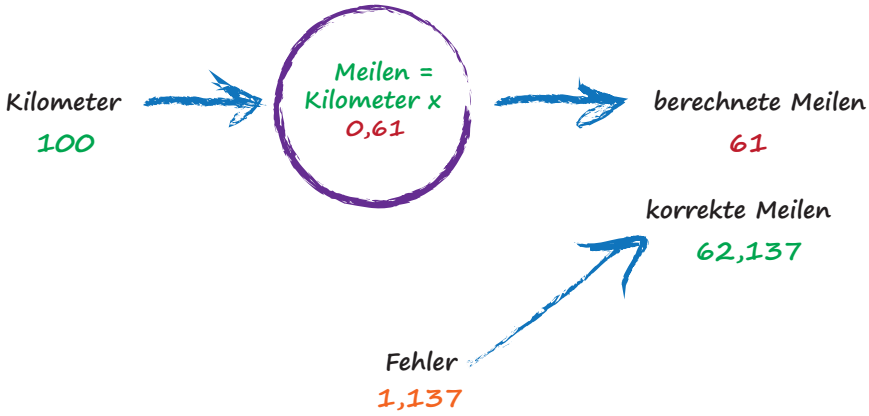

Abbildung 1-10: Die Konstante c wird jetzt in kleineren Schritten verändert.

Das ist schon viel besser als vorher. Wir haben einen Ausgabewert von 61, der nur 1,137 vom exakten Wert 62,137 abweicht.

Dieser letzte Versuch hat uns also gelehrt, dass wir den Wert von c moderat verändern sollten. Wenn die Ausgaben der korrekten Antwort näher kommen – d. h. der Fehler kleiner wird –, sollte man die veränderbare Komponente nicht so stark

anheben. Auf diese Weise lässt sich vermeiden, dass über den richtigen Wert hinausgeschossen wird, wie es weiter oben passiert ist.

Ohne uns nun dadurch ablenken zu lassen, wie denn c im Detail ermittelt wird, bleiben wir beim Konzept der sukzessiven Verfeinerung und wählen als Korrekturwert einen bestimmten Bruchteil des Fehlers. Das ist intuitiv richtig – ein großer Fehler bedeutet, dass eine größere Korrektur erforderlich ist, und ein winziger Fehler heißt, wir brauchen die kleinsten Anpassungsschritte für c.

Ob Sie es glauben oder nicht, wir sind gerade den prinzipiellen Lernprozess in einem neuronalen Netz durchgegangen – wir haben die Maschine trainiert, damit sie immer besser dabei wird, die richtige Antwort zu geben.

Es lohnt sich, kurz innezuhalten und darüber nachzudenken – wir haben ein Problem nicht in einem einzigen Schritt genau gelöst, wie wir es oftmals in der Schulmathematik oder bei wissenschaftlichen Problemen tun. Stattdessen haben wir einen gänzlich anderen Weg eingeschlagen, indem wir eine Antwort ausprobiert und sie wiederholt verbessert haben. Man spricht auch von *iterativ* und meint damit, dass eine Antwort wiederholt Stück für Stück verbessert wird.

Kernideen

- Alle nützlichen Computersysteme übernehmen eine Eingabe und liefern eine Ausgabe, wobei dazwischen bestimmte Berechnungen stattfinden. Neuronale Netze unterscheiden sich hiervon nicht.
- Wenn wir nicht genau wissen, wie etwas funktioniert, können wir versuchen, es mithilfe eines Modells abzuschätzen, dessen Parameter wir anpassen können. Hätten wir nicht gewusst, wie wir Kilometer in Meilen umrechnen, könnten wir eine lineare Funktion mit einem anpassbaren Gradienten als Modell verwenden.
- Eine gute Möglichkeit, dieses Modell zu verfeinern, besteht darin, die Parameter anzupassen, und zwar basierend darauf, wie falsch das Modell verglichen mit bekannten wahren Beispielen ist.

Klassifizieren unterscheidet sich nicht sehr vom Vorhersagen

Die obige einfache Maschine bezeichnen wir als *Prädiktor*, weil sie eine Eingabe übernimmt und eine Vorhersage darüber trifft, wie die Ausgabe sein sollte. Um diese Vorhersage zu verfeinern, haben wir einen internen Parameter angepasst. Dabei haben wir uns an dem Fehler orientiert, der gegenüber dem korrekten Wert bei einem bekannten wahren Beispiel auftritt.

Sehen Sie sich nun die Darstellung in Abbildung 1-11 an, die Messungen der Breite und Länge von Insekten zeigt.

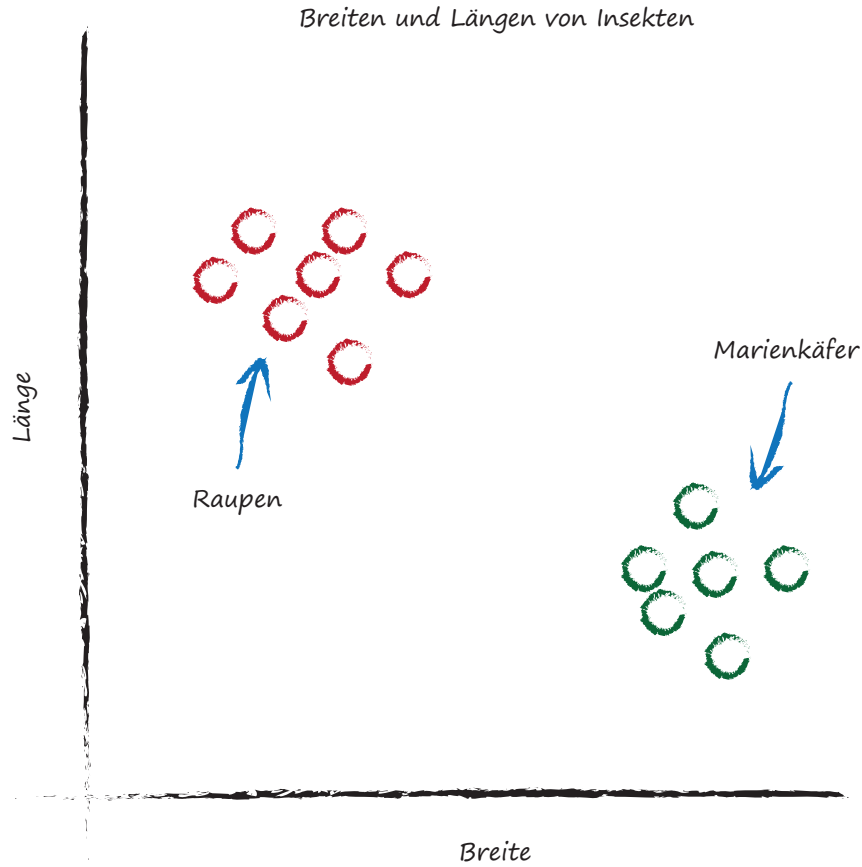

Abbildung 1-11: Diagramm mit Messdaten für die Breite und Länge von Raupen und Marienkäfern

Es lassen sich ganz klar zwei Gruppen ausmachen. Die Raupen sind dünn und lang, die Marienkäfer breit und kurz. (Um das Prinzip zu verdeutlichen, haben wir hier etwas nachgeholfen, Marienkäfer sind natürlich nicht ganz so breit).

Erinnern Sie sich noch an den Prädiktor, der versucht hat, die richtige Anzahl von Meilen für eine gegebene Anzahl von Kilometern herauszufinden? Dieser Prädiktor bestand im Kern aus einer anpassbaren linearen Funktion. Wie Sie wissen, ergeben lineare Funktionen gerade Linien, wenn man ihre Ausgabewerte über den Eingaben als Diagramm darstellt. Der anpassbare Parameter `c` hat den Anstieg dieser Geraden verändert.

Was passiert, wenn wir über diese Punkte eine gerade Linie legen?

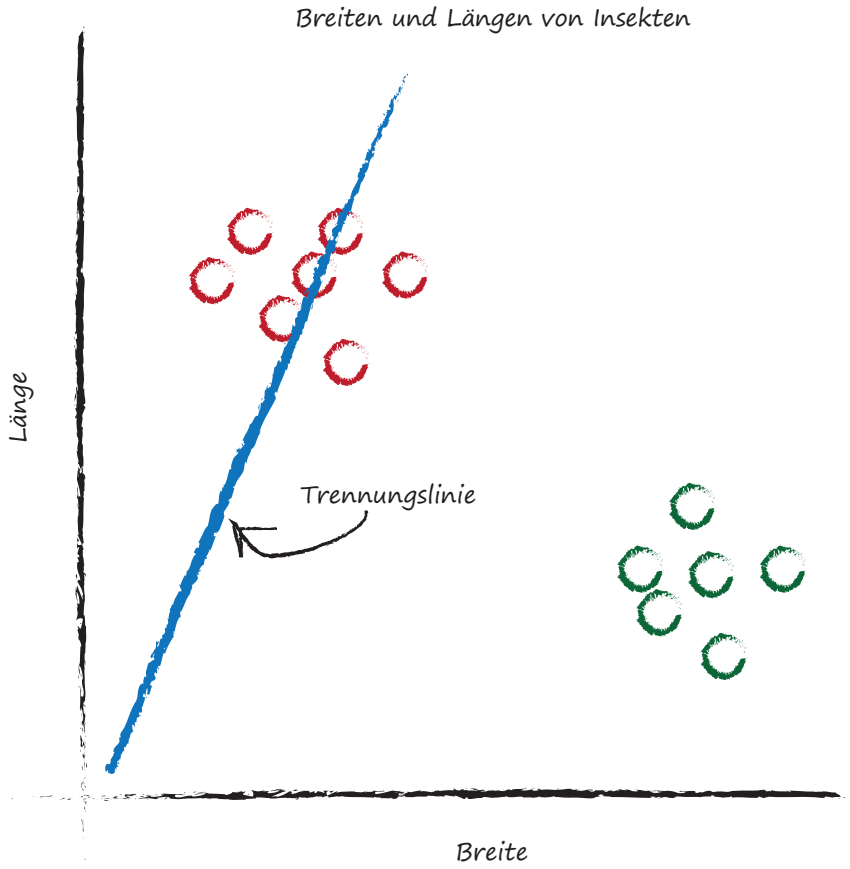

Abbildung 1-12: Trennungslinie durch eine Punktwolke von Messdaten

Die Linie können wir nicht in der gleichen Weise wie zuvor verwenden – um eine Zahl (Kilometer) in eine andere (Meilen) zu konvertieren –, doch vielleicht können wir die Linie nutzen, um verschiedene Arten von Dingen zu trennen.

Wenn die Linie in der obigen Punktdarstellung die Raupen von den Marienkäfern trennen würde, könnte man sie verwenden, um anhand der Messwerte ein unbekanntes Insekt zu *klassifizieren*. Die in Abbildung 1-12 eingezeichnete Linie leistet das noch nicht, da die Hälfte der Raupen auf derselben Seite der Trennungslinie wie die Marienkäfer liegt.

Probieren wir eine andere Linie aus, indem wir wieder den Anstieg der Geraden anpassen, und sehen wir uns die Wirkung an (siehe Abbildung 1-13).

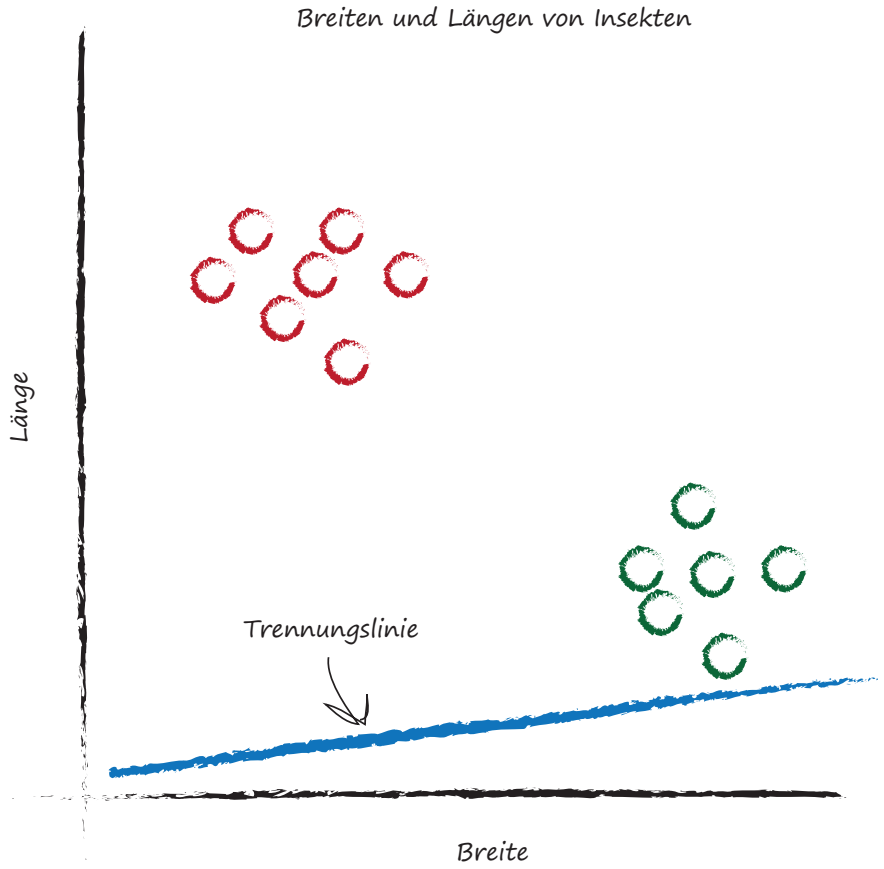

Abbildung 1-13: Trennungslinie mit anderem Anstieg

Dieses Mal ist die Trennungslinie sogar völlig unnütz! Sie trennt die beiden Insektenarten überhaupt nicht.

Abbildung 1-14 zeigt einen weiteren Versuch.

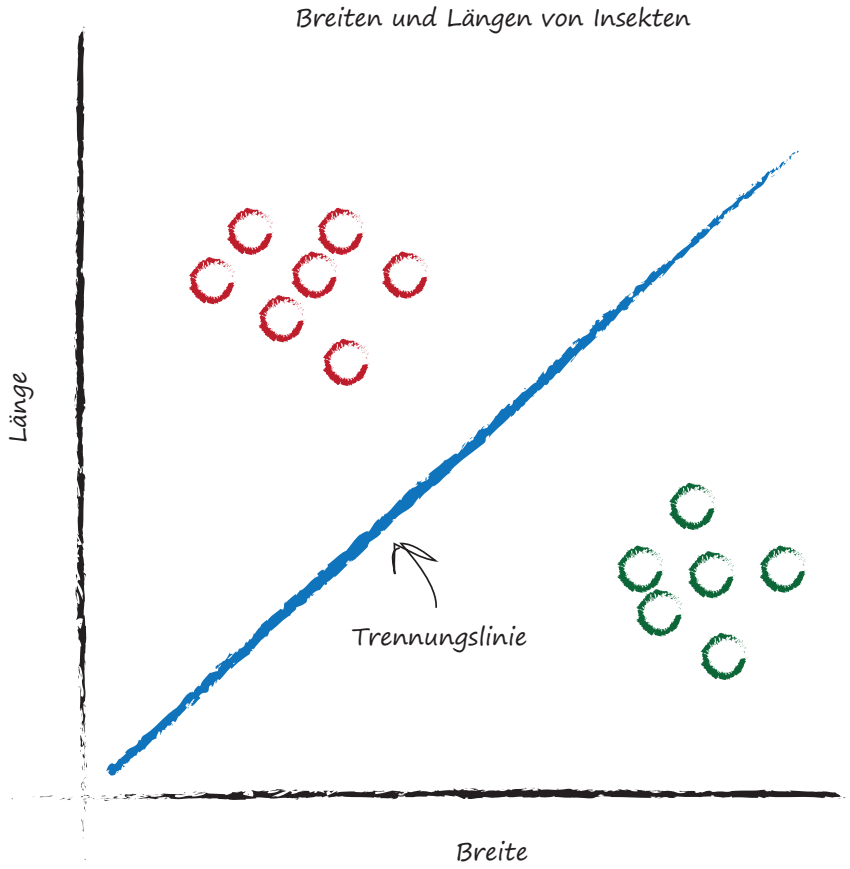

Abbildung 1-14: Dieses Mal ist die Trennungslinie brauchbar.

Das ist wesentlich besser! Diese Linie trennt die Raupen ganz sauber von den Marienkäfern. Jetzt können wir diese Linie als *Klassifizierer* von Insekten verwenden.

Wir nehmen hier an, dass es in unserer Welt nur zwei Arten von Insekten gibt – doch das ist fürs Erste in Ordnung, denn wir wollen lediglich das Konzept eines einfachen Klassifizierers veranschaulichen.

Stellen wir uns als Nächstes vor, dass sich unser Computer mit einem Roboterarm ein neues Insekt greift, seine Breite und Höhe misst und es dann mithilfe der obigen Linie korrekt als Raupe oder Marienkäfer klassifizieren kann.

In Abbildung 1-15 wurden die Daten für ein unbekanntes Insekt eingetragen. Es ist zweifelsfrei eine Raupe, weil der Punkt oberhalb der Linie liegt. Diese Klassifizierung ist zwar einfach, aber schon ziemlich leistungsfähig!

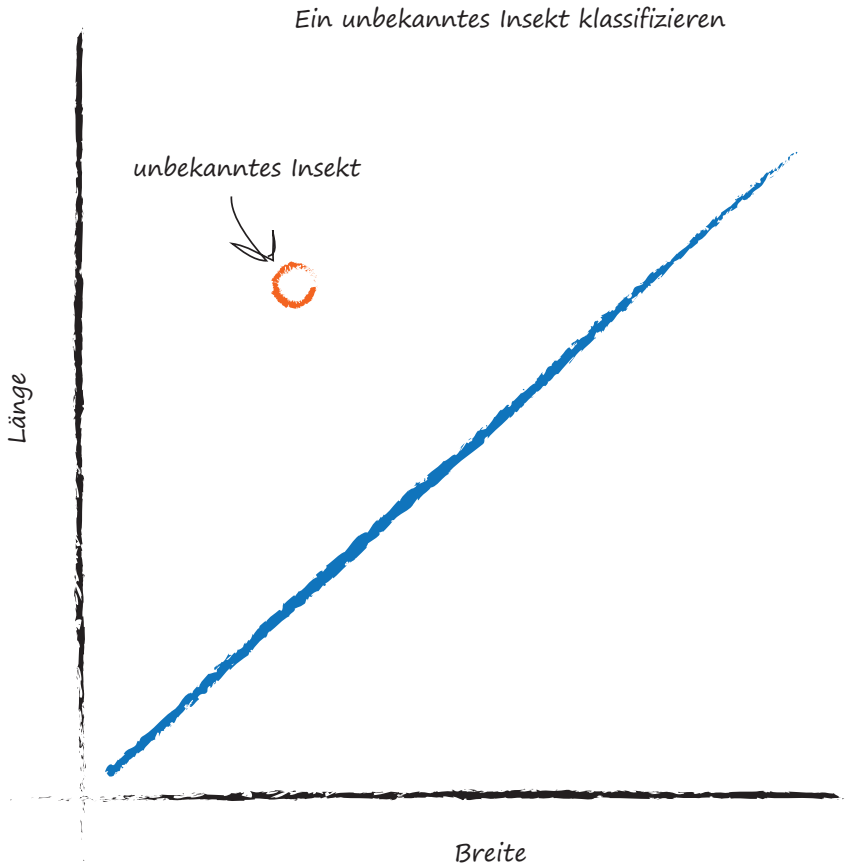

Abbildung 1-15: In das Diagramm wurden die Daten für ein unbekanntes Insekt eingetragen.

Wir haben nun gesehen, wie eine lineare Funktion innerhalb eines einfachen Prädiktors genutzt werden kann, um vorher noch nicht gesehene Daten zu klassifizieren.

Doch wir sind über ein entscheidendes Element hinweggegangen. Wie kommen wir zum richtigen Anstieg? Wie verbessern wir eine Linie, die bekanntermaßen kein guter Teiler zwischen den beiden Insektenarten ist?

Die Antwort darauf bringt uns wieder zu dem Grundprinzip, wie neuronale Netze lernen. Und das schauen wir uns als Nächstes an.

Einen einfachen Klassifizierer trainieren

Wir wollen unseren linearen Klassifizierer *trainieren*, um Insekten richtig als Marienkäfer oder Raupe klassifizieren zu können. Wie wir oben gesehen haben, ist dazu lediglich der Anstieg der Trennungslinie anzupassen, um die beiden Gruppen von Punkten in einem Diagramm mit großer Breite und Höhe zu trennen.

Wie bewerkstelligen wir das?

Anstatt im Voraus irgendeine mathematische Theorie zu entwickeln, wollen wir uns durch Versuche vorantasten. Auf diese Weise werden wir die Mathematik besser verstehen.

Wir brauchen einige Beispiele, um daraus zu lernen. Tabelle 1-3 zeigt lediglich zwei Beispiele, um diese Übung einfach zu halten.

Tabelle 1-3: Zwei Beispiele, um daraus zu lernen

Beispiel	Breite	Länge	Insekt
1	3,0	1,0	Marienkäfer
2	1,0	3,0	Raupe

Wir haben ein Beispiel für ein Insekt, das 3,0 breit und 1,0 lang ist und das uns als Marienkäfer bekannt ist. Außerdem haben wir ein Beispiel für ein Insekt, das mit 3,0 länger und mit 1,0 dünner ist und das wir als Raupe kennen. (Die Maße sind in Längeneinheiten angegeben, die von den automatischen Messsensoren gemeldet werden. Bei Bedarf können Sie sie in Millimeter oder Zoll umrechnen.)

Bei diesem Beispieldatensatz wissen wir, dass er die Wahrheit widerspiegelt. Anhand solcher Beispiele lässt sich der Anstieg der Klassifizierungsfunktion verfeinern. Wahre Beispiele für die Lernphase eines Prädiktors oder Klassifizierers werden als *Trainingsdaten* bezeichnet.

Wir tragen die beiden Trainingsdaten in ein Diagramm ein (siehe Abbildung 1-16). Oftmals ist es sehr hilfreich, Daten zu visualisieren, um sie besser zu verstehen und ein Gefühl für sie zu bekommen. Wenn man sich lediglich eine Liste oder Tabelle mit Zahlen ansieht, ist das nicht so einfach zu erreichen.

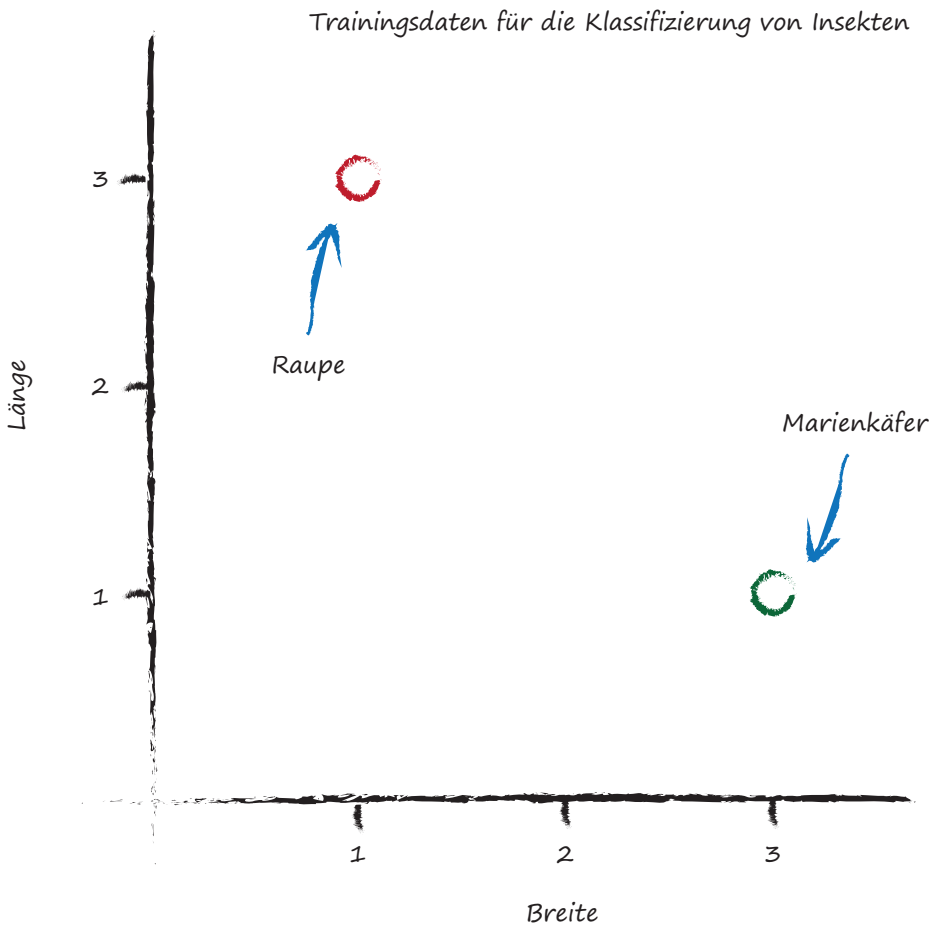

Abbildung 1-16: Sichere Messwerte für zwei Arten von Insekten, die als Trainingsdaten dienen

Wir beginnen mit einer zufälligen Trennungslinie, nur um irgendwo anfangen zu können. Bei unserem Prädiktor für die Umrechnung von Kilometern in Meilen hatten wir eine lineare Funktion, deren Parameter wir anpassten. Hier können wir genauso vorgehen, weil die Trennungslinie eine Gerade ist:

```
y = Ax
```

Wir haben bewusst die Namen `y` und `x` anstelle von Länge und Breite verwendet, weil hier die Linie genau genommen kein Prädiktor ist. Sie konvertiert nicht Breite in Länge, wie sie weiter oben Kilometer in Meilen umgerechnet hat. Stattdessen ist sie eine Trennungslinie, ein Klassifizierer.

Außerdem haben Sie sicherlich bemerkt, dass dieses `y = Ax` einfacher ist als die vollständigere Form einer Geraden: `y = Ax + B`. Dieses Szenario für Insekten haben wir absichtlich so einfach wie möglich gehalten. Lässt man für `B` einen Wert ungleich null zu, heißt das nur, dass die Gerade nicht durch den Ursprung des Diagramms geht, was in unserem Szenario nichts Brauchbares beisteuert.

Wie bereits zu sehen war, steuert der Parameter `A` den Anstieg der Geraden. Je größer `A` ist, desto größer ist der Anstieg.

Für den Anfang wählen wir `A = 0,25`. Die Gleichung der Trennungslinie lautet dann `y = 0,25x`. Diese Linie tragen wir in das Diagramm der Trainingsdaten ein, wie Abbildung 1-17 zeigt.

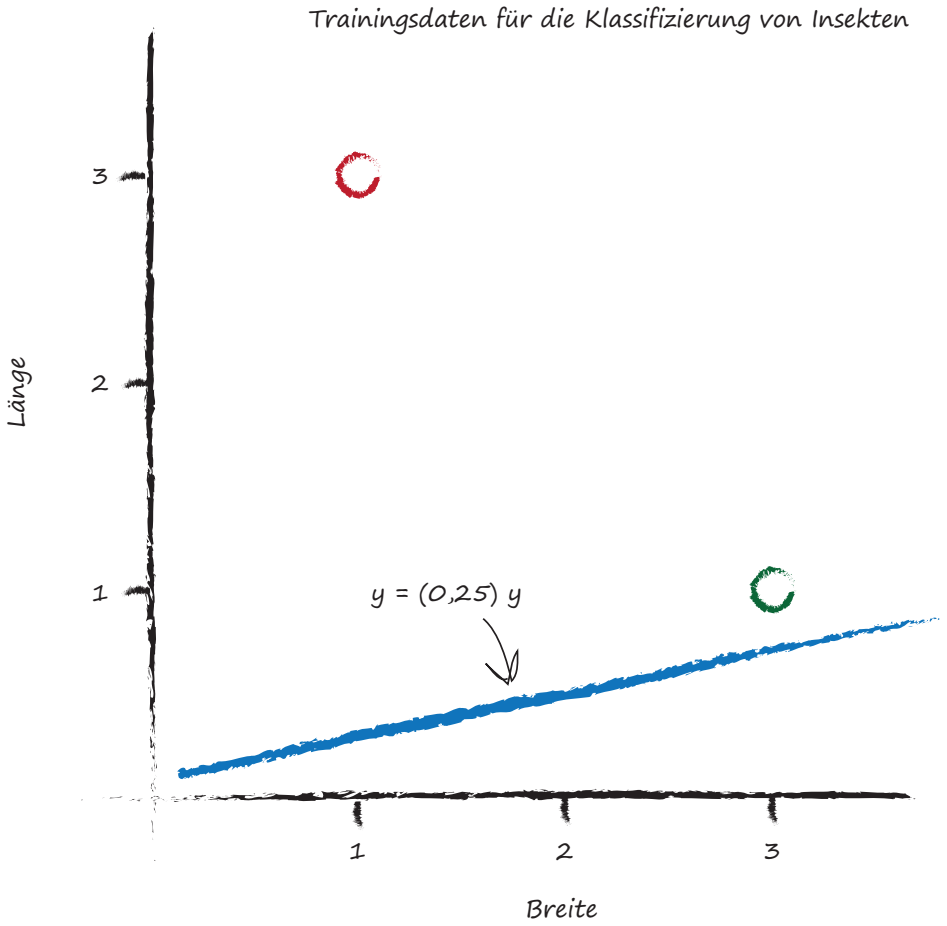

Abbildung 1-17: Trainingsdaten mit der anfänglichen Trennungslinie

Es ist auf einen Blick zu sehen, dass die Linie `y = 0,25x` ohne weitere Berechnungen noch kein guter Klassifizierer ist. Die Gerade teilt die beiden Insektenarten nicht. Wir können nicht sagen: »Wenn das Insekt oberhalb der Linie liegt, ist es eine Raupe«, weil auch der Marienkäfer über der Linie liegt.

Es liegt also nahe, die Linie etwas nach oben zu verschieben. Wir widerstehen der Versuchung, das zu tun, indem wir mit einem Blick auf das Diagramm eine geeignete Linie zeichnen. Wir wollen ja schließlich sehen, ob wir für dieses Vorgehen ein wiederholbares Rezept finden, d. h. eine Reihe von Computerbefehlen, die Informatiker als *Algorithmus* bezeichnen.

Sehen wir uns das erste Trainingsbeispiel an: Für einen Marienkäfer beträgt die Breite 3,0 und die Länge 1,0. Wenn wir die Funktion `y = Ax` mit diesem Beispiel testen, wobei `x` gleich 3,0 ist, erhalten wir:

```
y = (0,25) * (3,0) = 0,75
```

Die Funktion, bei der der Parameter `A` auf den zufällig gewählten Anfangswert 0,25 gesetzt ist, legt nahe, dass für ein Insekt mit der Breite 3,0 die Länge 0,75 betragen sollte. Wir wissen aber, dass das zu klein ist, weil aus dem Trainingsdatenbeispiel hervorgeht, dass es eine Länge von 1,0 haben muss.

Wir haben also eine Differenz, einen *Fehler*. Genau wie zuvor beim Prädiktor »Kilometer in Meilen« können wir uns an diesem Fehler orientieren, wenn wir den Parameter `A` anpassen.

Doch bevor wir das tun, sollten wir noch einmal über den Wert von `y` nachdenken. Wenn `y` gleich 1,0 ist, verläuft die Gerade direkt durch den Punkt (`x`, `y`) = (3,0; 1,0), an dem der Marienkäfer sitzt. Wir wollen das eigentlich nicht, die Linie soll oberhalb dieses Punkts verlaufen. Warum? Weil die Punkte für Marienkäfer unterhalb der Linie liegen sollen und nicht auf ihr. Die Gerade soll eine Trennungslinie zwischen Marienkäfern und Raupen sein, kein Prädiktor für die Länge eines Insekts bei gegebener Breite.

Probieren wir also, auf `y` = 1,1 abzuzielen, wenn `x` = 3,0 ist. Es ist lediglich eine kleine Zahl über 1,0; wir hätten auch 1,2 oder sogar 1,3 wählen können, doch wir wollen keine größere Zahl wie zum Beispiel 10 oder 100, weil die Gerade dabei höchstwahrscheinlich sowohl oberhalb der Marienkäfer als auch oberhalb der Raupen liegen würde, was eine Trennungslinie ergäbe, die überhaupt nicht brauchbar ist.

Der gesuchte Sollwert ist also 1,1, und der Fehler `E` beträgt:

```
Fehler = (Sollwert - Istwert)
```

Damit erhalten wir

```
E = 1,1 - 0,75 = 0,35
```

Legen wir eine kurze Pause ein und machen wir uns klar, was der Fehler, der Sollwert und der berechnete Istwert visuell bedeuten.

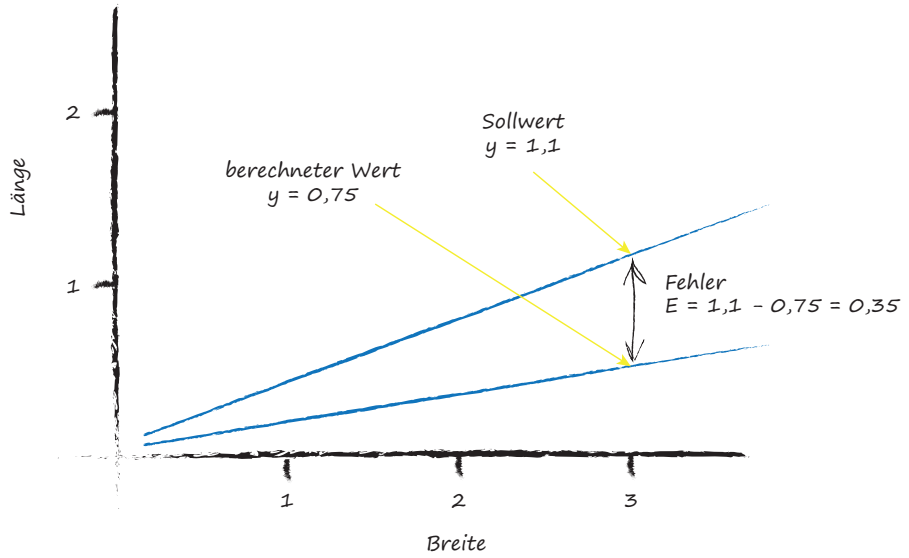

Abbildung 1-18: Visualisierung von Fehler, Sollwert und berechnetem Istwert

Was stellen wir nun mit diesem `E` an, um zu einem besseren, verfeinerten Parameter `A` zu gelangen? Das ist die entscheidende Frage.

Überlegen wir noch einmal. Wir wollen den Fehler, den wir `E` nennen, in `y` verwenden, um die erforderliche Änderung für Parameter `A` zu liefern. Dazu müssen wir wissen, wie die beiden miteinander in Beziehung stehen. Wie steht `A` in Beziehung zu `E`? Wenn wir das wüssten, könnten wir auch verstehen, wie sich die eine Änderung auf die andere auswirkt.

Beginnen wir mit der linearen Funktion für den Klassifizierer:

```
y = Ax
```

Wir wissen, dass dies bei anfänglichen Schätzungen von `A` die falsche Antwort für `y` ergibt, was der Wert entsprechend den Trainingsdaten sein sollte. Die Korrektur zum Sollwert nennen wir `t` für *target value* (Sollwert). Um diesen Wert `t` zu erhalten, müssen wir `A` um einen geringen Betrag anpassen. Mathematiker verwenden den griechischen Großbuchstaben Δ in der Bedeutung »eine kleine Änderung in«. Ausgeschrieben, sieht das so aus:

```
t = (A + ΔA)x
```

Bildlich lässt sich das Ganze viel einfacher veranschaulichen. In Abbildung 1-19 ist der neue Anstieg (`A + ΔA`) zu sehen.

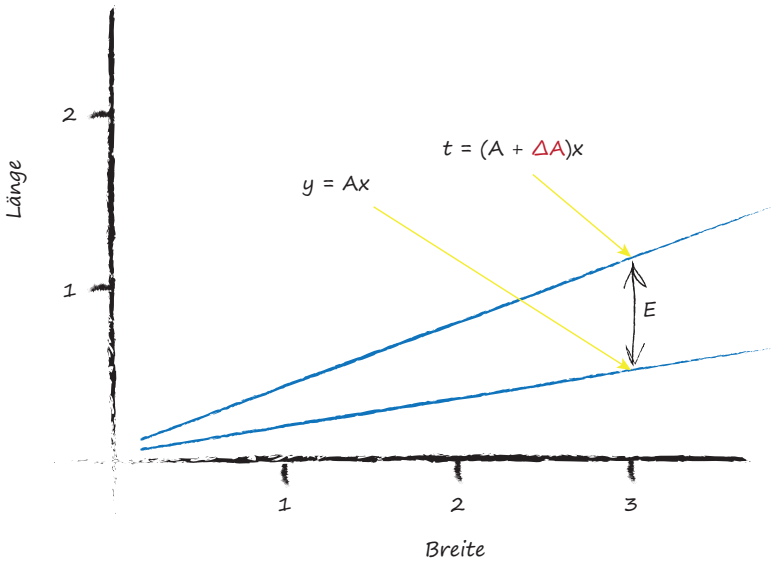

Abbildung 1-19: Neuer Anstieg für die Trennungslinie

Denken Sie daran, dass der Fehler `E` die Differenz zwischen dem korrekten Sollwert und dem Wert ist, den wir basierend auf unserer aktuellen Schätzung für `A` berechnet haben. Das heißt, dass `E` gleich `t - y` ist.

Um das deutlich zu machen, schreiben wir wie folgt:

```
t - y = (A + ΔA)x - Ax
```

Wir erweitern die Terme und vereinfachen:

```
E = t - y = Ax + (ΔA)x - Ax
E = (ΔA)x
```

Das ist bemerkenswert! Der Fehler `E` ist mit `ΔA` ganz einfach verknüpft. Und zwar so einfach, dass ich zunächst glaubte, falsch zu liegen – doch es stimmt tatsächlich. Auf jeden Fall macht uns diese einfache Beziehung das Leben leichter.

Bei dieser Algebra kann man leicht durcheinanderkommen oder abgelenkt werden. Erinnern wir uns daran, was wir von dem Ganzen erwarten, und zwar in Umgangssprache.

Wir wollten wissen, um welchen Betrag `A` anzupassen ist, sodass die Linie durch den geänderten Anstieg ein besserer Klassifizierer wird. Diese Berechnung soll auf Basis des Fehlers `E` geschehen. Dazu stellen wir einfach die letzte Gleichung nach `ΔA` um:

```
ΔA = E / x
```

Genau das ist es! Das ist der magische Ausdruck, nach dem wir gesucht haben. Anhand des Fehlers `E` können wir den Anstieg `A` der Klassifizierungslinie um einen Betrag `ΔA` verfeinern.

Los geht's – aktualisieren wir diesen anfänglichen Anstieg.

Der Fehler hatte einen Wert von 0,35, und x war 3,0. Damit machen wir ΔA = E / x zu 0,35 / 3,0 = 0,1167. Das heißt, wir müssen den aktuellen Anstieg A = 0,25 um 0,1167 ändern. Für den neuen, verbesserten Wert A ergibt sich also (A + ΔA) und somit 0,25 + 0,1167 = 0,3667. Tatsächlich ist der berechnete Wert von y mit diesem neuen Anstieg A von 1,1 – wie zu erwarten – der gewünschte Zielwert.

Geschafft! Nach der ganzen Arbeit haben wir eine Methode in der Hand, um den Parameter A zu verfeinern, und zwar gesteuert vom aktuellen Fehler.

Legen wir nach.

Ein erstes Trainingsbeispiel haben wir durchgenommen, nun wollen wir vom nächsten lernen. Hier haben wir eine bekannte wahre Paarung von x = 1,0 und y = 3,0.

Was passiert, wenn wir x = 1,0 in die lineare Funktion einsetzen, die jetzt den aktualisierten Wert A = 0,3667 verwendet? Wir erhalten y = 0,3667 × 1,0 = 0,3667. Das liegt nun gleich gar nicht in der Nähe des Trainingsbeispiels mit y = 3,0.

Mit der gleichen Argumentation wie zuvor – die Gerade sollte nicht die Trainingsdaten kreuzen, sondern unmittelbar darüber oder darunter verlaufen – können wir den Sollwert auf 2,9 setzen. Auf diese Weise liegt das Trainingsbeispiel einer Raupe unmittelbar über der Linie und nicht auf ihr. Der Fehler E beträgt (2,9 – 0,3667) = 2,5333.

Der Fehler ist größer als zuvor. Doch wenn man es recht bedenkt, hatten wir für die lineare Funktion nichts weiter zum Lernen als ein einzelnes Trainingsbeispiel, was zweifellos den Verlauf der Linie in Bezug auf dieses einzelne Beispiel verzerrt.

Wir aktualisieren A erneut, genau wie wir es zuvor getan haben. Die Änderung ΔA ist E / x, was 2,5333 / 1,0 = 2,5333 bedeutet. Das heißt, dass sogar der neuere Anstieg A gleich 0,3667 + 2,5333 = 2,9 ist. Und das bedeutet, dass die Funktion für x = 1,0 den Wert 2,9 als Antwort liefert, was dem Sollwert entspricht.

Das ist schon eine ganze Menge, was wir uns erarbeitet haben. Legen wir also eine Pause ein und stellen wir grafisch dar, was wir getan haben. Das Diagramm in Abbildung 1-20 zeigt die anfängliche Gerade, die Gerade nach dem Lernen aus dem ersten Trainingsbeispiel und die letzte Gerade nach dem Lernen aus dem zweiten Trainingsbeispiel.

Inhalt