

Machine Learning 3: Neuronale Netze I

Ergänzungsfach Informatik, 2021/2022, pro@kswe.ch

19. November 2021

1 Motivation und Ziel

Wir möchten nun verstehen, was ein neuronales Netzwerk ist, wie es arbeitet und wozu man es einsetzen kann. Das Ziel ist es, ein neuronales Netzwerk zu erstellen, welches **handgeschriebene Ziffern** (siehe Abbildung 1) erkennen kann.

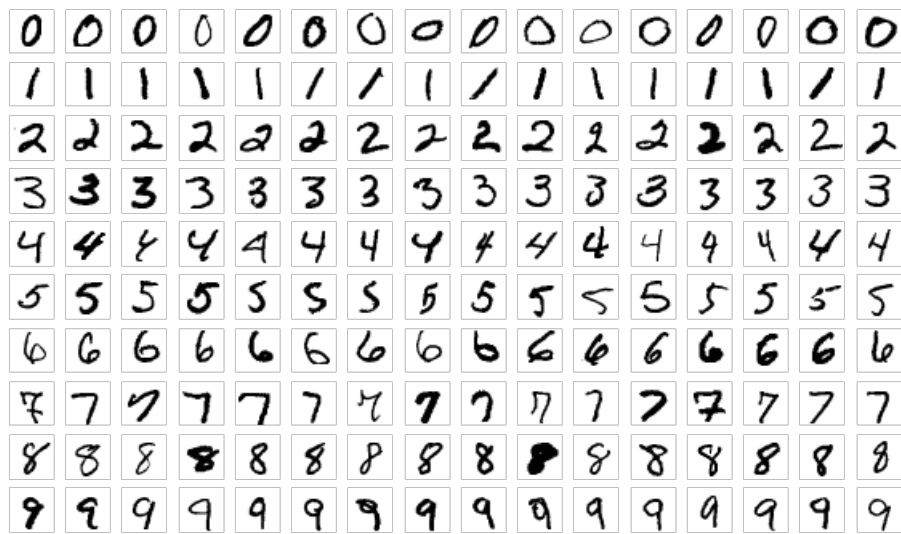


Abbildung 1: MNIST beinhaltet Graustufenbilder mit einer Auflösung von 28×28 Pixel.

Wir verwenden dafür das bekannte **MNIST** (Modified National Institute of Standards and Technology) Daten-Set. Es besteht aus 70 000 Samples. Typischerweise werden 60 000 Samples für das Training benutzt und 10 000 Samples für das Testen. Ein Sample beinhaltet das Label (Ziffer von 0 bis 9) und das 28×28 Pixel-Graustufenbild.

2 Lernprinzip

Wir betrachten den Machine-Learning-Task nochmals aus einer anderen Perspektive. Wir möchten eine einfache “Vorhersagemaschine” erstellen. Computer arbeiten nach dem EVA-Prinzip.

1. Beispiel: Computer soll $3 \cdot 4$ berechnen.

2. Beispiel: Computer soll Kilometer in Meilen umrechnen.

Trainingsdaten: Die Vorgabe ist es, einen linearen Zusammenhang zu erlernen: $MI = KM \cdot c$.

	Kilometer (KM)	Meilen (MI)
Sample 1	0	0
Sample 2	100	62,137

Training:

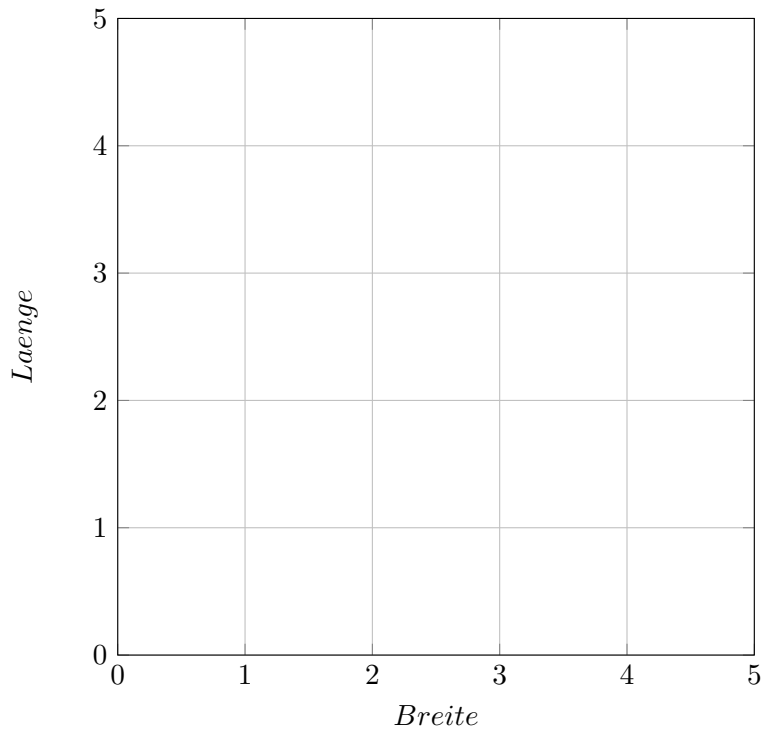
Schritt #	c	Kommentar	“Test”	Fehler (error): $E =$ Wahrer Wert - Berechneter Wert

Lessons Learned: Das Lernprinzip eines neuronalen Netzes funktioniert wie folgt:

- Der Fehler wird schrittweise verbessert \Rightarrow iteratives Prinzip
- Ein “grosser” Fehler benötigt eine “grosse” Anpassung der Parameter (im Beispiel c).
- Ein “kleiner” Fehler benötigt eine “kleine” Anpassung der Parameter (im Beispiel c).

3 Linearer Klassifizierer

Wir möchten durch eine Linie Marienkäfer und Raupen klassifizieren.



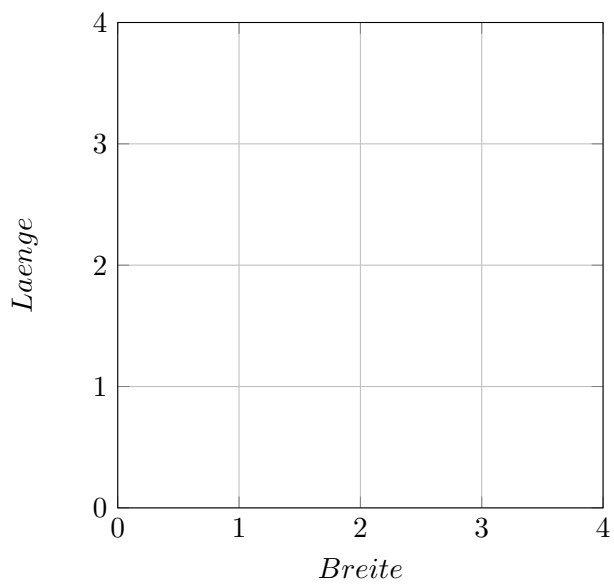
Wir verwenden einen iterativen Ansatz mit Trainingsdaten.

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

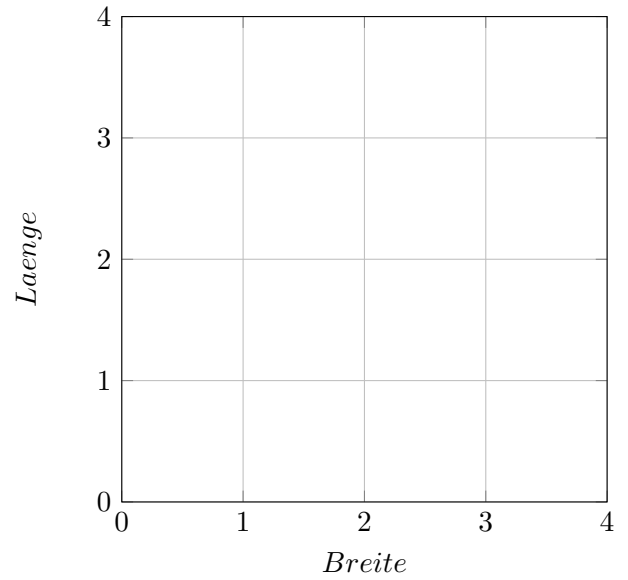
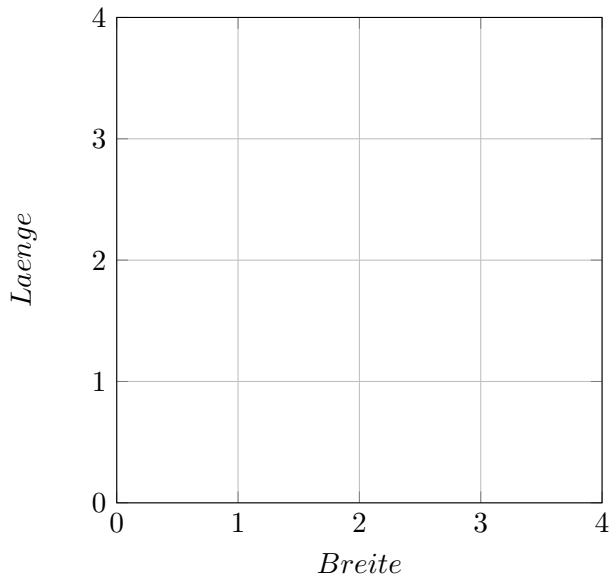
Die Trennungslinie ist eine Gerade durch den Ursprung.

$$f(x) = y = w_1 \cdot x + w_0 = w_1 \cdot x + 0 = w_1 \cdot x$$

1. Schritt: $w_1 =$

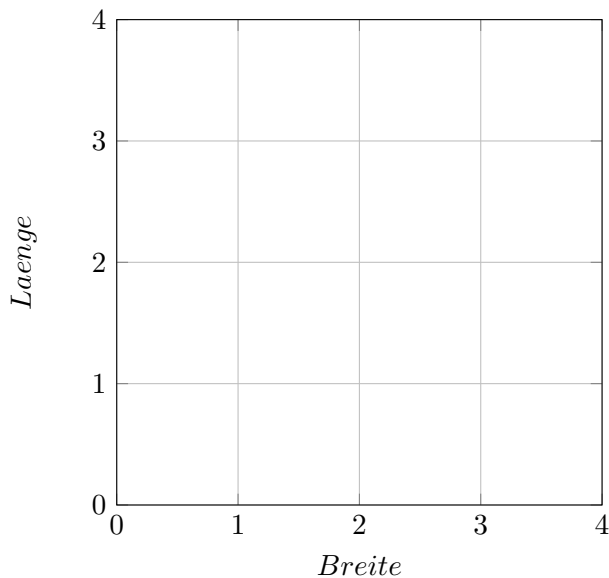


$f(3,0) = 1,0$ ergibt folgende Trennlinie:



3.1 Wie können wir den Parameter w_1 durch den Fehler E verbessern?

Um den gewünschten Wert t zu erhalten, müssen wir w_1 anpassen. Mathematisch schreibt man Δ für “eine kleine Änderung in”. Der Fehler E ist die Differenz zwischen Zielwert und dem berechneten Wert: $E = t - y$



3.2 Was bedeutet das Ergebnis?

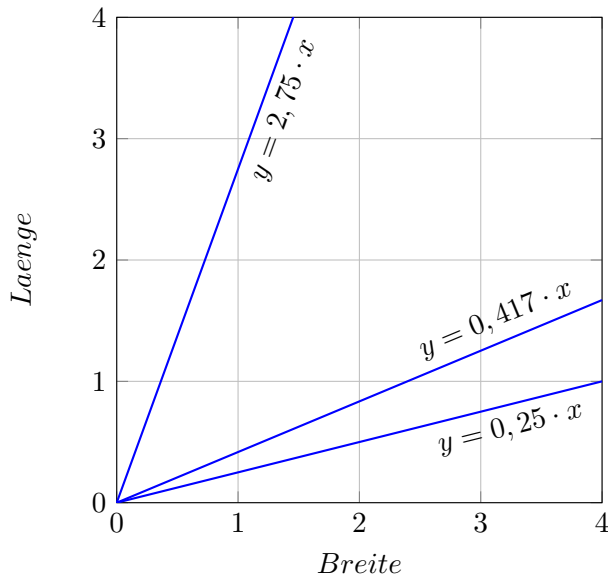
Eingangsfrage: Wie muss man w_1 anpassen, so dass eine “bessere” Linie entsteht? Der Fehler E soll dabei berücksichtigt werden.

Lösung: Passe w_1 um $\Delta w_1 = \frac{E}{x}$ an.

2. Schritt: Anpassung von w_1 um $\Delta w_1 = \frac{E}{x} = \frac{0,5}{3,0} = \frac{1}{6} \approx 0,167$. Wir erhalten dadurch folgende Trennungslinie: $y = (w_1 + \Delta w_1) \cdot x = (0,25 + 0,167) \cdot x = 0,417 \cdot x$.

Lessons Learned: Wir haben die Trennlinie durch einen Sample und den damit verbundenen Fehler anpassen können.

3. Schritt: Trennlinie durch den 2. Sample anpassen. $y = 0,417 \cdot 1,0 = 0,417 \neq 3,0$. Der Wert liegt nicht bei 3,0. Wir nehmen als Zielwert nun 2,75 und erhalten den Fehler: $E = 2,75 - 0,417 \approx 2,333$. Wir passen w_1 an: $\Delta w_1 = \frac{E}{x} = \frac{2,3333}{1,0} = 2,3333$. $\Rightarrow w_1 = 0,417 + 2,333 = 2,75$



3.3 Moderation

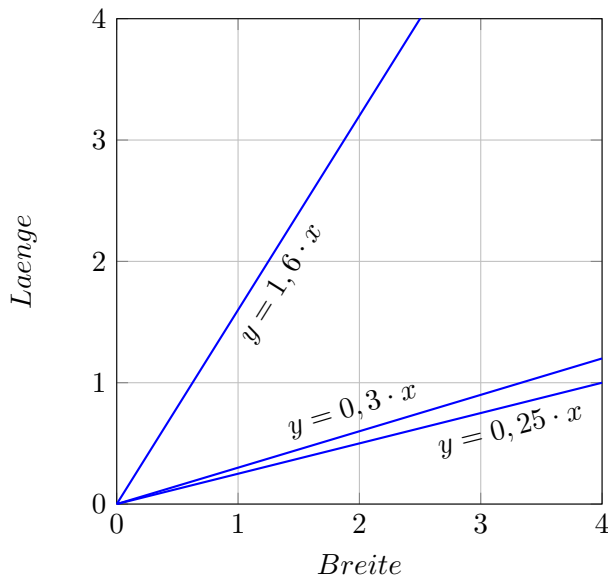
Ein zentrales Machine Learning-Konzept ist die **Lernrate**:

$$\Delta w_1 = L \cdot \frac{E}{x}$$

Die Lernrate L bedeutet, dass man nur einen Bruchteil der Änderung verwendet und nicht alles. Man spricht von einer **Moderation der Aktualisierungen**.

Beispiel: Erneutes Training mit Lernrate $L = 0,5$. Wir starten mit $f(x) = y = 0,25 \cdot x$.

(Breite / Länge)	f(x)	Fehler	Δw_1	f(x) "neu"
(3,0/1,0)				
(1,0/3,0)				



Lessons Learned:

- “Automatisierte Lernmethode”.
- Fehler und Sample erlauben den Parameter anzupassen.
- Moderation (Lernrate) berücksichtigt “alle” Samples.

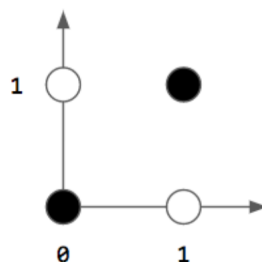
Wichtig!. Diese “automatisierte Lernmethode” funktioniert so noch nicht! Es soll viel mehr das Prinzip zeigen.

3.4 Aufgaben

1. Laden Sie die Dateien `marienkaefer_raupen_beispiel.py` und `raupen_marienkaefer.csv` auf Ihren Computer und fügen Sie die Dateien in das PyCharm-Projekt ein. Programmieren Sie dann in der Python-Datei die “automatisierte Lernmethode”. Starten Sie das Programm und schauen Sie was bisher funktioniert. Sie müssen “nur” die `for`-Schleife programmieren.

Lösungsvorschlag: Sie finden den Code in der Python-Datei `marienkaefer_raupen_beispiel.py`.

2. Zeichnen Sie einen Plot (x und y), sodass die “automatisierte Lernmethode” nicht funktioniert.



3. Was berücksichtigt die “automatisierte Lernmethode” bei der Aktualisierung der Gewichte eigentlich gar nicht?

Lösungsvorschlag: Die “automatisierte Lernmethode” funktioniert nicht wirklich. Es kommt darauf an in welcher Reihenfolge die Samples eingelesen werden. Auch werden die Klassen der Samples gar nicht berücksichtigt. Dies muss ein neuronales Netz alles lösen.