

Sie dürfen zu diesem Übungsblatt eine Abgabe machen bis spätestens zum 11.4., damit könnten Sie eine der erforderlichen fünf Abgaben schaffen für die ULP.

Sie dürfen in Gruppen bis max. drei Personen zusammen arbeiten, aber bitte geben Sie die Zusammenarbeit auf der Abgabe an und bitte geben Sie **jede(r) individuell** ab.

Besuchen Sie gerne das Praktikum am 4.4. und/oder am 11.4. In einer der nächsten Übungen können wir auch gerne ein „Fachgespräch“ zu diesen Aufgaben führen.

Implementieren Sie das einfache künstliche neuronale Netz aus der Vorlesung, das die Summe von zwei Zahlen berechnet, also ist

$$y = x_1 + x_2$$

und

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

Implementieren Sie bitte (zunächst ohne Bibliotheken wie *PyTorch* o.ä.) das zugehörige Training mittels Gradientenabstieg. Die partiellen Ableitungen ergeben sich aus der Fehlerfunktion

$$\begin{aligned} L(\hat{y}, y) &= \frac{1}{2} (\hat{y} - y)^2 \\ &= \frac{1}{2} (x_1 w_1 + x_2 w_2 - y)^2 \end{aligned}$$

Zum Trainieren können Sie einen Datensatz fester Größe mit zufälligen Eingaben erzeugen. Ihre Update-Schritte folgen dann der Logik

$$w_i \leftarrow w_i - \eta \cdot \frac{\partial L}{\partial w_i}$$

Sie sollten beobachten, dass die Fehler tendenziell kleiner werden und die Parameter w_1 und w_2 konvergieren. Für welche Werte von η funktioniert das Training besser / schlechter / nicht?

Visualisieren Sie, wie sich die Gewichte, ausgehend von verschiedenen Initialisierungen im Laufe des Trainings verändern, ein Beispiel ist hier unten in Abbildung 1 dargestellt. Visualisieren Sie auch, wie sich der Fehler während des Trainings verändert.

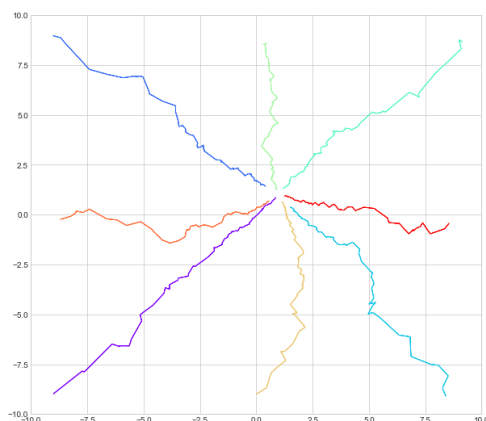


Abbildung 1: Entwicklung der Parameter w_1 und w_2 im Laufe des Trainings für acht unterschiedliche Initialisierungen (außen)