

# Übung 2 zur Vorlesung Deep Learning

## Aufgabenstellung

- Erlaubte Python-Pakete sind ausschließlich die Python-Standardbibliothek und NumPy!
- Berechnungen, mit Ausnahme der Aktivierungsfunktion, ausschließlich in Matrixschreibweise!
- Aufbauend auf Übung 1 soll das neuronale Netzwerk mittels Backpropagation und Gradientenabstieg trainiert werden. Initialisieren Sie dafür die Gewichte und Bias zufällig aus einer Gleichverteilung mit geeignetem Minimal- und Maximalwert, z.B. zwischen -0,1 und +0,1.
- Erstellen Sie einen Trainingsdatensatz mit Ein- und Ausgaben für AND, OR und XOR. Der Trainingsdatensatz besteht dabei aus je einer NumPy-Matrix für Ein- und Ausgaben. Erste Dimension: Trainingsbeispiele, zweite Dimension: Ein- bzw. Ausgabevektor.
- Implementieren Sie ein Backpropagation-Training für das neuronale Netz. Zielfunktion ist der Mean Squared Error (MSE). Wählen Sie eine geeignete Schrittweite ( $>0$  und  $<1$ ) für das Training. Das Training ist dabei eine Schleife mit folgenden Teilschritten:
  - Kopieren der Ein- und Ausgabevektoren des aktuellen Trainingsbeispiels.
  - Forward-Pass (Vorhersage) durch das neuronale Netzwerk.
  - Berechnung der MSE-Zielfunktion.
  - Ausgabe der Vorhersage, MSE und Zähler der aktuellen Iteration auf der Kommandozeile.
  - Backpropagation zur Berechnen der partiellen Ableitungen für das aktuelle Sample.
  - Aktualisierung der Gewichte und Biase (Gradientenabstieg).
  - Sprung zum nächsten Trainingssample und Beginn der Schleife. Wählen Sie eine geeignete Abbruchbedingung auf Basis des MSE.
- Der Trainingsdatensatz soll im gesamten Training mehrfach durchlaufen werden.
- Ableitung der logistischen Sigmoidfunktion:  $\phi'(x) = \phi(x)(1 - \phi(x))$

## Abgabekriterien

- Lösung in einer einzelnen Python-Datei.
- Zufällige Initialisierung der Parameter des neuronalen Netzes.
- Ein- und Ausgabe, MSE und Zähler der aktuellen Iteration werden für das aktuelle Trainingsbeispiel innerhalb der Trainings-Schleife ausgegeben.
- Die Ausgaben der vier verschiedenen Eingaben werden am Ende des Trainings korrekt vom neuronalen Netz berechnet.

- Abnahme durch E-Mail mit einer Python-Datei an [martin.schall@htwg-konstanz.de](mailto:martin.schall@htwg-konstanz.de) und  
Besprechung in der nächsten Übung mit Marc-Peter Schambach oder Martin Schall.

## **Abgabetermin:**

05.07.2018