

# Deep Learning

Wintersemester 2024/25

## Übungsblatt 3

Besprechung am 29.10.2024

Den Code für dieses Übungsblatt finden Sie wieder als Jupyter Notebooks im Repository:

<https://gitlab.hs-coburg.de/fei/education/master/deep-learning/2024-wise/material-und-aufgaben/deepl-computation-graphs>

### Aufgabe 1 – Lineare Regression mit dem EDF

Im Jupyter Notebook `linear_regression.ipynb` finden Sie ganz unten zwei Abschnitte zum Thema Lineare Regression mit dem EDF (Educational Framework).

- a) Implementieren Sie die Methoden `forward` und `backward` für den L2Loss-Knoten des Computation Graphs. Hierbei soll der L2Loss den elementweisen quadrierten Fehler zwischen  $y$  und  $\hat{y}$  berechnen.  
Das EDF arbeitet auf Batches, also sind  $y$  und  $\hat{y}$  Matrizen der Größe  $B \times 1$  und die Ausgabe muss ebenfalls eine Matrix der Größe  $B \times 1$  sein. Sie müssen keinen Durchschnitt berechnen oder summieren!
- b) Um Ihre Implementierung von L2Loss zu testen, müssen Sie den korrekten Computation Graph zusammenbauen. Definieren Sie dazu die Variablen `affine_params`, `x_node`, `y_node` und `loss_node`, welche von der Funktion `train` als Parameter erwartet werden.

Am Ende sollte das gleiche Ergebnis wie im letzten Übungsblatt herauskommen bzw. Sie können die Punkte und die sich ergebende Kurve plotten, um zu sehen, ob sie die Daten gut abbildet.

Tipp: In der Datei `edf.py` sehen Sie Implementierungen für andere Knoten. Lassen Sie sich davon inspirieren.

### Aufgabe 2 – Logistische Regression mit und ohne EDF

Im Jupyter Notebook `logistic_regression.ipynb` finden Sie Code zum Laden des MNIST-Datensatzes zur Ziffernerkennung.

Implementieren Sie die Methode `compute_derivative` für die logistische Regression und trainieren Sie das Modell. Schauen Sie, wie weit Sie kommen ;)

Viel Erfolg!