Fakultät Elektrotechnik und Informatik
Prof. Dr. Florian Mittag

Deep Learning

Wintersemester 2024/25

Übungsblatt 2

Besprechung am 22.10.2024

Den Code für dieses Übungsblatt finden Sie 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

a) Die Loss Function $\mathcal{L}$ für ein Model mit linearer Regression lautet:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(w_0 + w_1 x_i - y_i)^2$$

Das Model entspricht also eine Gerade mit Steigung w_1, welche die y-Achse bei w_0 schneidet. Berechnen Sie jeweils die partielle Ableitung bzgl. w_0 und w_1, also $\frac{\partial \mathcal{L}}{\partial w_0}$ und $\frac{\partial \mathcal{L}}{\partial w_1}$.

b) Im Jupyter Notebook `linear_regression.ipynb` finden Sie Code zum Laden eines gegebenen Datensatzes und ein unvollständiges Code-Gerüst zum Training eines Linear Regression Models.
Implementieren Sie die Funktion `compute_derivatives(x, y, slope, offset_y)`, wobei x und y hier NumPy-Arrays gleicher Größe sind, welche die Eingabedaten $(x_1, x_2, \dots, x_N)$ und die Soll-Werte $(y_1, y_2, \dots, y_N)$ enthalten. `slope` ist die Steigung w_1 und `offset_y` entspricht w_0.
Die Funktion soll ein Tuple zurückgeben, welches den partiellen Ableitungen $\left(\frac{\partial \mathcal{L}}{\partial w_0}, \frac{\partial \mathcal{L}}{\partial w_1}\right)$ entspricht, also dem Gradienten. Überprüfen Sie, ob Ihre Implementierung richtig ist, indem Sie ein Linear Regression Model trainieren lassen und das Model plotten.

c) Wie in der Vorlesung gezeigt, gibt es für dieses Problem auch eine analytische Lösung. Implementieren Sie diese Lösung in der Funktion `compute_weights_cf(X, y)`, welche `slope` und `offset_y` als Tupel zurückgibt.
Sie können Ihre Lösung überprüfen, indem Sie sie plotten und mit der Lösung aus b) vergleichen.

Viel Erfolg!