

# IV Übung zur Vorlesung Deep Learning

Neural Networks und Gewichts Anpassungen mittels SGD

## Aufgabe 1

Gegeben sei ein neuronales Netz mit zwei hidden layers, welches die Außentemperatur auf Basis von sonstigen Wetterdaten, wie beispielsweise Tageszeit, Jahreszeit, Luftfeuchtigkeit, Bewölkungsgrad, etc..., vorhersagt. Der Output des Netzes  $y^*$  ist linear (keine Aktivierung des Outputlayers) und wird durch den MSE-Loss bewertet  $l(y^*, y) = \frac{1}{N} \sum_i (y_i^* - y_i)^2$ , wobei  $y$  die tatsächlich gemessene Temperatur ist, und  $N$  die Anzahl an Vorhersagen, die bewertet werden. In der ersten hidden layer des Netzes wird die tanh-Aktivierung verwendet, in der zweiten ReLU.

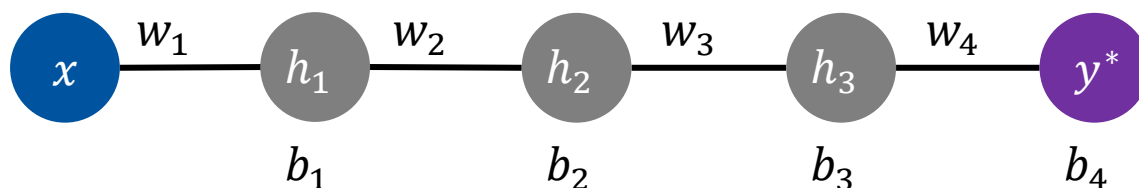
Berechne die Anpassung des Gewichts  $w_{1,1}^{(1)}$ , welches den Input  $x_1$  des Inputvektors  $x$  vom ersten Neuron des Inputlayers zum ersten Neuron des ersten hidden layers weitergibt. Die Abhängigkeit des Loss vom Gewicht  $w_{1,1}^{(1)}$  ist gegeben durch

$$l(y^*, y) = l(y^*(h^{(2)}), y) = l(y^*(h^{(2)}(h^{(1)})), y) = l(y^*(h^{(2)}(h^{(1)}(w_{1,1}^{(1)}))), y)$$

wobei  $h^{(i)}$  die Aktivierungsfunktion der Neuronen im  $i$ -ten hidden layer ist.

## Aufgabe 2

Betrachte folgenden Strang innerhalb eines neuronalen Netzes:



Es gilt

$$y^* = w_4 \cdot \left( \text{ReLU} \left( w_3 \cdot \left( \text{ReLU} \left( w_2 \cdot \left( \text{ReLU} \left( w_1 x + b_1 \right) \right) + b_2 \right) \right) + b_3 \right) \right) + b_4$$

Gegeben sind folgende Werte:

$$x = 2.0, \quad w_1 = 0.25, \quad w_2 = 0.5, \quad w_3 = 0.75, \quad w_4 = 0.8, \\ b_1 = 1.0, \quad b_2 = 0.25, \quad b_3 = 0.25, \quad b_4 = 0.7$$

Die Zielgröße  $y$  ist  $y = x = 2.0$ . Das Netz gibt mit den aktuellen Parametern allerdings ein  $y^* = 1.5$ . Korrigiere eines der Gewichte um den entsprechenden Wert, sodass die Vorhersage des Netzes korrekt ist, also  $y = y^* = x = 2.0$ . Berechne zunächst, welches Gewicht den größten Einfluss auf den Loss  $l(y^*, y) = (y_i^* - y_i)^2$  hat und korrigiere nur dieses Gewicht. Bestimme dazu die passende Lernrate  $\alpha$ , sodass die Gewichts Anpassung in einem Schritt geschieht.

## Aufgabe 3 – Fortsetzung von Aufgabe 3 aus der Übung 3

Aufgabe 3, Übung 3 endete mit einem trainierten MLP mit einer Klassifizierungsgenauigkeit von  $> 95\%$  für das MNIST Dataset. Für Fortsetzung steht der Code als Musterlösung im Moodle zur Verfügung.

- a) Reduziere die Größe des Trainingsdatensatzes in 10% Schritten solange, bis nur noch 10% der Trainingsdaten übrig sind. Evaluiere dein trainiertes Modell nach jedem Reduktionsschritt und plote den Verlauf der Modellgenauigkeit gegen die Menge an verwendeten Trainingsdaten.
- b) Untersuche, ob es Klassenspezifische Unterschiede bei der Abhängigkeit der Modellgenauigkeit von der Menge an Trainingsdaten gibt. Gibt es Klassen, die selbst bei geringer Datenmenge noch gut vorhergesagt werden? Welche Klasse benötigt die meisten Trainingsdaten und welche die wenigsten?
- c) Wähle die minimal Menge an Trainingsdaten pro Klasse, sodass jede Klasse mit einer Genauigkeit von  $> 90\%$  vorhergesagt wird.