



Diese Übung wird am **01.07.2020** um **13:30 Uhr** besprochen und **nicht** bewertet.

## 9.1 Backpropagation und Stochastic Gradient Descent

Gehen Sie von folgendem Netzwerk aus, wie in Abbildung 9.1 zu sehen. Verwenden Sie dabei die Identitätsfunktion  $f(x) = x$  als Aktivierungsfunktion. Zur Vereinfachung verwendet das Netzwerk keine Bias-Parameter. Nehmen Sie für den aktuellen Zielausgabewert  $y^* = 0.5$  und eine Definition der Fehlerfunktion von  $L(W) = \frac{1}{2} \|g - y^*\|^2$  mit einer Lernrate von  $\alpha = 0.5$  an.

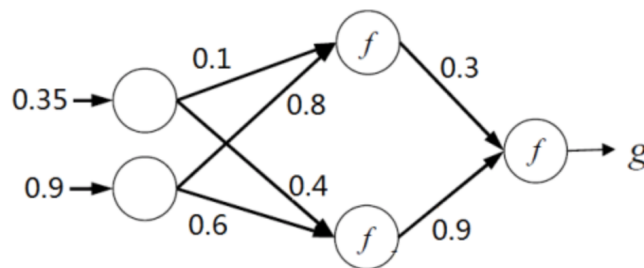


Abbildung 1: Netzstruktur eines „Multi-Layer Perceptron“

Die Eingabe und Gewichte werden wie folgt beschrieben:

$x_1 = 0.35$ ,  $x_2 = 0.9$ ,  $w_1 = 0.1$ ,  $w_2 = 0.8$ ,  $w_3 = 0.4$ ,  $w_4 = 0.6$ ,  $w_5 = 0.3$ ,  $w_6 = 0.9$ .

### a) Backpropagation

Wenden Sie Backpropagation an, um die Gewichte  $w_1$  bis  $w_6$  zu aktualisieren.



**b) Stochastic Gradient Descent (SGD)**

Beschreiben Sie SGD und seine Beziehung zu Backpropagation.

**c) Backpropagation mit Batches**

Wie verhält sich der Fehler, wenn man die Backpropagation statt für einen einzelnen Datenpunkt auf einer Menge von Daten, einem Batch, gleichzeitig durchführt.

Tabelle 1: Matrixdarstellung eines Bildes

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Tabelle 2: Kernel-Filter

1	0	1
0	1	0
1	0	1

## 9.2 2D Convolution

In dieser Übung führen wir eine 2D-Faltung auf dem Bild mit dem gegebenen Filter durch.

a) **Resultierende Größe**

Gehen Sie von quadratischer Eingabegröße und quadratischer Kernelgröße aus. Gegeben ist die Eingabegröße  $i$ , Kernelgröße  $k$ , Stride  $s$  und Zero-Padding  $p$ . Was ist die resultierende Größe der Ausgabe  $o$  nach Anwendung der Faltungsoperation?

b) **Anwendung I**

Zeigen Sie das Ergebnis der 2D-Faltung, bei  $stride = 1$  entlang beider Achsen und ohne  $zero - padding$ .

c) **Anwendung II**

Zeigen Sie das Ergebnis der 2D-Faltung, bei  $stride = 2$  entlang beider Achsen und  $zero - padding = 1 \times 1$ .

---

### 9.3 Neuronale Netze

---

In dieser Übung erforschen wir neuronale Netzwerke mit dem **tensorflow playground**:

<https://playground.tensorflow.org>

Wir wählen den Spiral-Datensatz für den Problemtyp-Klassifizierung, stellen das Trainings-/Testverhältnis 50 % und wählenden folgende Parametereinstellungen: Rauschen 20, Batch Size 10, Lernrate 0.03, Aktivierungsfunktion ReLU, Regularisierung L2, Regularisierungsrate 0.001.

a) **Konfiguration I**

Wählen Sie ein zweischichtiges Modell(4, 2) und verwenden Sie nur die Merkmale  $X_1$  und  $X_2$ , was ist der Trainings-/Testverlust nach 200 Epochen?

b) **Konfiguration II**

Wählen Sie ein vierschichtiges Modell (6, 4, 2, 2) und verwenden Sie nur die Merkmale  $X_1$  und  $X_2$ , wie hoch ist der Trainings-/Testfehler nach 200/1600 Epochen?

c) **Konfiguration III**

Wählen Sie ein dreischichtiges Modell (4, 4, 2) und nutzen Sie alle Features. Wie hoch ist der Trainings-/Testfehler nach 200 Epochen? Wie ist er bei 500 Epochen?