

**Übung Nr. 11 zur Vorlesung Algorithmische Mathematik II
Sommersemester 2020**

Abgabe bis zum 3. Juli 2020

Evaluation Wir wurden darauf hingewiesen, dass der Rücklauf der Evaluation bisher sehr gering ist. Falls die Evaluation (link wurde per Moodle verschickt) noch freigeschaltet ist, bitten wir um Teilnahme.

Künstliche neuronale Netze Bei der Parameteridentifikation haben wir ein Modell gegeben, also eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, welches von zusätzlichen Parametern $q \in Q$ abhängt, also $f(q; x)$. Unser Ziel war es, die zunächst unbekannt Parameter $q \in Q$ optimal zu bestimmen, so dass $f(q; \cdot)$ für gegebene Daten möglichst gute Übereinstimmung ergibt. Dabei haben wir (um es einfach zu halten) die Güte der Approximation immer im Least-Squares-Sinne bestimmt, d.h.

$$\min \frac{1}{2} \sum_{i=1}^N |f(q; x_i) - y_i|^2 \rightarrow \min,$$

wobei (x_i, y_i) für $i = 1, \dots, N$ die Testdaten sind.

Im Folgenden machen wir einen einfachen Einstieg in das *maschinelle Lernen*. Wir gehen weiter davon aus, dass wir Datensätze (x_i, y_i) , $i = 1, \dots, N$ haben, das zugrundeliegende Modell f aber nicht kennen, sondern es aus den Daten *lernen* müssen. So wie wir das Problem angehen, handelt es sich am Ende wieder einfach um das Parameteridentifikationsproblem. Wir geben auch nach wie vor ein Modell vor, dieses wird jedoch eine extrem einfache Gestalt haben, dafür über sehr viele Parameter verfügen.

Klassifikation Man lese die Abschnitt 8.3.3 *Klassifikation* und gebe kurze Antworten:

1. Wieviele Parameter sind notwendig zur Beschreibung einer Hyperebene in 2 Dimensionen (also einer Linie) und wieviele sind in 3 Dimensionen notwendig (hier ist es eine Ebene). Ganz allgemein, wieviele Parameter sind in d Dimensionen notwendig.
2. Warum ersetzen wir zur Identifikation der Parameter die Heaviside-Funktion $\Theta(s)$ mit der Sigmoid-Funktion $\sigma(s)$?

3. Wie ist die Ableitung der Sigmoid-Funktion $\sigma'(s)$.
4. Was gilt für $|\sigma'(s)|$ für $s \rightarrow \infty$ und warum ist das evtl. ein Problem in der Anwendung des Gradientenverfahrens?

Künstliche neuronale Netze Das einführende Beispiel des vorherigen Abschnittes lässt sich als ein sehr einfaches künstliches neuronales Netz mit einem künstlichen Neuron beschreiben. Dieses hat 2 Eingänge (km/h und dB) und einen Ausgabe (Auto, Fahrrad). Wir entwickeln nun einen Mechanismus zur Erweiterung dieser Idee: künstliche Neuronen werden vernetzt, d.h., mehrere Neuronen arbeiten parallel und die Ausgabe eines künstlichen Neurons kann die Eingabe von weiteren Schichten künstlicher Neuronen sein.

Man lese Abschnitt 8.3.4 *künstliche neuronale Netze* bis einschließlich Satz 8.24 (*Gradient des einschichtigen neuronalen Netzes*) und gebe kurze Antworten

1. Man skizziere ein künstliches neuronales Netzwerk mit 3 Eingaben und 4 Ausgaben.
2. Wieviele freie Parameter hat dieses künstliche neuronale Netzwerk?
3. Es sei $f : \mathbb{R} \rightarrow \mathbb{R}$ eine skalare Funktion, z.B. $f(x) = x^2$. Was meint man mit der *komponentenweisen Auswertung* von f ? Was ist das Ergebnis von $f(\mathbf{x})$ für $\mathbf{x} = (1, 2, 3)$ im Falle von $f(x) = x^3$?

Aufgabe 11.1:

Man beweise Sätze 8.22 sowie 8.23.

Programmieraufgabe 11.2:

Entsprechend zu Abschnitt 8.3.3 *Klassifikation* wollen wir ein einfaches Klassifikationsproblem lösen:

- Es wird ein Datensatz `fisch.txt` mit der Beschreibung von 100 Fischen eingelesen. Für jeden Fisch i ist `fisch[i,0]` das Alter, `fisch[i,1]` das Gewicht (in kg), `fisch[i,2]` die Länge (in cm) und `fisch[i,3]` gibt an, um welchen Fisch es sich handelt, dabei steht 0 für einen Zander, 1 für einen Hecht.
- Wir versuchen nun zu lernen, um welchen Fisch es sich jeweils handelt.
- Hierzu bilden wir ein einfaches Modell mit einem künstlichen Neuron mit 3 Eingängen (Alter, Gewicht, Länge) und einer Ausgabe (Fisch-Typ) sowie einem Bias, d.h.

$$f(\mathbf{x}) = b + \sum_{i=1}^3 x_i w_i.$$

a) Man programmiere das Gradientenverfahren zum Lösen dieser Aufgabe. Dabei orientiere man sich am Skript sowie an der Vorlage `template_11.py`. Am Sonntag wird darüber hinaus ein lauffähiges Programm zum ersten Klassifikationsbeispiel (Autos und Fahrräder) online gestellt. Darüber hinaus wird auch das Template bis Sonntag noch überarbeitet. Wieviele der Fische aus dem Trainingsdatensatz können Sie korrekt identifizieren?

Sie können das Gradientenverfahren ohne Schrittweitenkontroll bei fester Schrittweite, z.B. $s=0.5$ programmieren.

b) Man wende das trainierte Modell auf den Datensatz `testfische.txt` an. Wieviele dieser Fische werden jetzt korrekt identifiziert? Wie ändert sich das Ergebnis, wenn Sie die Toleranz des Gradientenverfahrens ändern? Gute Werte sind $tol=1.e-3$, $tol=0.5e-3$ oder $tol=1.e-4$.