Technische Universität Berlin
Fakultät IV – Elektrotechnik und Informatik

Künstliche Intelligenz: Grundlagen und Anwendungen

Albayrak, Fricke (AOT) – Opper, Thiel (KI)
Wintersemester 2014 / 2015

8. Aufgabenblatt

Abgabetermin: 08.02.2017 (2 Wochen Bearbeitungszeit!)

Aufgabe 1 – Neuronales Netz (40%)

Ein neuronales Netz soll die XOR-Funktion $y = x_1 \oplus x_2$ lernen, wobei die Wahrheitswerte durch $+1$ (wahr) und -1 (falsch) dargestellt werden:

x_1	-1	$+1$	-1	$+1$
x_2	-1	-1	$+1$	$+1$
$y = x_1 \oplus x_2$	-1	$+1$	$+1$	-1

Die Neuronen haben je zwei Eingänge e_{k1}, e_{k2} mit den Gewichten w_{k1}, w_{k2} und dem Bias w_{k0}. Sie verwenden die Schwellwert-Aktivierungsfunktion

$$a_k = \mathrm{sgn}(h_k) = \begin{cases} +1 & \text{für} \quad h_k > 0 \\ -1 & \text{für} \quad h_k \leq 0 \end{cases}$$

mit $h_k = w_{k1}\, e_{k1} + w_{k2}\, e_{k2} - w_{k0}$ zur Berechnung der eigenen Ausgabe a_k.

(a) Ein Perzeptron (einzelnes Neuron) startet mit $w_{11} = w_{12} = 1$ und $w_{10} = -1$. Berechnen Sie die Gewichte nach einmaligem Training mit jedem der vier Beispiele, wenn die Perzeptron-Lernregel mit Schrittweite $\lambda = 0.6$ verwendet wird! Vergessen Sie nicht, beim Lernen auch den Bias w_{10} anzupassen.

(b) Kann ein Perzeptron die oben angegebene XOR-Funktion überhaupt so lernen, dass alle vier Beispiele korrekt wiedergegeben werden? Begründen Sie ihre Antwort!

(c) Konstruieren Sie ein neuronales Netz aus mehreren Neuronen, dass die XOR-Funktion richtig berechnet! Welche booleschen Funktionen berechnen die einzelnen Neuronen in diesem Netz?

Aufgabe 2 – Lineare Aktivierungsfunktion **(20%)**

Betrachten Sie ein neuronales Netzwerk mit N Eingaben e_j, einer verborgenen Schicht mit K Neuronen und einem Ausgabeneuron. Da alle Neuronen eine lineare Aktivierungsfunktion haben, ist die Ausgabe des i-ten Neurons in der Zwischenschicht durch

$$z_i = -\alpha_{i0} + \sum_{j=1}^{N} \alpha_{ij} e_j$$

und die Gesamtausgabe des Netzwerks durch

$$a = -\beta_0 + \sum_{i=1}^{K} \beta_i z_i$$

gegeben. Zeigen Sie, dass es ein neuronales Netz ohne verborgene Einheiten gibt, das dieselbe Funktion berechnet!

Aufgabe 3 – Nichtdeterministische Beispiele **(40%)**

Der Trainingsdatensatz für ein neuronales Netz besteht aus N Beispielen, die jeweils aus derselben Eingabe x und einer zufällig gewählte Soll-Ausgabe $y_i \in \{-1, 1\}$ bestehen. Der Lernalgorithmus des Netzwerkes wählt die Gewichte so, dass die tatsächlichen Ausgaben $a(x) \in [-1; 1]$ die Fehlerfunktion

$$E_n = \sum_{i=1}^{N} |y_i - a(x_i)|^n$$

minimieren. Für die relative Häufigkeit positiver Beispiele gilt:

$$p = \frac{1}{N} \sum_{i=1}^{N} \frac{1 + y_i}{2}$$

(a) Welche Ausgabe $a(x)$ ordnet ein optimal trainiertes neuronales Netz den Beispielen zu, wenn der absolute Fehler E_1 minimiert wird?

(b) Welche Fehlerfunktion E_n sollte für den Lernalgorithmus gewählt werden, damit für die Beispielseingabe $a(x) = 2p - 1$ erreicht wird? Zeigen Sie, dass es genau ein n mit dieser Eigenschaft gibt!

(c) Berechnen Sie für $p = 0.9$ und $N = 100$ das Minimum der quadratischen Fehlerfunktion E_2 und der kubischen Fehlerfunktion E_3!