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!) Musterlösung

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

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

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.

(1) $x_1 = -1$, $x_2 = -1$, $y = -1$

$$a = \mathrm{sgn}(1 \cdot (-1) + 1 \cdot (-1) - (-1)) = \mathrm{sgn}(-1) = -1 = y$$

Keine Anpassung der Gewichte!

(2) $x_1 = +1$, $x_2 = -1$, $y = -1$

$$a = \mathrm{sgn}(1 \cdot (+1) + 1 \cdot (-1) - (-1)) = \mathrm{sgn}(+1) = +1 = y$$

Keine Anpassung der Gewichte!

(3) $x_1 = -1$, $x_2 = +1$, $y = -1$

$$a = \text{sgn}(1 \cdot (-1) + 1 \cdot (+1) - (-1)) = \text{sgn}(+1) = +1 = y$$

Keine Anpassung der Gewichte!

(4) $x_1 = +1$, $x_2 = +1$, $y = -1$

$$a = \text{sgn}(1 \cdot (+1) + 1 \cdot (+1) - (-1)) = \text{sgn}(+3) = +1 \neq y$$

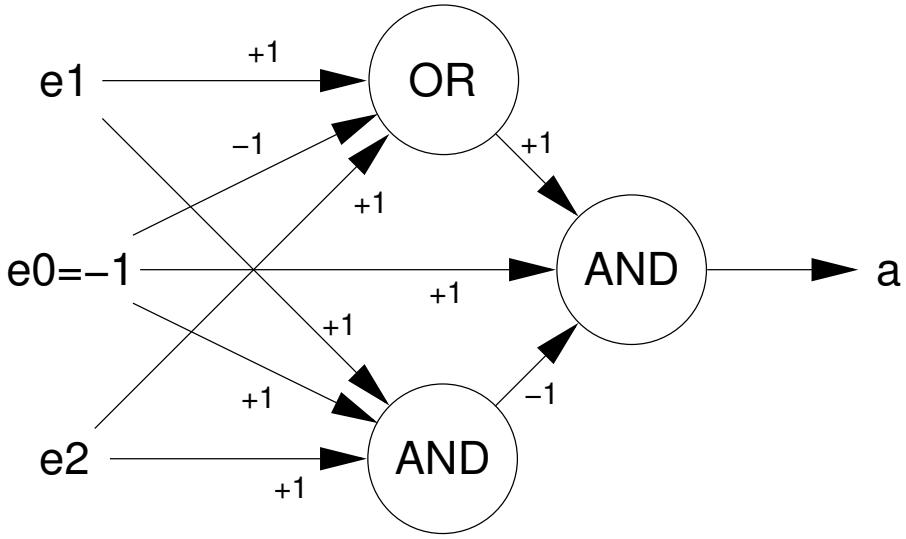
Anpassung der Gewichte erforderlich:

$$\begin{array}{rclcrcl} w_{10}^{+} & = & w_{10} + \lambda\, y\, e_{10} & = & -1 + 0.6 \cdot (-1) \cdot (-1) & = & -0.4 \\ w_{11}^{+} & = & w_{11} + \lambda\, y\, e_{11} & = & 1 + 0.6 \cdot (-1) \cdot (+1) & = & 0.4 \\ w_{12}^{+} & = & w_{12} + \lambda\, y\, e_{12} & = & 1 + 0.6 \cdot (-1) \cdot (+1) & = & 0.4 \end{array}$$

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

Da die Beispiele nicht linear separabel sind, kann ein Perzeptron diese nicht vollständig lernen. Es wird immer mindestens ein Beispiel geben, das falsch klassifiziert wird.

(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!

$$\begin{aligned} a &= -\beta_0 + \sum_{i=1}^{K} \beta_i \left(-\alpha_{i0} + \sum_{j=1}^{N} \alpha_{ij} e_j \right) \\ &= -\left(\beta_0 + \sum_{i=1}^{K} \beta_i \alpha_{i0} \right) + \sum_{j=1}^{N} \left(\sum_{i=1}^{K} \beta_i \alpha_{ij} \right) e_j \\ &= -w_0 + \sum_{j=1}^{N} w_j e_j \end{aligned}$$

Neue Gewichte für das Einschicht-Netz:

$$\begin{aligned} w_0 &= \beta_0 + \sum_{i=1}^{K} \beta_i \alpha_{i0} \\ w_j &= \sum_{i=1}^{K} \beta_i \alpha_{ij} \end{aligned}$$

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?

Da alle Beispiele die gleiche Eingabe aufweisen, ist auch die Ausgabe konstant $a_i = a$.

$$\begin{aligned} E_1 &= \sum_{i=1}^{N} |y_i - a| \\ &= Np|1 - a| + N(1 - p)| - 1 - a| \\ &= Np(1 - a) + N(1 - p)(1 + a) \\ \frac{\partial}{\partial a} E_1 &= N(1 - p) - Np \\ &= N(1 - 2p) \end{aligned}$$

Nur für $p = 0.5$ wird die Ableitung im Inneren des Wertebereichs Null. Das Minimum liegt sonst am Rand.

$$a = \begin{cases} 1 & \text{für } p > 0.5 \\ -1 & \text{für } p < 0.5 \end{cases}$$

(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!

$$\begin{aligned} E_n &= \sum_{i=1}^{N} |y_i - a|^n \\ &= Np|1 - a|^n + N(1 - p)| - 1 - a|^n \\ &= Np(1 - a)^n + N(1 - p)(1 + a)^n \\ \frac{\partial}{\partial a} E_n &= -nNp(1 - a)^{n-1} + nN(1 - p)(1 + a)^{n-1} \end{aligned}$$

Notwendige Bedingung für ein Minimum:

$$\frac{\partial}{\partial a} E_n = 0 \Longleftrightarrow \frac{p}{1 - p} = \left(\frac{1 + a}{1 - a}\right)^{n-1}$$

Einsetzen der Bedingung $a = 2p - 1$ führt zu:

$$\frac{\partial}{\partial a} E_n = 0 \iff \frac{p}{1 - p} = \left(\frac{p}{1 - p}\right)^{n-1}$$

$$
\begin{aligned}
&\iff \log\left(\frac{p}{1-p}\right) = (n-1)\log\left(\frac{p}{1-p}\right) \\
&\iff (n-2)[\log p - \log(1-p)] = 0 \\
&\iff n = 2 \vee p \in \{0, 0.5, 1\}
\end{aligned}
$$

Damit $a = 2p - 1$ erreicht wird, muss für den Lernalgorithmus die quadratische Fehlerfunktion E_2 verwendet werden.

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

Für E_2 gilt:

$$
\begin{aligned}
a &= 2p - 1 \\
&= 2 \cdot 0.9 - 1 \\
&= 0.8
\end{aligned}
$$

Für E_3 gilt:

$$
\begin{aligned}
\frac{p}{1-p} = \left(\frac{1+a}{1-a}\right)^2 &\iff \frac{1+a}{1-a} = \sqrt{\frac{0.9}{1-0.9}} \\
&\iff (1+a) = \sqrt{9}(1-a) \\
&\iff 4a = 2 \\
&\iff a = \frac{1}{2} = 0.5
\end{aligned}
$$