Einführung in die
Künstliche Intelligenz
SS09 - Prof. Dr. J. Fürnkranz

TECHNISCHE UNIVERSITÄT DARMSTADT

Beispiellösung für das 6. Übungsblatt (14.07.2009)

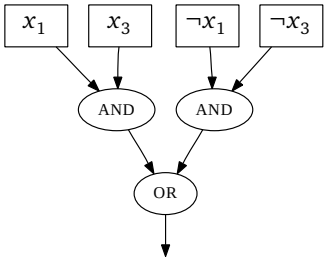
Aufgabe 1 Perceptrons, Neuronale Netze

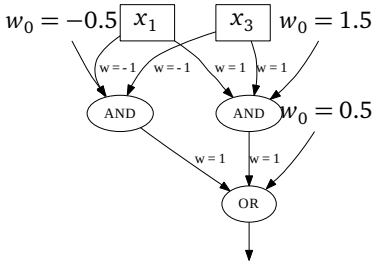
a)
- Der Parameter x_2 hat keinen Einfluß auf das Ergebnis. (Bei der Erstellung des neuronalen Netzes kann dieser deshalb weggelassen werden.)
- Die Funktion ist nicht linear separabel (f stellt eine negierte XOR Operation dar, die genauso wie XOR nicht mittels einer linearen Trennebene realisierbar ist.)

b) Die Wahrheitstabelle ohne x_2 sieht folgendermassen aus:

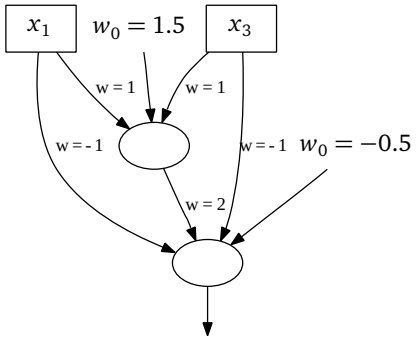
x_1	x_3	$f(x_1, x_3)$
0	0	1
0	1	0
1	0	0
1	1	1

Es handelt sich also um eine negierte XOR Operation. Als eine einfache Realisierung mittels zweistelligen logischen Operationen könnte man sich das Schema auf der linken Seite vorstellen. Da die Perceptron Darstellung für AND,OR und NOT bereits aus der Vorlesung bekannt ist, erhält man sofort ein Neuronales Netz, dass die Funktion f realisiert (rechte Graphik).





Eine besonders kompakte Realisierung stellt folgende Lösung dar:



c) Ausgabe des Netzes (im Folgenden wird auf 3 Dezimalstellen gerundet):

$I_1 = a_1 = 1$ $\quad I_2 = a_2 = 0$
$in_3 = 0.4 \cdot 1 + 0.2 \cdot 0 = 0.4$ $\quad a_3 = g(0.4) = 0.6$
$in_4 = 0.5 \cdot 1 - 0.7 \cdot 0 = 0.5$ $\quad a_4 = g(0.5) = 0.623$
$in_5 = -0.2 \cdot 0.6 + 0.7 \cdot 0.623 = 0.316$ $\quad a_5 = g(0.316) = 0.578$

Fehler: $Err_5 = T - a_5 = 0 - 0.578 = -0.578$

$\Delta_5 = Err_5 \cdot g'(in_5) = Err_5 \cdot g(in_5) \cdot (1 - g(in_5)) = -0.578 \cdot 0.578 \cdot 0.422 = -0.141$
$w_{3,5}^{neu} = w_{3,5} + \alpha \cdot \Delta_5 \cdot a_3 = -0.2 + 0.6 \cdot -0.141 \cdot 0.6 = -0.251$
$\Delta_3 = w_{3,5}^{neu} \cdot \Delta_5 \cdot g'(in_3) = -0.251 \cdot -0.141 \cdot 0.6 \cdot 0.4 = 0.008$
$w_{1,3}^{neu} = w_{1,3} + \alpha \cdot \Delta_3 \cdot a_1 = 0.4 + 0.6 \cdot 0.008 \cdot 1 = 0.405$

Aufgabe 2 Reinforcement Learning

a) Jeder Schritt kostet 0.1 Punkte und für das Erreichen des Feldes f erhält man einen Reward von 1. Das bedeutet, dass Zustands-Aktionspaare (s,a), deren Ausführung das Erreichen von Feld f zur Folge hat, einen Reward von $1 - 0.1 = 0.9$ erhalten. Für alle restlichen Paare erhält man einen Reward von -0.1.

$r(a,u) = -0.1$	$r(d,o) = -0.1$	$r(b,l) = -0.1$	$r(a,r) = -0.1$
$r(b,u) = -0.1$	$r(e,o) = -0.1$	$r(c,l) = -0.1$	$r(b,r) = -0.1$
$r(c,u) = 0.9$	$r(g,o) = -0.1$	$r(e,l) = -0.1$	$r(d,r) = -0.1$
$r(d,u) = -0.1$	$r(h,o) = -0.1$	$r(h,l) = -0.1$	$r(e,r) = 0.9$
$r(e,u) = -0.1$	$r(i,o) = 0.9$	$r(i,l) = -0.1$	$r(g,r) = -0.1$
			$r(h,r) = -0.1$

b) Überlegen wir uns beispielhaft die Bewertung des Feldes g. Im allgemeinen wird der akkumulierte erwartete Reward berechnet als $V^\pi = \sum_{k=0}^{\infty} \gamma^k \cdot r_k$. Laut Policy bewegt sich der Agent ausgehend von g wie folgt: $\rightarrow d \rightarrow a \rightarrow b \rightarrow c \rightarrow f$, wobei im Feld f keine Aktion mehr möglich ist.

Die Bewertung $V^\pi(g)$ ergibt sich also als:

$$\begin{aligned} V^\pi(g) &= \gamma^0 \cdot r(g,o) + \gamma^1 \cdot r(d,o) + \gamma^2 \cdot r(a,r) + \gamma^3 \cdot r(b,r) + \gamma^4 \cdot r(c,u) \\ &= (-0.1) + 0.9 \cdot (-0.1) + 0.9^2 \cdot (-0.1) + 0.9^3 \cdot (-0.1) + 0.9^4 \cdot 0.9 \\ &= 0.24659 \approx 0.25 \end{aligned}$$

Analog berechnet man die Bewertungen der restlichen Felder:

$V^\pi(a) = 0.54$	$V^\pi(b) = 0.71$	$V^\pi(c) = 0.90$
$V^\pi(d) = 0.39$	$V^\pi(e) = 0.54$	$V^\pi(f) = 0$
$V^\pi(g) = 0.25$	$V^\pi(h) = 0.39$	$V^\pi(i) = 0.90$

c) PolicyImprovement ändert die aktuelle Policy π für einen Zustand s um, indem sie die Aktion a selektiert, die folgendes maximiert:

$$\max_a r(s,a) + \gamma \cdot V^\pi(s') \qquad \text{wobei } s' = \delta(s,a)$$

Die aktuelle Policy $\pi(e)$ für den Zustand e würde einen Schritt nach **oben** vorgeben, mit der Gesamt-Bewertung 0.54. Für die anderen Aktionen ergibt sich:

links: $r(e,l) + \gamma \cdot V^\pi(d) = -0.1 + 0.9 \cdot 0.39 = 0.251$
unten: $r(e,u) + \gamma \cdot V^\pi(h) = -0.1 + 0.9 \cdot 0.39 = 0.251$
rechts: $r(e,r) + \gamma \cdot V^\pi(f) = 0.9$

Da die Aktion *rechts* im Zustand e die beste Bewertung hat, würde die aktuelle Policy für den Zustand e mit der Anweisung $\pi'(e) = r$ überschrieben werden.

d) Wir überlegen uns für jedes Feld, welches ein optimaler Weg wäre. Für das Feld g beispielsweise würde der Weg $\rightarrow h \rightarrow i \rightarrow f$ einen optimalen Reward erzielen, genauso wie $\rightarrow d \rightarrow e \rightarrow f$, nämlich:

$$= -0.1 + 0.9 \cdot (-0.1) + 0.9^2 \cdot (0.9) = 0.54$$

Analog berechnet man die Bewertungen der restlichen Felder und erhält:

$V^*(a) = 0.54$	$V^*(b) = 0.71$	$V^*(c) = 0.90$
$V^*(d) = 0.71$	$V^*(e) = 0.90$	$V^*(f) = 0$
$V^*(g) = 0.54$	$V^*(h) = 0.71$	$V^*(i) = 0.90$

Die optimale Q-Funktion für alle Zustandspaare lässt sich nun mit den berechneten optimalen Bewertungsfunktionen recht einfach berechnen. Wie aus der Vorlesung bekannt, gilt für die optimale Q-Funktion :

$$Q(s,a) = r(s,a) + \gamma \cdot V^*(s')$$

Im Feld d erhalten wir beispielsweise :
$Q(d,u) = r(d,u) + \gamma \cdot V^*(g) = -0.1 + 0.9 \cdot 0.54 = 0.386 \approx 0.39$
$Q(d,o) = r(d,o) + \gamma \cdot V^*(a) = -0.1 + 0.9 \cdot 0.54 = 0.386 \approx 0.39$
$Q(d,r) = r(d,r) + \gamma \cdot V^*(e) = -0.1 + 0.9 \cdot 0.9 = 0.71$

Insgesamt ergibt dies:

$Q(a,u) = 0.54$	$Q(d,o) = 0.39$	$Q(b,l) = 0.39$	$Q(a,r) = 0.54$
$Q(b,u) = 0.71$	$Q(e,o) = 0.54$	$Q(c,l) = 0.54$	$Q(b,r) = 0.71$
$Q(c,u) = 0.9$	$Q(g,o) = 0.54$	$Q(e,l) = 0.54$	$Q(d,r) = 0.71$
$Q(d,u) = 0.39$	$Q(h,o) = 0.71$	$Q(h,l) = 0.39$	$Q(e,r) = 0.9$
$Q(e,u) = 0.54$	$Q(i,o) = 0.9$	$Q(i,l) = 0.54$	$Q(g,r) = 0.54$
			$Q(h,r) = 0.71$

e) Die optimale Policy wählt in jedem Feld diejenige Aktion aus, die den höchsten akkumulierten erwarteten Reward verspricht:

$$\begin{aligned}\pi^*(s) &= \text{argmax}_a\, r(s,a) + \gamma \cdot V^*(s') \\ &= \text{argmax}_a\, Q(s,a)\end{aligned}$$

Mithilfe der vorigen Teilaufgaben lässt sich einfach die optimale Policy ablesen, indem man für jeden Zustand die Aktion wählt, die den höchsten Q-Wert aufweist. Insgesamt ergibt das folgende graphisch dargestellte Policy:

$\downarrow\rightarrow$	$\downarrow\rightarrow$	$\downarrow$
$\rightarrow$	$\rightarrow$	
$\uparrow\rightarrow$	$\uparrow\rightarrow$	$\uparrow$

f) Alle Werte $\hat{Q}(s,a)$ werden zunächst auf null gesetzt. Wir verwenden folgende graphische Ansicht der $\hat{Q}$-Werte:

a $\hat{Q}(a,u) = 0$	$\hat{Q}(a,r) = 0$	$\hat{Q}(b,l) = 0$	b $\hat{Q}(b,u) = 0$	$\hat{Q}(b,r) = 0$	$\hat{Q}(c,l) = 0$	c $\hat{Q}(c,u) = 0$
$\hat{Q}(d,o) = 0$ d $\hat{Q}(d,u) = 0$	$\hat{Q}(d,r) = 0$	$\hat{Q}(e,l) = 0$	$\hat{Q}(e,o) = 0$ e $\hat{Q}(e,u) = 0$	$\hat{Q}(e,r) = 0$	$\hat{Q}(f,l) = 0$	$\hat{Q}(f,o) = 0$ f $\hat{Q}(f,u) = 0$
$\hat{Q}(g,o) = 0$ g	$\hat{Q}(g,r) = 0$	$\hat{Q}(h,l) = 0$	$\hat{Q}(h,o) = 0$ h	$\hat{Q}(h,r) = 0$	$\hat{Q}(i,l) = 0$	$\hat{Q}(i,o) = 0$ i

Wir wählen zufällig ein Feld aus, sagen wir g. Da die beiden Aktionen o und r gleich bewertet sind, wählen wir erneut zufällig die Aktion r.

Nun ergibt sich der neue Wert $\hat{Q}(g,r) = \hat{Q}(g,r) + \alpha[r(g,r) + \gamma \cdot \max_a \hat{Q}(h,a) - \hat{Q}(g,r)]$. Da α auf 1 gesetzt wurde, wird der alte Wert nicht berücksichtigt, d.h. als Update-Regel wird $\hat{Q}(g,r) = r(g,r) + \gamma \cdot \max_a \hat{Q}(h,a)$ verwendet.

Darüberhinaus sind alle $\hat{Q}$-Werte von h auf 0 gesetzt, so dass $\hat{Q}(g,r) = -0.1 + 0.9 \cdot 0 = -0.1$.

Im Feld h wählen wir zufällig die Aktion o : $\hat{Q}(h,o) = -0.1$
Im Feld e wählen wir zufällig die Aktion r : $\hat{Q}(e,r) = 0.9$

Die momentanen $\hat{Q}$-Werte sehen dann wie folgt aus:

a	0.0		0.0	b	0.0		0.0	c
0.0				0.0				0.0
0.0				0.0				0.0
d	0.0		0.0	e	0.9		0.0	f
0.0				0.0				0.0
0.0				-0.1				0.0
g	-0.1		0.0	h	0.0		0.0	i

In einer weiteren Iteration starten wir von b und wählen die Aktion u. Es ergibt sich nun $\hat{Q}(b,u) = r(b,u) + \gamma \cdot \max_a \hat{Q}(e,a)$. Laut unserer aktuellen Q-Funktion ist im Feld e die optimale Aktion mit 0.9 bewertet, deshalb erhalten wir $\hat{Q}(b,u) = -0.1 + 0.9 \cdot 0.9 = 0.71$. Im Feld e wird daraufhin die Aktion r gewählt und die Q-Werte ändern sich nicht.

a	0.0		0.0	b	0.0		0.0	c
0.0				0.71				0.0
0.0				0.0				0.0
d	0.0		0.0	e	0.9		0.0	f
0.0				0.0				0.0
0.0				-0.1				0.0
g	-0.1		0.0	h	0.0		0.0	i

Ein weiterer Durchlauf sei folgenden Weg gegangen : $i \to h \to g \to d \to e \to f$ (wobei die Wahl der nächsten Aktion im Feld g und e eindeutig war).

a	0.0		0.0	b	0.0		0.0	c
0.0				0.71				0.0
0.0				0.0				0.0
d	0.71		0.0	e	0.9		0.0	f
0.0				0.0				0.0
-0.1				-0.1				0.0
g	-0.1		-0.1	h	0.0		-0.1	i

Weitere Durchläufe ergaben folgende Wege : $a \to b \to e \to f$ und $g \to d \to e \to f$

a	0.54		0.0	b	0.0		0.0	c
0.0				0.71				0.0
0.0				0.0				0.0
d	0.71		0.0	e	0.9		0.0	f
0.0				0.0				0.0
0.54				-0.1				0.0
g	-0.1		-0.1	h	0.0		-0.1	i

Nach den anschließenden Durchläufen $c \to f$, $i \to f$ und $h \to i \to f$ finden keine weiteren Änderungen an den Q-Werten mehr statt. Insgesamt wurde eine (pseudo-)optimale Policy gefunden, die im Folgenden auf der rechten Seite zu sehen ist. Beachten Sie, dass die ermittelten Q-Werte nicht immer optimal sein müssen (Ein Beispiel zum Nachrechnen wäre, wenn die Pfade $e \to f$, $b \to e \to f$, $a \to b \to e \to f$ und $d \to a \to b \to e \to f$ am Anfang traversiert würden.). Die Konvergenz von Q-Learning an die optimale Q-Funktion gilt im Allgemeinen nur, wenn jedes Zustands-Aktions Paar beliebig oft besucht wird.

a	0.54		0.0	b	0.0		0.0	c
0.0				0.71				0.9
0.0				0.0				0.0
d	0.71		0.0	e	0.9		0.0	f
0.0				0.0				0.0
0.54				-0.1				0.9
g	-0.1		-0.1	h	0.71		-0.1	i

→	↓	↓
→	→	
↑	→	↑