

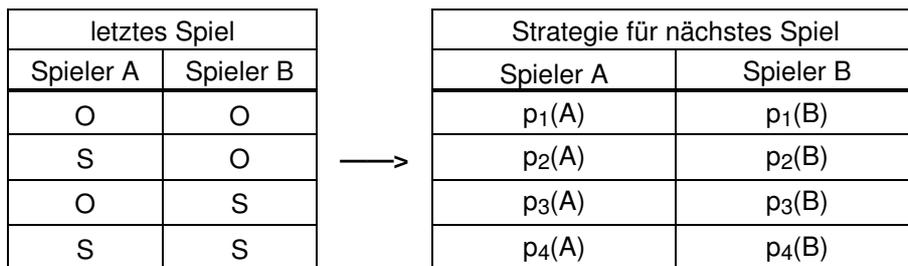
## Informationsverarbeitung in neuronalen Netzwerken

Frühjahrssemester 2009

### Musterlösung Übung 5

Bei dieser Übung ging es darum, das Verhalten von stochastischen Neuronen im "Salon-Problem" zu untersuchen, erstens in einem Spiel zwischen zwei Neuronen und zweitens gegen einen menschlichen Gegenspieler.

Die Strategie der Neuronen (und auch diejenige des menschlichen Gegners) kann durch 4 Wahrscheinlichkeiten  $\{p_1, p_2, p_3, p_4\}$  charakterisiert werden. Diese beziehen sich auf das Verhalten der beiden Spieler im letzten Spiel und bezeichnen die Wahrscheinlichkeit, mit der beim nächsten Spiel die Variante S (= Salon) gewählt wird [und mit Wahrscheinlichkeit  $1 - p_i$  die Variante O (= Oper)]:



#### Neuron A gegen Neuron B:

Es wurden im Wesentlichen zwei verschiedene Verhaltensweisen beobachtet, auf die sich die beiden Neuronen nach jeweils etwa 2500 Spielen einspielen (wobei natürlich A und B vertauscht sein können).

Die Gewinne  $g_A$  und  $g_B$  beziehen sich auf den Mittelwert aus 100 Spielen (nach Konvergenz zum Grenzverhalten).

	<b>p<sub>1</sub></b>	<b>p<sub>2</sub></b>	<b>p<sub>3</sub></b>	<b>p<sub>4</sub></b>	<b>g</b>
<u>Verhalten I:</u>					
( $\eta = 0.5, \alpha = 0$ )					
Spieler A:	0.82	0.91	0.00	0.09	20.00
Spieler B:	0.64	0.36	1.00	0.97	100.00
 <u>Verhalten II:</u>					
( $\eta = 0.5, \alpha = 0$ )					
Spieler A:	0.94	0.01	1.00	0.06	60.00
Spieler B:	0.17	1.00	0.01	0.98	60.00

	<b>p<sub>1</sub></b>	<b>p<sub>2</sub></b>	<b>p<sub>3</sub></b>	<b>p<sub>4</sub></b>	<b>g</b>
<u>Verhalten Ia:</u> ( $\eta = 0.5, \alpha = 0.05$ )					
Spieler A:	0.32	0.99	0.91	0.98	85.00
Spieler B:	0.91	0.16	0.31	0.19	17.00

<u>Verhalten IIa:</u> ( $\eta = 0.5, \alpha = 0.05$ )					
Spieler A:	0.88	0.16	0.99	0.99	52.80
Spieler B:	0.40	0.99	0.16	0.16	46.40

Bemerkung: Die Wahrscheinlichkeiten  $p_k$  sind nur auf 2 Stellen genau angegeben. Werte von 1.00 oder 0.00 bedeuten deshalb nur, dass die entsprechenden  $p_k$  sehr nahe bei 1 oder 0 liegen!

### Beobachtungen:

- Verhalten I entspricht dem Fall, dass ein Spieler immer in den Salon geht und der andere immer in die Oper, Verhalten II dem Fall, dass sich die beiden Spieler mit Salon- und Opernbesuch abwechseln.
- Für  $\alpha = 0$  (keine Bestrafung) entwickeln sich diese Varianten fast perfekt, für  $\alpha = 0.05$  hingegen nur näherungsweise:
  - Bei Variante Ia beträgt die Wahrscheinlichkeit, dass A in den Salon und B in die Oper geht, etwa 5/6, aber mit Wahrscheinlichkeit etwa 1/6 gehen beide in den Salon.
  - Bei Variante IIa ist die Wahrscheinlichkeit für (S, O) etwa 0.46, für (O, S) etwa 0.38 und für (S, S) etwa 0.16.

### Neuron (Spieler A) gegen Mensch (Spieler B):

In Versuchen mit einfachen (periodischen) Strategien des Menschen (Spieler B) hat das Neuron (Spieler A) wie folgt reagiert (Verhalten des Neurons nach 100 Spielen):

(Es wurde immer  $\eta = 2.0$  und  $\alpha = 0$  gewählt.)

<u>Strategie Mensch:</u>	<u>Strategie Neuron:</u>	<b>p<sub>1</sub></b>	<b>p<sub>2</sub></b>	<b>p<sub>3</sub></b>	<b>p<sub>4</sub></b>
(1) S S S S S S S S S ...		0.50	0.50	0.04	0.19
(2) O O O O O O O O ...		0.80	0.99	0.50	0.73
(3) S O S O S O S O ...		0.29	0.09	0.97	0.96
(4) S S O S S O S S O ...		0.37	0.10	0.85	0.98
(5) S S O O S S O O ...		0.84	0.96	0.50	0.96

### Beobachtungen:

- Auf die einfachen Strategien (1), (2) und (3) reagiert das Neuron praktisch optimal, auf die komplexeren Strategien (4) und (5) hingegen nicht. Das ist auf das beschränkte Gedächtnis des Neurons zurückzuführen (nur das letzte Spiel wird in Betracht gezogen):
  - Auf SSO antwortet das Neuron meistens mit OSS [ca. 78%] oder mit SSS [ca. 10%], und nur relativ selten mit OOS [ca. 12%], was optimal wäre.
  - Bei der Strategie SSOO geht das Neuron praktisch immer in den Salon.

## **Anhang: Programm-Beispiele (Matlab)**

### **Neuron A gegen Neuron B:**

```

rand('state', sum(100*clock));
e0 = 0.5;
a1 = 0.0;
nt = 50;
nts = 100;
wg1 = zeros(4,1);
pg1 = zeros(4,1);
wg2 = zeros(4,1);
pg2 = zeros(4,1);
iz1 = 1;
iz2 = 1;
gt1 = 0;
gt2 = 0;
for it=1:nt
    g1 = 0;
    g2 = 0;
    for kt=1:nts
        ng = 1+iz1+2*iz2;
        po1 = wg1(ng);
        po1 = exp(-po1);
        po1 = 1/(1+po1);
        po2 = wg2(ng);
        po2 = exp(-po2);
        po2 = 1/(1+po2);
        on1 = 0;
        cr = rand(1);
        if cr <= po1
            on1 = 1;
        end
        on2 = 0;
        cr = rand(1);
        if cr <= po2
            on2 = 1;
        end
        if on1 == 1
            if on2 == 1
                dg1 = 0.0;
                dg2 = 0.0;
            else
                dg1 = 1.0;
                dg2 = 0.2;
            end
        else
            if on2 == 1
                dg1 = 0.2;
                dg2 = 1.0;
            else
                dg1 = 0.2;
                dg2 = 0.2;
            end
        end
        end
        g1 = g1+dg1;
        g2 = g2+dg2;
        dw1 = dg1*(on1-po1)-a1*(1-dg1)*(on1-1+po1);
        wg1(ng) = wg1(ng)+e0*dw1;
        dw2 = dg2*(on2-po2)-a1*(1-dg2)*(on2-1+po2);
        wg2(ng) = wg2(ng)+e0*dw2;
        iz1 = on1;
        iz2 = on2;
    end
end

```

```

gt1 = gt1+g1;
gt2 = gt2+g2;
for k=1:4
    pg1(k) = 1/(1+exp(-wg1(k)));
    pg2(k) = 1/(1+exp(-wg2(k)));
end
[→ g1, gt1, pg1(k), k=1,...,4 und g2, gt2, pg2(k), k=1,...,4 ausdrucken]
end

```

### Neuron (Spieler A) gegen Mensch (Spieler B):

```

rand('state',sum(100*clock));
e0 = 2.0;
a1 = 0.0;
nt = 100;
wg1 = zeros(4,1);
pg1 = zeros(4,1);
iz1 = 1;
iz2 = 1;
gt1 = 0;
gt2 = 0;
for it=1:nt
    ng = 1+iz1+2*iz2;
    pol = wg1(ng);
    pol = exp(-pol);
    pol = 1/(1+pol);
    on1 = 0;
    cr = rand(1);
    if cr <= pol
        on1 = 1;
    end
    on2 = input('next ');
    if on1 == 1
        if on2 == 1
            g1 = 0.0;
            g2 = 0.0;
        else
            g1 = 1;
            g2 = 0.2;
        end
    else
        if on2 == 1
            g1 = 0.2;
            g2 = 1;
        else
            g1 = 0.2;
            g2 = 0.2;
        end
    end
    end
    dw1 = g1*(on1-pol)-a1*(1-g1)*(on1-1+pol);
    wg1(ng) = wg1(ng)+e0*dw1;
    iz1 = on1;
    iz2 = on2;
    gt1 = gt1+g1;
    gt2 = gt2+g2;
    for k=1:4
        pg1(k) = 1/(1+exp(-wg1(k)));
    end
    gg = [g1,g2];
    ggtp = [gt1,gt2,pg1(1),pg1(2),pg1(3),pg1(4)];
    disp(gg)
    disp(ggtp)
end

```

---