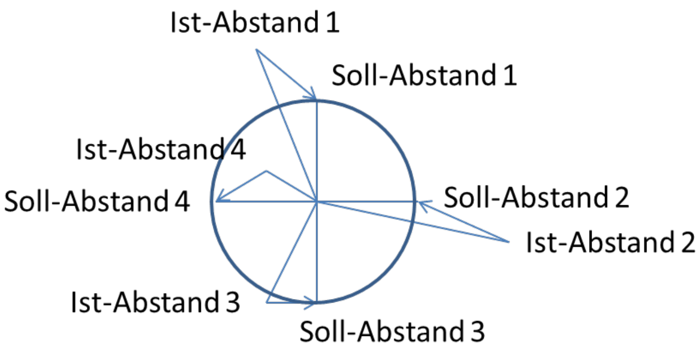
Aufgabe: Modellieren Sie das Schwarmverhalten des Vogelflugs anhand eines neuronalen Netzes.

Lösung: Da ein Vogel den Abstand zu seinen nächsten Nachbarn im Schwarm nicht berechnen kann, schätzt er ihn einfach und regelt ihn so aus, daß er zu allen einen etwa gleich großen Abstand einhält und somit auch das Risiko einer Kollision kleinhält. Dieses Verhalten kann durch ein eindimensionales neuronales Netz ohne verdeckte Schicht sehr gut simuliert werden.

Die Ausgangs-Sollwerte im Potentialminimum sind dabei gegeben durch eine Aktivierungsschwelle von 0,5, d.h. wenn dieser Wert von der Sigmoidfunktion überschritten wird, feuert das entsprechende Neuron. Dazu muß der Netzinput für jeden Schwarmteilnehmer aufgrund der waagrechten Tangente des Potentialminimums null sein, was jeweils dem Zustand in einem Gitterpunkt des Raums entspricht. Die Ausgangsgewichte $w_{ij,0}$ entsprechen dabei den zu Beginn des Verfahrens vorliegenden, individuell unterschiedlichen Abständen (siehe Abb.).



Der Optimierungsprozeß mit Laufindex k hängt also nur von den Ausgangsentfernungen aller 4 nächsten Nachbarn ab, die durch die Inputneuronen $x_i,\ \ i = 1, 2, 3, 4$ modelliert werden, d.h.

$$\begin{pmatrix} y_{1,k} \\ y_{2,k} \\ \vdots \\ y_{i,k} \\ \vdots \\ y_{m,k} \end{pmatrix} = \begin{pmatrix} w_{11,k} & w_{12,k} & w_{13,k} & w_{14,k} \\ w_{21,k} & w_{22,k} & w_{23,k} & w_{24,k} \\ \vdots & \vdots & \vdots & \vdots \\ w_{i1,k} & w_{i2,k} & w_{i3,k} & w_{i4,k} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1,k} & w_{m2,k} & w_{m3,k} & w_{m4,k} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

Da es nur ein bekanntes Muster gibt, das eintrainiert werden muß, gilt für den Fehler

$$E_k = \frac{1}{2} \sum_{i=1}^{m} \left(\eta_i - f\left(\sum_{j=1}^{4} w_{ij,k} x_j \right) \right)^2 \to \text{Minimum},$$

wobei der Netzinput durch

$$y_{i,k} = \sum_{j=1}^{4} w_{ij,k} x_j$$

und die Aktivierungsfunktion durch eine Sigmoidfunktion gegeben ist, i.e.

$$f(y_{i,k}) = \frac{1}{1+e^{-y_{i,k}}}.$$

In einem quadratischen Schwarmgitter mit Kantenlange $x_j = 1 \, \forall j \in \{1, 2, 3, 4\}$ soll also für den Sollabstand eines jeden Schwarmteilnehmers i die Relation $\eta_i = 0{,}5 \, \forall i \in \{1, 2, \ldots, 16\}$ gelten. Man beachte, daß sich bei einem Formationsflug mit vier nächsten Nachbarn alle Nachbarn im gleichen Abstand vom jeweiligen Schwarmteilnehmer befinden müssen. Das ist nur möglich, wenn sie einen Winkel von 90° relativ zum Betrachter einschließen und ihre Verbindungsgeraden paarweise orthogonal sind.

Wir nehmen nun $m = 16$ Vögel an, die in unserer Formation mitfliegen sollen. Die Startgewichte zum Zeitpunkt t_0, d.h. die realen Anfangsabstände zum Schwerpunkt jeder Zelle seien beispielsweise durch folgende Matrix gegeben:

$$W_0 = \left(\left(w_{ik,0}\right)\right) = \begin{pmatrix} 0{,}5 & 0{,}6 & 0{,}1 & -0{,}3 \\ -0{,}4 & 0{,}2 & 0{,}3 & 0{,}7 \\ 0{,}4 & -0{,}2 & -0{,}1 & 0{,}3 \\ -0{,}5 & 0 & -0{,}3 & 0{,}2 \\ 0{,}3 & -0{,}1 & 0{,}2 & -0{,}2 \\ -0{,}5 & 0{,}5 & 0{,}6 & -0{,}1 \\ 0{,}2 & 0{,}1 & -0{,}2 & 0{,}2 \\ -0{,}3 & -0{,}1 & -0{,}4 & 0{,}1 \\ 0{,}5 & 0{,}3 & 0 & 0{,}4 \\ 0{,}4 & 0{,}2 & -0{,}1 & 0{,}3 \\ 0{,}3 & 0{,}1 & -0{,}2 & 0{,}2 \\ 0{,}4 & 0{,}5 & 0 & -0{,}4 \\ -0{,}6 & 0{,}7 & 0 & 0 \\ 0{,}2 & 0{,}2 & 0{,}2 & 0{,}2 \\ -0{,}2 & -0{,}2 & -0{,}2 & -0{,}2 \\ -0{,}3 & -0{,}1 & -0{,}1 & -0{,}3 \end{pmatrix}.$$

Daraus ergeben sich dann die modifizierten Gewichte durch die Iterationen

$$w_{ij,k+1} = w_{ij,k} + \Delta w_{ij,k},$$

wobei die Änderungen nach dem Gradientenabstiegsverfahren gegeben sind durch

$$\Delta w_{ij,k} = -\lambda \frac{\partial E}{\partial w_{ij,k}} = \lambda\left(\eta_i - f(y_{i,k})\right) f(y_{i,k})\left(1 - f(y_{i,k})\right) x_j .$$

Die Gewichte werden so lange verändert, bis der Netzinput null ist. Das ist zugleich der Punkt des Potentialminimums. Anbei ein entsprechendes MATLAB-Programm:

```
% Programm Schwarmverhalten

% Fehlerberechnung für ein zweistufiges neuronales Netz ohne verdeckte
% Schicht mit beliebig vielen Ein- und Ausgabeneuronen und einer
% Sigmoidkurve als Aktivierungsfunktion
clear all

% Der Lernparameter sei lambda
lambda = 5;

% p ist die Laufvariable des jeweiligen Trainingsmusters
% pmax ist die maximale Zahl vorliegender Trainingsmuster
pmax = 1;

% Eingangs-, Soll-Ausgangs- und Ist-Ausgangsvektor hängen jeweils von der
% Laufvariablen p des Musters ab.

% Der Soll-Ausgangsvektor eta(p,i) hat imax Komponenten, wobei i die
% Zeilennummer der Gewichtsmatrix angibt
imax = 16;

% Die Spaltennumer der Gewichtematrix sei j
jmax = 4;

% Die 16 Komponenten des Soll-Ausgangsvektors sind gegeben durch
for i=1:imax
    eta(1,i) = 0.5;
end

% Die 16 Komponenten des Trainingsmuster x(1,j) des Eingangsvektors sind
% gegeben durch die Sollabstände
for j=1:jmax
    x(1,j) = 1;
end

% Die Laufvariable der kmax Iterationsschritte sei k
kmax = 9;

% Die imax Startgewichte w(i,j,1)seien gegeben durch
w(1,1,1)=0.5;   w(1,2,1)=0.6;   w(1,3,1)=0.1;   w(1,4,1)=-0.3;
w(2,1,1)=-0.4;  w(2,2,1)=0.2;   w(2,3,1)=0.3;   w(2,4,1)=0.7;
w(3,1,1)=0.4;   w(3,2,1)=-0.2;  w(3,3,1)=-0.1;  w(3,4,1)=0.3;
w(4,1,1)=-0.5;  w(4,2,1)=0;     w(4,3,1)=-0.3;  w(4,4,1)=0.2;

w(5,1,1)=0.3;   w(5,2,1)=-0.1;  w(5,3,1)=0.2;   w(5,4,1)=-0.2;
w(6,1,1)=-0.5;  w(6,2,1)=0.5;   w(6,3,1)=0.6;   w(6,4,1)=-0.1;
w(7,1,1)=0.2;   w(7,2,1)=0.1;   w(7,3,1)=-0.2;  w(7,4,1)=0.2;
w(8,1,1)=-0.3;  w(8,2,1)=-0.1;  w(8,3,1)=-0.4;  w(8,4,1)=0.1;

w(9,1,1)=0.5;   w(9,2,1)=0.3;   w(9,3,1)=0;     w(9,4,1)=0.4;
w(10,1,1)=0.4;  w(10,2,1)=0.2;  w(10,3,1)=-0.1; w(10,4,1)=0.3;
```



```
w(11,1,1)=0.3;  w(11,2,1)=0.1;  w(11,3,1)=-0.2; w(11,4,1)=0.2;
w(12,1,1)=0.4;  w(12,2,1)=0.5;  w(12,3,1)=0;    w(12,4,1)=-0.4;

w(13,1,1)=-0.6; w(13,2,1)=0.7;  w(13,3,1)=0;    w(13,4,1)=0;
w(14,1,1)=0.2;  w(14,2,1)=0.2;  w(14,3,1)=0.2;  w(14,4,1)=0.2;
w(15,1,1)=-0.2; w(15,2,1)=-0.2; w(15,3,1)=-0.2; w(15,4,1)=-0.2;
w(16,1,1)=-0.3; w(16,2,1)=-0.1; w(16,3,1)=-0.1; w(16,4,1)=-0.3;


for k=1:kmax
    E = 0;
    for i=1:imax
        % Berechnung der Aktivierungsfunktion
        for p=1:pmax
            ypsilon = 0;
            for j=1:jmax
                y(p,i,k) = ypsilon + w(i,j,k)*x(p,j);
                ypsilon = y(p,i,k);
            end
            f(p,i,k)=1/(1+exp(-y(p,i,k)));
        end
        % Änderung des Verbindungsgewichts der iten Zeile und jten Spalte
        for j=1:jmax
            Deltawij = 0;
            for p=1:pmax
                Deltaw(i,j,k) = Deltawij + lambda*(eta(p,i)-
f(p,i,k))*f(p,i,k)*(1-f(p,i,k))*x(p,j);
                Deltawij = Deltaw(i,j,k);
            end
            w(i,j,k+1) = w(i,j,k) + Deltaw(i,j,k);
        end
    end
end

% Fehlerrechnung
for k=1:kmax
    for i=1:imax
        for p=1:pmax
            EM(p,k) = 0.5*(eta(p,i)-f(p,i,k))^2;
        end
        EM0 = 0;
        for p=1:pmax
            Error(i,k) = EM0 + EM(p,k);
            EM0 = Error(i,k);
        end
    end
    E0 = 0;
    for i=1:imax
        E(k) = E0 + Error(i,k);
        E0 = E(k);
    end
end

% Ausdruck
for p=1:pmax
    for i=1:imax
        disp(['y(',num2str(p),',',num2str(i),') = ' num2str(y(p,i,kmax))])
    end
end
disp(' ');
```

```
for i=1:imax
    for j=1:jmax
        disp(['w(',num2str(i),',',num2str(j),') = ' num2str(w(i,j,kmax))])
    end
end
disp(' ');
for i=1:imax
    for j=1:jmax
        disp(['Deltaw(',num2str(i),',',num2str(j),') = '
num2str(Deltaw(i,j,kmax))])
    end
end
disp(' ');
for p=1:pmax
    for i=1:imax
        disp(['f(',num2str(p),',',num2str(i),') = ' num2str(f(p,i,kmax))])
    end
end
disp(' ');
disp(['E = ' num2str(E)])
```

Wir erkennen, daß sich der Netzinput nach nur 9 Iterationen der Null bereits bis auf die sechste Dezimale angenähert hat, und zwar für alle Vögel simultan. Damit hat jeder der 16 Schwarmteilnehmer seinen Gitterplatz bezogen, die Formation ist hergestellt.

```
>> schwarmverhalten (für kmax = 9)
y(1,1) = -2.01e-06
y(1,2) = 7.7783e-07
y(1,3) = 4.4855e-06
y(1,4) = -4.0257e-06
y(1,5) = 2.839e-06
y(1,6) = 4.5656e-06
y(1,7) = 3.8756e-06
y(1,8) = -2.7843e-06
y(1,9) = -1.3387e-05
y(1,10) = 7.7783e-07
y(1,11) = 4.4855e-06
y(1,12) = 4.5656e-06
y(1,13) = 1.4989e-06
y(1,14) = 7.7783e-07
y(1,15) = -7.7783e-07
y(1,16) = -7.7783e-07
```

Die Gewichte nach diesen 9 Iterationen haben dabei folgende Werte angenommen:

```
w(1,1) = 0.275
w(1,2) = 0.375
w(1,3) = -0.125
w(1,4) = -0.525
w(2,1) = -0.6
w(2,2) = 1.9446e-07
w(2,3) = 0.1
w(2,4) = 0.5
w(3,1) = 0.3
w(3,2) = -0.3
w(3,3) = -0.2
w(3,4) = 0.2
```

```
w(4,1) = -0.35
w(4,2) = 0.15
w(4,3) = -0.15
w(4,4) = 0.35
w(5,1) = 0.25
w(5,2) = -0.15
w(5,3) = 0.15
w(5,4) = -0.25
w(6,1) = -0.625
w(6,2) = 0.375
w(6,3) = 0.475
w(6,4) = -0.225
w(7,1) = 0.125
w(7,2) = 0.025001
w(7,3) = -0.275
w(7,4) = 0.125
w(8,1) = -0.125
w(8,2) = 0.074999
w(8,3) = -0.225
w(8,4) = 0.275
w(9,1) = 0.2
w(9,2) = -3.3469e-06
w(9,3) = -0.3
w(9,4) = 0.099997
w(10,1) = 0.2
w(10,2) = 1.9446e-07
w(10,3) = -0.3
w(10,4) = 0.1
w(11,1) = 0.2
w(11,2) = 1.1214e-06
w(11,3) = -0.3
w(11,4) = 0.1
w(12,1) = 0.275
w(12,2) = 0.375
w(12,3) = -0.125
w(12,4) = -0.525
w(13,1) = -0.625
w(13,2) = 0.675
w(13,3) = -0.025
w(13,4) = -0.025
w(14,1) = 1.9446e-07
w(14,2) = 1.9446e-07
w(14,3) = 1.9446e-07
w(14,4) = 1.9446e-07
w(15,1) = -1.9446e-07
w(15,2) = -1.9446e-07
w(15,3) = -1.9446e-07
w(15,4) = -1.9446e-07
w(16,1) = -0.1
w(16,2) = 0.1
w(16,3) = 0.1
w(16,4) = -0.1
```

Diese ändern sich nur mehr in der 6ten Dezimale.

```
Deltaw(1,1) = 6.2814e-07
Deltaw(1,2) = 6.2814e-07
Deltaw(1,3) = 6.2814e-07
Deltaw(1,4) = 6.2814e-07
Deltaw(2,1) = -2.4307e-07
Deltaw(2,2) = -2.4307e-07
Deltaw(2,3) = -2.4307e-07
Deltaw(2,4) = -2.4307e-07
Deltaw(3,1) = -1.4017e-06
Deltaw(3,2) = -1.4017e-06
Deltaw(3,3) = -1.4017e-06
Deltaw(3,4) = -1.4017e-06
Deltaw(4,1) = 1.258e-06
Deltaw(4,2) = 1.258e-06
Deltaw(4,3) = 1.258e-06
Deltaw(4,4) = 1.258e-06
Deltaw(5,1) = -8.8718e-07
Deltaw(5,2) = -8.8718e-07
Deltaw(5,3) = -8.8718e-07
Deltaw(5,4) = -8.8718e-07
Deltaw(6,1) = -1.4268e-06
Deltaw(6,2) = -1.4268e-06
Deltaw(6,3) = -1.4268e-06
Deltaw(6,4) = -1.4268e-06
Deltaw(7,1) = -1.2111e-06
Deltaw(7,2) = -1.2111e-06
Deltaw(7,3) = -1.2111e-06
Deltaw(7,4) = -1.2111e-06
Deltaw(8,1) = 8.7008e-07
Deltaw(8,2) = 8.7008e-07
Deltaw(8,3) = 8.7008e-07
Deltaw(8,4) = 8.7008e-07
Deltaw(9,1) = 4.1836e-06
Deltaw(9,2) = 4.1836e-06
Deltaw(9,3) = 4.1836e-06
Deltaw(9,4) = 4.1836e-06
Deltaw(10,1) = -2.4307e-07
Deltaw(10,2) = -2.4307e-07
Deltaw(10,3) = -2.4307e-07
Deltaw(10,4) = -2.4307e-07
Deltaw(11,1) = -1.4017e-06
Deltaw(11,2) = -1.4017e-06
Deltaw(11,3) = -1.4017e-06
Deltaw(11,4) = -1.4017e-06
Deltaw(12,1) = -1.4268e-06
Deltaw(12,2) = -1.4268e-06
Deltaw(12,3) = -1.4268e-06
Deltaw(12,4) = -1.4268e-06
Deltaw(13,1) = -4.684e-07
Deltaw(13,2) = -4.684e-07
Deltaw(13,3) = -4.684e-07
Deltaw(13,4) = -4.684e-07
Deltaw(14,1) = -2.4307e-07
```



```
Deltaw(14,2) = -2.4307e-07
Deltaw(14,3) = -2.4307e-07
Deltaw(14,4) = -2.4307e-07
Deltaw(15,1) = 2.4307e-07
Deltaw(15,2) = 2.4307e-07
Deltaw(15,3) = 2.4307e-07
Deltaw(15,4) = 2.4307e-07
Deltaw(16,1) = 2.4307e-07
Deltaw(16,2) = 2.4307e-07
Deltaw(16,3) = 2.4307e-07
Deltaw(16,4) = 2.4307e-07
```

Die Aktivierungsfunktion hat nun ohne weitere Rundungsfehler einheitlich den Wert 1/2 angenommen, bei dem alle Neuronen feuern, einige wohl auch schon früher.[1]

```
f(1,1) = 0.5
f(1,2) = 0.5
f(1,3) = 0.5
f(1,4) = 0.5
f(1,5) = 0.5
f(1,6) = 0.5
f(1,7) = 0.5
f(1,8) = 0.5
f(1,9) = 0.5
f(1,10) = 0.5
f(1,11) = 0.5
f(1,12) = 0.5
f(1,13) = 0.5
f(1,14) = 0.5
f(1,15) = 0.5
f(1,16) = 0.5
```

Der Fehler des Verfahrens ist dabei auf die elfte Dezimale abgesunken:

```
E = 0.20234   0.0030296  0.00016751  1.0408e-05  6.5026e-07   4.064e-08
2.54e-09  1.5875e-10   9.922e-12
```

Zu Beginn der Iterationen lagen die Verhältnisse noch ganz anders. Aufgrund der tatsächlich geschätzten Entfernungen lag der Netzinput für alle 16 Vögel bei folgenden Startwerten.

```
>> schwarmverhalten (für k = 1)
y(1,1) = 0.9
y(1,2) = 0.8
y(1,3) = 0.4
y(1,4) = -0.6
y(1,5) = 0.2
y(1,6) = 0.5
y(1,7) = 0.3
y(1,8) = -0.7
y(1,9) = 1.2
y(1,10) = 0.8
y(1,11) = 0.4
y(1,12) = 0.5
y(1,13) = 0.1
```

[1] Diese Zeitdifferenzen sind jedoch unerheblich.

```
y(1,14) = 0.8
y(1,15) = -0.8
y(1,16) = -0.8
```

Die zugehörigen Gewichte sind empirisch gewonnen. Sie beruhen auf keiner Berechnung.

```
w(1,1) = 0.5
w(1,2) = 0.6
w(1,3) = 0.1
w(1,4) = -0.3
w(2,1) = -0.4
w(2,2) = 0.2
w(2,3) = 0.3
w(2,4) = 0.7
w(3,1) = 0.4
w(3,2) = -0.2
w(3,3) = -0.1
w(3,4) = 0.3
w(4,1) = -0.5
w(4,2) = 0
w(4,3) = -0.3
w(4,4) = 0.2
w(5,1) = 0.3
w(5,2) = -0.1
w(5,3) = 0.2
w(5,4) = -0.2
w(6,1) = -0.5
w(6,2) = 0.5
w(6,3) = 0.6
w(6,4) = -0.1
w(7,1) = 0.2
w(7,2) = 0.1
w(7,3) = -0.2
w(7,4) = 0.2
w(8,1) = -0.3
w(8,2) = -0.1
w(8,3) = -0.4
w(8,4) = 0.1
w(9,1) = 0.5
w(9,2) = 0.3
w(9,3) = 0
w(9,4) = 0.4
w(10,1) = 0.4
w(10,2) = 0.2
w(10,3) = -0.1
w(10,4) = 0.3
w(11,1) = 0.3
w(11,2) = 0.1
w(11,3) = -0.2
w(11,4) = 0.2
w(12,1) = 0.4
w(12,2) = 0.5
w(12,3) = 0
w(12,4) = -0.4
```



```
w(13,1) = -0.6
w(13,2) = 0.7
w(13,3) = 0
w(13,4) = 0
w(14,1) = 0.2
w(14,2) = 0.2
w(14,3) = 0.2
w(14,4) = 0.2
w(15,1) = -0.2
w(15,2) = -0.2
w(15,3) = -0.2
w(15,4) = -0.2
w(16,1) = -0.3
w(16,2) = -0.1
w(16,3) = -0.1
w(16,4) = -0.3
```

Die zugehörigen Änderungen liegen um Größenordnungen über den Werten der 9. Iteration, sind also noch voll im Gradientenabstieg begriffen.

```
Deltaw(1,1) = -0.21675
Deltaw(1,2) = -0.21675
Deltaw(1,3) = -0.21675
Deltaw(1,4) = -0.21675
Deltaw(2,1) = -0.20319
Deltaw(2,2) = -0.20319
Deltaw(2,3) = -0.20319
Deltaw(2,4) = -0.20319
Deltaw(3,1) = -0.11855
Deltaw(3,2) = -0.11855
Deltaw(3,3) = -0.11855
Deltaw(3,4) = -0.11855
Deltaw(4,1) = 0.16662
Deltaw(4,2) = 0.16662
Deltaw(4,3) = 0.16662
Deltaw(4,4) = 0.16662
Deltaw(5,1) = -0.061674
Deltaw(5,2) = -0.061674
Deltaw(5,3) = -0.061674
Deltaw(5,4) = -0.061674
Deltaw(6,1) = -0.14389
Deltaw(6,2) = -0.14389
Deltaw(6,3) = -0.14389
Deltaw(6,4) = -0.14389
Deltaw(7,1) = -0.09099
Deltaw(7,2) = -0.09099
Deltaw(7,3) = -0.09099
Deltaw(7,4) = -0.09099
Deltaw(8,1) = 0.18645
Deltaw(8,2) = 0.18645
Deltaw(8,3) = 0.18645
Deltaw(8,4) = 0.18645
Deltaw(9,1) = -0.23885
Deltaw(9,2) = -0.23885
```



```
Deltaw(9,3) = -0.23885
Deltaw(9,4) = -0.23885
Deltaw(10,1) = -0.20319
Deltaw(10,2) = -0.20319
Deltaw(10,3) = -0.20319
Deltaw(10,4) = -0.20319
Deltaw(11,1) = -0.11855
Deltaw(11,2) = -0.11855
Deltaw(11,3) = -0.11855
Deltaw(11,4) = -0.11855
Deltaw(12,1) = -0.14389
Deltaw(12,2) = -0.14389
Deltaw(12,3) = -0.14389
Deltaw(12,4) = -0.14389
Deltaw(13,1) = -0.031146
Deltaw(13,2) = -0.031146
Deltaw(13,3) = -0.031146
Deltaw(13,4) = -0.031146
Deltaw(14,1) = -0.20319
Deltaw(14,2) = -0.20319
Deltaw(14,3) = -0.20319
Deltaw(14,4) = -0.20319
Deltaw(15,1) = 0.20319
Deltaw(15,2) = 0.20319
Deltaw(15,3) = 0.20319
Deltaw(15,4) = 0.20319
Deltaw(16,1) = 0.20319
Deltaw(16,2) = 0.20319
Deltaw(16,3) = 0.20319
Deltaw(16,4) = 0.20319
```

Die Funktionswerte streuen zu Beginn der Iteration noch stark um den Mittelwert der Sigmoidfunktion von 0,5.

```
f(1,1) = 0.71095
f(1,2) = 0.68997
f(1,3) = 0.59869
f(1,4) = 0.35434
f(1,5) = 0.54983
f(1,6) = 0.62246
f(1,7) = 0.57444
f(1,8) = 0.33181
f(1,9) = 0.76852
f(1,10) = 0.68997
f(1,11) = 0.59869
f(1,12) = 0.62246
f(1,13) = 0.52498
f(1,14) = 0.68997
f(1,15) = 0.31003
f(1,16) = 0.31003
```

Der Fehler des Verfahrens muß naturgemäß zu Beginn der Iteration am größten sein und wird von nun an sukzessive kleiner:

```
E = 0.20234
```


Wir haben vorstehend zwar eine relativ simple quadratische Elementarzelle[2] angenommen, aber es sind auch andere Zellbildungen möglich, nämlich solche mit 6 nächsten Nachbarn, welche die Form eines Sechsecks (Wabenstruktur) aufweisen. Eine solche Struktur läßt sich in 6 gleichseitige Dreiecke zerlegen, wobei jeder Vogel nur seine beiden Vordermänner anzupeilen braucht, und das unter einem Winkel von genau 60°. Wird eine solche Formation mit zwei Flügelmännern gebildet, läßt sie sich zu einer sogenannten Deltaformation ausbauen. Auch hier gehen also trotz des kleineren Blickfeldes immer noch drei Schätzgrößen in die Optimierung ein. Diesbezüglich weist das neuronale Netz also keinerlei Beschränkungen auf.

[2] Auf drei Dimensionen erweitert wäre es die eines Natriumchloridgitters.

