

# Übungsblatt 9

## „Mustererkennung“

J. Cavojska, N. Lehmann, R. Toudic

06.07.2015

### Inhaltsverzeichnis

<b>1</b>	<b>Aufgabe 1 - XOR-Netzwerk mit Backpropagation trainieren</b>	<b>2</b>
<b>2</b>	<b>Aufgabe 2 - Handschriftbuchstaben klassifizieren</b>	<b>5</b>
2.1	k = 2 . . . . .	5
2.2	k = 4 . . . . .	9
2.3	k = 8 . . . . .	13
2.4	k = 10 . . . . .	17

# 1 Aufgabe 1 - XOR-Netzwerk mit Backpropagation trainieren

```

1  W1_init    = rand(3,2);           % random weights 3x2 from input ↔
   to layer 1
2  W2_init    = rand(3,1);           % random weights 3x1 from layer ↔
   1 to layer 2
3  LTD        = [1 1 0; 1 0 1; 0 1 1; 0 0 0]; % labeled training data
4  L          = [0; 1; 1; 0];        % labels
5  ATD        = [1 1 1; 1 0 1; 0 1 1; 0 0 1]; % augmented data without labels
6  E1         = [];                  % error history for plot
7  e1         = 1;                   % just for e != 0
8
9  % set initial values
10 W1          = W1_init;
11 W2          = W2_init;
12 alpha       = 0.1;                % learning rate
13 eq1         = 0.0001;              % error quality
14
15 % learning
16 for runs=1:100000
17     random    = randi(4);
18     L0        = ATD(random,:);
19     label     = L(random,:);
20
21     % forward pass
22     % layer 1
23     t         = L0 * W1;
24     perceptron1_layer1 = 1.0 / (1.0 + exp(-t(:,1)));
25     perceptron2_layer1 = 1.0 / (1.0 + exp(-t(:,2)));
26     out_layer1 = [perceptron1_layer1, perceptron2_layer1];
27
28     % layer 2
29     t         = [perceptron1_layer1, perceptron2_layer1, 1]*W2;
30     perceptron1_layer2 = 1.0 / (1.0 + exp(-t));
31     out_layer2 = perceptron1_layer2;
32
33     % error calculation
34     e1        = perceptron1_layer2 - label;
35     E1        = horzcat(E1,e1);
36
37     % backward pass
38     t1        = L0 * W1;
39     s11       = (1.0 / (1+exp(-t1(:,1))))*(1-(1 / (1+exp(-t1(:,1))))↔
   ));
40     s12       = (1.0 / (1+exp(-t1(:,2))))*(1-(1 / (1+exp(-t1(:,2))))↔
   ));
41     D1        = [s11, 0; 0, s12];
42
43     t2        = [out_layer1, 1] * W2;
44     s2        = (1 / (1+exp(-t2)))*(1-(1 / (1+exp(-t2))));
45     D2        = s2;
46

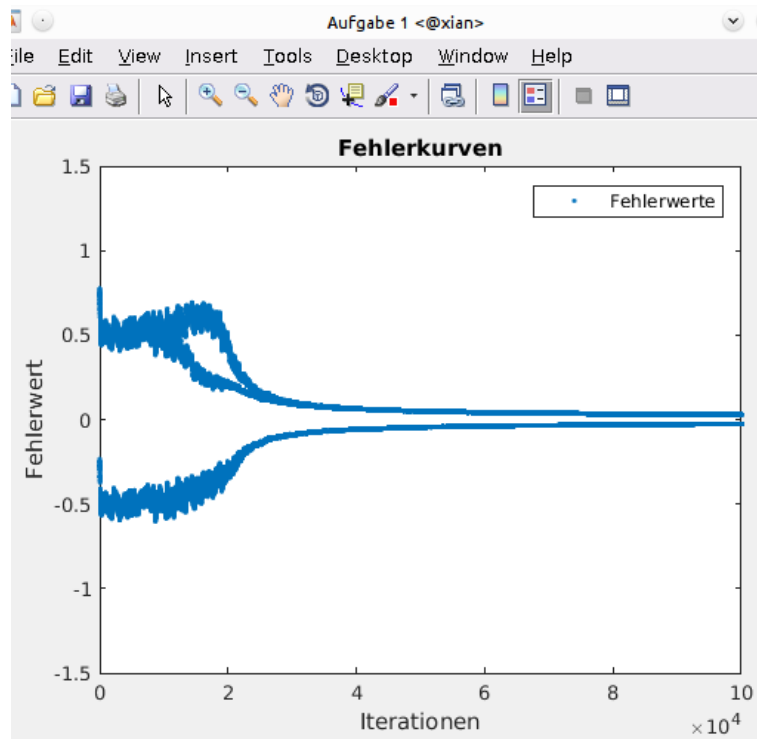
```

```

47     W2_          = W2(1:2,:);
48     dW1          = -alpha*D1*W2_*D2*e1'*L0;
49     dW2          = -alpha*D2*e1'*[out_layer1, 1];
50     W1           = W1 + dW1';
51     W2           = W2 + dW2';
52
53 end
54
55 needed_iterations = length(E1);
56
57 % plot
58 figure('NumberTitle','off','Name','Aufgabe 1');
59 plot(E1, 'r')
60 title('Fehlerkurven');
61 xlabel('Iterationen');
62 ylabel('Fehlerwert');
63 axis([-0.1 needed_iterations -1.5 1.5]);
64 legend('Fehlerwerte');
65
66 % gelernte Gewichte:
67 W1
68 %      6.2294      4.2282
69 %      6.2397      4.2218
70 %     -2.7229     -6.4831
71
72 W2
73 %      8.6958
74 %     -9.4197
75 %     -3.9711

```

Wir haben alle errors  $e_1$  aufgesammelt und in einem Scatter-Plot dargestellt:



## 2 Aufgabe 2 - Handschriftbuchstaben klassifizieren

Datenaufbereitung fuer Aufg. 2

```
1 LTD      = horzcat(cTraining,labelsTraining); % labeled training data
2 ATD      = horzcat(cTraining,ones(7494,1)); % augmented data without labels
3
4 LTDtest   = horzcat(cTesting,labelsTesting); % labeled test data
5 ATDtest   = horzcat(cTesting,ones(3498,1)); % augmented data without labels
6
7 predictedClassk2 = [];
8 predictedClassk4 = [];
9 predictedClassk8 = [];
10 predictedClassk10 = [];
```

### 2.1 $k = 2$

Der Code fuer eine hidden layer size von 2:

```
1 % k = 2, Training
2 E2      = []; % error history for plot
3 e2      = 1; % just for e != 0
4
5 % set initial values
6 W1      = rand(17,2); % random weights 17x2 from layer 0 to layer 1
7 W2      = rand(3,10); % random weights 3x10 from layer 1 to layer 2
8 alpha   = 0.01; % learning rate
9 eq2     = 0.0001; % error quality
10
11
12 for runs = 1:length(ATD)
13
14     L0      = ATD(runs,:);
15     label   = labelsTraining(runs);
16
17     % forward pass
18     % layer 1
19     t       = L0 * W1;
20     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
21     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
22
23     out_layer1 = [perceptron01_layer1,perceptron02_layer1];
24
25     % layer 2
```

```

26     t                = [out_layer1, 1]*W2;
27     perceptron01_layer2 = 1 / (1 + exp(-t(1)));
28     perceptron02_layer2 = 1 / (1 + exp(-t(2)));
29     perceptron03_layer2 = 1 / (1 + exp(-t(3)));
30     perceptron04_layer2 = 1 / (1 + exp(-t(4)));
31     perceptron05_layer2 = 1 / (1 + exp(-t(5)));
32     perceptron06_layer2 = 1 / (1 + exp(-t(6)));
33     perceptron07_layer2 = 1 / (1 + exp(-t(7)));
34     perceptron08_layer2 = 1 / (1 + exp(-t(8)));
35     perceptron09_layer2 = 1 / (1 + exp(-t(9)));
36     perceptron10_layer2 = 1 / (1 + exp(-t(10)));
37
38     out_layer2        = [perceptron01_layer2,perceptron02_layer2,↵
        perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵
        perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
        perceptron09_layer2,perceptron10_layer2];
39
40     % error calculation
41     labelVector = zeros(1,10);
42     for labelIndex = 1:10
43         if label == labelIndex
44             labelVector(:,labelIndex) = 1;
45         end
46     end
47
48
49     e2                = out_layer2 - labelVector;
50     E2                = horzcat(E2,sum(e2));
51
52
53     % backward pass
54     t1                = L0 * W1;
55     s11                = (1 / (1+exp(-t1(:,1))))*(1-(1 / (1+exp(-t1(:,1)))))↵
        ;
56     s12                = (1 / (1+exp(-t1(:,2))))*(1-(1 / (1+exp(-t1(:,2)))))↵
        ;
57
58     D1                = [s11,0;
59         0,s12];
60
61     t2                = [out_layer1, 1] * W2;
62     s201               = (1 / (1+exp(-t2(1))))*(1-(1 / (1+exp(-t2(1)))))↵
63     s202               = (1 / (1+exp(-t2(2))))*(1-(1 / (1+exp(-t2(2)))))↵
64     s203               = (1 / (1+exp(-t2(3))))*(1-(1 / (1+exp(-t2(3)))))↵
65     s204               = (1 / (1+exp(-t2(4))))*(1-(1 / (1+exp(-t2(4)))))↵
66     s205               = (1 / (1+exp(-t2(5))))*(1-(1 / (1+exp(-t2(5)))))↵
67     s206               = (1 / (1+exp(-t2(6))))*(1-(1 / (1+exp(-t2(6)))))↵
68     s207               = (1 / (1+exp(-t2(7))))*(1-(1 / (1+exp(-t2(7)))))↵
69     s208               = (1 / (1+exp(-t2(8))))*(1-(1 / (1+exp(-t2(8)))))↵
70     s209               = (1 / (1+exp(-t2(9))))*(1-(1 / (1+exp(-t2(9)))))↵
71     s210               = (1 / (1+exp(-t2(10))))*(1-(1 / (1+exp(-t2(10)))))↵
72
73     D2                = [s201,0,0,0,0,0,0,0,0,0;
74         0,s202,0,0,0,0,0,0,0,0;
75         0,0,s203,0,0,0,0,0,0,0;
76         0,0,0,s204,0,0,0,0,0,0;
77         0,0,0,0,s205,0,0,0,0,0;

```

```

78         0,0,0,0,0,s206,0,0,0,0;
79         0,0,0,0,0,0,s207,0,0,0;
80         0,0,0,0,0,0,0,s208,0,0;
81         0,0,0,0,0,0,0,0,s209,0;
82         0,0,0,0,0,0,0,0,0,s210];
83
84     W2_          = W2(1:2,:);
85     dW1          = -alpha*D1*W2_*D2*e2'*L0;
86     dW2          = -alpha*D2*e2'*[out_layer1, 1];
87     W1           = W1 + dW1';
88     W2           = W2 + dW2';
89
90 end
91
92 needed_iterations = length(E2);
93
94 % plot
95 figure('NumberTitle','off','Name','Aufgabe 2, k=2');
96 plot(E2, 'b.')
97 title('Fehlerkurven');
98 xlabel('Iterationen');
99 ylabel('Fehlerwert');
100 axis([-0.1 needed_iterations -10 10]);
101 legend('Fehlerwerte');
102
103
104 % k = 2, Testing
105
106 correctly_predicted = 0;
107 for runs = 1:length(ATDtest)
108
109     L0          = ATDtest(runs,:);
110     label       = labelsTesting(runs);
111
112     % forward pass
113     % layer 1
114     t           = L0 * W1;
115     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
116     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
117
118     out_layer1  = [perceptron01_layer1,perceptron02_layer1];
119
120     % layer 2
121     t           = [out_layer1, 1]*W2;
122     perceptron01_layer2 = 1 / (1 + exp(-t(1)));
123     perceptron02_layer2 = 1 / (1 + exp(-t(2)));
124     perceptron03_layer2 = 1 / (1 + exp(-t(3)));
125     perceptron04_layer2 = 1 / (1 + exp(-t(4)));
126     perceptron05_layer2 = 1 / (1 + exp(-t(5)));
127     perceptron06_layer2 = 1 / (1 + exp(-t(6)));
128     perceptron07_layer2 = 1 / (1 + exp(-t(7)));
129     perceptron08_layer2 = 1 / (1 + exp(-t(8)));
130     perceptron09_layer2 = 1 / (1 + exp(-t(9)));
131     perceptron10_layer2 = 1 / (1 + exp(-t(10)));
132
133     out_layer2  = [perceptron01_layer2,perceptron02_layer2,↵
        perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵

```

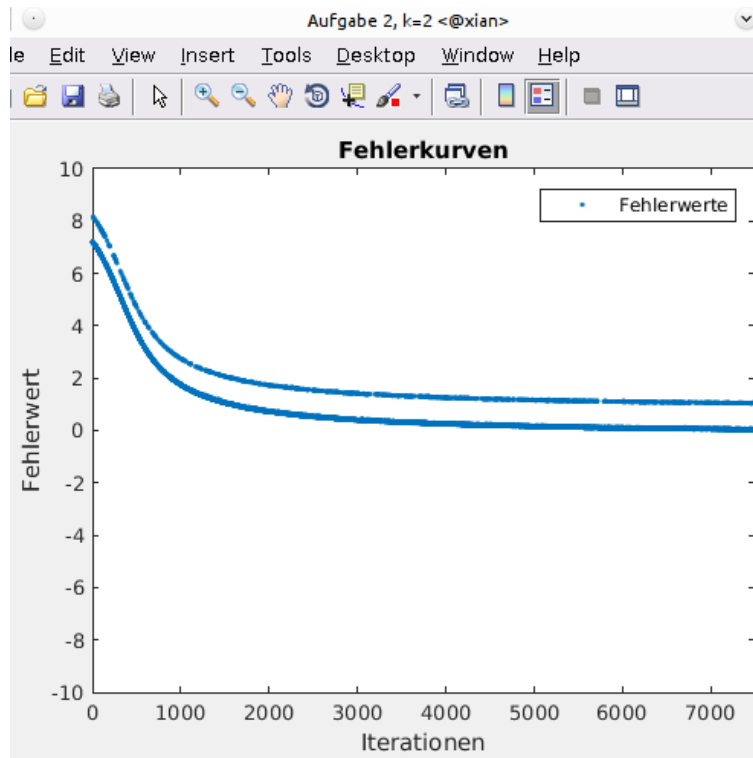
```

134         perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,←
            perceptron09_layer2,perceptron10_layer2];
135
136     % prediction calculation
137     prediction = 999; % initial value
138     predictionVal = max(out_layer2);
139     for index = 1:length(out_layer2)
140         if out_layer2(1, index) == predictionVal
141             prediction = index - 1;
142         end
143     end
144     predictedClassk2 = vertcat(predictedClassk2, prediction);
145     if prediction == label
146         correctly_predicted = correctly_predicted + 1;
147     end
148 end
149
150 confusionMatrix_k2 = confusionmat(labelsTesting, predictedClassk2)
151
152 %confusionMatrix_k2 =
153 %
154 %      0      0      0      0      0      0      0      363      0      0      0
155 %      0      0      0      0      0      0      0      364      0      0      0
156 %      0      0      0      0      0      0      0      364      0      0      0
157 %      0      0      0      0      0      0      0      336      0      0      0
158 %      0      0      0      0      0      0      0      364      0      0      0
159 %      0      0      0      0      0      0      0      335      0      0      0
160 %      0      0      0      0      0      0      0      336      0      0      0
161 %      0      0      0      0      0      0      0      364      0      0      0
162 %      0      0      0      0      0      0      0      336      0      0      0
163 %      0      0      0      0      0      0      0      336      0      0      0
164
165     klass_guete = correctly_predicted / size(ATDtest, 1)
166
167 %klass_guete =
168 %
169 %      0.0961

```

Ein Plot des beim Training mit  $k = 2$  entstandenen Errors:





## 2.2 $k = 4$

```

1 % k = 4, Training
2 E4      = [];           % error history for plot
3 e4      = 1;           % just for e != 0
4
5 % set initial values
6 W1      = rand(17,4);   % random weights 17x2 from layer 0 to layer 1
7 W2      = rand(5,10);   % random weights 5x10 from layer 1 to layer 2
8 alpha   = 0.01;        % learning rate
9
10 for runs = 1:length(ATD)
11
12     LO          = ATD(runs,:);
13     label       = labelsTraining(runs);
14
15     % forward pass
16     % layer 1
17     t           = LO * W1;
18     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
19     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
20     perceptron03_layer1 = 1 / (1 + exp(-t(:,3)));
21     perceptron04_layer1 = 1 / (1 + exp(-t(:,4)));

```

```

22
23 out_layer1 = [perceptron01_layer1,perceptron02_layer1,↵
    perceptron03_layer1,perceptron04_layer1];
24
25 % layer 2
26 t = [out_layer1, 1]*W2;
27 perceptron01_layer2 = 1 / (1 + exp(-t(1)));
28 perceptron02_layer2 = 1 / (1 + exp(-t(2)));
29 perceptron03_layer2 = 1 / (1 + exp(-t(3)));
30 perceptron04_layer2 = 1 / (1 + exp(-t(4)));
31 perceptron05_layer2 = 1 / (1 + exp(-t(5)));
32 perceptron06_layer2 = 1 / (1 + exp(-t(6)));
33 perceptron07_layer2 = 1 / (1 + exp(-t(7)));
34 perceptron08_layer2 = 1 / (1 + exp(-t(8)));
35 perceptron09_layer2 = 1 / (1 + exp(-t(9)));
36 perceptron10_layer2 = 1 / (1 + exp(-t(10)));
37
38 out_layer2 = [perceptron01_layer2,perceptron02_layer2,↵
    perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵
    perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
    perceptron09_layer2,perceptron10_layer2];
39
40 % error calculation
41 labelVector = zeros(1,10);
42 for labelIndex = 1:10
43     if label == labelIndex
44         labelVector(:,labelIndex) = 1;
45     end
46 end
47 e4 = out_layer2 - labelVector;
48 E4 = horzcat(E4,sum(e4));
49
50
51 % backward pass
52 t1 = L0 * W1;
53 s11 = (1 / (1+exp(-t1(:,1))))*(1-(1 / (1+exp(-t1(:,1)))))↵
    ;
54 s12 = (1 / (1+exp(-t1(:,2))))*(1-(1 / (1+exp(-t1(:,2)))))↵
    ;
55 s13 = (1 / (1+exp(-t1(:,3))))*(1-(1 / (1+exp(-t1(:,3)))))↵
    ;
56 s14 = (1 / (1+exp(-t1(:,4))))*(1-(1 / (1+exp(-t1(:,4)))))↵
    ;
57
58 D1 = [s11,0,0,0;
59       0,s12,0,0;
60       0,0,s13,0;
61       0,0,0,s14];
62
63 t2 = [out_layer1, 1] * W2;
64 s201 = (1 / (1+exp(-t2(1))))*(1-(1 / (1+exp(-t2(1)))))↵
65 s202 = (1 / (1+exp(-t2(2))))*(1-(1 / (1+exp(-t2(2)))))↵
66 s203 = (1 / (1+exp(-t2(3))))*(1-(1 / (1+exp(-t2(3)))))↵
67 s204 = (1 / (1+exp(-t2(4))))*(1-(1 / (1+exp(-t2(4)))))↵
68 s205 = (1 / (1+exp(-t2(5))))*(1-(1 / (1+exp(-t2(5)))))↵
69 s206 = (1 / (1+exp(-t2(6))))*(1-(1 / (1+exp(-t2(6)))))↵
70 s207 = (1 / (1+exp(-t2(7))))*(1-(1 / (1+exp(-t2(7)))))↵

```

```

71     s208 = (1 / (1+exp(-t2(8))))*(1-(1 / (1+exp(-t2(8)))));
72     s209 = (1 / (1+exp(-t2(9))))*(1-(1 / (1+exp(-t2(9)))));
73     s210 = (1 / (1+exp(-t2(10))))*(1-(1 / (1+exp(-t2(10)))));
74
75     D2 = [s201,0,0,0,0,0,0,0,0,0;
76           0,s202,0,0,0,0,0,0,0,0;
77           0,0,s203,0,0,0,0,0,0,0;
78           0,0,0,s204,0,0,0,0,0,0;
79           0,0,0,0,s205,0,0,0,0,0;
80           0,0,0,0,0,s206,0,0,0,0;
81           0,0,0,0,0,0,s207,0,0,0;
82           0,0,0,0,0,0,0,s208,0,0;
83           0,0,0,0,0,0,0,0,s209,0;
84           0,0,0,0,0,0,0,0,0,s210];
85
86     W2_ = W2(1:4,:);
87     dW1 = -alpha*D1*W2_*D2*e4'*L0;
88     dW2 = -alpha*D2*e4'*[out_layer1, 1];
89     W1 = W1 + dW1';
90     W2 = W2 + dW2';
91
92 end
93
94 needed_iterations = length(E4);
95
96 % plot
97 figure('NumberTitle','off','Name','Aufgabe 2, k=4');
98 plot(E4, '.');
99 title('Fehlerkurven');
100 xlabel('Iterationen');
101 ylabel('Fehlerwert');
102 axis([-0.1 needed_iterations -10 10]);
103 legend('Fehlerwerte');
104
105
106 % k = 4, Testing
107
108 correctly_predicted = 0;
109 for runs = 1:length(ATDtest)
110
111     L0 = ATDtest(runs,:);
112     label = labelsTesting(runs);
113
114     % forward pass
115     % layer 1
116     t = L0 * W1;
117     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
118     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
119     perceptron03_layer1 = 1 / (1 + exp(-t(:,3)));
120     perceptron04_layer1 = 1 / (1 + exp(-t(:,4)));
121
122     out_layer1 = [perceptron01_layer1,perceptron02_layer1,↵
123                  perceptron03_layer1,perceptron04_layer1];
124
125     % layer 2
126     t = [out_layer1, 1]*W2;
127     perceptron01_layer2 = 1 / (1 + exp(-t(1)));

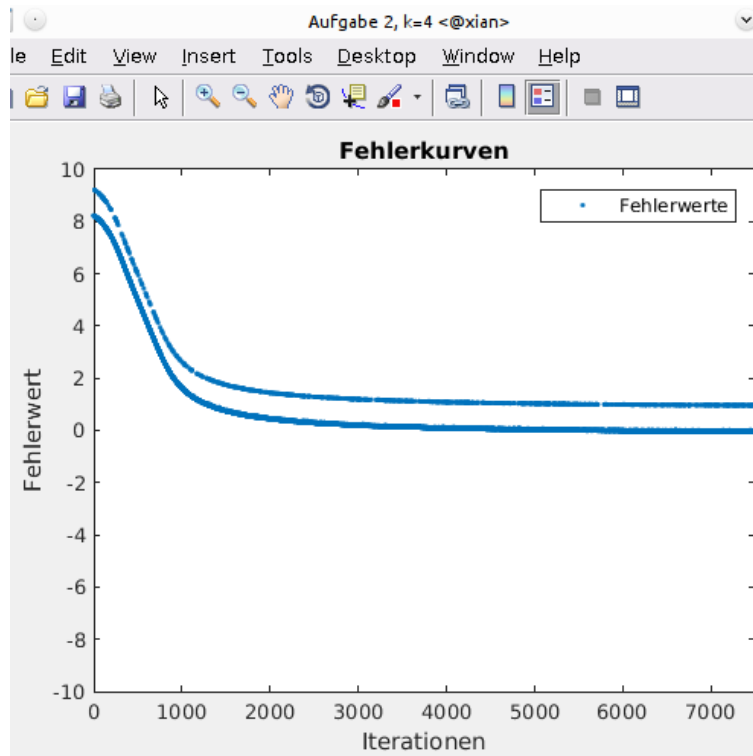
```

```

127 perceptron02_layer2 = 1 / (1 + exp(-t(2)));
128 perceptron03_layer2 = 1 / (1 + exp(-t(3)));
129 perceptron04_layer2 = 1 / (1 + exp(-t(4)));
130 perceptron05_layer2 = 1 / (1 + exp(-t(5)));
131 perceptron06_layer2 = 1 / (1 + exp(-t(6)));
132 perceptron07_layer2 = 1 / (1 + exp(-t(7)));
133 perceptron08_layer2 = 1 / (1 + exp(-t(8)));
134 perceptron09_layer2 = 1 / (1 + exp(-t(9)));
135 perceptron10_layer2 = 1 / (1 + exp(-t(10)));
136
137 out_layer2 = [perceptron01_layer2,perceptron02_layer2,↵
    perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵
    perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
    perceptron09_layer2,perceptron10_layer2];
138
139 % prediction calculation
140 prediction = 999; % initial value
141 predictionVal = max(out_layer2);
142 for index = 1:length(out_layer2)
143     if out_layer2(1, index) == predictionVal
144         prediction = index - 1;
145     end
146 end
147 predictedClassk4 = vertcat(predictedClassk4, prediction);
148 if prediction == label
149     correctly_predicted = correctly_predicted + 1;
150 end
151 end
152
153
154 confusionMatrix_k4 = confusionmat(labelsTesting, predictedClassk4)
155
156 %confusionMatrix_k4 =
157 %
158 %      0      0      0      0      0      0      0      363      0      0      0
159 %      0      0      0      0      0      0      0      364      0      0      0
160 %      0      0      0      0      0      0      0      364      0      0      0
161 %      0      0      0      0      0      0      0      336      0      0      0
162 %      0      0      0      0      0      0      0      364      0      0      0
163 %      0      0      0      0      0      0      0      335      0      0      0
164 %      0      0      0      0      0      0      0      336      0      0      0
165 %      0      0      0      0      0      0      0      364      0      0      0
166 %      0      0      0      0      0      0      0      336      0      0      0
167 %      0      0      0      0      0      0      0      336      0      0      0
168
169 klass_guete = correctly_predicted / size(ATDtest, 1)
170
171 %klass_guete =
172 %
173 %      0.0961

```

Ein Plot des beim Training mit  $k = 4$  entstandenen Errors:



## 2.3 $k = 8$

```

1 % k = 8, Training
2 E8      = [];           % error history for plot
3 e8      = 1;           % just for e != 0
4
5 % set initial values
6 W1      = rand(17,8);   % random weights 17x2 from layer 0 to layer 1
7 W2      = rand(9,10);   % random weights 3x10 from layer 1 to layer 2
8 alpha   = 0.01;        % learning rate
9
10 for runs = 1:length(ATD)
11
12     L0    = ATD(runs,:);
13     label = labelsTraining(runs);
14
15     % forward pass
16     % layer 1
17     t     = L0 * W1;
18     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
19     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
20     perceptron03_layer1 = 1 / (1 + exp(-t(:,3)));
21     perceptron04_layer1 = 1 / (1 + exp(-t(:,4)));

```

```

22 perceptron05_layer1 = 1 / (1 + exp(-t(:,5)));
23 perceptron06_layer1 = 1 / (1 + exp(-t(:,6)));
24 perceptron07_layer1 = 1 / (1 + exp(-t(:,7)));
25 perceptron08_layer1 = 1 / (1 + exp(-t(:,8)));
26
27 out_layer1 = [perceptron01_layer1,perceptron02_layer1,↵
                perceptron03_layer1,perceptron04_layer1,perceptron05_layer1,↵
                perceptron06_layer1,perceptron07_layer1,perceptron08_layer1];
28
29 % layer 2
30 t = [out_layer1, 1]*W2;
31 perceptron01_layer2 = 1 / (1 + exp(-t(1)));
32 perceptron02_layer2 = 1 / (1 + exp(-t(2)));
33 perceptron03_layer2 = 1 / (1 + exp(-t(3)));
34 perceptron04_layer2 = 1 / (1 + exp(-t(4)));
35 perceptron05_layer2 = 1 / (1 + exp(-t(5)));
36 perceptron06_layer2 = 1 / (1 + exp(-t(6)));
37 perceptron07_layer2 = 1 / (1 + exp(-t(7)));
38 perceptron08_layer2 = 1 / (1 + exp(-t(8)));
39 perceptron09_layer2 = 1 / (1 + exp(-t(9)));
40 perceptron10_layer2 = 1 / (1 + exp(-t(10)));
41
42 out_layer2 = [perceptron01_layer2,perceptron02_layer2,↵
                perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵
                perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
                perceptron09_layer2,perceptron10_layer2];
43
44 % error calculation
45 labelVector = zeros(1,10);
46 for labelIndex = 1:10
47     if label == labelIndex
48         labelVector(:,labelIndex) = 1;
49     end
50 end
51
52 e8 = out_layer2 - labelVector;
53 E8 = horzcat(E8,sum(e8));
54
55 % backward pass
56 t1 = L0 * W1;
57 s11 = (1 / (1+exp(-t1(:,1))))*(1-(1 / (1+exp(-t1(:,1)))))↵
    ;
58 s12 = (1 / (1+exp(-t1(:,2))))*(1-(1 / (1+exp(-t1(:,2)))))↵
    ;
59 s13 = (1 / (1+exp(-t1(:,3))))*(1-(1 / (1+exp(-t1(:,3)))))↵
    ;
60 s14 = (1 / (1+exp(-t1(:,4))))*(1-(1 / (1+exp(-t1(:,4)))))↵
    ;
61 s15 = (1 / (1+exp(-t1(:,5))))*(1-(1 / (1+exp(-t1(:,5)))))↵
    ;
62 s16 = (1 / (1+exp(-t1(:,6))))*(1-(1 / (1+exp(-t1(:,6)))))↵
    ;
63 s17 = (1 / (1+exp(-t1(:,7))))*(1-(1 / (1+exp(-t1(:,7)))))↵
    ;
64 s18 = (1 / (1+exp(-t1(:,8))))*(1-(1 / (1+exp(-t1(:,8)))))↵
    ;
65

```

```

66     D1          = [s11,0,0,0,0,0,0,0;
67                   0,s12,0,0,0,0,0,0;
68                   0,0,s13,0,0,0,0,0;
69                   0,0,0,s14,0,0,0,0;
70                   0,0,0,0,s15,0,0,0;
71                   0,0,0,0,0,s16,0,0;
72                   0,0,0,0,0,0,s17,0;
73                   0,0,0,0,0,0,0,s18];
74
75     t2          = [out_layer1, 1] * W2;
76     s201        = (1 / (1+exp(-t2(1))))*(1-(1 / (1+exp(-t2(1)))));
77     s202        = (1 / (1+exp(-t2(2))))*(1-(1 / (1+exp(-t2(2)))));
78     s203        = (1 / (1+exp(-t2(3))))*(1-(1 / (1+exp(-t2(3)))));
79     s204        = (1 / (1+exp(-t2(4))))*(1-(1 / (1+exp(-t2(4)))));
80     s205        = (1 / (1+exp(-t2(5))))*(1-(1 / (1+exp(-t2(5)))));
81     s206        = (1 / (1+exp(-t2(6))))*(1-(1 / (1+exp(-t2(6)))));
82     s207        = (1 / (1+exp(-t2(7))))*(1-(1 / (1+exp(-t2(7)))));
83     s208        = (1 / (1+exp(-t2(8))))*(1-(1 / (1+exp(-t2(8)))));
84     s209        = (1 / (1+exp(-t2(9))))*(1-(1 / (1+exp(-t2(9)))));
85     s210        = (1 / (1+exp(-t2(10))))*(1-(1 / (1+exp(-t2(10)))));
86
87     D2          = [s201,0,0,0,0,0,0,0,0,0;
88                   0,s202,0,0,0,0,0,0,0,0;
89                   0,0,s203,0,0,0,0,0,0,0;
90                   0,0,0,s204,0,0,0,0,0,0;
91                   0,0,0,0,s205,0,0,0,0,0;
92                   0,0,0,0,0,s206,0,0,0,0;
93                   0,0,0,0,0,0,s207,0,0,0;
94                   0,0,0,0,0,0,0,s208,0,0;
95                   0,0,0,0,0,0,0,0,s209,0;
96                   0,0,0,0,0,0,0,0,0,s210];
97
98     W2_         = W2(1:8,:);
99     dW1         = -alpha*D1*W2_*D2*e8'*L0;
100    dW2         = -alpha*D2*e8'*[out_layer1, 1];
101    W1          = W1 + dW1';
102    W2          = W2 + dW2';
103
104 end
105
106 needed_iterations = length(E8);
107
108 % plot
109 figure('NumberTitle','off','Name','Aufgabe 2, k=8');
110 plot(E8, 'r')
111 title('Fehlerkurven');
112 xlabel('Iterationen');
113 ylabel('Fehlerwert');
114 axis([-0.1 needed_iterations -10 10]);
115 legend('Fehlerwerte');
116
117
118 % k = 8, Testing
119
120 correctly_predicted = 0;
121 for runs = 1:length(ATDtest)
122

```

```

123     LO      = ATDtest(runs,:);
124     label   = labelsTesting(runs);
125
126     % forward pass
127     % layer 1
128     t                = LO * W1;
129     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
130     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
131     perceptron03_layer1 = 1 / (1 + exp(-t(:,3)));
132     perceptron04_layer1 = 1 / (1 + exp(-t(:,4)));
133     perceptron05_layer1 = 1 / (1 + exp(-t(:,5)));
134     perceptron06_layer1 = 1 / (1 + exp(-t(:,6)));
135     perceptron07_layer1 = 1 / (1 + exp(-t(:,7)));
136     perceptron08_layer1 = 1 / (1 + exp(-t(:,8)));
137
138     out_layer1      = [perceptron01_layer1,perceptron02_layer1,↵
        perceptron03_layer1,perceptron04_layer1,perceptron05_layer1,↵
        perceptron06_layer1,perceptron07_layer1,perceptron08_layer1];
139
140     % layer 2
141     t                = [out_layer1, 1]*W2;
142     perceptron01_layer2 = 1 / (1 + exp(-t(1)));
143     perceptron02_layer2 = 1 / (1 + exp(-t(2)));
144     perceptron03_layer2 = 1 / (1 + exp(-t(3)));
145     perceptron04_layer2 = 1 / (1 + exp(-t(4)));
146     perceptron05_layer2 = 1 / (1 + exp(-t(5)));
147     perceptron06_layer2 = 1 / (1 + exp(-t(6)));
148     perceptron07_layer2 = 1 / (1 + exp(-t(7)));
149     perceptron08_layer2 = 1 / (1 + exp(-t(8)));
150     perceptron09_layer2 = 1 / (1 + exp(-t(9)));
151     perceptron10_layer2 = 1 / (1 + exp(-t(10)));
152
153     out_layer2      = [perceptron01_layer2,perceptron02_layer2,↵
        perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵
        perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
        perceptron09_layer2,perceptron10_layer2];
154
155     % prediction calculation
156     prediction = 999; % initial value
157     predictionVal = max(out_layer2);
158     for index = 1:length(out_layer2)
159         if out_layer2(1, index) == predictionVal
160             prediction = index - 1;
161         end
162     end
163     predictedClassk8 = vertcat(predictedClassk8, prediction);
164     if prediction == label
165         correctly_predicted = correctly_predicted + 1;
166     end
167
168 end
169
170
171 confusionMatrix_k8 = confusionmat(labelsTesting, predictedClassk8)
172
173 %confusionMatrix_k8 =
174 %

```

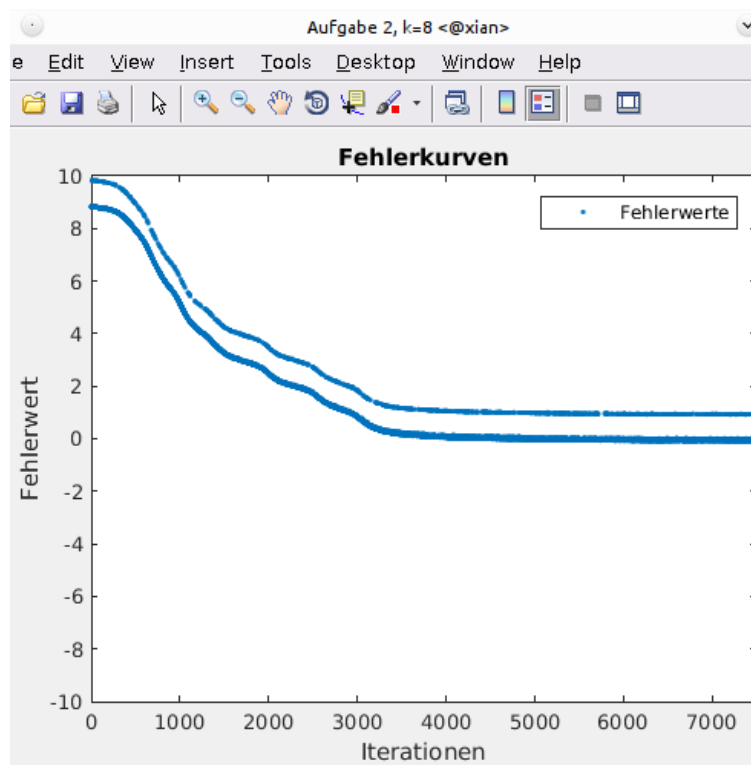


```

175 %      0      0      0      0      0      0      0      363      0      0      0
176 %      0      0      0      0      0      0      0      364      0      0      0
177 %      0      0      0      0      0      0      0      364      0      0      0
178 %      0      0      0      0      0      0      0      336      0      0      0
179 %      0      0      0      0      0      0      0      364      0      0      0
180 %      0      0      0      0      0      0      0      335      0      0      0
181 %      0      0      0      0      0      0      0      336      0      0      0
182 %      0      0      0      0      0      0      0      364      0      0      0
183 %      0      0      0      0      0      0      0      336      0      0      0
184 %      0      0      0      0      0      0      0      336      0      0      0
185
186
187 klass_guete = correctly_predicted / size(ATDtest, 1)
188
189 %klass_guete =
190 %
191 %      0.0961

```

Ein Plot des beim Training mit  $k = 8$  entstandenen Errors:



## 2.4 $k = 10$

```

1 % k = 10, Training
2 E10      = [];          % error history for plot
3 e10      = 1;          % just for e != 0
4
5 % set initial values
6 W1       = rand(17,10); % random weights 17x2 from layer 0 to layer 1
7 W2       = rand(11,10); % random weights 3x10 from layer 1 to layer 2
8 alpha    = 0.01;       % learning rate
9
10 for runs = 1:length(ATD)
11
12     LO          = ATD(runs,:);
13     label       = labelsTraining(runs);
14
15     % forward pass
16     % layer 1
17     t           = LO * W1;
18     perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
19     perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
20     perceptron03_layer1 = 1 / (1 + exp(-t(:,3)));
21     perceptron04_layer1 = 1 / (1 + exp(-t(:,4)));
22     perceptron05_layer1 = 1 / (1 + exp(-t(:,5)));
23     perceptron06_layer1 = 1 / (1 + exp(-t(:,6)));
24     perceptron07_layer1 = 1 / (1 + exp(-t(:,7)));
25     perceptron08_layer1 = 1 / (1 + exp(-t(:,8)));
26     perceptron09_layer1 = 1 / (1 + exp(-t(:,9)));
27     perceptron10_layer1 = 1 / (1 + exp(-t(:,10)));
28
29     out_layer1   = [perceptron01_layer1,perceptron02_layer1,↵
        perceptron03_layer1,perceptron04_layer1,perceptron05_layer1,↵
        perceptron06_layer1,perceptron07_layer1,perceptron08_layer1,↵
        perceptron09_layer1,perceptron10_layer1];
30
31     % layer 2
32     t           = [out_layer1, 1]*W2;
33     perceptron01_layer2 = 1 / (1 + exp(-t(1)));
34     perceptron02_layer2 = 1 / (1 + exp(-t(2)));
35     perceptron03_layer2 = 1 / (1 + exp(-t(3)));
36     perceptron04_layer2 = 1 / (1 + exp(-t(4)));
37     perceptron05_layer2 = 1 / (1 + exp(-t(5)));
38     perceptron06_layer2 = 1 / (1 + exp(-t(6)));
39     perceptron07_layer2 = 1 / (1 + exp(-t(7)));
40     perceptron08_layer2 = 1 / (1 + exp(-t(8)));
41     perceptron09_layer2 = 1 / (1 + exp(-t(9)));
42     perceptron10_layer2 = 1 / (1 + exp(-t(10)));
43
44     out_layer2   = [perceptron01_layer2,perceptron02_layer2,↵
        perceptron03_layer2,perceptron04_layer2,perceptron05_layer2,↵
        perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
        perceptron09_layer2,perceptron10_layer2];
45
46     % error calculation
47     labelVector = zeros(1,10);

```

```

48     for labelIndex = 1:10
49         if label == labelIndex
50             labelVector(:,labelIndex) = 1;
51         end
52     end
53     e10          = out_layer2 - labelVector;
54     E10          = horzcat(E10,sum(e10));
55
56     % backward pass
57     t1           = L0 * W1;
58     s101         = (1 / (1+exp(-t1(:,1))))*(1-(1 / (1+exp(-t1(:,1)))))←
59     ;
60     s102         = (1 / (1+exp(-t1(:,2))))*(1-(1 / (1+exp(-t1(:,2)))))←
61     ;
62     s103         = (1 / (1+exp(-t1(:,3))))*(1-(1 / (1+exp(-t1(:,3)))))←
63     ;
64     s104         = (1 / (1+exp(-t1(:,4))))*(1-(1 / (1+exp(-t1(:,4)))))←
65     ;
66     s105         = (1 / (1+exp(-t1(:,5))))*(1-(1 / (1+exp(-t1(:,5)))))←
67     ;
68     s106         = (1 / (1+exp(-t1(:,6))))*(1-(1 / (1+exp(-t1(:,6)))))←
69     ;
70     s107         = (1 / (1+exp(-t1(:,7))))*(1-(1 / (1+exp(-t1(:,7)))))←
71     ;
72     s108         = (1 / (1+exp(-t1(:,8))))*(1-(1 / (1+exp(-t1(:,8)))))←
73     ;
74     s109         = (1 / (1+exp(-t1(:,9))))*(1-(1 / (1+exp(-t1(:,9)))))←
75     ;
76     s110        = (1 / (1+exp(-t1(:,10))))*(1-(1 / (1+exp(-t1(:,10)))))←
77     ));
78
79     D1          = [s101,0,0,0,0,0,0,0,0,0;
80                   0,s102,0,0,0,0,0,0,0,0;
81                   0,0,s103,0,0,0,0,0,0,0;
82                   0,0,0,s104,0,0,0,0,0,0;
83                   0,0,0,0,s105,0,0,0,0,0;
84                   0,0,0,0,0,s106,0,0,0,0;
85                   0,0,0,0,0,0,s107,0,0,0;
86                   0,0,0,0,0,0,0,s108,0,0;
87                   0,0,0,0,0,0,0,0,s109,0;
88                   0,0,0,0,0,0,0,0,0,s110];
89
90     t2          = [out_layer1, 1] * W2;
91     s201         = (1 / (1+exp(-t2(1))))*(1-(1 / (1+exp(-t2(1)))))←
92     ;
93     s202         = (1 / (1+exp(-t2(2))))*(1-(1 / (1+exp(-t2(2)))))←
94     ;
95     s203         = (1 / (1+exp(-t2(3))))*(1-(1 / (1+exp(-t2(3)))))←
96     ;
97     s204         = (1 / (1+exp(-t2(4))))*(1-(1 / (1+exp(-t2(4)))))←
98     ;
99     s205         = (1 / (1+exp(-t2(5))))*(1-(1 / (1+exp(-t2(5)))))←
100    ;
101    s206         = (1 / (1+exp(-t2(6))))*(1-(1 / (1+exp(-t2(6)))))←
102    ;
103    s207         = (1 / (1+exp(-t2(7))))*(1-(1 / (1+exp(-t2(7)))))←
104    ;
105    s208         = (1 / (1+exp(-t2(8))))*(1-(1 / (1+exp(-t2(8)))))←
106    ;
107    s209         = (1 / (1+exp(-t2(9))))*(1-(1 / (1+exp(-t2(9)))))←
108    ;
109    s210         = (1 / (1+exp(-t2(10))))*(1-(1 / (1+exp(-t2(10)))))←
110    ;
111
112     D2          = [s201,0,0,0,0,0,0,0,0,0;
113                   0,s202,0,0,0,0,0,0,0,0;
114                   0,0,s203,0,0,0,0,0,0,0;

```

```

95         0,0,0,s204,0,0,0,0,0,0;
96         0,0,0,0,s205,0,0,0,0,0;
97         0,0,0,0,0,s206,0,0,0,0;
98         0,0,0,0,0,0,s207,0,0,0;
99         0,0,0,0,0,0,0,s208,0,0;
100        0,0,0,0,0,0,0,0,s209,0;
101        0,0,0,0,0,0,0,0,0,s210];
102
103        W2_          = W2(1:10,:);
104        dW1          = -alpha*D1*W2_*D2*e10'*L0;
105        dW2          = -alpha*D2*e10'*[out_layer1, 1];
106        W1           = W1 + dW1';
107        W2           = W2 + dW2';
108
109    end
110
111    needed_iterations = length(E10);
112
113    % plot
114    figure('NumberTitle','off','Name','Aufgabe 2, k=10');
115    plot(E10, '.');
116    title('Fehlerkurven');
117    xlabel('Iterationen');
118    ylabel('Fehlerwert');
119    axis([-0.1 needed_iterations -10 10]);
120    legend('Fehlerwerte');
121
122
123    % k = 10, Testing
124    correctly_predicted = 0;
125    for runs = 1:length(ATDtest)
126
127        L0          = ATDtest(runs,:);
128        label       = labelsTesting(runs);
129
130        % forward pass
131        % layer 1
132        t           = L0 * W1;
133        perceptron01_layer1 = 1 / (1 + exp(-t(:,1)));
134        perceptron02_layer1 = 1 / (1 + exp(-t(:,2)));
135        perceptron03_layer1 = 1 / (1 + exp(-t(:,3)));
136        perceptron04_layer1 = 1 / (1 + exp(-t(:,4)));
137        perceptron05_layer1 = 1 / (1 + exp(-t(:,5)));
138        perceptron06_layer1 = 1 / (1 + exp(-t(:,6)));
139        perceptron07_layer1 = 1 / (1 + exp(-t(:,7)));
140        perceptron08_layer1 = 1 / (1 + exp(-t(:,8)));
141        perceptron09_layer1 = 1 / (1 + exp(-t(:,9)));
142        perceptron10_layer1 = 1 / (1 + exp(-t(:,10)));
143
144        out_layer1    = [perceptron01_layer1,perceptron02_layer1,↵
145            perceptron03_layer1,perceptron04_layer1,perceptron05_layer1,↵
146            perceptron06_layer1,perceptron07_layer1,perceptron08_layer1,↵
147            perceptron09_layer1,perceptron10_layer1];
148
149        % layer 2
150        t           = [out_layer1, 1]*W2;
151        perceptron01_layer2 = 1 / (1 + exp(-t(1)));

```

```

149 perceptron02_layer2 = 1 / (1 + exp(-t(2)));
150 perceptron03_layer2 = 1 / (1 + exp(-t(3)));
151 perceptron04_layer2 = 1 / (1 + exp(-t(4)));
152 perceptron05_layer2 = 1 / (1 + exp(-t(5)));
153 perceptron06_layer2 = 1 / (1 + exp(-t(6)));
154 perceptron07_layer2 = 1 / (1 + exp(-t(7)));
155 perceptron08_layer2 = 1 / (1 + exp(-t(8)));
156 perceptron09_layer2 = 1 / (1 + exp(-t(9)));
157 perceptron10_layer2 = 1 / (1 + exp(-t(10)));
158
159 out_layer2 = [perceptron01_layer2,perceptron02_layer2,↵
    perceptron03_layer2,perceptron04_layer2,perceptron05_layer2, ↵
    perceptron06_layer2,perceptron07_layer2,perceptron08_layer2,↵
    perceptron09_layer2,perceptron10_layer2];
160
161 % prediction calculation
162 prediction = 999; % initial value
163 predictionVal = max(out_layer2);
164 for index = 1:length(out_layer2)
165     if out_layer2(1, index) == predictionVal
166         prediction = index - 1;
167     end
168 end
169 predictedClassk10 = vertcat(predictedClassk10, prediction);
170 if prediction == label
171     correctly_predicted = correctly_predicted + 1;
172 end
173 end
174
175
176 confusionMatrix_k10 = confusionmat(labelsTesting, predictedClassk10)
177
178 %confusionMatrix_k10 =
179 %
180 %      0      363      0      0      0      0      0      0      0      0
181 %      0      364      0      0      0      0      0      0      0      0
182 %      0      364      0      0      0      0      0      0      0      0
183 %      0      336      0      0      0      0      0      0      0      0
184 %      0      364      0      0      0      0      0      0      0      0
185 %      0      335      0      0      0      0      0      0      0      0
186 %      0      336      0      0      0      0      0      0      0      0
187 %      0      364      0      0      0      0      0      0      0      0
188 %      0      336      0      0      0      0      0      0      0      0
189 %      0      336      0      0      0      0      0      0      0      0
190
191
192 klass_guete = correctly_predicted / size(ATDtest, 1)
193
194 %klass_guete =
195 %
196 %      0.1041

```

Ein Plot des beim Training mit  $k = 10$  entstandenen Errors:

