

Dr. Jakob Bernasconi

Informationsverarbeitung in neuronalen Netzwerken

Frühjahrssemester 2009

Musterlösung Übung 4

Aufgabe 1

Für den Gewichtssatz

$$\begin{array}{lll} W_{10} = 0.5 & W_{11} = 1 & W_{12} = 0 \\ W_{20} = 0.5 & W_{21} = -1 & W_{22} = 0 \\ W_{30} = 0.5 & W_{31} = 0 & W_{32} = 1 \\ W_{40} = 0.5 & W_{41} = 0 & W_{42} = -1 \end{array}$$

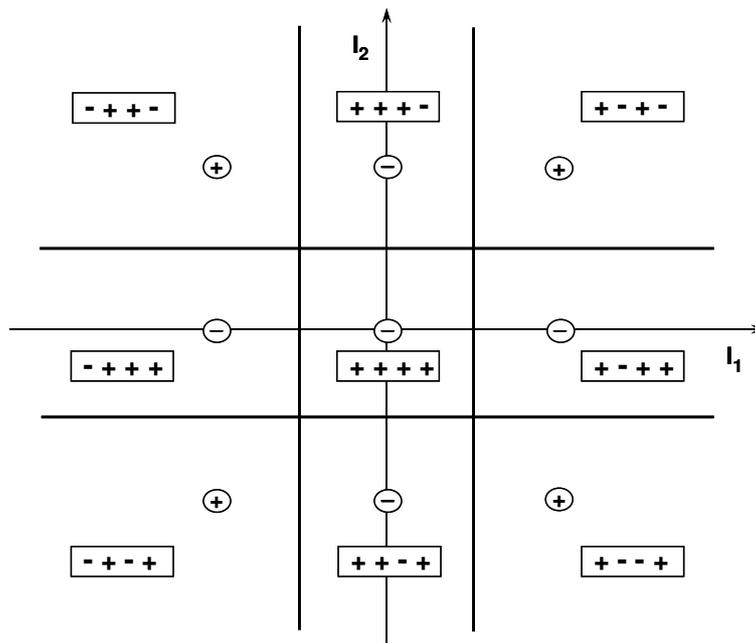
sind in der folgenden Figur die durch die 4 Hidden Units definierten Trenngeraden graphisch dargestellt. Die entsprechende Unterteilung des Inputraumes ist so, dass die Teilgebiete nur je ein Lernbeispiel enthalten. In den rechteckigen Boxen sind zudem die zugehörigen Werte der 4 Hidden Units S_1, S_2, S_3 und S_4 angegeben.

Um zu entscheiden, ob die Gewichte V_0 bis V_4 so gewählt werden können, dass der Output O alle 7 Lernbeispiele richtig klassifiziert, muss der entsprechende Satz von Ungleichungen untersucht werden:

Für jedes Teilgebiet ergibt sich eine Ungleichung der Form

$$V_0 + V_1 \cdot S_1 + V_2 \cdot S_2 + V_3 \cdot S_3 + V_4 \cdot S_4 \geq 0 \quad \text{bzw.} \quad < 0,$$

je nachdem, ob der gewünschte Output D in diesem Gebiet den Wert $+1$ oder -1 hat.



Die resultierenden Ungleichungen sind die folgenden:

$$V_0 - V_1 + V_2 + V_3 - V_4 \geq 0$$

$$V_0 + V_1 - V_2 + V_3 - V_4 \geq 0$$

$$V_0 - V_1 + V_2 - V_3 + V_4 \geq 0$$

$$V_0 + V_1 - V_2 - V_3 + V_4 \geq 0$$

$$V_0 - V_1 + V_2 + V_3 + V_4 < 0$$

$$V_0 + V_1 + V_2 + V_3 - V_4 < 0$$

$$V_0 + V_1 + V_2 + V_3 + V_4 < 0$$

$$V_0 + V_1 + V_2 - V_3 + V_4 < 0$$

$$V_0 + V_1 - V_2 + V_3 + V_4 < 0$$

Diese Ungleichungen lassen sich z.B. recht einfach mit dem Ansatz

$$V_0 = a, \quad V_1 = V_2 = V_3 = V_4 = b$$

lösen. Es ergeben sich dann noch die folgenden Ungleichungen:

$$a \geq 0$$

$$a + 2b < 0$$

$$a + 4b < 0$$

Mit der Wahl $a = 1$, $b = -1$ (zum Beispiel) sind dann also alle Ungleichungen für die V_i erfüllt.

Aufgabe 2

Die beigelegten Graphiken beziehen sich auf drei Lösungen (A, B und C), die mit den folgenden Konfigurations-Files gefunden wurden:

config_file_A:

```
inputs=2
h1=4
h2=0
outputs=1
```

```
output_function=1
hidden_function=1
A=1
B=1
```

```
eta=0.05
alpha=0
w0=2
```

```
seed=35031
```

```
mode=0;
iterations=1'000'000;
```

config_file_B:

```
inputs=2
h1=4
h2=0
outputs=1
```

```
output_function=2
hidden_function=1
A=1
B=1
```

```
eta=0.1
alpha=0
w0=2
```

```
seed=83731
```

```
mode=0;
iterations=1'000'000;
```

config_file_C:

```
inputs=2
h1=3
h2=0
outputs=1
```

```
output_function=1
hidden_function=1
A=1
B=1
```

```
eta=0.05
alpha=0
w0=2
```

```
seed=35031
```

```
mode=0;
iterations=500'000;
```

Die folgende Tabelle zeigt die für die 9 Lernbeispiele erzielten Output-Werte der entsprechenden Lösungen:

v	I ₁	I ₂	D	O _A	O _B	O _C
1	-1	-1	+1	+0.99	+0.23	+0.96
2	0	-1	-1	-0.99	+0.25	-0.97
3	+1	-1	+1	+0.99	+0.27	+0.97
4	-1	0	-1	-0.99	-1.10	-0.97
5	0	0	-1	-1.00	-1.10	-1.00
6	+1	0	-1	-0.99	-1.10	-0.98
7	-1	+1	+1	+0.99	-0.23	+0.97
8	0	+1	-1	-0.99	-0.17	-0.98
9	+1	+1	+1	+0.99	-0.12	+0.99

Bemerkungen:

zu Lösung A:

Eine fast perfekte symmetrische Lösung.

Diesen Typ von Lösungen mit 4 hidden Units würde man auch aufgrund von analytischen Überlegungen erwarten.

Mit nur 10'000 Iterationen ergeben sich bereits gute Approximationswerte von etwa ± 0.85 .

zu Lösung B:

Offenbar Steckenbleiben in einem schlechten lokalen Minimum ("Badewannen-Lösung").

Auch nach 1'000'000 Iterationen sieht man keine Anzeichen für ein eventuelles Herauskommen aus dieser Lösung.

Zu Lösung C:

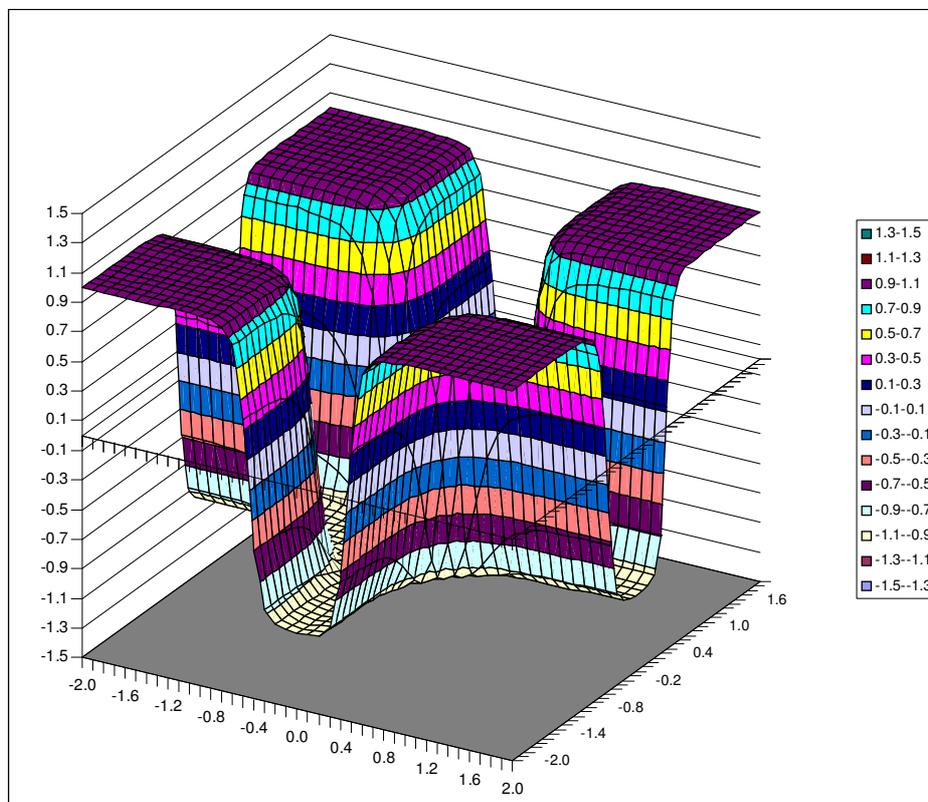
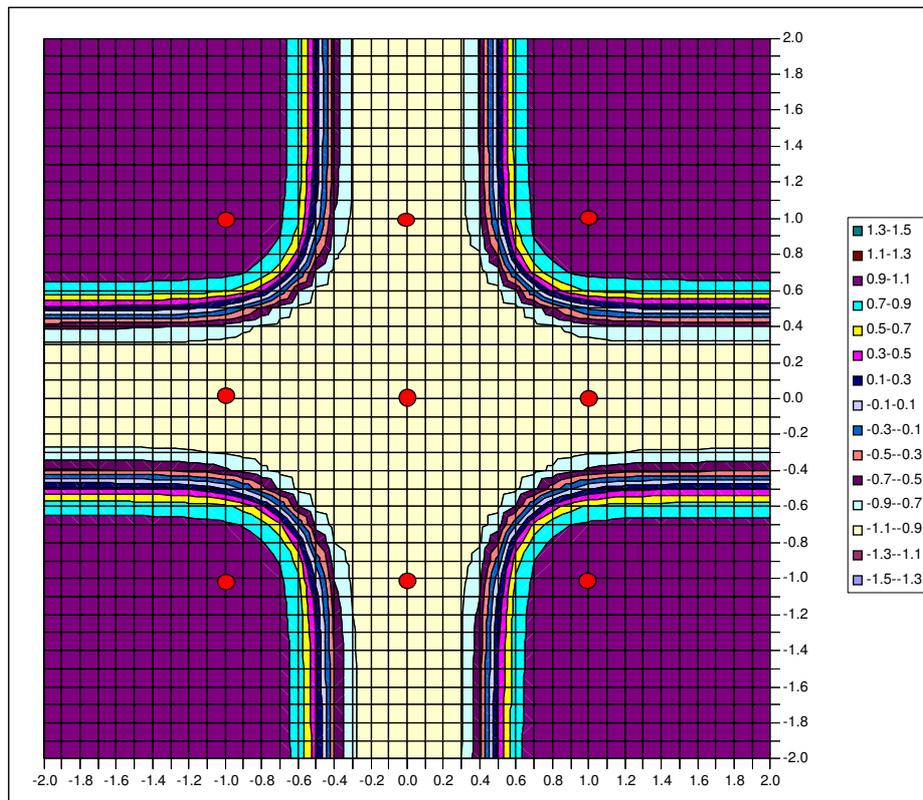
Fast perfekte Lösung mit nur 3 hidden Units nach 500'000 Iterationen.

Eine Lösung, auf die man mit analytischen Überlegungen kaum kommen würde.

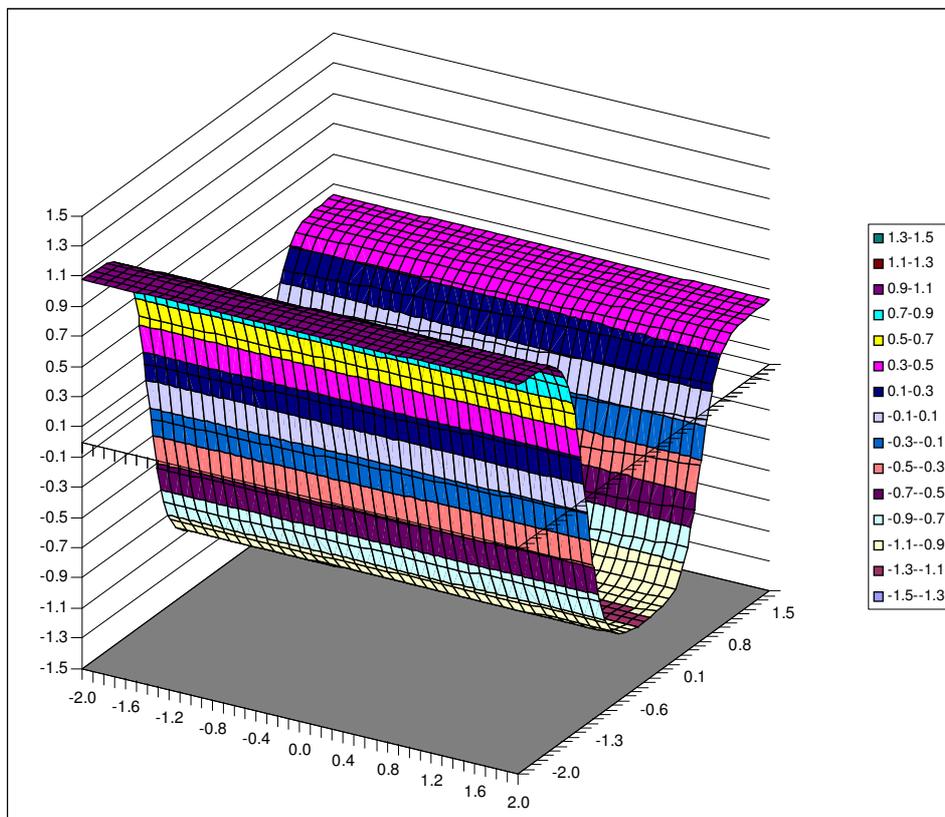
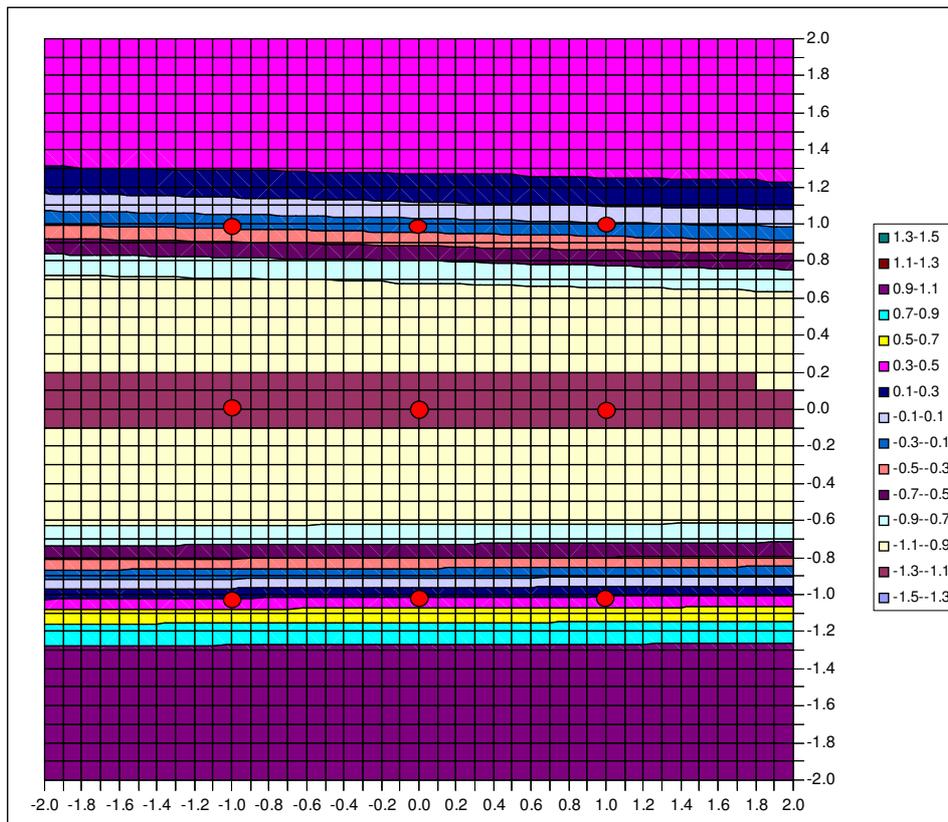
Allgemein:

- Man beobachtet eine grosse Abhängigkeit vom Random Seed für die Wahl der Anfangsgewichte. Nur ein Bruchteil der Versuche führt zu guten Lösungen, was auf das Vorhandensein von lokalen Minima hindeutet.
 - Eine lineare Aktivierungsfunktion für das Output-Neuron (`output_function=2`) führt manchmal zu einer etwas schnelleren Konvergenz als eine sigmoide Aktivierungsfunktion, die zwischen -1 und +1 variiert (`output_function=1`).
 - Auch eine Wahl von $A > 1$ [z.B. $A = 1.1$], $B > 1$ [z.B. $B = 2$] oder $\alpha > 0$ [z.B. $\alpha = 0.7$] kann die Konvergenz des Lernprozesses beschleunigen.
 - Mit dem Batch-Lernmodus [`mode=1`] erreicht man ähnliche Resultate wie beim Example-by-Example Verfahren [`mode=0`] mit etwa 10 mal weniger Iterationen. (Was zu erwarten ist, da in einem Batch-Durchgang alle 9 Lernbeispiele präsentiert werden.)
 - Eine grössere Anzahl von hidden Units oder die Hinzunahme eines zweiten hidden Layers scheinen keine signifikanten Vorteile zu bringen.
-

Lösung A:



Lösung B:



Lösung C:

