## Home | Startseite | Impressum | Kontakt | Gästebuch

Aufgabe: Berechnen Sie mit Hilfe des neuronalen Netzwerk-Tools von MATLAB die Inversen von drei binären Eingangsneuronen für jede der insgesamt acht Permutationen. Verwenden Sie als Aktivierungsfunktion eine Heaviside-Sprungfunktion.

Lösung: Die Tabelle der acht Permutationen aller Eingangs- und Ausgangsneuronen lautet:

Nr.	$x_{1}^{(1)}$	$x_{2}^{(1)}$	$x_{3}^{(1)}$	$x_1^{(3)}$	$x_2^{(3)}$	$x_{3}^{(3)}$
1	0	0	0	1	1	1
2	0	0	1	1	1	0
3	0	1	0	1	0	1
4	0	1	1	1	0	0
5	1	0	0	0	1	1
6	1	0	1	0	1	0
7	1	1	0	0	0	1
8	1	1	1	0	0	0

Nach Eintippen von *nntool* zum Starten des Tools wechseln Sie im Fenster Create Network or Data zu **Data**, geben Sie den Input-Vektor p ein, welcher der Matrix  $(x_1^{(1)}, x_2^{(1)}, x_3^{(1)})$  entspricht,

☆ Create Network or Data           Network         Data	
Name P	
Value	Data Type
[0 0 0 0 1 1 1 1;0 0 1 1 0 0 1 1;0 1 0 1	<ul> <li>Inputs</li> <li>Targets</li> <li>Input Delay States</li> <li>Layer Delay States</li> <li>Outputs</li> <li>Errors</li> </ul>
V Help	😤 Create 🛛 🔇 Close

und generieren Sie mit **Create** das Input file. Verfahren Sie mit dem Zielvektor t, welcher der Matrix  $(x_1^{(3)}, x_2^{(3)}, x_3^{(3)})$  entspricht, genauso:

Sector Create Network or Data		
Network Data		
Name		
t		
Value	Data Type	
[1 1 1 1 0 0 0 0;1 1 0 0 1 1 0 0;1 0 1 0	Inputs	
	Targets	
	Input Delay States	
	Cayer Delay States	
	Outputs	
	Errors	
W Help	😤 Create 🛛 🙆 Close	

Erzeugen Sie auf dem Reiter **Network** im Fenster Create Network or Data ein neues Netzwerk mit dem Namen INVNet, setzen Sie **Network Type** auf Perceptron und wählen Sie mit dem Abwärtspfeil als **Input data** p aus, als **Target data** t und geben Sie die **Transfer function** HARDLIM und die **Learning function** LEARNP ein. Das Fenster Create Network or Data sieht nun so aus:

😤 Create Network or Data	
Network Data	
Name	
INVNet	
Network Properties	
Network Type:	Perceptron 👻
Input data:	p 🔻
Target data:	t 👻
Transfer function:	HARDLIM 👻
Learning function:	LEARNP 👻
	View 😪 Restore Defaults
() Help	😤 Create 🛛 🙆 Close

Um das Netzwerk zu trainieren, markieren Sie die Zeile, in der INVNet steht, klicken Sie danach auf **Open** und wählen Sie auf dem Reiter **Train** als **Inputs** p und als **Targets** t.

aining Data			Training Results	
puts	p	•	Outputs	INVNet_outputs
argets	t	•	Errors	INVNet_errors
it Input Delay States	(zeros)	-	Final Input Delay States	INVNet_inputStates
it Layer Delay States	(zeros)	<b>v</b>	Final Layer Delay States	INVNet_layerStates

Die **Training Parameters** können Sie wie immer ändern. Danach klicken Sie auf den Button **Train Network** und es erscheinen die Trainingsergebnisse:

📣 Neural Network Training (nntraintool)						
Neural Network						
Layer Input 3 b 3						
Algorithms	Algorithms					
Training: Cyclica Performance: Mean A Derivative: Default	Training:         Cyclical Weight/Bias Rule (trainc)           Performance:         Mean Absolute Error (mae)           Derivative:         Default (defaultderiv)					
Progress						
Epoch: 0	2 iterations	1000				
Time:	0:00:00					
Performance: 0.500	0.00	0.00				
Plots						
Performance (	Performance (plotperform)					
Training State (plottrainstate)						
Plot Interval:						
V Performance goa	al met.					
	Stop Train	ing Cancel				

Gehen Sie im INVNet-Fenster auf den Reiter **Simulate**, geben Sie dort als **Inputs** p und als **Targets** t ein und klicken Sie danach auf **Simulate Network**.

imulation Data			Simulation Results	
nputs	p	•	Outputs	INVNet_outputs
nit Input Delay States	(zeros)	~	Final Input Delay States	INVNet_inputStates
nit Layer Delay States	(zeros)	~	Final Layer Delay States	INVNet_layerStates
upply Targets				
argets	t	•	Errors	INVNet_errors
	n			

Im Neural Network/Data Manager erscheinen die neuen Variablen INVNet\_outputs und IN-VNet\_errors.

📣 Neural Network/Data Manager (nntool)		
P Input Data:	Vetworks INV/Net	Output Data:     INVNet_outputs
<b>⊚</b> Target Data: t		Error Data:
⊙ Input Delay States:		⊗ Layer Delay States:
Simport 😤 New 🔲 Oper	S Export Export	V Help Close

Doppelklicken Sie diese Variablen in den kleinen Fenstern und Sie erhalten die Ergebnisse

🛃 Data: INVNet_outputs 🗖 🗉 🖾	🛃 Data: INVNet_errors 🗖 🗉 🖾
Value	Value
[1 1 1 1 0 0 0 0;	[0 0 0 0 0 0 0;
11001100; 10101010]	000000;
OK Cancel	OK Cancel

Klicken Sie nun auf **Export**, dann auf **Select All** und geben Sie im MATLAB-Editor *who* ein. Danach können Sie die einzelnen Variablen der Reihe nach auslesen.

>> INVNet\_outputs