Home | Startseite | Impressum | Kontakt | Gästebuch

Aufgabe: Erstellen Sie ein Perzeptron-Netzwerk für die logische UND-Verknüpfung mit Hilfe des MATLAB Tools.

T		<i>a</i> :	C 1	T 1	. 1	• •		0 1 1	-
Lösung:	Inppen	Sie zum	Starten des	Tools	s nntool e	in Es	erscheint	tolgendes	Fenster:
Losang	1 pp en		Starten del	10010	11110010		ensemenne	1015011400	1 01150011

📣 Neural Network/Data Manager (nntool)		
P Input Data:	Vetworks	Output Data:
Target Data:		Krror Data:
𝔆 Input Delay States:		🕑 Layer Delay States:
Simport 🔅 New 🔲 Open.	💧 Export	🕖 Help 🛛 🙆 Close

Klicken Sie auf **New** und es erscheint das Fenster Create Network or Data. Wechseln Sie zu **Data**, setzen Sie **Name** auf p, **Value** auf [0 0 1 1;0 1 0 1] und stellen Sie sicher, daß **Data Type** auf **Inputs** steht. Danach drücken Sie **Create**, um das Input file p zu generieren. Der Create Network or Data Manager erscheint und zeigt p als Eingabe.

Create Network or Data	
Network Data	
Name	
p	
Value	Data Type
[0 0 1 1;0 1 0 1]	Inputs
	Targets
	Input Delay States
	Cayer Delay States
	Outputs
	Errors
🕖 Help	😤 Create 🛛 🔇 Close

Als nächstes erzeugen wir das Netz-Target. Klicken Sie erneut auf **New** und geben Sie diesmal für **Name** t ein, für **Value** den Vektor [0 0 0 1] und klicken Sie auf den **Data Type** Targets.

😤 Create Network or Data	
Network Data	
Name	
t	
Value	Data Type
[0 0 0 1]	◎ Inputs
	Targets
	Input Delay States
	Cayer Delay States
	Outputs
	© Errors
V Help	😤 Create 🛛 🔇 Close

Im Neural Network/Data Manager erscheint daraufhin p als Input Data und t als Target Data. Wir erzeugen nun ein neues Netzwerk mit dem Namen ANDNet. Gehen Sie auf **Network** im Fenster Create Network or Data, geben Sie ANDNet als **Name** ein und setzen Sie **Network Type** auf Perceptron. Wählen Sie mit dem Abwärtspfeil unter **Input data** p aus und unter **Target data** t und geben Sie als **Transfer function** HARDLIM und als **Learning function** LEARNP ein. Das Fenster Create Network or Data sieht nun wie folgt aus:

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

Klicken Sie auf View und es erscheint das Fenster



Dann klicken Sie auf **Create.** Das Netzwerk ANDNet ist nun im Neural Network/Data Manager gelistet.

📣 Neural Network/Data Manager (nntool)		
P	ANDNet	• Output Data:
❷ Target Data: t		K Error Data:
Solution States:		♥ Layer Delay States:
Simport 😤 New 🔲 Open	🔇 Export 🗶 Delete	V Help Olose

Um das Netzwerk zu trainieren, markieren Sie die Zeile, in der ANDNet steht, und klicken danach auf **Open**. Wählen Sie den Reiter **Train** und geben Sie unter **Inputs** p und unter **Targets** t ein.

🗱 Network: ANDNet					
View Train Simulate Ada	apt Reinitialize Weights View	/Edit Weights			
Training Info Training Parameters					
Training Data		Training Results			
Inputs	p	▼ Outputs	ANDNet_outputs		
Targets	t	Errors	ANDNet_errors		
Init Input Delay States	(zeros)	 Final Input Delay States 	ANDNet_inputStates		
Init Layer Delay States	(zeros)	Final Layer Delay States	ANDNet_layerStates		
			🐚 Train Network		

Copyright © 2017, Manfred Hiebl. Alle Rechte vorbehalten.

An der Stelle gehen Sie auf den Reiter **Training Parameters**. Diese Werte können Sie ändern. Danach klicken Sie auf den Button **Train Network**. Es erscheinen die Trainingsergebnisse:

📣 Neural Network Training (nntraintool)	
Neural Network	
Layer Input 2 1	Output
Algorithms Training: Cyclical Weight/Bias Rule (trainc) Performance: Mean Absolute Error (mae) Derivative: Default (defaultderiv)	
Epoch: 0 5 iterations Time: 0:00:00 0:00 Performance: 0.750 0.00	0.00
Plots Performance (plotperform) Training State (plottrainstate) Plot Interval:	1 epochs
Performance goal met.	ng 🖉 Cancel

Bestätigen Sie nunmehr, daß das trainierte Netzwerk wirklich einen verschwindenden Fehler liefert, indem Sie auf das ANDNet-Fenster gehen und den **Simulate**-Reiter auswählen. Geben Sie unter **Inputs** p ein und klicken Sie dann auf **Simulate Network**.

		Simulation Results	
p	-	Outputs	ANDNet_outputs
(zeros)	-	Final Input Delay States	ANDNet_inputStates
Init Layer Delay States (zeros)		Final Layer Delay States	ANDNet_layerStates
(zeros)	-	Errors	ANDNet_errors
	p (zeros) (zeros) (zeros)	p ▼ (zeros) ▼ (zeros) ▼ [zeros) ▼	p • (zeros) • (zeros) • (zeros) • (zeros) • (zeros) • Errors •

Im Neural Network/Data Manager erscheint die neue Variable ANDNet_outputs. Doppelklicken Sie die Zeile in dem kleinen Fenster und es erscheinen die ANDNet_outputs-Daten

💑 Data: ANDNet_outputs	
Value	
[0 0 0 1]	
UK VK	Cancel

Gehen Sie nun zurück zum Neural Network/Data Manager-Fenster, markieren Sie die AND-Net_outputs und klicken Sie auf Export. Klicken Sie auf **Select All**. Es erscheint folgendes:

Evport from Network/Data Manager	
C-Let Verille	
Select variables	
D	
t	
ANDNet	
ANDNet outputs	
ANDNet errors	
Colored and a survey of the last Theory (Free add the survivables	
select one or more variables. Then [Export] the variables	
to the MATERD workspace of [Save] them to a disk file	
Select All Select None 🗞 Export 🛃 Save	e 🙆 Close

Danach drücken Sie Export.

Geben Sie nun in der Kommandozeile des MATLAB-Editors

>> who

ein und Sie erhalten die Anzeige:

Your variables are:

ANDNet ANDNet_errors ANDNet_outputs p t

Tippen Sie nun der Reihe nach ANDNet_outputs, ANDNet_errors, t und p ein und Sie erhalten das finale Ergebnis:

```
>> ANDNet_outputs
ANDNet_outputs =
    0     0     0     1
>> ANDNet_errors
ANDNet_errors =
    0     0     0
```

Copyright © 2017, Manfred Hiebl. Alle Rechte vorbehalten.

>	> t					
t	=					
	0	0	0	1		
>	> p					
p	=					
	0	0	1	1		
	0	1	0	1		