



Einführungsübungen mit **JavaNNS**

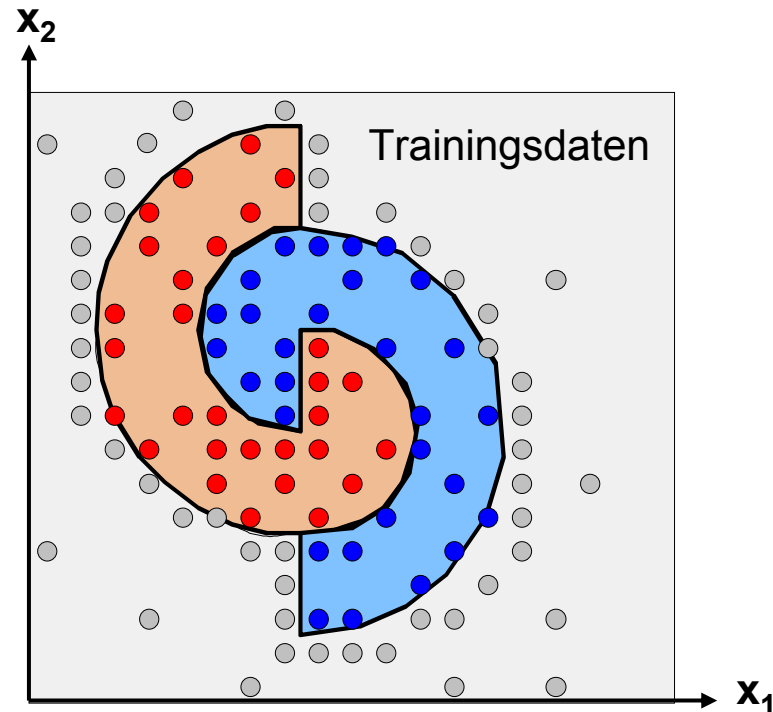
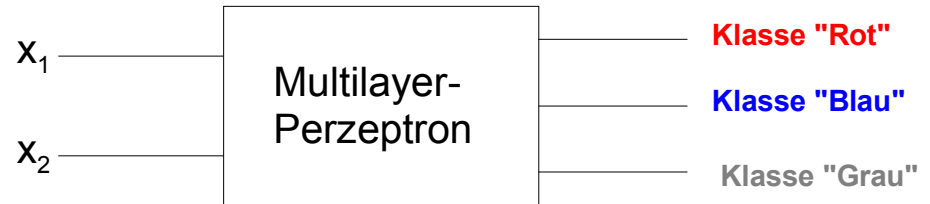
Prof. Dr.-Ing. Andreas Meisel

ÜBUNG 1:

Netztyp: Multilayer-Perzeptron mit
2 Input-Neuronen
20 Hidden1-Neuronen
10 Hidden2-Neuronen
3 Output-Neuronen

Trainingsalgorithmus:
Resilient-Propagation

Zu lösende Aufgabe:
Spiralproblem



Neuronales Netz erzeugen: Erzeugen des Input-Layer

[File] → [New]

[Tools] → [Create] → [Layers...]

Create layers

Size & Position

Width: 1 Height: 2

Top left position: 1 1 1

Unit detail

Unit type: Input

Activation function: Act_Identity

Output function: Out_Identity

Layer number: 1 Subnet number: 0

Create Close

Neuronales Netz erzeugen: Erzeugen der Hidden-Layer

Create layers

Size & Position

Width: 2 Height: 10

Top left position: 3 1 1

Unit detail

Unit type: Hidden

Activation function: Act_Logistic

Output function: Out_Identity

Layer number: 2 Subnet number: 0

Create Close

Create layers

Size & Position

Width: 1 Height: 10

Top left position: 5 1 1

Unit detail

Unit type: Hidden

Activation function: Act_Logistic

Output function: Out_Identity

Layer number: 3 Subnet number: 0

Create Close

The figure consists of two side-by-side screenshots from a software interface.

The left screenshot shows the 'Create layers' dialog box. It has two main sections: 'Size & Position' and 'Unit detail'. In the 'Size & Position' section, the 'Width' is set to 1 and the 'Height' is set to 3. In the 'Unit detail' section, the 'Unit type' is set to 'Output', the 'Activation function' is set to 'Act_Logistic', and the 'Output function' is set to 'Out_Identity'. The 'Layer number' is 4 and the 'Subnet number' is 0. There are 'Create' and 'Close' buttons at the bottom.

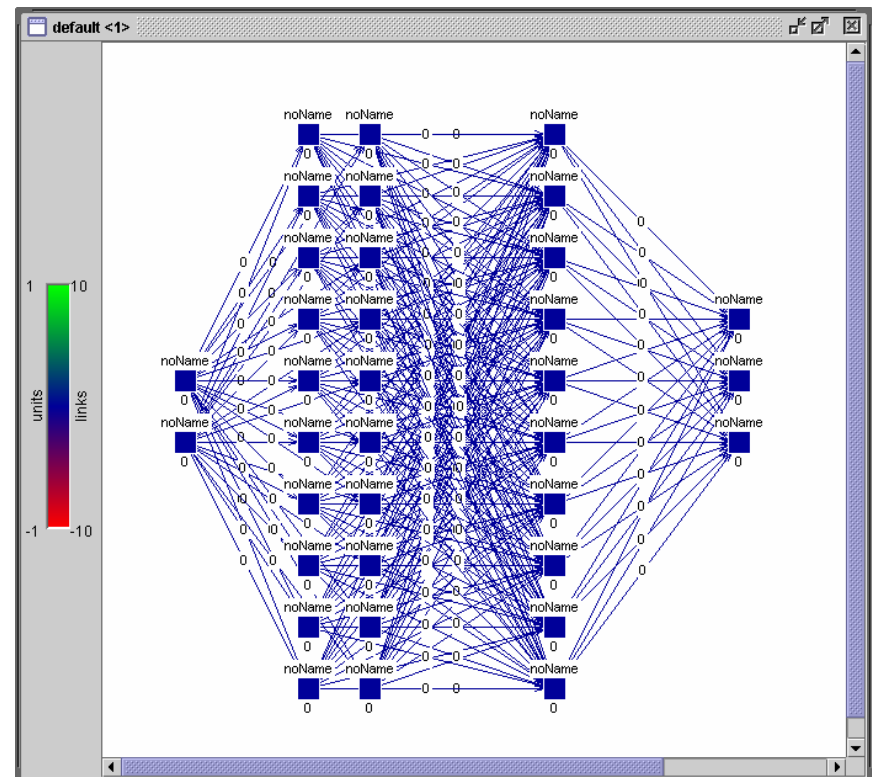
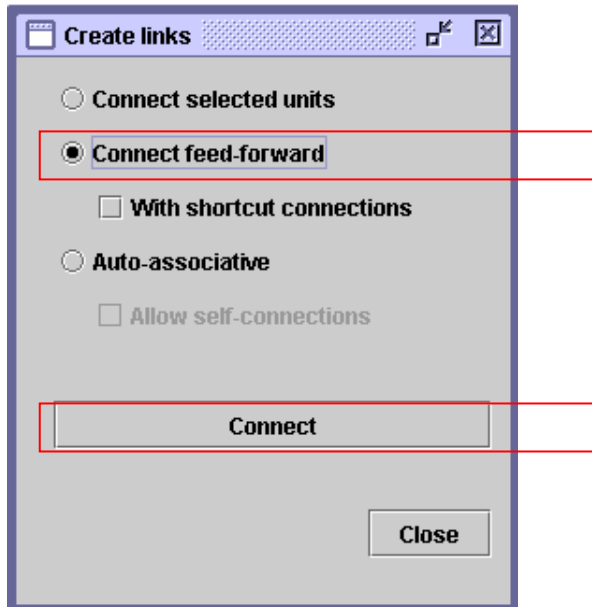
The right screenshot shows the 'default <1>' window, which displays a neural network diagram. The diagram consists of several layers of nodes, each represented by a blue square with the text 'noName' above it and a value '0' below it. A color bar on the left indicates the scale for 'units' (from -1 to 1) and 'links' (from -10 to 10).

5

Neuronales Netz erzeugen: Verbindungen anlegen

Alle Neuronen auswählen (werden dann gelb angezeigt).

[Tools] → [Create] → [Connections...]



Neuronales Netz erzeugen: Beschriften der Neuronen

Input-Neuronen anwählen

rechte Maustaste → Popup-Menü erscheint → [EditUnits...]

Edit units

Numbers: 1;2

Names: Input

Unit type: Input

Activation function: Act_Identity

Output function: Out_Identity

Activation: 0.0

Initial activation: 0.0

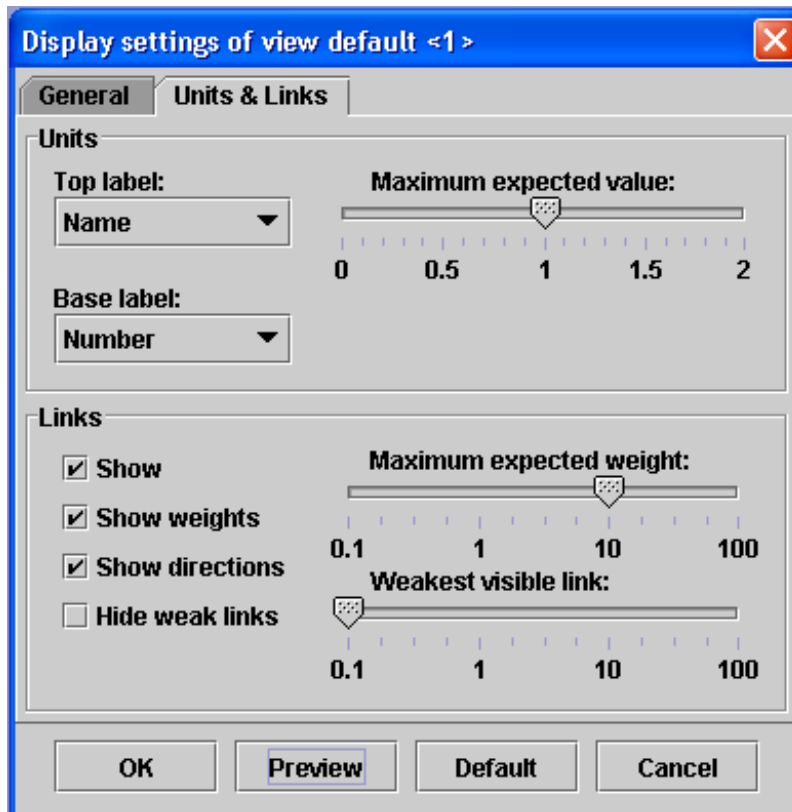
Bias: 0.0

Layer number: 1 Subnet number: 0

Apply Close

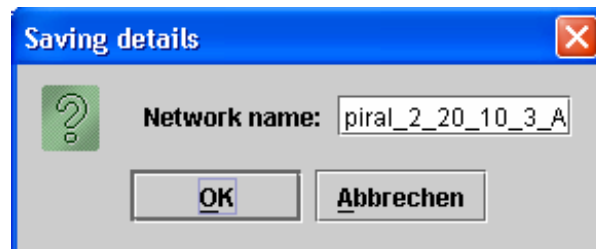
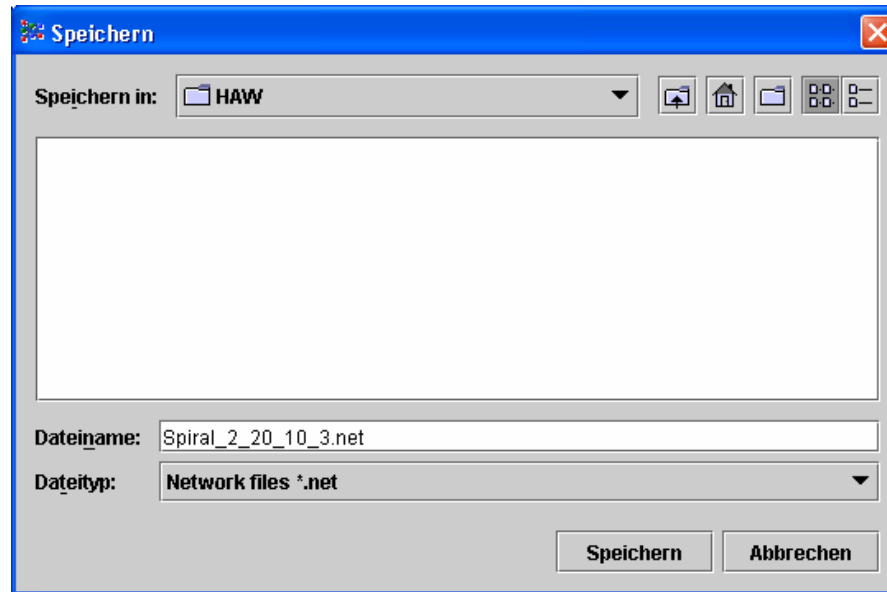
Neuronales Netz erzeugen: Anzeigooptionen einstellen

[View] → [Display Settings]



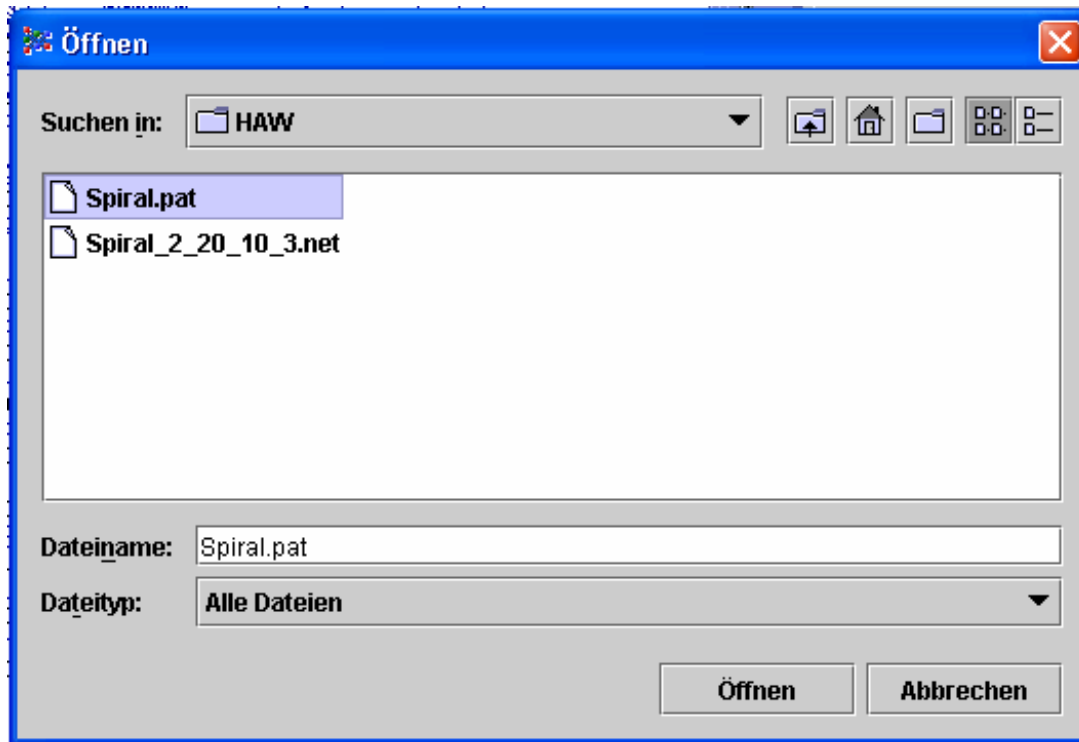
Neuronales Netz erzeugen: Netzwerk abspeichern

[File] → [Save]



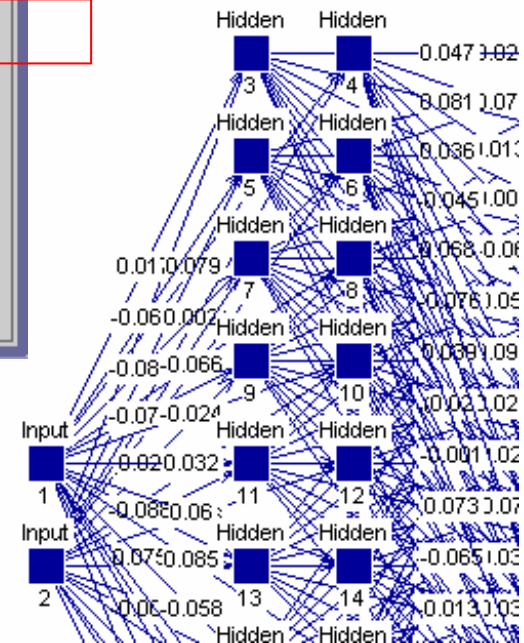
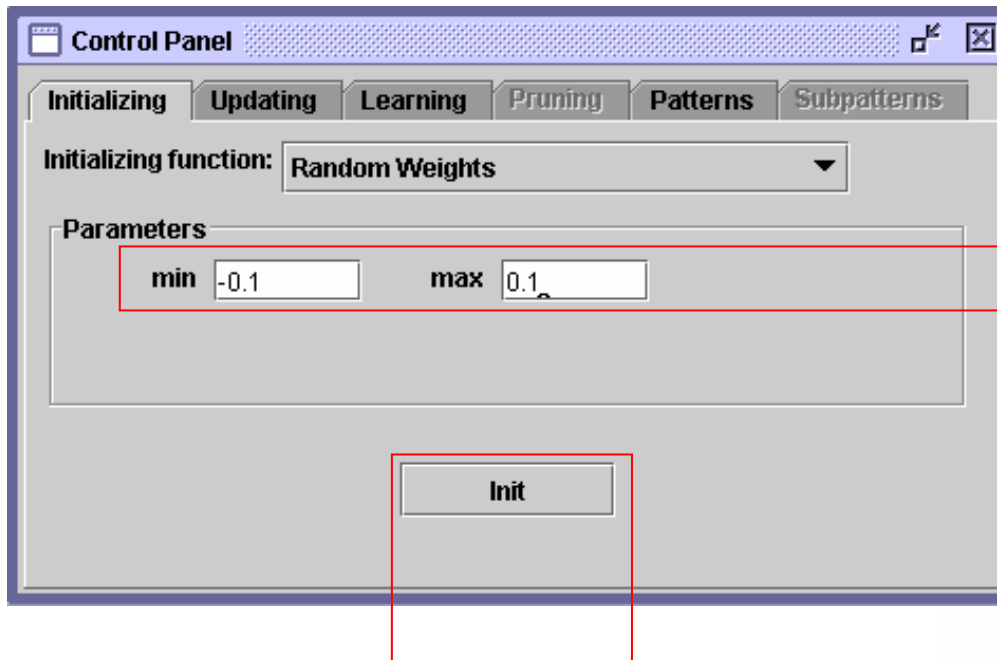
Training: Existierende Trainingsdaten laden

[File] → [Open]

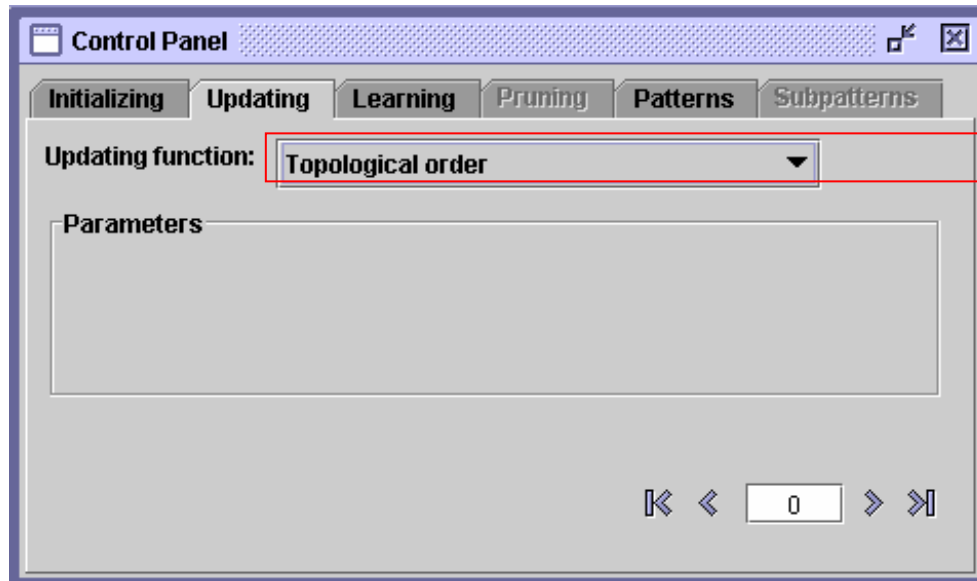


Training: Netzgewichte mit Zufallswerten initialisieren

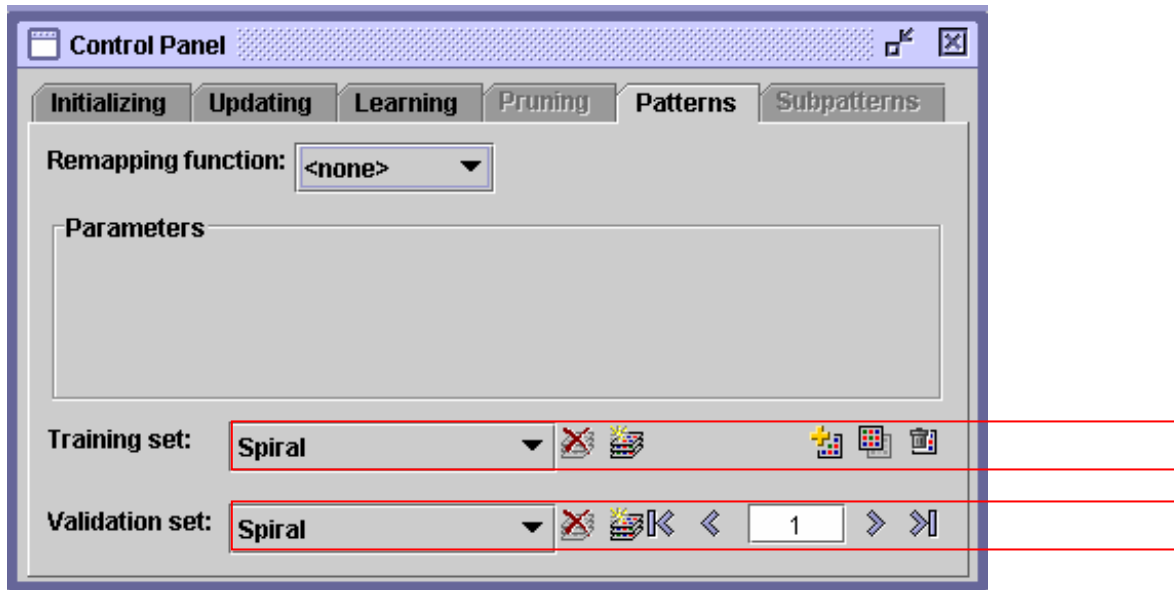
[Tools]→[Control]



Training: Trainingsreihenfolge einstellen



Training: Trainings- und Testdatensatz festlegen



Training: Lernalgorithmus und –parameter einstellen

Control Panel

Initializing Updating **Learning** Pruning Patterns Subpatterns

Learning function: **Resilient Propagation**

Parameters

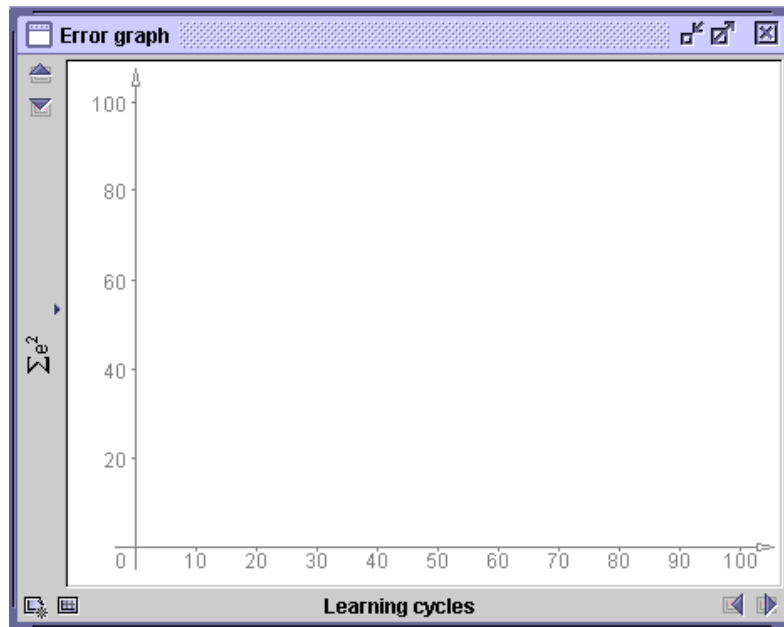
δ_0 0.1 δ_{max} 50.0 α 4.0

Cycles: 1000 Steps: 1 ☒ Shuffle

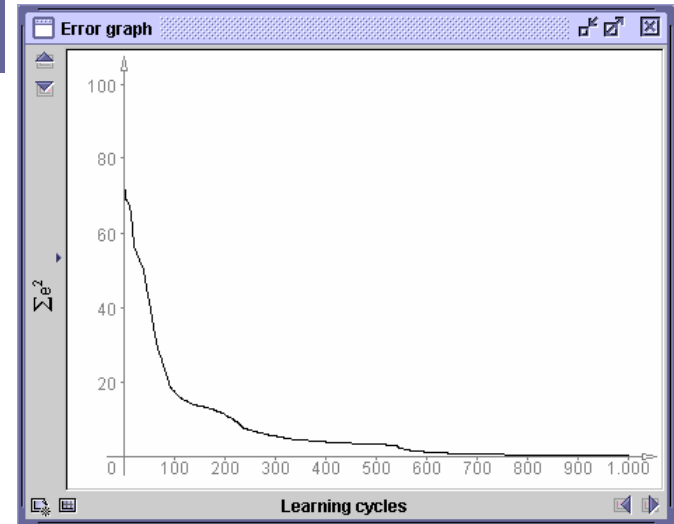
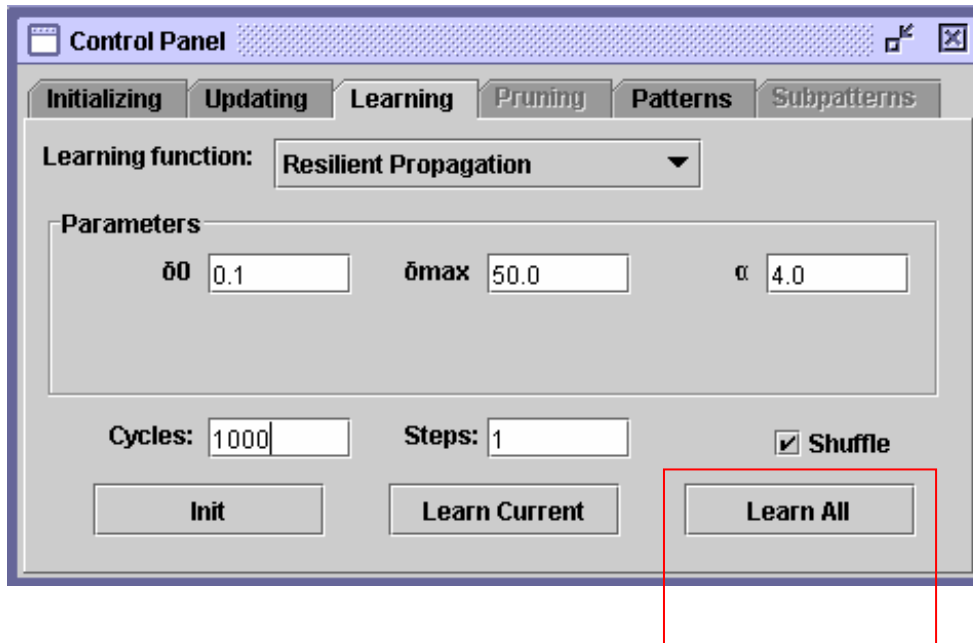
Init Learn Current Learn All

Training: Anzeigen des Fehlergraphen

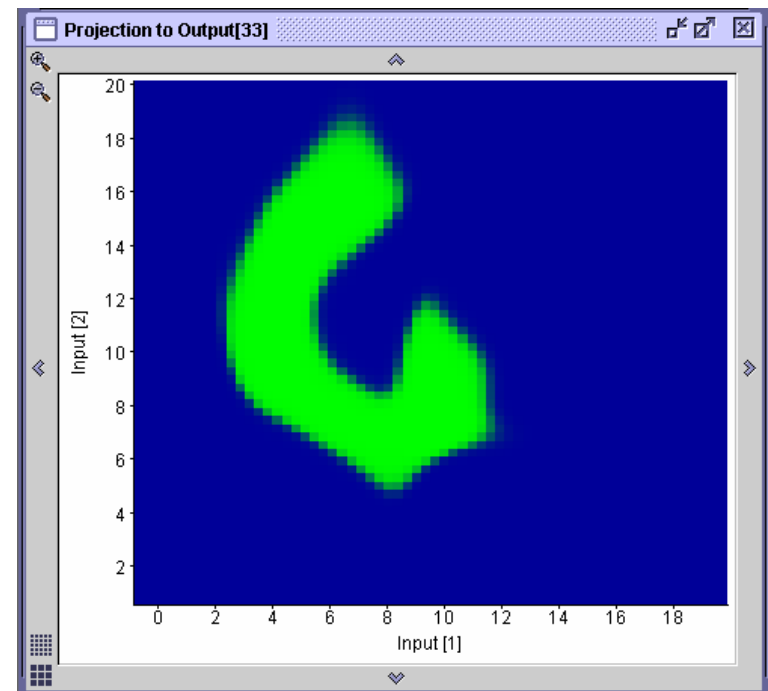
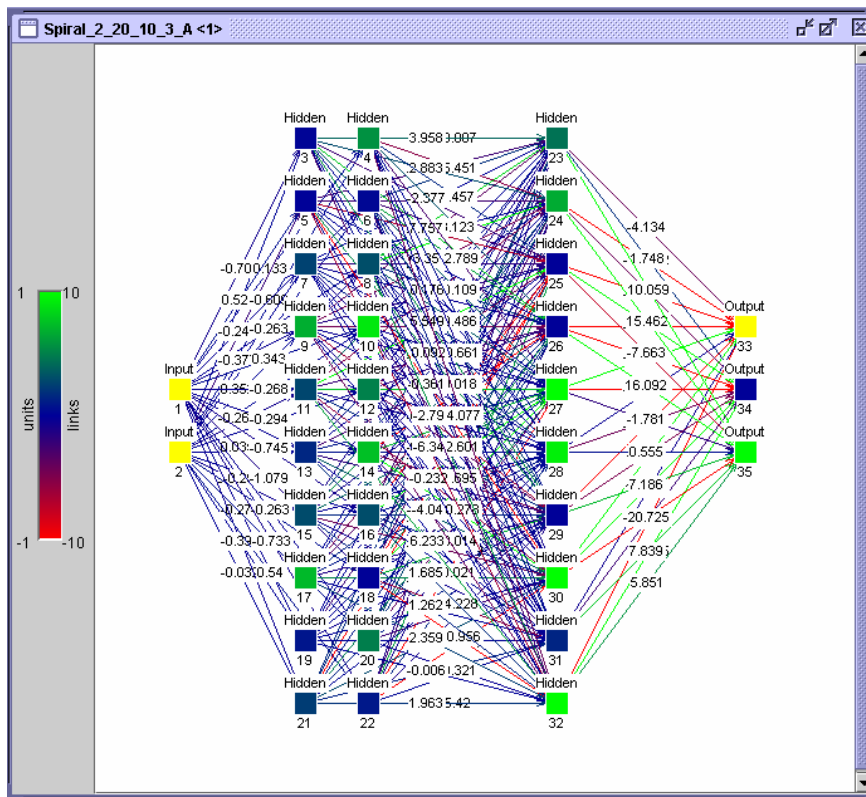
[View] → [Error Graph]



Training: Start des Trainings



Training: Projektion ansehen



ÜBUNG 2:

Netztyp: Multilayer-Perzeptron mit

160 Input-Neuronen (10 x 16)

50 Hidden1-Neuronen

32 Hidden2-Neuronen

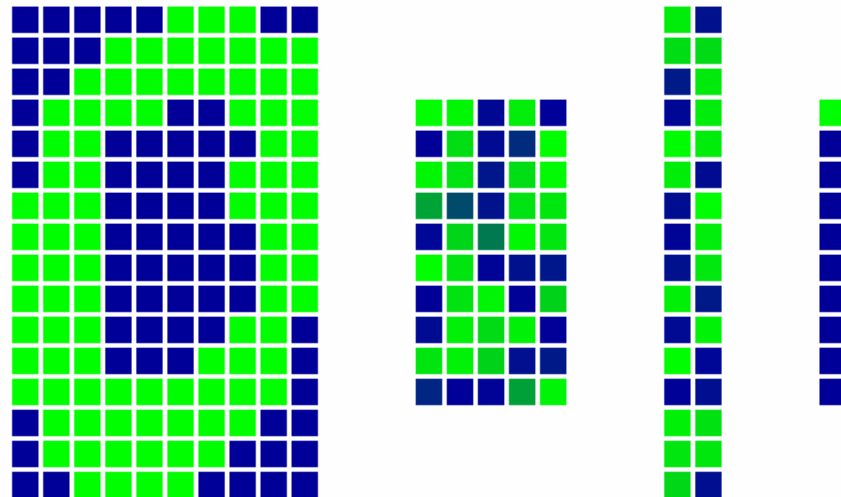
10 Output-Neuronen

Trainingsalgorithmus:

Backpropagation-Momentum / Resilient Propagation

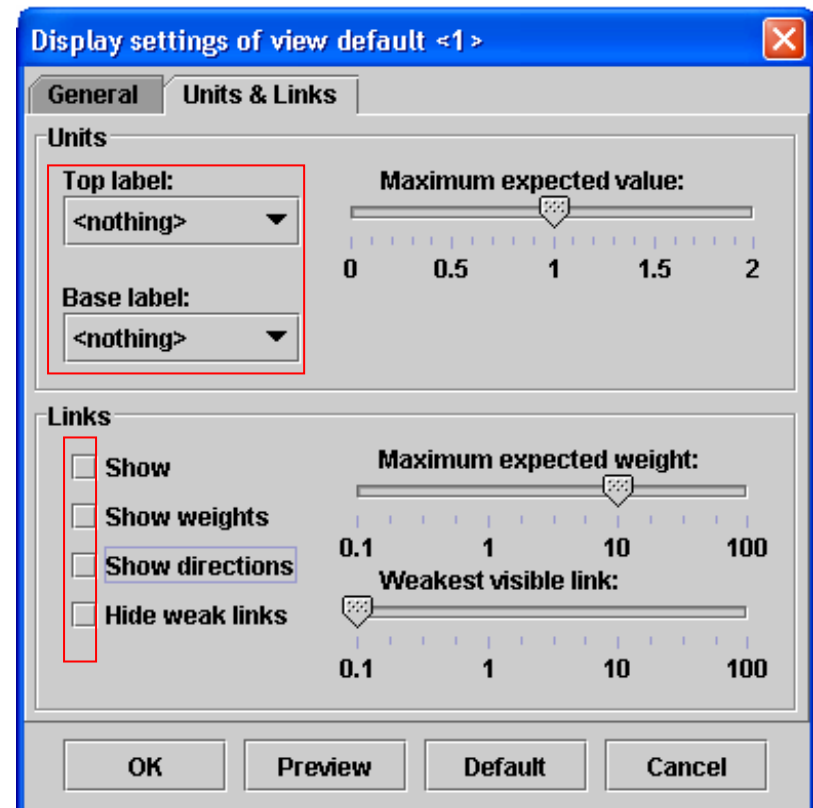
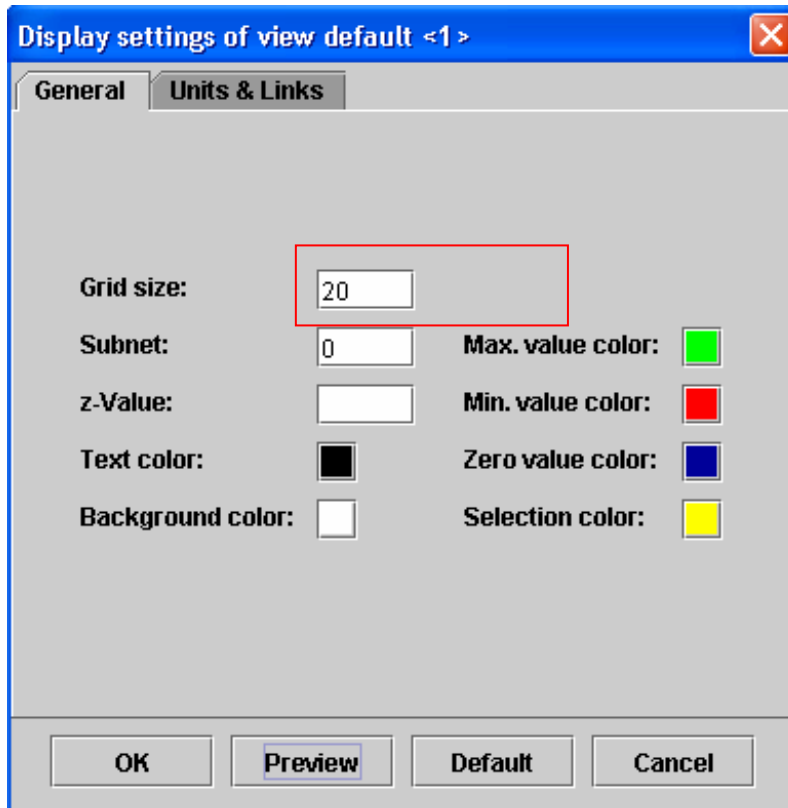
Zu lösende Aufgabe:

Zeichenklassifikator



Neuronales Netz anlegen: Anzeigeeinstellung festlegen

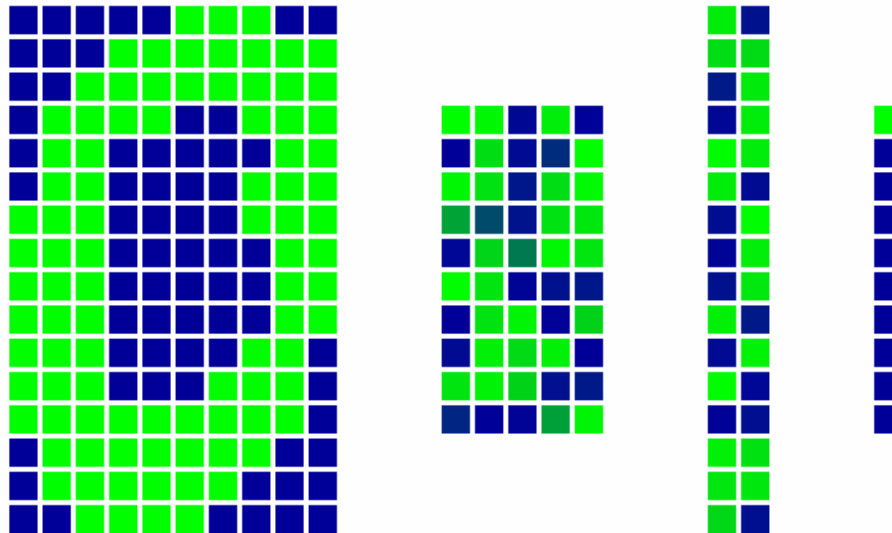
Anzeigeeinstellungen wie folgt einstellen:



Neuronales Netz anlegen: Neuronen und Verbindungsnetz anlegen

Netz anlegen und abspeichern

(Aktivierungs- und Ausgabefunktionen sowie Verbindungsnetz wie in Ü1)



Trainings- und Testdaten laden (`Train.pat` + `Test.pat`)

Neuronales Netz trainieren: Einstellungen

Control Panel

Initializing Updating Learning Pruning Patterns Subpatterns

Initializing function: Random Weights

Parameters

min -0.1 max 0.1

Init

Control Panel

Initializing Updating Learning Pruning Patterns Subpatterns

Updating function: Topological order

Parameters

35

Control Panel

Initializing Updating Learning Pruning Patterns Subpatterns

Learning function: Backprop-Momentum

Parameters

η 0.2 μ 0.5 c 0.1

d_{max} 0.1

Cycles: 500 Steps: 1 ☒ Shuffle

Init Learn Current Learn All

Control Panel

Initializing Updating Learning Pruning Patterns Subpatterns

Remapping function: <none>

Parameters

Training set: Train

Validation set: Test

1