
Neuronale Netze

Neuronale Netze bestehen aus künstlichen Neuronen, die durch synaptische Verbindungen gekoppelt sind. Mithilfe des Lern-Programms **Netz32** lassen sich solche Netze erstellen und trainieren. Eine graphische Benutzeroberfläche erlaubt es, (nahezu) beliebige Strukturen von Netzen zu erzeugen; dabei kann man zwischen verschiedenen Aktivierungsfunktionen mit frei wählbaren Werten für die Schwellenwerte auswählen. Obendrein können spezielle Netztypen (3-schichtiges Perzeptron und Hopfieldnetz) automatisch generiert werden. Die Anzahl der Neuronen ist auf 60 beschränkt.

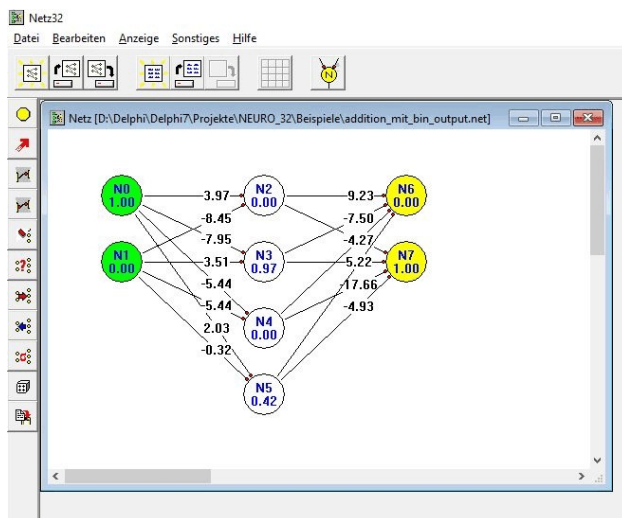


Abbildung 1

1. Neuronale Netze erstellen

Um ein neuronales Netz zu erstellen, öffnen wir zunächst das Netz-Fenster, indem wir die linke Schaltfläche in der Menüleiste; alternativ können wir auch im Datei-Menü auf <Neues Netz> klicken.

In diesem Fenster wollen wir nun ein Netz wie in Abb. 2 erstellen. Zunächst erzeugen wir die 3 Neuronen. Dazu betätigen wir die oberste Schaltfläche in der Werkzeugleiste; diese Schaltfläche rastet ein und bleibt so lange aktiviert, bis sie erneut angeklickt wird (oder eine andere Schaltfläche aus der Werkzeugleiste betätigt wird). Anschließend klicken wir mit der linken Maustaste diejenige Position an, an der das erste Neuron positioniert werden soll. Es erscheint ein Kreis mit der Bezeichnung N0; die Bezeichnung wird automatisch vom Programm erzeugt und kann nicht verändert werden. Die Zahl darunter gibt den Aktivierungs- bzw. Ausgangswert an. Beim Erzeugen eines Neurons ist dieser Wert standardmäßig 0.

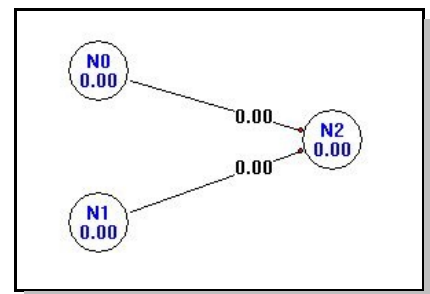




Abbildung 2

Neuronale Netze


Für die beiden nächsten Neuronen klicken wir im Netzfenster auf die entsprechenden Positionen.



Nun sorgen wir für die Verbindungen: Wir klicken zuerst auf die Schaltfläche ; diese bleibt wieder eingerastet und wir können nun solange Verbindungen herstellen, bis diese Schaltfläche wieder deaktiviert wird. Dann positionieren wir den Mauszeiger auf dem Neuron N0 und ziehen ihn mit gedrückter linker Maustaste auf das Neuron N2. Hier lassen wir die Maustaste los und die Verbindung wird mitsamt ihrem Gewicht angezeigt. Der Gewichtswert ist standardmäßig 0.

Der rote Punkt am Neuron N2 zeigt an, dass hier ein Axon/Ausgang von N0 an ein Dendrit/Eingang von N2 ankoppelt. Möchte man eine umgekehrte Kopplung vornehmen, also ein Axon/Ausgang von N2 an ein Dendrit/Eingang von N0 koppeln, müsste der Mauszeiger entsprechend von N2 nach N0 gezogen werden.

Die Verbindung von N1 nach N2 wird auf die gleiche Weise hergestellt. Anschließend deaktivieren wir die Schaltfläche .

2. Neuronale Netze speichern und öffnen

Unser erstes Netz ist fertig und soll jetzt mit der Schaltfläche  (oder mit <Datei>-<Netz speichern unter...>) gesichert werden; wir benutzen hier den Namen Bsp1a und legen die Datei in einem eigenen Ordner für dieses Kapitel ab. **Beachten Sie, dass beim Speichern nur die Struktur des Netzes und seiner Neuronen, nicht aber Signalwerte gesichert werden.**

Nach dem Sichern können wir unser Netz löschen; das geschieht mit der Schaltfläche . Mit der Schaltfläche  können wir es wieder öffnen.

3. Neuronale Netze editieren


Unser gerade geöffnetes Netz können wir ergänzen oder ändern: Mit <Bearbeiten> - <Letztes Neuron löschen> können wir das letzte Neuron, also das mit dem höchsten Index, löschen. Verbindungen zwischen zwei Neuronen kann man löschen, indem man genau so verfährt wie beim Herstellen der Verbindung: Erneutes Zeichnen löscht die Verbindung.

Auch die Schwellenwerte und Aktivierungsfunktionen können wir editieren. Dazu aktivieren wir in der Werkzeugleiste die Schaltfläche mit dem roten Zeiger und klicken mit der *rechten* Maustaste auf das zu editierende Neuron. Es erscheint ein Fenster mit umfangreichen Editiermöglichkeiten, vgl. Abb. 4.

Wie die Gewichte verändert werden können, wird in dem Kapitel über Lernprozesse ausführlich erläutert.

Neuronale Netze

4. Propagieren eines Netzes

Bei biologischen neuronalen Netzen werden die Signale unmittelbar von einem zum anderen Neuron weitergeleitet. Bei unserem künstlichen neuronalen Netz geschieht dies der Reihe nach (entsprechend dem Index der Neuronen); diese Signalweitergabe bezeichnet man als **Propagieren**. Eine solches Propagieren wird bei unserem Programm ausgeführt, indem man die Schaltfläche  betätigt.

Bei unserem Beispiel aus Abb. 2 wollen wir dies praktisch durchführen. Zunächst öffnen wir die Datei Bsp1.c aus dem Ordner Beispiell. Dass die beiden Neuronen N0 und N1 hier grün eingefärbt sind, kennzeichnet sie als **Eingabe-Neuronen**. Bei biologischen Neuronen entspricht das Sinneszellen, deren Ausgabewert durch einen äußeren Einfluss bestimmt wird. Hier bedeutet es, dass der Ausgabewert durch den Benutzer vorgegeben werden kann. Die Farbe Gelb kennzeichnet das Neuron N2 als **Ausgabe-Neuron**. Dies wird erst im Zusammenhang mit Lernprozessen eine Rolle spielen.

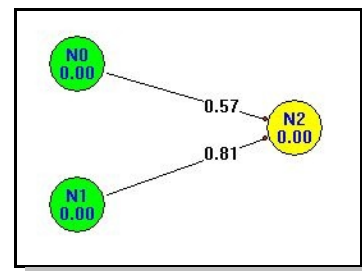



Abbildung 3

Die Gewichte 0,57 und 0,81 sind hier zufällig festgelegt worden; dies geschieht mit der Schaltfläche .

Welchen Wert erhält das Neuron N2 als Ausgangswert, wenn die Neuronen N0 und N1 die Werte 0,8 bzw. 0,2 ausgeben? Zunächst editieren wir die beiden Neuronen N0 und N1, um deren Ausgabewerte festzulegen. Dazu aktivieren wir die Schaltfläche mit dem roten Pfeil und klicken das Neuron N0 mit der rechten Maustaste an. Es erscheint das Fenster aus Abb. 4. Hier geben wir als Sollwert die Zahl 0,8 ein; die restlichen Angaben spielen dann keine Rolle. Dann schließen wir das Fenster mit der OK-Schaltfläche. Ähnlich verfahren wir mit dem Neuron N1, um ihm den Ausgangswert 0,2 zu geben. Die beiden Neuronen N0 und N1 im Netzfenster weisen jetzt die neuen Werte 0,8 und 0,2 auf.

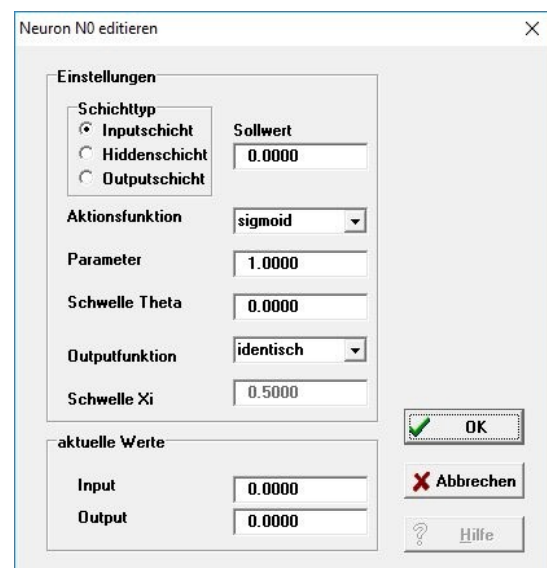


Abbildung 4

Nun können wir eine Propagation mit Hilfe der Schaltfläche  durchführen. Das Neuron N2 erhält den Wert 0,62. Dies können wir nachrechnen; dazu müssen wir die speziellen Eigen-

Neuronale Netze

schaften des Neurons N2 kennen. Im Editierfenster von N2 können wir sehen, dass die Schwelle ϑ den Wert 0 hat; die Aktivierungsfunktion ist “linear” mit der Steigung 1, vgl. Abb. 3 des Kapitels *Neuronen*. Das Feld für die Schwelle ξ ist hier deaktiviert; es spielt nur eine Rolle, wenn die Outputfunktion “binär” ist.

Die Rechnung lautet daher: $0,9 \cdot 0,57 + 0,2 \cdot 0,81 = 0,618 \approx 0,62$.

Ist der Wert von N0 nicht 0,8, sondern 2,0, dann erhält das Neuron N2 nach dem Propagieren den Wert 1,0. Hier macht sich bemerkbar, dass die Werte der Aktivierungsfunktion (betragsmäßig) maximal den Wert 1 haben 1, vgl. Abb. 3 des Kapitels *Neuronen*.

5. Eine erste Lernaufgabe

Wie die Ausgangs-Neuronen auf die Einstellungen bei den Eingabe-Neuronen reagieren, hängt entscheidend von den Gewichten des neuronalen Netzes ab. Im letzten Abschnitt hatten wir diese willkürlich vorgegeben und damit den Ausgabewert berechnen lassen. In der Praxis ist es aber häufig so, dass zunächst nur eine Reihe von Zielverhaltensweisen bekannt sind und daraus passende Gewichte ermittelt werden müssen. Diese Zielverhaltensweisen können wir mit Hilfe von **Lernaufgaben** beschreiben. Diese legen fest, welche Ausgabewerte gewissen Eingabewerten zugeordnet werden sollen.

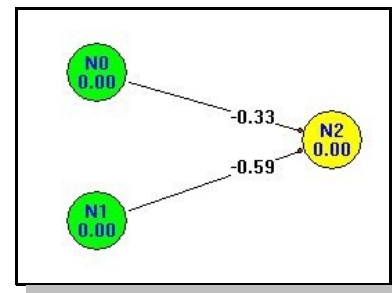


Abbildung 5

Unser neuronales Netz aus Abb. 5 findet man in der Datei Bsp1b. Es soll folgende Aufgaben erfüllen:

Aufgabe	Eingabe		Ausgabe
	N0	N1	N2
A1	0,7	0	0,2
A2	0,2	0,6	0,9

Man überzeugt sich leicht, dass es weder die erste noch die zweite Aufgabe erfüllt. Welche Werte müssen nun die Gewichte haben? Für dieses einfache Netz lässt sich dies leicht ausrechnen. Da die Aktivierungsfunktion von Neuron N2 wieder “linear” mit der Steigung 1 ist, muss für die beiden Gewichte $g_{0,2}$ und $g_{1,2}$ das folgende lineare Gleichungssystem gelten:

$$\begin{cases} 0,7 \cdot g_{0,2} + 0 = 0,2 \\ 0,2 \cdot g_{0,2} + 0,6 \cdot g_{1,2} = 0,9 \end{cases}$$

Neuronale Netze

Aus der ersten Zeile ergibt sich unmittelbar: $g_{0,2} = \frac{0,2}{0,7} \approx 0,2857$. Setzen wir diesen Wert in die zweite Gleichung ein, erhalten wir für $g_{1,2}$ den Wert 1,4048.

Die Situation wird deutlich komplizierter, wenn das Netz komplexer ist und die Anzahl der Aufgaben größer ist. Eine zusätzliche Schwierigkeit taucht auf, wenn die Aktivierungsfunktion nicht linear ist. Für solche Fälle stellt das Programm **Netz32** ein Verfahren zur Verfügung, welches die gesuchten Gewichte durch ein **Iterationsverfahren** näherungsweise bestimmt.


Wir wollen dieses Verfahren einmal auf den eben betrachteten Fall anwenden: Zunächst geben wir die Lernaufgaben ein. Dazu betätigen wir die Schaltfläche . Es erscheint das Fenster aus Abb. 6. Hier tragen wir in die A1-Zeile als erstes die Zahl 0,7, in das nächste Feld den Wert 0 und in das letzte Feld den Wert 0,2 ein. In die A2-Zeile tragen wir genauso die Werte für die zweite Aufgabe ein.



Abbildung 6


Da unser Netz alle Aufgaben beherrschen soll, müssen alle “trainiert” werden; wir betätigen deswegen die Schaltfläche “Alle”. Nun beginnt das Iterationsverfahren.

Bei jedem Iterationsschritt werden aus den vorliegenden Gewichten verbesserte Werte für die Gewichte gebildet. Mit diesen neuen Werten wird für beide Aufgaben das Netz propagiert und kontrolliert, wie stark die jeweiligen Ausgangswerte von N2 nun von dem jeweiligen Sollwert abweichen. Diese Abweichung wird in dem Fehler-Feld angezeigt.

Sie können mit der Stop-Taste die Iteration abbrechen, wenn der Fehlerwert klein genug ist, z. B. kleiner als $1E-6$ ist.

Neuronale Netze

Während der Iteration werden die Gewichte im Netzfenster laufend aktualisiert; so erhalten Sie am Ende der Iteration die gesuchten Werte. Sie stimmen mit unseren berechneten Werten überein.

Was geschieht, wenn die Iteration mit anderen Startwerten für die Gewichte beginnt? Mit der Schaltfläche  sorgen wir für zufällige Gewichtswerte und starten die Iteration neu. Nach kurzer Zeit werden wieder die bekannten Gewichtswerte angezeigt.

Es dürfte klar sein, dass nicht zu jedem Satz von Lernaufgaben passende Gewichte gefunden werden können; insbesondere können keine passenden Gewichte gefunden werden, wenn sich einzelne Aufgaben widersprechen.

Übrigens: Ersetzt man beim Neuron N2 die Aktivierungsfunktion “linear” durch “sigmoid” oder “tanh”, erhält man ganz andere Werte für die beiden Gewichte.