NNetSimu

Eine Simulation natürlicher neuronaler Netzwerke

Anwenderdokumentation

P. Kraus

Stand 10.04.2020

**Änderungen in dieser Version:**

1. **Ein Fehler beim Zurücksetzen des Modells wurde beseitigt. Dieser Fehler hat dazu geführt, dass in manchen Situationen nach dem Ändern der Pulsfrequenz eines Inputneurons das Modell nicht synchronisiert neu gestartet wurde.**
2. **Eine neue Funktion „Center model“ ist verfügbar, siehe 4.1.3. Sie ist quasi ein Abfallprodukt der Analyse-Funktionen.**
3. **Neue Funktionen zur Umwandlung von Neuronen in Input-Neuronen und umgekehrt wurden implementiert. Sie sind über das Kontextmenü anwählbar, siehe 4.5.3. Die Funktionen wurden im Hinblick auf das geplante Kopieren und Zusammenfügen größerer Einheiten geschaffen. Die Input-Neuronen der nachgeordneten Einheit werden dann zu normalen Neuronen, die ihren Input von der vorgeschalteten Einheit erhalten. Ohne die Funktion „Input-Neuron in normales Neuron umwandeln“ müsste man das Input-Neuron löschen, an dieser Stelle ein neues Neuron erzeugen und mit dem Axon des gelöschten Input-Neurons verbinden.**
4. **Eine weitere Funktion zur Analyse des Modells, die „entartete“ Dendriten findet, wurde implementiert, siehe 4.1.2**
5. **Der Editor hat verschiedene Methoden zum Erstellen und Bearbeiten einer „permanenten Auswahl“ bekommen. Die Möglichkeit zur permanenten Auswahl eine Menge von Objekten (Neuronen, Dendriten-Teilstücke) ist Voraussetzung für die Manipulation einer solchen Auswahl, z.B. Kopieren.**

# Einleitung

NNetSimu ist eine Simulation natürlicher neuronaler Netzwerke, wie zum Beispiel des mensch­lichen Gehirns. Das Programm erlaubt es, ein Modell zu erstellen, das aus Neuronen und Ver­bindungen zwischen diesen Neuronen besteht, und das dynamische Verhalten des Modells zu beobachten. NNetSimu muss notwendigerweise die Komplexität biologischer Systeme stark vereinfachen (siehe Kapitel 4.6).

Das vorliegende Dokument bezieht sich ausschließlich auf die in NNetSimu implementierten Elemente. Aussagen der Art „es gibt drei Arten von …“ sind deshalb nicht als Behauptungen über biologische Systeme zu verstehen (dort ist die Vielfalt meist höher), sondern lediglich als Beschreibungen der aktuell im Programm NNetSimu implementierten Funktionalität.

# Die Elemente des Modells

## Statisches und dynamisches Modell

NNetSimu zeigt ein Modell eines neuronalen Netzwerks an, das statische und dynamische Aspek­te enthält. Das statische Modell beschreibt eine Anzahl verschiedener Neuronen mit ihren jewei­ligen Positionen auf der Arbeitsfläche sowie Verbindungen zwischen den Neuronen.

Das **statische Modell** kann durch den Benutzer verändert werden, z.B. durch das Hinzufügen wei­terer Neuronen oder das Verändern ihrer Position, siehe Kapitel 4.3.

Das **dynamische Modell** beschreibt die Verteilung und Veränderung der elektrischen Spannung in den Elementen des statischen Modells über die Zeit. Input-Neuronen geben in einem be­stimmten Takt elektrische Impulse ab, die durch die Verbindungen zu Neuronen laufen. Die ein­gehenden Impulse können ein Neuron veranlassen, selbst Impulse abzugeben, usw. Diese dyna­mischen Vorgänge laufen in NNetSimu automatisch ab, sie können vom Benutzer aber durch be­stimmte Parameter beeinflusst werden.

Es gibt drei Arten von Neuronen, die sich optisch durch ihre äußere Form (Rechteck, Kreis mit oder ohne Ausbuchtung) unterscheiden. In den folgenden Abbildungen sind neben der Form der verschiede­nen Arten von Neuronen noch weitere Elemente wie Farben, und Beschriftungen zu erkennen, die für das dynamischen Modell Bedeutung haben.

## Input-Neuronen

Input-Neuronen werden durch Rechtecke dargestellt, die in eine Richtung geöffnet sind. Sie haben keine eingehenden Dendriten und ei­nen Ausgang, das Axon.

Input-Neuronen stellen quasi den Eingang in das neuronale Netzwerk dar. Sie werden durch externe Ereignisse getriggert, in der Regel Sinnesreize, wie einfallendes Licht beim Auge oder akustische Signale bei den Neuronen im Innenohr.

Das dynamische Verhalten eines Inputneurons wird durch die individuell einstellbare (siehe 4.5.3) Pulsfrequenz und die für alle Neuronen identische Auslösespannung, „threshold“ (siehe 3.2) bestimmt. Das elektrische Potential steigt linear an, bis die Auslösespannung erreicht ist. Die Geschwindigkeit des Spannungsanstiegs ist so bemessen, dass genau nach einer Pulsdauer die Auslösespannung erreicht wird. Dann wird im Axon ein Impuls ausgelöst, dessen Amplitude durch den Parameter „peak voltage“ bestimmt ist. Das Potential im Inputneuron wird auf 0 Volt zurückgesetzt. Nun beginnt der Ablauf von vorne, das Potential steigt linear an, usw.

Der aktuelle Zustand eines Input-Neurons wird durch die Farbe des Inneren visualisiert, die sich mit ansteigendem elektrischen Potential von Schwarz zu einem helleren Farbton (z.Z. rot) verän­dert.

## „Normale“ Neuronen

„Normale“ Neuronen werden durch einen Kreis dargestellt, der an einer Stelle eine Ausbuchtung hat. Sie haben eine oder mehrere eingehende Ver­bindungen und einen Ausgang, das Axon. Das Axon entspringt der Ausbuch­tung, auch Axon-Hügel genannt. Die eingehenden Ver­bindungen sind Dendri­ten, die über diverse Verzweigungen von anderen Neuronen ausgehen.

Das dynamische Verhalten eines „normalen“ Neurons wird durch die Poten­tialverläufe in den eingehenden Verbindungen und die für alle Neuronen identische Auslösespannung („threshold“) bestimmt. Bei jedem Takt (zurzeit 100 Mikrosekun­den) wird die anstehende Spannung in allen eingehenden Dendriten abgefragt und aufsummiert. Falls die Auslösespannung erreicht ist, wird im Axon ein Impuls mit der Amplitude „peak voltage“ ausgelöst. Im Gegensatz zum Inputneuron ist die Zeitdauer bis zum Auslösen nicht konstant, sondern hängt von der Anzahl der eingehenden Dendriten sowie vom Verlauf der eingehenden Impulse ab (Frequenz, Phase).

Wie bei den Input-Neuronen wird das aktuelle elektrische Potential von Neuronen farblich visua­lisiert. Zusätzlich wird das Potential als numerische Anzeige in Prozent der Auslösespannung dar­gestellt.

## Output-Neuronen

****Output-Neuronen werden durch einen Kreis ohne Ausbuchtung darge­stellt. Sie haben eine oder mehrere eingehende aber keine ausgehen­den Verbindungen. Auf der Ausgangsseite stehen Output-Neuronen nicht mit anderen Neuronen in Verbindung, sondern sie steuern Akto­ren an, z.B. Muskelzellen.

Wird an ein Output-Neuron ein Axon angehängt, so verwandelt es sich automatisch in ein normales Neuron. Umgekehrt wird ein normales Neuron automatisch zu einem Output-Neuron, wenn das Axon entfernt wird.

Das dynamische Verhalten von Output-Neuronen entspricht im Prinzip dem der normalen Neuronen. Falls die Summe der Spannungspotentiale in den eingehenden Verbindungen die Auslösespannung überschreitet, feuert das Output-Neuron. Da es jedoch kein Axon hat, wird der Impuls nicht an ande­re Neuronen weitergegeben. Stattdessen wird ein Aktor angesteuert, was aber in NNetSimu zur­zeit nicht dargestellt wird.

Wie bei normalen Neuronen wird das aktuelle elektrische Potential von Output-Neuronen farblich und durch eine numerische Anzeige visualisiert.

## Verbindungen

Die Verbindungen zwischen Neuronen heißen Axone oder Dendriten. Axone sind die Ausgänge von Input-Neuronen und normalen Neuronen. Wenn sich diese weiter verzweigen, heißen sie Dendriten. Im Rahmen dieses Dokuments wird meistens der generische Begriff „Verbindungen“ benutzt, wenn nicht speziell auf die Eigenschaften z.B. von Axonen hingewiesen werden soll.

Die Verbindungen zwischen den Neuronen sind grundsätzlich gerichtet, d.h. sie haben definierte Start- und Endpunkte. Zur Visualisierung können wahlweise Richtungspfeile eingeblendet werden, wie in der Abbildung zu Kapitel 2.7. Die Richtungspfeile können über die Menüleiste („View“ – „Arrows on“ bzw. „Arrows off“), sowie über das Kontextmenü (siehe Kapitel 4.5.3) ein- und ausgeschaltet werden.

## Verzweigungen

Verzweigungen im statischen Modell von NNetSimu können z.B. durch Neuronen erfolgen, die mehrere Eingänge haben. Es sind aber auch Verzweigungen von Dendriten möglich. Die ne­benstehende Abbildung zeigt verschiedene Arten von Verzweigungen. Mehrere eingehende Verbin­dungen können zu einer ausgehenden Verbindung zusammenlaufen (links oben) oder eine eingehen­de Verbindung kann sich in mehrere ausgehenden Verbindungen aufspalten (rechts).

Dynamisches Verhalten von Verzweigungen:

Bei zusammenlaufenden Verzweigungen werden die eingehenden Signale addiert und das Sum­mensignal auf den ausgehenden Ast gesendet.

Bei aufspaltenden Verzweigungen wird das eingehende Signal unverändert an alle ausgehenden Äste weitergegeben.

## Schleifen

Ein NNetSimu-Modell kann auch Schleifen enthalten. Elektrische Impulse laufen dann über mehrere Neuronen hinweg sozusagen im Kreis.

Die nebenstehende Abbildung zeigt einen einfachen Fall einer solchen Schleife. Der Output des unteren Neurons läuft über die rechts dargestell­te Verzweigung sowohl in das Output-Neuron oben rechts als auch in das Neuron in der Mitte und erreicht von dort aus wieder das untere Neuron.

# Aspekte des dynamischen Modells

## Impulsform

In der Beschreibung der verschiedenen Arten von Neuronen ist jeweils von einer „Auslösung“ die Re­de. Damit ist gemeint, dass das Neuron einen Impuls auf seinem Axon erzeugt, dass dann durch den Dendritenbaum zu weiteren Neuronen wandert.

In biologischen Systemen hat der Impuls laut Wikipedia die in der nebenstehenden Abbildung dargestellte Form.

Zurzeit verwendet NNetSimu eine sehr grobe Approximation dieses Signals, das über eine einfach zu berechnende Parabel zweiten Grades dargestellt wird. Diese Funktion ist über zwei Parameter determiniert:

1. Die Amplitude, im Programm „peak voltage“ – die Differenz zwischen der maximalen Span­nung und dem Ruhepotential.
2. Die Pulsbreite „pulse width“ – die Zeitdiffe­renz zwischen dem Verlassen des Ruhepotenti­als und der Rückkehr zum Ruhepotential am Ende des Pulses.

An das Ende des Pulses schließt sich eine Phase (im Programm „refractory period“ genannt) an, in der keine neue Auslösung möglich ist, selbst wenn die eingehenden Signale die Schwellenspannung übersteigen.

Das Programm abstrahiert von einigen Eigenschaften biologischer Systeme:

1. Dass das Ruhepotential in biologischen Systemen nicht 0, sondern einen negativen Span­nungswert hat, wird in der Simulation nicht berücksichtigt
2. Die Hyperpolarisation während der Ruhephase wird nicht berücksichtigt.
3. Bei manchen biologischen Neuronen ist eine Auslösung während der Ruhephase nicht völlig ausgeschlossen, sondern die Schwellenspannung für eine erneute Auslösung ist lediglich stark erhöht. Das heißt, sehr starke Impulse können sich auch während der Ruhephase durchsetzen.

Anmerkung: In der obigen Abbildung ist die Amplitude des Pulses deutlich größer als die Schwellenspannung. Wenn dieser Puls ohne Abschwächung zu nachfolgenden Neuronen läuft, müsste er auf jeden Fall auch dort sofort eine Auslösung verursachen. Das Gesamtsystem würde demnach immer feuern, sobald auch nur ein einziger auslösender Impuls auftritt. Um eine sinnvolle Simulation zu gewährleisten, sollte die Amplitude des Pulses im Gegensatz zu der Abbildung niedriger als die Schwellenspannung gewählt werden. Zurzeit ist der Widerspruch zwischen der Abbildung in Wikipedia und den Erfordernissen der Simulation noch ungeklärt.

## Einstellbare Parameter

Die globalen Parameter des dynamischen Modells können über den Dialog „Global parameters“ verändert werden. Dies sind

* Peak voltage: Die Amplitude der Signale in mV
* Threshold: Die Schwellenspannung für die Auslösung eines Impulses in mV
* Pulse width: Die Pulsbreite in Millisekunden
* Refractory period: Die Dauer der Ruhephase nach einem Impuls in Millisekunden
* Time resolution: Die zeitliche Auflösung mit der das Modell berechnet wird
* Pulse speed: Die Ausbreitungsgeschwindigkeit der Impulse in den Verbindungen in

Änderungen in den Feldern des Parameter-Dialogs werden wirksam, wenn der „Apply“-Button ak­tiviert wird.

Wie die Bezeichnung **globale** Parameter bereits andeutet, gelten diese Werte für alle Elemente des Modells. Individuelle Einstellungen für einzelne Neuronen oder Verbindungen sind zurzeit nicht möglich. Im Gegensatz dazu kann die Pulsrate jedes Input-Neurons separat eingestellt wer­den, siehe 2.2.

Falls der Parameter-Dialog stört, kann er mit dem Kreuz rechts oben geschlossen werden. Bei Bedarf kann der Dialog mit dem Menüpunkt „View“ – „Windows“ - „Show parameter window“ wieder geöffnet werden.

## Visuelles und akustisches Feedback

Die Fortpflanzung der Signale in den Dendriten wird durch die farbliche Veränderung der Abschnitte des Dendriten visualisiert. Im unerregten Zustand ist das Innere eines Dendriten schwarz. Je höher das Spannungspotential ist, umso stärker ist der Rotanteil des Abschnitts. Sind diese Abschnitte relativ kurz, dann wirkt die Bewegung flüssig und gleichmäßig. Sind sie relativ lang, dann sind Abstufungen zu erkennen und die Bewegung wirkt eckig und ungleichförmig. Die Länge der Abschnitte hängt ab von der eingestellten Ausbreitungsgeschwindigkeit (Pulse speed) und der zeitlichen Auflösung (Time resolution) ab, siehe auch Kap. 3.2., sowie Kap. 5 Punkt 3.

Das Auslösen eines Neurons kann sowohl visuell als auch akustisch signalisiert werden.

1. Wie auch bei den Dendriten wird das Innere von Neuronen mit zunehmendem Spannungspotential mit immer größerem Rotanteil dargestellt.
2. Außerdem wird das Verhältnis das aktuellen Spannungspotentials zur Auslösespannung (Threshold) als Prozentwert eingeblendet.
3. Bei der Auslösung wird statt dem Prozentwert 100% das Wort „TRIGGER“ ausgegeben.
4. Zusätzlich kann bei jedem Neuron individuell ein akustisches Signal mit dem Auslöseereignis verknüpft werden. Dazu wird mit der rechten Maustaste auf dem Neuron im Kontextmenü der Punkt „Trigger Sound“ angewählt. Es öffnet sich ein kleines Dialogfenster, in dem Frequenz und Dauer des akustischen Signals eingegeben werden können.

# Benutzerinteraktion

Die grundsätzlichen Möglichkeiten der Benutzerinteraktion sind:

* Die Menüleiste oben
* Die Statuszeile unten
* Aktionen mit der Maus im Hauptbereich
* Tastatureingaben

## Menüleiste

Die Menüleiste bietet auf der obersten Ebene die Bereiche

* File Abspeichern und Laden von Modellen, Beenden des Programms
* Action verschiedene Funktionen, z.B. „Analyze“, „Center model“, siehe unten
* View Funktionen, die mit der Darstellung des Programms zu tun haben
* Options weitere Optionen, z.B. um das akustische Feedback ein- und auszuschalten
* Help Informationen zum Programm

Unter dem Menü „View“ ist die Option „Window refresh rate“ enthalten. Sie öffnet einen Dialog, mit dem die Bildwiederholrate einge­stellt werden kann. Damit ist nicht die Hardware-Bildwiederholrate des Monitors gemeint, son­dern die Frequenz, mit der von der Software die Darstellung des Netzes erneuert wird. Im Nor­malfall ist es nicht nötig, den voreingestellten Wert zu verändern.

Im Menü „Options“ können folgende Eigenschaften des Programms ein- und ausgeschaltet werden:

* Arrows: Die Anzeige der Richtungspfeile an Verbindungen
* Sound: Akustische Signale beim Andocken oder Löschen von Objekten im Editor, sowie TriggerSounds
* AutoOpen: Wenn diese Option aktiv ist, wird das zuletzt bearbeitete Modell sowie die Zoomstufe und die Position auf der Arbeitsfläche abgespeichert und beim nächsten Programmstart automatisch wiederhergestellt. Wenn die Option deaktiviert ist, startet das Programm mit einem Minimalmodell bestehend aus zwei Neuronen.

### „Action“ - „Analyze“ – „Find loops“

Diese Funktion untersucht das Modell und sucht nach Schleifen (siehe 2.7). Kleine Schleifen, an denen nur wenige Neuronen beteiligt sind, können während der Erstellung eines Modells versehentlich erzeugt werden und sind oft mit bloßem Auge kaum zu entdecken. Sie können aber das Verhalten des Modells auf unerwünschte Weise beeinflussen.

Bei dieser Erkennungsfunktion gibt es neben dem Rechenaufwand bei größeren Modellen vor allem das Problem der Interaktion mit dem Benutzer.

Wie viele Elemente darf eine Schleife, die von der Analyse erfasst wird, maximal umfassen? Interessant sind vor allem sehr kleine Schleifen, die nur aus wenige Neuronen bestehen, da diese Schleifen höchstwahrscheinlich unbeabsichtigt sind. Bewusst gesetzte Rückkopplungsschleifen, die über mehrere Faserbündel verlaufen, bestehen aus einer größeren Anzahl von Neuronen. Aber wo soll die Grenze gesetzt werden? Eine Einstellung durch den Benutzer wäre möglich, aber umständlich. Die Funktion wurde nun so implementiert, dass zunächst kleine Schleifen gesucht werden. Wenn nichts gefunden wird, beginnt die Suche erneut mit einer etwas größeren Obergrenze und so weiter, bis schließlich nach Schleifen gesucht wird, die alle Neuronen des Modells beinhalten. Die ersten Durchläufe gehen sehr schnell. Mit zunehmender Obergrenze benötigt die Suche immer mehr Rechenzeit. Im Meldungsbereich der Statuszeile (siehe Kap. 4.1.2) wird angezeigt, welche Schleifengröße aktuell gesucht wird. Der Benutzer kann die Suche jederzeit mit der ESC-Taste abbrechen.

Wenn eine Schleife gefunden wurde, ergibt sich die Frage, wie diese dem Benutzer angezeigt werden soll. In Regionen mit sehr vielen Neuronen und Dendriten, die dicht beieinander liegen, sich teilweise sogar überdecken, ist es schwer eine kleine Schleife zu entdecken. Außerdem kann es sein, dass nicht das ganze Modell im sichtbaren Teil des Fensters angezeigt wird, und die Schleife gerade außerhalb des Fensters liegt. Für die Anzeige einer gefundenen Schleife wird deshalb folgendes Verfahren angewandt:

* Zunächst wird die Aktion „Modell zentrieren“ (siehe Kap. 4.1.2) ausgeführt, so dass das gesamte Modell sichtbar ist und etwa in der Mitte des Fensters liegt,
* Gleichzeitig werden die nicht an der Schleife beteiligten Neuronen und Dendriten farblich gedimmt, sodass sie weniger auffallen. Die an der Schleife beteiligten Elemente werden kräftig rot hervorgehoben.
* Dann wird wieder soweit hineingezoomt, dass die gefundene Schleife noch ganz sichtbar ist und möglichst groß in der Mitte des Fensters dargestellt wird.

Der Benutzer kann nun die Schleife auflösen, z.B. indem er Neuronen löscht oder Verbindungen auflöst.

Um wieder in den normalen Modus zurückzukehren, in dem das Modell berechnet und dynamisch visualisiert wird, benutzt man die Funktion „Run“ in der Statuszeile.

Danach sollte nochmals mit der Funktion "Analyze“ getestet werden, ob noch weitere Schleifen im Modell enthalten sind.

### „Action“ - „Analyze“ – „Find anomalies“

Neben Schleifen gibt es noch weitere Anomalien, die sich unbeabsichtigt in ein Modell einschleichen können. Dazu gehören „entartete“ Dendriten, bei denen z.B. zwei Abschnitte in eine Knickstelle einmünden, aber keine herausführt, oder umgekehrt. Derartige Unsauberkeiten haben zwar keinen Einfluss auf das Verhalten des Modells, da in ihnen niemals Impulse fließen können, sie verursachen aber unnötigen Rechenaufwand und machen das Modell unübersichtlich. Die Funktion sucht nach solchen entarteten Dendriten-Teilstücken und hebt sie genauso wie „Find loops“ optisch hervor. Nach der Bereinigung eines solchen Fehlers sollte die Funktion erneut aufgerufen werden, solange bis keine weitere Anomalie mehr gefunden wurde.

### „Action“ - „Center model“

Wenn das Modell komplett außerhalb des sichtbaren Fensters liegt, was durch versehentliches Verschieben oder durch einen Fehler beim Einlesen gespeicherter Daten auftreten kann, ist es schwer möglich, das Modell wieder aufzufinden. Die Funktion "Center model" holt das Modell in die Mitte des Fensters und passt die Zoomstufe so an dass das Modell komplett und ausreichend groß sichtbar ist.

## Statuszeile



Die Statuszeile bietet von links nach rechts

* Eine Anzeige der abgelaufenen Zeit im Modell. Je nach eingestellter Slow-Motion-Stufe läuft diese Zeit erheblich langsam als die Realzeit.
* Buttons um das Modell in Einzelschritten (Taktrate der Modellberechnung) laufen zu lassen, das Modell zu starten und das Modell anzuhalten
* Die Anzeige der aktuellen Slow-Motion Stufe.
* Buttons, um die Berechnung langsamer oder schneller ablaufen zu lassen.
* Ein Bereich, in dem Meldungen angezeigt werden können.

## Tastatureingaben

Wenn sich der Maus-Cursor über einem Input-Neuron befindet, kann die Pulsrate des Input-Neurons mit den „+“- und „-“-Tasten in Schritten von 1/100 Hertz erhöht bzw. erniedrigt werden. Diese Funktion wirkt sich unmittelbar auf das Modell aus und ist eine schnellere Alternative zu dem Dialog, der über das Kontextmenü aufgerufen wird.

## Mausaktionen im Hauptbereich

Benutzt werden

* die linke Maustaste
* die rechte Maustaste
* das Mausrad (falls vorhanden)
* und natürlich die Bewegung des Maus-Cursors

Um eine Operation, wie z.B. Verschieben oder Aufruf des Kontextmenüs (siehe 4.5.3) auf ein Ob­jekt anwenden zu können, muss dieses ausgewählt sein. Die Auswahl erfolgt automatisch so­bald sich der Maus-Cursor über dem Objekt befindet. Das jeweils ausgewählte Objekt wird durch eine veränderte Farbe der Objektumrandung visualisiert, siehe auch Kapitel 4.6.

## Funktionen des Editors

Der Editor ermöglicht es dem Benutzer, das statische Modell interaktiv zu verändern. Das we­sentliche Instrument dabei sind Kontextmenüs, die über die rechte Maustaste aufgerufen wer­den, siehe 4.5.3. und ziehen mit der Maus (siehe Kapitel 4.5.1).

Verfügbare Funktionen:

* Ein neues (Input-)Neuron erzeugen, das zunächst keine Verbindungen besitzt
* An ein Neuron eine ein- oder ausgehende Verbindung anhängen
* Von einer Verbindung eine ein- oder ausgehende Verbindung verzweigen
* In eine Verbindung ein Neuron einfügen
* Ein (Input-)Neuron oder eine Verbindung löschen.

Das Programm sorgt dafür, dass sich das Modell immer in einem konsistenten Zustand befindet. Das bedeutet, dass

* Input-Neuronen keine eingehenden und Output-Neuronen keine ausgehenden Verbindun­gen haben (letztere werden automatisch zu normalen Neuronen, wenn sie einen Ausgang bekommen).
* kein Neuron mehr als einen unmittelbaren Ausgang hat. Der einzige Ausgang, das Axon, kann sich zwar verzweigen, aus dem Neuron selbst tritt aber unmittelbar nur das Axon selbst aus.

Durch diese Bedingungen sind nicht für alle Objekte immer alle denkbaren Operationen verfüg­bar. Zum Beispiel wird bei einem Neuron, das bereits einen Ausgang hat, die Option „Add outgoing dendrite“ gar nicht angeboten.

Die Elemente des statischen Modells können prinzipiell beliebig auf einer zweidimensionalen Ebene angeordnet werden. Es ist in der Regel sinnvoll, die Input-Neuronen an einer oder mehre­ren Stellen in der Peripherie des Modells zu konzentrieren, das Programm selbst erlegt dem An­wender hierbei jedoch keine Einschränkungen auf. Gleiches gilt für die Output-Neuronen.

### Linke Maustaste (drücken und ziehen)

Mit der linken Maustaste kann das gesamte Modell oder einzelne Neuronen verschoben werden:

Verschieben des Modells als Ganzes:

* den Maus-Cursor auf eine Stelle bewegen, an der sich keine Objekte (Neuron etc.) befin­det
* linke Maustaste drücken und festhalten
* den Maus-Cursor bewegen. Das gesamte Modell bewegt sich mit dem Maus-Cursor.
* Wenn die gewünschte Position erreicht ist, die linke Maustaste loslassen

Einzelne Neuronen oder Dendritenäste verschieben:

* den Maus-Cursor auf das zu verschiebende Objekt bewegen
* wenn das Objekt erkannt wurde, ändert sich als optisches Feedback seine Farbe
* linke Maustaste drücken und festhalten
* den Maus-Cursor bewegen. Das gewählte Objekt bewegt sich mit dem Maus-Cursor. Objekte, die unmittelbar mit dem bewegten Objekt verbunden sind werden mitgezogen. Dendritenäste verändern dabei in der Regel ihre Richtung und Länge.
* Wenn die gewünschte Position erreicht ist, die linke Maustaste loslassen

### Linke Maustaste (Doppelklick)

Mit einem Doppelklick auf die linke Maustaste können häufig gebrauchte Funktionen direkt auf­gerufen werden, ohne den Umweg über ein Menue.

Die Funktion hängt ab von dem Objekt, das sich aktuell unter dem Maus-Cursor befindet.

**Zurzeit sind keine Doppelklick-Funktionen aktiv**.

### Rechte Maustaste

Mit der rechten Maustaste wird das **Kontextmenü** aufgerufen, das je nachdem über welchem Objekt sich der Maus-Cursor gerade befindet, unterschiedliche Funktionen anbietet.

So kann man z.B. mit dem Kontextmenü

* eines Input-Neurons die Pulsfrequenz dieses Neurons einstellen
* eines Neurons diesem neue Dendriten hinzufügen
* einer Verbindung ein Neuron einfügen. Die ursprüngliche Verbindung wird dabei in zwei Ver­bindungen aufgespalten von denen eine in das neu erzeugte Neuron einmündet, während die andere aus dem Neuron hervorgeht.
* eines Neurons dessen Verbindungen lösen „Disconnect“,
* ein Input-Neuron in ein normales Neuron umwandeln und umgekehrt. Die Umwandlung eines normalen Neurons in ein Input-Neuron ist allerdings nur möglich, wenn das Neuron keine eingehenden Dendriten besitzt.
* eines Neurons ein akustisches Signal definieren, das mit beim triggern des Neurons ausgelöst wird.
* etc.

Objekte des Modells können über das Kontextmenü auch wieder entfernt werden („Remove“).

Die nebenstehende Abbildung zeigt das Kon­textmenü der Ver­bindung zwischen zwei Neuro­nen.

Die Option „Add output neuron“ erzeugt an der Stelle, an der sich der Maus-Cursor gera­de befin­det, einen Verzeigungspunkt und hängt einen ausgehenden Dend­riten an.

Die Option „Add input neuron“ erzeugt auf gleiche Weise eine ausgehende Verzweigung.

Mit „Insert neuron“ wird ein normales Neu­ron erzeugt, das an der Cursor-Position in die Ver­bindung eingefügt wird.

### Mausrad

Mit dem Mausrad kann gezoomt werden.

* Das Mausrad nach vorne bewegen, um das Modell zu vergrößern
* Das Mausrad nach hinten bewegen, um das Modell zu verkleinern.

Beim Zoomen bleibt die Position, auf die der Maus-Cursor zeigt, unverändert. Dadurch kann auf intuitive Weise ein Objekt, für das man sich gerade interessiert genauer betrachtet werden, in­dem man mit dem Maus-Cursor auf das Objekt zeigt und das Mausrad nach vorne bewegt.

## Optisches und akustisches Feedback

Der NNetSimu-Editor benutzt visuelle und akustische Effekte, um dem Benutzer die Orientierung zu erleichtern und Feedback zu ausgeführten Operationen zu geben.

Als visuelles Feedback ändert sich die Farbe des ausgewählten Objekts, auf das der Maus-Cursor gerade zeigt. Das Kontextmenü, das mit der rechten Maustaste aktiviert wird (siehe Kapitel 4.5.3), bezieht sich immer auf dieses Objekt. Es gibt Situationen, wie z.B. bei nahe beieinander liegenden oder sich sogar überlagernden Objekten, in denen eine eindeutige Erkennung des aus­gewählten Objekts ohne dieses visuelle Feedback schwierig wäre.

Wenn Objekte miteinander verbunden werden sollen, z.B. das lose Ende eines Dendriten an ein Neuron angeschlossen werden soll, sind zwei unterschiedliche Farbmarkierungen notwendig. Das durch den Maus-Cursor bewegte Objekt, in diesem Fall das Ende des Dendriten wird wie im vor­hergehenden Absatz beschrieben farblich markiert. Sobald sich dieses Objekt über einem ande­ren Objekt befindet, an das es andocken kann, wird das Zielobjekt in einer anderen, auffälligeren Farbe markiert. Diese Zielmarkierung erfolgt nur, wenn das Andocken wirklich möglich ist. Dadurch erkennt der Benutzer frühzeitig, wenn er im Begriff ist, eine nicht zulässige Operation durchzuführen, z.B. einen eingehenden Dendriten an ein Inputneuron anzuschließen.

Ein akustisches Feedback erfolgt bei einigen Operationen, wie Andocken von Objekten, Lösen von Verbindungen („Disconnect“) und Löschen von Objekten. Das akustische Signal kann über das Hauptmenü „Options“ – „Sound on“ bzw. „Sound off“ ein- und ausgeschaltet werden.

## Auswahl von Objekten (Select/Deselect)

Für verschiedene Operationen ist es notwendig, zunächst das oder die Objekt(e) auszuwählen (zu selektieren), auf dem/denen die Operation ausgeführt werden soll.

Im einfachsten Fall bezieht sich eine Operation auf ein einzelnes Objekt (Neuron, Dendriten-Teilstück, Knickstelle). In diesem Fall geschieht die Auswahl einfach indem der Cursor über dem Objekt platziert wird. Die korrekte Positionierung wird durch eine farbliche Veränderung des betreffenden Objekts angezeigt (siehe 4.6). Durch den Aufruf des Kontextmenüs können dann verschiedene Operationen auf das ausgewählte Objekt angewandt werden. Diese Art der Auswahl heißt **temporäre Selektion**. Sie erfolgt automatisch, wenn der Cursor über einem Objekt steht und wird automatisch wieder aufgehoben, wenn der Cursor das Objekt verlässt.

Für Operationen, die auf mehreren Objekten angewandt werden soll ist die temporäre Auswahl nicht ausreichend. Der Benutzer muss die Möglichkeit haben, die Auswahl zunächst nach seinen Vorstellungen zu erweitern oder einzuschränken und erst wenn die gewünschte Menge von Objekten definiert ist die Operation auszulösen. Diese Art der Auswahl heißt **permanente Selektion**. Für die Erstellung und Manipulation einer permanenten Auswahl stehen eine ganze Reihe von Werkzeugen zur Verfügung:

1. Alle Objekte des Modells

CTRL-A

ESC

Hauptmenu

Kontextmenü Hintergrund

1. Einzelne Objekte

Kontextmenü Select/Deselect

Doppelklick links

1. Teilbäume

Kontextmenü Select subtree

1. Rechteckigen Ausschnitt

Rechte Maustaste gedrückt halten und ziehen

1. Auswahl durch Analysefunktion

Siehe Kapitel 4.1.1 und 4.1.2

# Einschränkungen und Grenzen des Modells

NNetSimu kann nicht das menschliche Nervensystem in seiner vollen Komplexität darstellen. Ein­schränkungen sind in mehrfacher Hinsicht notwendig:

1. Einschränkungen im Detaillierungsgrad: Zum Beispiel werden zurzeit keine Synapsen model­liert, sondern die Verbindungen zwischen Neuronen werden so behandelt, als würden die Dendriten unmittelbar in ein nachfolgendes Neuron einmünden. Auch von Details der Axone, wie die Myelinscheide oder die Ranvier-Schnürringe wird abstrahiert.
2. Quantitative Einschränkungen: Während das menschliche Gehirn aus ca. 86 Milliarden Neu­ronen besteht, muss die Kapazität von NNetSimu drastisch geringer sein. Beim aktuellen Entwicklungstand kann noch keine Aussage über die mögliche Maximalzahl getroffen wer­den, sie wird aber wahrscheinlich kleiner als eine Million Neuronen sein.
3. Einschränkungen in der zeitlichen Auflösung: NNetSimu muss die dynamischen Abläufe im Netzwerk in einer gewissen zeitlichen Taktung sequenziell berechnen. Je kürzer dieser Zeit­takt ist, umso größer ist der Rechenaufwand, um das Modell mit einer bestimmten Ge­schwindigkeit laufen zu lassen. Prinzipiell kann man in beliebig kleinen Schritten berechnen, das Modell wird dann aber nur in extremer Zeitlupe laufen. Umgekehrt ist ein zu lang ge­wählter Takt ungünstig für die Darstellung schnell ablaufender Vorgänge, z.B. Anstieg und Abfall des Aktionspotentials beim Auslösen eines Neurons (siehe Nyquist-Shannon-Abtasttheorem). Zurzeit ist ein Takt von 100 Mikrosekunden voreingestellt. Der Wert kann im Parameterfenster verändert werden, siehe Kapitel 3.2
4. Einschränkungen in der Vielfalt der darstellbaren Neuronen, Dendriten usw. NNetSimu ent­hält zurzeit nur eine geringe Anzahl von elementaren Typen von Neuronen, die im Kapitel 2 „Die Elemente des Modell“ beschrieben sind. Bei Bedarf können weitere Sonderfälle, wie sie in der Na­tur vorkommen hinzugefügt werden.

Grenzen für die Größe des darstellbaren Modells ergeben sich aus der begrenzten Rechenkapazität des verwendeten Computers. Insbesondere die Berechnung des Modells für jedem Zeittakt stellt eine Grenze für die Größe des Modells dar. Einflussgrößen hierfür sind:

* Die Rechenleistung der CPU
* Die Anzahl der Objekte des Modells. Jedes Neuron oder Input-Neuron ist in diesem Sinne ein Objekt, aber auch jeder Verzeigungs- oder Knickpunkt von Dendriten und jeder Dendritenabschnitt zwischen solchen Punkten, bzw. Neuronen. Das in der Abbildung zu Kapitel 2.7 dargestellte Modell besteht z.B. aus 10 Objekten: einem Input-Neuron, zwei normalen Neuronen, einem Output-Neuron, einem Verzeigungspunkt und 5 Dendriten-Abschnitten.
* Die zeitliche Auflösung des Modells (time resolution).
* Die eingestellte Zeitlupe

Auf einem Rechner mit Intel Core i5 4670 CPU folgende Ergebnisse erzielt werden:

Testobjekt war ein Modell mit insgesamt 187 Objekten, darunter 30 Neuronen. Um den Rechner an seine Grenzen zu bringen wurde der Parameter TimeResolution auf 2µs (Faktor 50 gegenüber dem Standardwert) gesetzt. Die höchstmögliche Geschwindigkeit war ein Zeitlupenfaktor von etwa 10 (Faktor 10 schneller als der Standardwert). Damit bestehen noch Reserven von etwa Faktor 500.

Anders herum ausgedrückt: Wenn man die Anzahl der Neuronen als Maßstab nimmt, müsste ein Modell mit etwa 30 \* 500 = 15000 Neuronen auf diesem Rechner möglich sein, auf einem langsameren Rechner entsprechend weniger.

# Änderungshistorie

**Änderungen in Version 2020-03-23:**

1. **Das Kapitel „Offene Punkte“, das mögliche Erweiterungen, Fragen und bekannte Fehler enthalten hat, wurde aus dem Dokument entfernt. Die offenen Punkte werden jetzt ausschließlich in GitHub verwaltet.**
2. **Am Ende des Dokuments werden in der Änderungshistorie die Beschreibungen früherer Versionen dokumentiert.**
3. **GitHub Issue #17 „Center model“ wurde implementiert, siehe Kapitel 4.1.2. Ich habe den Verdacht, dass beim automatischen Öffnen des zuletzt gespeicherten Modells (siehe Issue #2), dieses nicht sichtbar ist, weil es aus irgendeinem Grund außerhalb des Fensters landet. Falls diese Vermutung zutrifft, kann das Modell mit dieser Funktion wieder sichtbar gemacht werden. Auch unabhängig von dieser Fehleranalyse kann die Funktion nützlich sein, wenn man versehentlich das Modell ins „Nirwana“ verschoben hat und Mühe hat, sie wieder zu finden.**
4. **Um dem Problem mit Issue #2 „Zuletzt gespeichertes Modell automatisch öffnen“ auf den Grund zu gehen wurden weitere Fehlerabfragen eingebaut. Es kann beim Starten des Programms ein Meldungsfenster erscheinen. Gegebenenfalls bitte mir melden.**
5. **Bei jeder Änderung der Pulsfrequenz eines Inputneurons wird das gesamte Modell zurückgesetzt, d.h. die Spannungspegel aller Neuronen und Dendriten werden auf 0 gesetzt. Damit starten das Modell immer mit reproduzierbaren Ausgangsbedingungen. Siehe Issue #14 „Andocken von Bereichen (kohärent oder different) - Eigenfrequenz“**

**Änderungen in Version 2020-03-20:**

1. **GitHub Issue #5: Ab dieser Version wird keine readme.txt-Datei mehr mit ausgeliefert, sondern alle Neuerungen werden in der Dokumentation (diese Word-Datei) beschrieben.**
2. **GitHub Issue #15: Die Pulsrate von Input-Neuronen kann jetzt sehr einfach und in kleinen Schritten verändert werden, indem man den Maus-Cursor über das Input-Neuron bringt und die +/- Tasten der Tastatur betätigt. Siehe Kapitel 4.3.**
3. **GitHub Issue #1: Eine neue Funktion wurde implementiert, die das Modell nach (unbeabsichtigten Schleifen) durchsucht. Siehe Kapitel 4.1.1**