PR: Programmierpraktikum II (SS2000)
Visualization of Artifical Neuronal Networks

Andreas Lackner
luke@cosy.sbg.ac.at

Version 2000-04-05

Gruppe 2
Zeit: Mittwoch 16.30 - 1700
Lehrveranstaltungsleiter: Schwaiger Roland

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

1.1 Darstellungsmöglichkeiten von ANNs

ANNs bestehen aus Knoten und Verbindungen. Eine Verbindung hat einen Start- und einen Endknoten, die ident sein können. Damit ist die graphische Darstellung logisch festgelegt. Was jedoch nicht feststeht ist die topographische Lage der Knoten in ihrer Umgebung. Bei der Erstellung von ANNs mit Hilfe des SNNS-Pakets muß die Position eines Knotens im Dreidimensionalen Raum vom Anwender eingegeben werden, um graphisches Arbeiten zu ermöglichen.

In der Literatur haben sich verschieden Darstellungsformen für Netze durchgesetzt, zumindest in Bezug auf die Topographische Darstellung. Die graphische Darstellung eines Knotes ist meist als Quadrat oder Kreis gehalten (bzw. Würfel oder Kugel im dreidimensionalen Darstellungen). Eine Verbindung wird als Linie vom Start- zum Endknoten gezeichnet.

1.2 Anforderungen an graphische Darstellungen

Grundsätzlich sollen Knoten so angeordnet werden, wie es den allgemeinen Standards der Literatur entspricht. In Abbildung 1.1 ist die typische Anordnung der Knoten eines Feed-Forward Netzes zu sehen

Bei der Anordnung der Knoten ist zu beachten, das die Verbindungslinien keine Kn oten schneiden sollen, um die 'Lesbarkeit' des Bildes nicht zu beeinträchtigen. Die Knoten-Daten müssen so platziert werden, daß klar erkennbar ist welche Da ten zu welchem Knoten gehören.

Zu jedem Knoten können viele verschiedene Daten interessant sein und sollten daher angezeigt werden. Werden zu viele Daten angezeigt leidet die Lesbarkeit, weiters ist es sowieso kaum möglich auf engstem Raum alle Daten darzustellen. Es muß also eine Problemlösung gefunden werden, die es dem Anwender ermöglicht schnell und

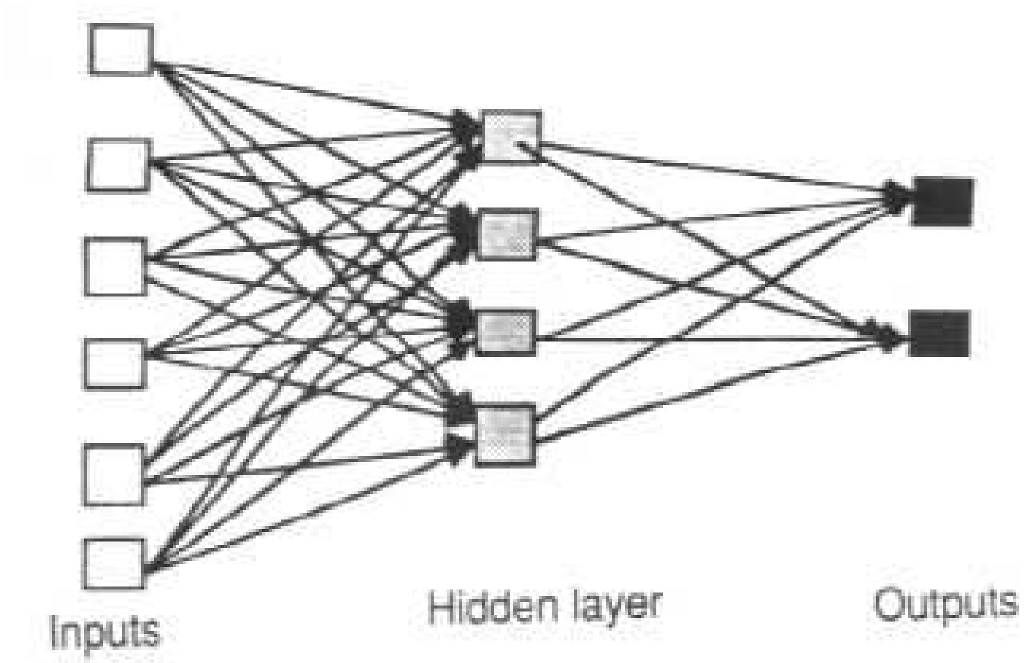

Abbildung 1.1: Beispiel eines einfachen Feed-Forward Netzes

effizient die Daten anzuzeigen, die gerade interessant sind.

Betreffend des Orts, an dem die Daten platziert werden, sollte jeder seinen eigenen Vorlieben folgen können. Das bedeutet, das jedes Feld (in dem Daten oder Knoten dargestellt werden) beliebig verschiebbar sein soll. Da die Datenfelder in Bezug zu einem Knoten stehen, sollen die Datenfelder mitverschoben werden, wenn der Knoten verschoben wird. Diese Funktion kann unter Umständen stören, z.B. wenn der Anwender Datenfelder am Rand der Graphik anordnet, in solchen Fällen kann die Verschiebung der Datenfelder unterdrückt werden.

Kapitel 2

Use Cases

Im folgenden Kapitel sollen die verschieden Use-Cases beschrieben werden, wie Sie in Abbildung A.1 auf Seite 6 dargestellt sind.

2.1 Verzeichnis auswählen

In einem Hauptfenster, das ähnlich einem Dateibrowser aufgebaut ist, wählt der Anwender ein Verzeichnis aus einem Verzeichnisbaum entweder durch Anklicken mit der Maus oder durch Cursorbewegungen mit den Pfeiltasten und der Enter-Taste au s. Die *.net-Dateien und die Unterverzeichnisse des gewählten Verzeichnisses we rden in einem eigenen Teil des Hauptfensers aufgelistet. Zu jedem *.net-File wir d ein Thumbnail erzeugt, das die Struktur des Netzes darstellt. Die Daten dazu w erden aus dem *.net-File gelesen.

2.2 Datei öffnen

Durch Doppelklick mit der linken Maustaste auf eines der Thumbnails wird eine Da tei geöffnet. Falls zur *.net Datei eine gleichnamige *.vis-Datei existiert, wi rd auch dies geöffnet, Falls nicht, wird eine solche Datei erzeugt und ein Algo rithmus zur topographischen Anordnung der Knoten wird aufgerufen. Die Daten dies er Files werden verwendet um eine Darstellung entsprechend der Voreinstellungen zu erzeugen, die in einem eigenen Fenster dargestellt werden (Graphikfenster).

2.3 Netz zeichnen

Die Daten zur Beschreibung einer Netzstruktur sind in *.net-Files abgelegt. Die Topographie des Netzes wird auch im *.net File gespeichert. Alle anderen Daten, wie

Farbetabellen, Position und Zuordnung der Datenfelder werden im *.vis-File g espeichert. Die graphische Darstellung der Knoten und Verbindungen unterscheidet sich je nac h Aktivierungs-Zustand der einzelnen Knoten.

2.4 Unit aktivieren

Der Anwender aktiviert im Graphikfenster einen Knoten durch Anklicken mit der li nken Maustaste. Das Anklicken einer bereits aktivierten Unit deaktiviert sie wie der. Der Aktivierungs-Zustand wird im *.vis gespeichert.

2.5 Unit-Daten in extra Fenster darstellen

Der Anwender öffnet im Graphikfenster einen Knoten durch doppelklicken mit der linken Maustaste. Dadurch öffnet sich ein extra Fenster, in dem sämtliche zur Verfügung stehenden Daten, die diesen Knoten und die zugehörigen Verbindungen betreffen, dargestellt werden.

2.6 Datenfeld verschieben

Alle Datenfelder (also Knoten, Textfelder usw.) können vom Anwender im Graphikf enster verschoben werden (Drag & Drop). Sämtliche Drag & Drop Bewegungen, die d er Anwender durchführt, werden abgespeichert, um sie bei Bedarf rückgängig zu machen.

2.7 Unit verschieben

Verschiebt der Anwender einen Knoten (s.a. Datenfelder umplatzieren), so kann er entscheiden, ob die zu diesem Knoten gehörigen Datenfelder mitverschoben werde n oder nicht.

2.8 Datei speichern

Der Anwender kann die Daten durch Anwählen eines Menüpunkts abspeichern. Dabei werden Koordinaten der Knoten und die Struktur des Netzes im *.net-File gespeic hert, alle anderen Daten werden im *.vis-File gespeichert. Der Anwender wird nac h dem Verzeichnis und Namen der Datei gefragt, unter der er das Netz abspeichern will und gewarnt, falls er bestehende Dateien überschreibt.

2.9 Screenshot exportieren

Die Darstellung des Netzes kann jederzeit als Screenshot exportiert werden, inde m der Anwender einen entsprechenden Menüpunkt im Graphikfenster anw:ahlt. Das S peichern der Graphik erfolgt durch einen eigenen Export-Filter. Je nach Vorhande nsein der entsprechenden Export-Filter können auch verschieden Datei-Typen erze ugt werden.

Anhang A

Diagramme

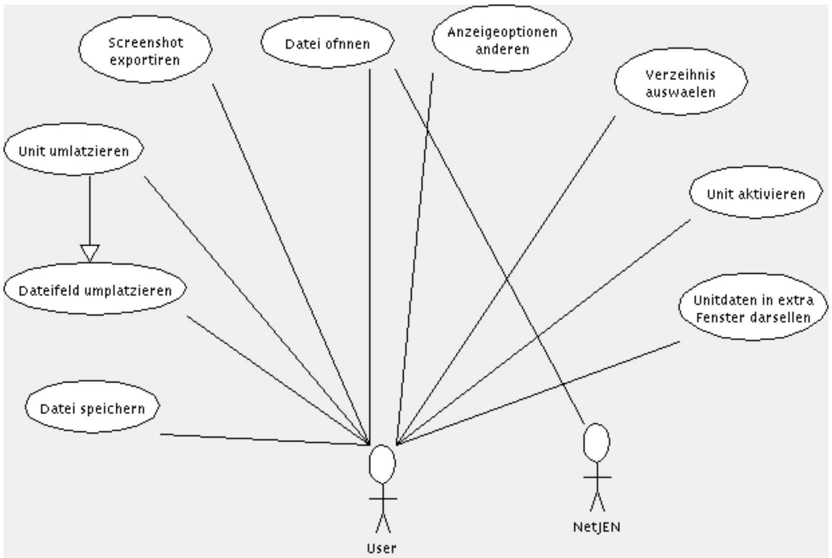

Abbildung A.1: Use-Case Diagramm

Anhang B

Protokolle

B.1 14.3.2000

Teilnehmende Personen: Schwaiger Roland
Lackner Andreas
Jestratijevic Marko

Protokollführer: Andreas Lackner
Ort: Büro Herr Schwaiger
Zeit: 16.30 - 17.00

Einführende Besprechung über das Projekt.
Programmfunktionen:

- Darstellung von ANNs, die aus SNNS-Daten gelesen werden.
- Manipulation der Darstellung am Bildschirm (mit der Maus).
- Drucken und Speichern der Daten in das SNNS-File.

Das Praktikum soll unter Zuhilfenahme folgender Softwarepakete durchgeführt werden:

- UML für den Entwurf und Design
- Java als Programmiersprache
- LaTeXfür die Dokumentation des Projekts

Bis zur Besprechung am 21.3.2000 sollten folgende Daten gesichtet werden:

- Dokumetation UML

- Beschreibung des SNNS Datenformats
- LateX Kurzbeschreibung

B.2 21.3.2000

Teilnehmende Personen: Schwaiger Roland
Lackner Andreas
Jestratijevic Marko

Protokollführer: Andreas Lackner
Ort: Büro Herr Schwaiger
Zeit: 16.30 - 17.00

Grundsätzliche Klärung der Ziele bei der Anwendung von UML

Erste Fragestellungen:

- Welche Daten sind in SNNS-Files gespeichert, was stellen diese Daten dar?
- Wie kann man ANNs graphisch darstellen?
- Welche Netztypen gibt es?

B.3 28.3.2000

Teilnehmende Personen: Schwaiger Roland
Lackner Andreas
Jestratijevic Marko

Protokollführer: Andreas Lackner
Ort: Büro Herr Schwaiger
Zeit: 16.30 - 17.00

Aussehen der Protokolle/Dokumentation.

Diskussion der bisher gefundenen use cases. Festlegung darauf, das das Programm in mehrere getrennt voneinander laufende Prozesse aufgegliedert wird. Damit soll auch ein Thread aus dem restlichen NetJEN-Paket unsere Threads aufrufen koennen (RMI, Remote Method Invocation).

Diskussion des Entwurfs der graphischen Darstellung.

UML-Anwendung bei:

- use cases
- sequence diagram
- colaboration diagram

Dazu soll ArgoUML verwendet werden. Ein Beispiel zur Anwendung von UML liegt als Artikel aus c't 1999, Heft 21, vor. ArgoUML ist auf den UNIX-Worksatations im Rechnerübungsraum installiert.

Im Rechnerübungsraum liegt eine Beschreibung in BNF der SNNS .net-Files auf.