Universität Salzburg, Institut für Computerwissenschaften

Proseminar Programmierpraktikum I
Sommersemester 1999

Spiele mit Künstlichen Neuronalen Netzwerken (KNN)

Wiesbauer Gerald	gerald.wiesbauer@sbg.ac.at
Stürzlinger Manuela	mstuerz@cosy.sbg.ac.at
Rettenbacher Christopher	cretten@cosy.sbg.ac.at
Mayer August	ags@gmx.net
Projektbetreuer:	
Dipl.Ing. Mag. Roland Schwaiger	rschwaiger@cosy.sbg.ac.at

Version vom 13. April 1999

Zusammenfassung

Dieses Dokument beschreibt die Planung und Implementierung eines Backgammon-Spiels mit Java, wobei der Computergegner durch ein neuronales Netzwerk möglichst spielstark sein soll.

Inhaltsverzeichnis

1 Einleitung

1.1 Spielregeln Backgammon[?]

1.1.1 Spielidee

Backgammon ist ein Spiel, bei dem zwei Spieler mit jeweils 15 Steinen auf einem Brett mit 24 Feldern spielen.Ziel des Spieles ist es, alle Steine vom Brett ins Aus zu spielen. Die Spielsteine haben eine bestimmte Startposition und müssen von dort aus alle in das Heimfeld gezogen werden. Das sind bei Weiß die Felder 1 bis 6, bei Schwarz die Felder 19 bis 24. Die Spieler ziehen deshalb immer in entgegengesetzte Richtung (Weiß von 24 nach 1; Schwarz von 1 nach 24), wobei sie auf ihrem Weg durch Blockaden des Gegners behindert werden können. Sind alle Steine im Heimfeld versammelt, dann können sie ausgespielt, das heißt durch würfeln aus dem Spiel genommen werden. Wer zuerst keine Steine mehr auf dem Brett hat, hat gewonnen.

1.1.2 Würfeln und Ziehen

Jeder Spieler würfelt vor seinem Zug mit zwei Würfeln und zieht die geworfene Augenzahl vorwärts (niemals rückwärts!).Als Beispiel nehmen wir an, Weiß hat in der Ausgangsstellung 6+5 geworfen. Er kann jetzt entweder zwei Steine mit jeweils einer Zahl (13 auf 7 und 13 auf 8) oder nur einen Stein mit beiden Zahlen spielen(24 auf 13). Welche der beiden Zahlen der Spieler dabei zuerst zieht, kann er sich aussuchen. Der Wurf muß aber immer vollständig ausgeführt werden, wenn das nicht durch eine Blockade des Gegners verhindert wird. Dabei darf man beliebig viele Steine seiner Farbe auf ein Feld stellen.
Würfelt man zwei gleiche Zahlen, so nennt man dies einen 'Pasch'. Wird ein Pasch geworfen, gilt folgende Sonderregel: Die Augenzahl ist viermal auszuführen- mit einem, zwei, drei, oder vier Steinen.

1.1.3 Blockierte Felder

Ein Feld ist blockiert, wenn dort zwei oder mehr Steine des Gegners stehen. Auf ein blockiertes Feld kann nicht gezogen werden, man darf es jedoch überspringen. Kann nur eine der beiden gewürfelten Zahlen ausgeführt werden, muß die höhere Zahl ziehen.

1.1.4 Schlagen

Steine können nur geschlagen werden, wenn sie einzeln auf einem Feld stehen. Ein geschlagener Stein wird auf die Bar gestellt, das ist der Balken zwischen den Bretthälften. Im nächsten Zug muß er -wenn möglich- wieder ins Spiel gebracht werden.

1.1.5 Wiedereinbringen geschlagener Steine

Wenn ein geschlagener Stein auf der Bar steht, muß er zunächst wieder ins Spiel gebracht werden, bevor man mit einem anderen ziehen darf. Geschlagene Steine müssen von der Bar ins Heimfeld des Gegners gespielt werden. Hierbei landet ein Stein bei einer 1 auf dem 1er(24er) Punkt, usw. Auf vom Gegner blockierte Felder darf nicht gezogen werden.

1.1.6 Ausspielen

Mit dem Ausspielen darf erst begonnen werden, wenn alle 15 Steine im eigenen Heimfeld sind. Man spielt dabei so als gäbe es ein Feld 0, auf das alle Steine ziehen müssen. Sind von den höheren Punkten alle Steine ausgespielt,so müssen die entsprechenden Zahlen vom nächst niedrigeren Punkt ins Aus gespielt werden.

1.1.7 Gammon und Backgammon

Ein Spiel ist Gammon gewonnen, wenn eine Partei alle 15 Steine, der Gegner aber noch keinen Stein ausgespielt hat. Ein mit Gammon gewonnenes Spiel zählt doppelt.
Hat der Sieger alle 15 Steine abgetragen, während er Gegner noch einen Stein auf der Bar, oder im Heimfeld des Siegers hat, dann zählt das Spiel dreifach, er hat Backgammon gewonnen.

1.1.8 Spielbeginn

Zu Beginn der Partie würfeln beide Spieler jeweils mit einem Würfel. Der Spieler der die höhere Zahl hat beginnt die Partie mit dem sich ergebenden Wurf.

1.1.9 Dopplerwürfel

Zu jedem Spiel gehört ein Würfel, der auf seinen Seiten die Zahlen 2, 4, 8, 16, 32 und 64 hat. Glaubt ein Spieler einen klaren Vorteil zu haben, nimmt er den Würfel und legt ihn mit der 2 oben auf die Spielfläche. Sein Gegner kann nun aufgeben, dies würde ihn einen Punkt kosten. Nimmt er die Herausforderung an, erhält er das alleinige Recht, bei einer Wende im weiteren Spielverlauf, ebenfalls zu verdoppeln. Auch diese Herausforderung kann angenommen, oder abgelehnt werden. Der Spielwert verdoppelt sich jedes mal.

1.2 Neuronale Netze[?]

1.2.1 Biologische Grundlagen

NN können als technische Umsetzung der Gehirnfunktionen verstanden werden. Der Aufbau der Nervenzellen in Gehirn weicht in auffälliger Weise von dem aller anderen Zellen des Körpers ab. Sie besitzen sogenannte Dendriten und Axone, die durch Synapsen verbunden sind. Dadurch entsteht zwischen den verschiedenen Neuronen eine hochgradige Vernetzung. Die synaptischen Verbindungen ändern sich im Laufe der Zeit, d.h. neue Synapsen entstehen und alte verkümmern. Wenn ein Neuron aktiv ist, so feuert es kurze Spannungsspitzen, sog. Aktionspotentiale, die sich über das Axon zu den Synapsen fortpflanzen,ab.
Zum Begriff des KNN gelangt man, wenn man die beschriebenen Vorgängen zu einem stark vereinfachten, abstrakten Modell zusammenfaßt:

1. Aufbau einer Nervenzelle: Eine Nervenzelle besitzt viele Eingänge, nämlich die synaptischen Verbindungen, und einen Ausgang, nämlich das Axon.
2. Zustände einer Nervenzelle: Eine Nervenzelle kann zwei Zustände annnehmen: den Ruhezustand und den Erregungszustand.
3. Verbindungen der Nervenzellen untereinander: der Ausgang einer Nervenzelle führt zu den Eingängen anderer Nervenzellen.
4. Eingänge des Nervensystems: Einige Nervenzellen (z.B. Sinneszellen) können durch Umweltreize erregt werden.
5. Ausgänge des Nervensystems: Einige Ausgänge von Nervenzellen wirken(über Muskelzellen) auf die Umwelt ein.
6. Erregungsbedingung. Eine Nervenzelle geht in den Erregungszustand über, wenn genügend viele ihrer Eingänge mit erregten Nervenzellen verbunden sind.
7. Unabhängigkeit der Nervenzellen: Der Zustand einer Nervenzelle ist allein durch die Verhältnisse an ihren Eingängen bestimmt. Die einzelnen Zellen arbeiten also unabhängig voneinander.

1.2.2 Künstliche neuronale Netze

1. Aufbau neuronaler Netze

   Die Nachahmungen von biologischen Neuronen-Netzen werden künstliche neuronale Netze (KNN) oder vereinfacht neuronale Netze (NN) genannt. Dabei werden in Anlehnung an die Natur die Dendriten, der Zellkern, das Axon und die Synapsen nachempfunden.

   Der künstliche Zellkern, auch Neuron, Element oder Unit genannt, besteht aus einem zeitabhängigen Aktivierungszustand $a_i(t)$, der den Grad der Zellaktivierung mit einem reellwertigen, skalaren Wert angibt. Diese Zahl läßt sich über die Aktivierungsfunktion f_a berechnen mit:

   $$a_i(t+1) = f_a(a_i(t), net_i(t), O_i)$$

   Wobei $a_i(t)$ die ursprüngliche Aktivierung des Neurons ist, $net_i(t)$ der Netzeingabe von anderen Neuronen und i dem Schwellenwert entspricht, bei dem das Neuron aktiv wird. Die Ausgabe O des Neurons i wird über die Ausgabefunktion f_o mit dem Parameter Aktivität bestimmt:

   Ein neuronales Netz besteht aus verschiedensten Verknüpfungen zwischen den einzelnen Neuronen und kann als ein gewichteter, gerichteter Graph angesehen werden, da er einen Eingabevektor auf einen Ausgabevektor abbildet.

   Die Verknüpfung von oben beschriebenen Neuronen mittels Verbindungen nennt man Netzwerkarchitektur, -struktur oder -topologie. Zur Beschreibung einer bestimmten Architektur gehören nicht nur die Neuronen und ihre Verbindungen, sondern auch das Gruppieren einer Anzahl von Neuronen innerhalb eines Netzwerkes. Diese Gruppierungen werden Schichten (engl. layers) oder Ebenen genannt.

2. Netzwerktypen

   Die unterschiedlichen Netzwerktopologien lassen sich in zwei Kategorien einteilen:

   (a) Netzwerke ohne Rückkopplung

   Bei dieser Architektur werden die Informationen von der ersten Schicht (Eingabeschicht) über die zweite Ebene (verdeckte Schicht, engl. hidden layer) zur Ausgabeschicht geleitet (siehe Abb. 2.2.3). Die Anzahl der verdeckten Schichten, deren Aktivität im Allgemeinen versteckt bleibt, kann dabei variieren und auch Null betragen. Es gibt jedoch keine Verbindung von einer höheren zu einer der vorgelagerten Ebenen, sowie keine in sich rückgekoppelten Informationsflüsse eines einzelnen Neurons. Solch ein Netzwerk wird als feedforward-Netz bezeichnet und kann auch Verknüpfungen besitzen, die eine Schicht überspringen. Diese Verbindungen werden shortcut connections bezeichnet.

   Ein vollständig ebenenweise verbundenes,vorwärtsgerichtetes, neuronales Netz (feedforward)

   (b) Netzwerke mit Rückkopplung

   Das abgebildete neuronale Netz besitzt im Gegensatz zum feedforward-Netz auch Verbindungen, die einen rückwärts gerichteten Informationsfluß zulassen. Sie werden auch Hopfield- oder feedback-Netze genannt und besitzen nur eine Ebene, die sowohl als Eingabe- als auch als Ausgabeschicht dient. Desweiteren existieren Netzwerke mit speziellen Rückkopplungen zwischen den einzelnen Neuronen einer Ebene (lateral feedback), nur zwischen nebeneinanderliegenden Schichten (indirect feedback) und einzelner Neuronen mit sich selbst (direct feedback).

   Ein vollständig verbundenes, neuronales Netz (feedback);

3. Lernverfahren

Wählt man eine bestimmte Netzwerkarchitektur aus, so muß dem neuronalen Netz mitgeteilt werden, wie es die Eingabeparameter behandeln soll und wie das Ergebnis an der Ausgabeschicht auszusehen hat. Das Angehen und Lösen dieser Aufgabenstellung wird, wie beim Menschen, durch eine Lernphase vollzogen. Die Trainingsphase eines neuronalen Netzes zur Bestimmung der Eingabe-Ausgabe-Beziehung läßt sich in drei Verfahren unterteilen:

- Überwachtes Lernen (engl. supervised learning)
  Dies ist die schnellste Lernmethode, bei der man dem neuronalen Netz Eingaben und die zugehörigen, gewünschten Ausgaben präsentiert. Durch die Minimierung des Fehlers zwischen berechneter und angegebener Ausgabe lernt es Ein- und Ausgabe aufeinander abzubilden und auf unbekannte, aber ähnliche Eingabemuster mit richtigen Ausgaben zu antworten (Generalisierung)

- Bestärkendes Lernen (engl. reinforcement learning)

  Bei diesem Verfahren wird dem neuronalen Netz nicht die exakte, zu erwartende Ausgabe mitgeteilt, sondern nur, ob die berechnete Lösung richtig oder falsch ist. Dies ist aus biologischer Sicht sicherlich eine plausible Methode, die jedoch mit sehr viel Zeitaufwand verbunden ist, da die zu trainierenden Netze weniger Informationen zur Änderung ihrer Parameter besitzen. In Zukunft sollen Untersuchungen durchgeführt werden, die dieses Lernverfahren auf Netze mit a priori Wissen anwenden, um die Trainingsphase zu verkürzen;

- Unüberwachtes Lernen (engl. unsupervised learning)
  Hier werden dem Netz nur die zu berechnenden Eingaben zugeführt. Ohne das Wissen einer gewünschten Ausgabe findet es selbständig Zusammenhänge zwischen den einzelnen Eingaben und entwickelt entsprechende Ausgaben. Das neuronale Netz versucht, Gruppen ähnlicher Eingabevektoren zu identifizieren und sie dann auf Gruppen ähnlicher oder benachbarter Zellen abzubilden. Ähnliche Strukturen, auch topologieerhaltende Karten genannte, Netzwerke sind schon biologisch bei Säuglingsgehirnen im visuellen Kortex nachgewiesen worden.

  Während des Lernvorgangs verändern sich interne Parameter des neuronalen Netzes, durch die die Erfahrungen, wie das Wissen beim biologischen Vorbild, gespeichert werden. Das anzulernende Netz ist also in der Trainingsphase ein dynamisches System, bei dem die Konvergenz auf ein gewünschtes Netzverhalten, also eine bestimmten Ausgabe, angestrebt wird. Um dies zu erreichen, können verschiedene Parameter eines Netzes angepaßt werden:

  (a) Modifikation der Verbindungsstärken
  (b) Änderung der Schwellenwerte und internen Funktionen
  (c) Hinzufügen oder Löschen von Neuronen

4. Einsatzmöglichkeiten u. Problematik

Der entscheidende Schritt ist, die Anzahl der Zellen und der Verknüpfungen festzulegen. Dies ist im allgemeinen der schwierigste und wichtigste Prozeß beim Aufbau eines neuronalen Netzes, da mit ihm festgelegt wird, wie viele unterschiedliche Klassen vom Netz separiert werden können. Es gibt leider nur empirische Studien über die notwendige Anzahl von Verknüpfungen und Neuronen, so daß es bislang nicht möglich ist, die exakte Anzahl der Elemente vorauszuberechnen.

Da neuronale Netze auf Grund ihres Generalisierungsvermögens immer eine gewisse Ungenauigkeit in ihren Ausgaben besitzen, werden sie im allgemeinen nicht für Probleme, die exakte Lösungen benötigen, eingesetzt. Vielmehr haben sie gegenüber gewöhnlich programmierten Programmen den Vorteil, hochdimensionale Aufgabenstellungen klassifizieren zu können. Dies betrifft Anwendungen, in denen der Zusammenhang der einfließenden Parameter schwer zu beschreiben oder gar nicht zu bestimmen ist. Zudem sind sie durch ihre Generalisierung in der Lage, zwischen den bekannten Daten zu interpolieren und bedingt zu extrapolieren.

2 OO-Design und Analyse

2.1 Use-Case-Diagramm

Abbildung 1: Use-Case-Analyse des Problems.

2.2 Klassendiagramm

Abbildung 2: Klassendiagramm von GammonApp.

A Glossar

“abfeuern“:
Das Umschalten des Ausgangs der Neuronen von 0 auf 1 wird als “feuern“ bezeichnet.

Algorithmus:
Eine Rechenvorschrift.

Back-Propagation Networks:
Neuronale Netze die das Fehlersignal speziell auf die einzelnen Neuronen aufteilen.

Exzitatorisches Signal:
von lat. excitare=anregen; Signal, das ein Neuron anregt, zu feuern. Siehe auch Inhibitorisches Signal.

Fehlersignal:
Ein Signal das den Neuronen anzeigt ob sie etwas falsch machen. Auch als Residualvektor bezeichnet.

Gewicht (weight)
Parameter, der die Stärke der Verbindung zweier Neuronen beschreibt.

GUI:
Graphical User Interface, die von Benutzer sichtbare Oberflache des Programmes.

Hopfield-Netz:
Neuronales Netz mit bidirektionalen Verbindungen zwischen den Neuronen.

Inhibitorisches Signal:
inhibit = hemmen, ein Signal welches ein Neuron vom “feuern“ abhalt. Gegensatz dazu: Exzitatorisches Signal.

JAVA:
Eine objektorientierte Programmiersprache.

Künstliche neuronale Netze:
Ein auf den Computer simuliertes neuronales Netz.

Neuron
Biologie: signalverarbeitende Nervenzelle, besteht aus Dendriten (Eingangsfasern), Axonen (Ausgangsfasern) und Zellkörper.

Rückkopplung:
Speziell bei den KNN eine Rückverbindung vom Ausgang eines oder mehrerer Neuronen zurück zu den Eingängen derselben. Rückgekoppelte Netze können nicht mehr seriell durchgerechnet werden, sondern müssen iteriert werden, d.h. es müssen alle Neuronen mehrfach mit den aktuellen Eingangswerten berechnet werden, bis eine definierte Endsituation eintritt (z.B. keine Änderung der Aktivierungszustände mehr).

TD-Gammon:
Implementierung des TD-Algorithmus in ein Backgammon Spiel, durchgeführt von Gerald Tesauro für IBM.

B Protokolle

B.1 Protokoll vom 10. März 1999

- **Projekt:** Backgammon mittels Artificial Neural Networks
- **Ort:** Hr. Schwaigers Büro
- **Zeit:** 18^{00}
- **Anwesende:**

  - DI Mag.Roland Schwaiger
  - August Mayer
  - Gerald Wiesbauer
  - Christopher Rettenbacher
  - Manuela Stürzlinger

- **Protokollführerin:** Manuela Stürzlinger

- **Themen:**

  - Wiederholung des administrativen Teils
  - Quellen fuer Lehrmaterial(Java,UML,HTML,Latex)
  - Präsentation der Web-Site unserer Projektgruppe
  - Informationsmaterial
    „Temporal Difference Learning and TD-Gammon“
    „Neural Networks “(s.S. 66) by Müller, Reinhardt
  - Besprechung des Dokumentaufbaus bis Punkt Protokolle

- **Aufgaben**

  - Fotos der Projektmitarbeiter (fur Web-Site)
  - Auseinandersetzen mit den Zeitungsartikel
    ev. auftauchende Fragen in Punkt Literaturuebersicht des Dokumentes vermerken
  - Ausarbeitung des Dokuments bis Punkt Einleitung konkret:
    * Titelseite
      Name des Projekts
      Name der Projektmitarbeiter

    * Abstract
      kurze und prägnante Beschreibung des Projekts

    * Einleitung
      Was ist Backgammon?
      Wie lauten die Spielregeln?
      Erscheinungsbild des Spielbrettes

- Ziele bis nächste Woche

  - Gestaltung der Web-Site(Fotos)
  - Lesen des Zeitungsartikels
  - Ausarbeitung des Dokuments bis Dienstag, den 16.03.99
    Abgabe per E-Mail(.dvi-File)an DI Mag. Schwaiger

B.2 Protokoll vom 17. März 1999

- **Ort:** Hr. Schwaigers Büro

- **Zeit:** 18^{00}

- **Anwesende:**

  - DI Mag.Roland Schwaiger
  - August Mayer

- Gerald Wiesbauer
- Christopher Rettenbacher
- Manuela Stürzlinger

- **Protokollführer:** August Mayer
- **Themen:**
  - Administrative Themen
    - Fotos für Webseite sind noch nicht eingetroffen. Vorschlag eines Gruppenfotos für die Projekt-Homepage.
    - Bei Mail an Hr. Schwaiger im Subject „PPKT 8: “ voranstellen
    - Bilder als GIF oder JPEG in Dokument einbinden bzw. als GZIPte Postscript-Datei schicken
  - Diskussion der Gestaltung des Papers
    - Titelseite sollte eine separate Seite sein
    - Inhaltsverzeichnis ist sehr gut
    - Quellenangabe bei der Einleitung hinzufügen (`\cite{...}`)
    - Bilder für Spielfelder fehlen; es gibt auf einer Seite im Internet geeignete Illustrationen → Webmaster dieser Seite muß informiert werden, Referenz auf diese Seite
    - Literaturverweise auf Bücher zu Backgammon-Taktiken hinzufügen; evtl. einige Taktiken selber lernen → Taktik-Spiel gegen Computer wäre vielleicht interessant
    - Inhaltsangabe TD-Gammon: Was sind diese „hand-crafted features“?
  - Viel Information im Internet zu TD-Gammon verfügbar
  - SNNS arbeitet nur bei mehr als 256 Farben
  - Fragen zum Text
    - Neuronen – verbundene Verarbeitungseinheiten
    - TD(λ) – Formel, neuronales Netz als Minimumsuche, Attraktoren
    - Diskussion der zwei „concept features“
- **Aufgaben**
  - Gestaltung des Papers aktualisieren, notwendige Änderungen durchführen
  - Einführung zu den Neuronalen Netzen (Oldstedt) auf der Projekt-Homepage durcharbeiten
  - Glossar zu Begriffen wie Neuron, . . . anlegen
  - Erste Überlegungen über die Umsetzung machen; GUI-Gestaltung (evt. Handzeichnungen); Repräsentation von Elementen, Zügen, Einstellungen (Gegner Mensch oder Computer) usw.
Verwendete Umgebung: Java 1.2 / Swing / Java2D
Aufteilung der Gruppe in GUI- und Engine-Team; 2 eigenständige Packages machen
  - Literatur: Müller, Reinhardt: „Neural Networks“

B.3 Protokoll vom 24. März 1999

- **Ort:** Hr. Schwaigers Büro
- **Zeit:** 18^{00}
- **Anwesende:**
  - DI Mag.Roland Schwaiger
  - August Mayer
  - Gerald Wiesbauer
  - Christopher Rettenbacher

  - Manuela Stürzlinger

- **Protokollführer:** Christopher Rettenbacher
- **Themen:**
  - Besprechung der Arbeit der letzten Woche
  - Einleitung zu ANN sollte aus 2 Unterpunkten bestehen;
    1. biologische Motivation
    2. künstl.neuronale Netze(technischer Ansatz)
       * Elemente
       * Funktionsweise der Neuronen
       * Prozesse,Neuron,Gewichte,Lernprozesse
       * Netztypen(TD,Hoppfield,...)

    Einleitung sollte ohne Zusatzinformation verständlich sein;
  - Protokolle sollen direkt vor Literatur gereiht werden
  - Fragestellungen
    * Gibt es automatisch/evolutionär generierte Netze?
    * Wer soll gegen unser Netz spielen?
    * Design der Input Daten?
    * Sind für andere Spiele (Go+Schach) bereits Lösungen vorhanden ?
  - Ziele
    * Objektorientierte Analyse u. Design
      basierend auf Anforderungsanalyse;Klassendiagramme der Packete (GUI,ANN,techn. Package)
    * graphische Oberfläche(Screenshot)
      basierend auf Kundenwünschen; Designansatz beim GUI;Kunde bestätigt Vorabentwurf mittels Unterschrift(bereits Vertragsbestandteil);
    * use case diagramm
      Interaktion user/GUI

C Bibliographie

C.1 Inhaltsangaben

C.1.1 Temporal Difference Learning and TD-Gammon [?]

Dieser Text behandelt das Programm TD-Gammon von Gerald Tesauro, ein „Backgammon-Lern-System“, das den TD(λ)-Algorithmus zum Lernen verwendet. Dabei spielt das Programm mit sich selbst, bewertet bei jedem Zug die aktuelle Spielsituation und modifiziert die Gewichte des Netzwerks zeitlich abgestuft anhand dieses Fehlersignals. Als Input des Netzes verwendet er neben der Spielposition auch einige „handgemachte“ Werte.

Im Vergleich zu Neurogammon, das ein von Experten mit menschlichen Regeln trainiertes neuronales Netz realisiert, aber nur gute Fortgeschrittenen-Spielstärke erreicht, erreicht das Programm Weltklasseniveau und findet sogar einige optimale Züge selbständig. So hat eine von TD-Gammon gefundene Zugkombination die alte, in dieser Spielsituation verwendete bereits abgelöst.

TD-Gammon scheint in diesem Zusammenhang tatsächlich zu lernen, und der Autor beobachtet auch verschiedene Stufen der Wissensaneignung beim Lernvorgang, wie das Lernen von Grundzügen und einfachen Strategien (erste paar tausend Spiele), später von komplexeren Strategien (nach einigen zehntausend Spielen).

Fragen zum Text:

- Was sagt die Gewichtung (weights) aus?

- Wie funktioniert das neuronales Netz von TD-Gammon?
- Wie wirken die -two concept features- zusammen?
- Wie entscheidet TD-Gammon über seinen nächsten Zug?

C.1.2 History and Principles of Neural Networks [?]

Die Theorie der KNN hat ihren Ursprung in einer Arbeit des amerikanischen Psychologen W. James, der Gedanken und Aktionen als Resultat von Spannungsverschiebungen im Gehirn interpretierte und Lernen als Modifikation der Pfade dieser Verschiebungen. In den nächsten Jahren bis 1910 wurde diese Theorie zum Standardmodell erweitert, das besagt, daß einzelne Gehirnzellen (Neuronen) Signale über Dendriten („Eingangsfasern") sammeln und bei Überschreitung einer Schwelle ein Signal abfeuern.

Zwischen 1938 und 1943 entstanden erste neurale Logiksysteme (N. Rashevsky – binäre logische Operationen im Gehirn; W. McCulloch, W. Pitts – „Schwellwert-Logik", „Mehrwertige Logik"; Erfindung der sog. Gewichte, adaptierbare Parametersysteme und Trennung von der mathematischen Logik). In weiterer Folge trennten sich die Neurowissenschaftler in Artificial-Intelligence- (AI-) und KNN-Forscher (etwa 1963); die AI-Forscher erweiterten und abstrahierten die binäre zu pro- und präpositionaler Logik (Entwicklung der Programmiersprache LISP), während die KNN-Forscher weiter parameterbasierte Klassifizierungsmodelle entwickelten.

1949 formulierte Hebb eine Lernregel, die in etwa besagt, daß bei häufigerer Benutzung einer Verbindung zwischen Neuronen die Effizienz der Verbindung zunimmt. Farley und Clark simulierten 1954 erstmals ein solches Netz am M.I.T. auf einem frühen Elektronenrechner, wobei die Neuronen zufällig verbunden waren. 1956 bauten N. Rochester im IBM-Forschungslabor Poughkeepsie, New York, ein rückgekoppeltes Netz, indem sie die Ausgänge von Neuronen wieder zurück ins Netz führten; es zeigten sich hier erstmals Formen von Selbstorganisation. 1958 baute F. Rosenblatt ein selbstlernendes Netz, das er Perceptron nannte und das Regelmäßigkeiten im Eingangsmuster erkannte. Es benutzte Subnetze, die untereinander konkurrieren. Allerdings bauten diese Netze alle auf einer zufälligen Verschaltung der Neuronen, die sich als zu aufwendig und ressourcenverbrauchend herausstellte, sodaß eine natürliche Verwendung wenig wahrscheinlich ist.

B. Widrow und M. Hoff entwickelten 1960 das ADALINE-System, bei dem vor der Summation eine Gewichtung erfolgt. Sie stellten 1962 eine Lernregel auf, nach der aktive Verbindungen mehr zum Fehler beitragen. F. Rosenblatt kombinierte ebenfalls 1962 das ursprüngliche Perceptron mit dem ADALINE-System zum „Klassischen Perceptron". M. Minsky und S. Papert, Vertreter der AI-Richtung, schrieben schließlich 1969 ein Buch über die inhärenten Grenzen der Perceptrons, das zu einer Stillegung der Forschung auf dem Gebiet der KNN in den USA für einige Zeit führte.

J. Anderson und T. Kohonen veröffentlichten 1972 Schriften über assoziative Netze, die zur nächsten Entwicklungsstufe führten. Dabei produzieren mehrere Eingangssignale mehrere analoge Ausgangssignale. Weiter entwickelte K. Fukushima 1975 das Cognitron, das ähnlich dem Perceptron funktionierte, aber bei einigen Problemstellungen durch spezielle inhibitorische Signale besser funktioniert. Dieses Modell erweiterte er 1980 zum Neocognitron, das auch Musterverschiebungen toleriert.

J. Hopfield entwickelte 1982 in den USA einen neuen rückgekoppelten Netztyp, indem er bidirektionale Verbindungen zwischen den Neuronen verwendete, und löste einen neuen Forschungsschub auf dem Gebiet aus. D. Reilly, L. Cooper und Ch. Elbaum entwickelten ebenfalls 1982 sog. hybride Netze, die mehrere Netzebenen kombinierten. Dabei bewirkt die erste Ebene eine Verallgemeinerung der Informationen, die zweite eine Spezialisierung.

Drei unabhängige Forschergruppen entwickelten 1986 den Typ der assoziativen Backpropagation-Netzwerke, die Fehlerwerte auf spezielle Weise im Netzwerk verteilen. Dabei wird angenommen, daß näher am Ausgang liegende Neuronen mehr für den Fehler verantwortlich sind als näher am Eingang liegende. Dieser Typ von Netzen lernt relativ langsam.

1987 entwickelten S. Grossberg und G. Carpenter Netzwerke zur Erkennung von Regelmäßigkeiten, die sog. Adaptive Resonance Networks (ART). Hier wurde versucht, zu erkennen, ob ein Netzteil ein Muster *n*icht erkennt, indem positive Rückkopplung verwendet wird.

1990 schließlich entwickelte D. Olmsted das Mehrwertig-Logische Neurale Netzwerk, das nach dem Vorbild der retikulären Strukturen im Gehirn arbeitet und später zur Mustererkennung verwendet wurde.

C.2 A 'neuronal' Network that learns to play Backgammon[?]

Backgammon ist ein komplexes Brettspiel, das besonders geeignet für die Anwendung Neuronale Netze scheint, da die Zugauswahl in erster Linie auf Basis von Mustererkennung im Gegensatz zur expliziten 'Look-ahead' Methode funktioniert. Dies folgt aus den nicht vorhersehbaren Würfelresultaten, die die Zugmöglichkeiten in einer typischen Spielsituation auf über 400 erhöhen.

Der wichtigste Faktor, der die Effektivität des Netzes entscheidet, war die Methode, die bei der Kodierung der Information angewandt wurde, wobei die Hinzunahme von 'precomputed features' eine signifikante Leistungssteigerung zur Folge haben.

Das hier beschriebene Netz greift auf eine Datenbank zurück, die ca. 3000 ,teilweise bewertete Brettsituationen sowie daraus resultierende optimale Züge [-/+100] enthält. Da einige Spielprinzipien verhältnismäßig oft zum Einsatz kommen, benötigt man einen intelligenten Auswahlmechanismus, um ein 'overlearning' zu verhindern.

Das verwendete NN besteht aus 459 Input-Neuronen(je 18 pro Feld+Bar, sowie einige für pre-computed information) Auf die Hinzunnahme von 'hand-crafted features' wurde bewußt verzichtet, einige einfache Bewertungsmechanismen jedoch hinzugenommen, da es für das Netz zu schwierig scheint in absehbarer Zeit aus der reinen 'Spielsituation' einen Lerneffekt zu erzielen. Der Lernprozeß unterscheidet sich hier jedoch insofern vom 'standard back-propagation learning algorithm', als dem Netz auch nicht bewertete Spielzüge vorgeschlagen werden. Dabei wird -unter der Annahme, daß i.A. ungünstige Szenarien nicht bewertet wurden- ein zufälliges 'teacher signal' [-65/+35]erzeugt.

Das Netzwerk wurde nach 4 Kriterien bewertet:

1. Performance beim Lernprozeß
2. Performance bei dem Netz unbekannten Situationen
3. Performance bei einem Wettkampf gegen ein konventionelles Programm
4. Performance bei einem Wettkampf gegen einen menschlichen Gegner

Das Netzwerk gewinnt in 60% der Fälle gegen Gammontool, ein zu Vergleichszwecken herangezogenes konventionelles Backgammonprogramm. Beim Entfernen der 'precomputed features' fällt die Erfolgsquote auf 41%, ohne die zufällig bewerteten Spielsituationen('noise') auf 45% und ohne die 'hidden units'(Zwischenschichten im NN) auf 50%. Im Gegensatz dazu beeinflußt eine Änderung etwa der Anzahl von 'hidden units', Schichten im NN oder Größe der Datenbank das Ergebnis kaum.

Verstärkt man das NN durch die Bewertungsmechanismen von Gammontool, so wird die Effizienz des Systems erhöht. In der Verbindung von digitaler Technologie und NN scheint die Zukunft der AI zu liegen.

Momentan erreichen NN jedoch nur mittelmäßiges Spielniveau da nur einzelne Spielsituationen bewertet, der daraus entstehende zukünftige Spielverlauf jedoch nicht berücksichtigt wird.

C.3 Bücherliste