

Neuronale Netze der 3. Generation und Anwendungsgebiete

Jan Klass

Angewandte Informatik, Hochschule Offenburg
jklass@stud.fh-offenburg.de

19. Dezember 2010

Abstract

Künstliche neuronale Netze sind den biologischen neuronalen Netzen nachempfunden. Sie sind sehr effizient beim generalisieren und klassifizieren und haben sich als effiziente Lösungsmodelle bewiesen. Die dritte Generation der neuronalen Netze führt eine temporale Informationsübertragung ein und nähert sich damit nochmals den biologischen neuronalen Netzen an. Dieses Paper stellt neuronale Netze im Allgemeinen und die dritte Generation im Speziellen vor. Dabei werden Konzepte und Modelle vorgestellt und sowohl die Schwierigkeiten wie auch die Chancen dargestellt.

1. Einführung

Das menschliche Gehirn ist eine der faszinierendsten, komplexesten und stärksten bekannten Berechnungsmaschinen. Durch das Gehirn konnten sich die Menschen von anderen Lebewesen klar abgrenzen. Durch Wissensdrang und Kollektivität kann der Mensch hoch komplexe Zusammenhänge verstehen und forscht sowohl im Kleinsten (Mikrokosmos; aktuell etwa Quarks oder dunkle Materie) wie auch im Größten (Makrokosmos; etwa dem Weltall). All dies wird ihm durch das Gehirn als Speicher und Berechnungsapparat ermöglicht. Kein Wunder also, dass das Gehirn und das Nervensystem ein sehr wichtiges Forschungsgebiet ist. Das Gehirn und damit die Funktionsweise des Menschen zu verstehen, ist eines der großen Forschungsgebiete unserer Zeit.

Im Hinblick auf die Mächtigkeit der Netze ist jedoch nicht nur das Verstehen interessant sondern vor allem auch das Verwenden dieser zur Problemlösung, zu Berechnungen, zur Klassifizierung, zum Speichern und Lernen. Ein

Modellieren und Erzeugen künstlicher neuronaler Netze also. Diese können nicht nur als Problemlöser verwendet werden, sondern vielmehr auch zum Verständnis biologischer neuronaler Netze. Verschiedene Modelle und Theorien zu den Berechnungsstrategien der Netze können mit relativ wenig Aufwand, gegenüber biologischen Netzen, ausprobiert und erforscht werden. Hierbei entsteht eine enge Wechselwirkung und Symbiose der beiden Forschungsgebiete der biologischen und künstlichen neuronalen Netze. Auch [KRE03] stellt die Notwendigkeit dieser Zusammenarbeit eindeutig fest. Auch in Zukunft sind so noch weitreichende Entwicklungen im Verständnis und der Modellierung zu immer leistungsfähigeren neuronalen Netzen („NN“) zu erwarten.

2. Allgemeine Konzepte

In diesem Abschnitt werden zunächst Grundlagen und Funktionsweisen biologischer neuronaler Netze erläutert, welche zum Verständnis künstlicher neuronaler Netze notwendig sind. Darauf folgend wird die Historie künstlicher neuronaler Netze in der Informatik vorgestellt, genauer die erste und zweite Generation dieser. Anschließend wird die dritte Generation präsentiert und deren Funktionsweise erläutert. Folgend wird erläutert welche Schwierigkeiten es beim Aufbau von NN gibt und wie man diese lösen kann.

2.1 Biologische Grundlagen

Biologische neuronale Netze bestehen aus zahlreichen Nervenzellen, den so genannten Neuronen. Abbildung 1 zeigt modellhaft ein solches Neuron und dessen Verbindungen zu einem weiteren Neuron.

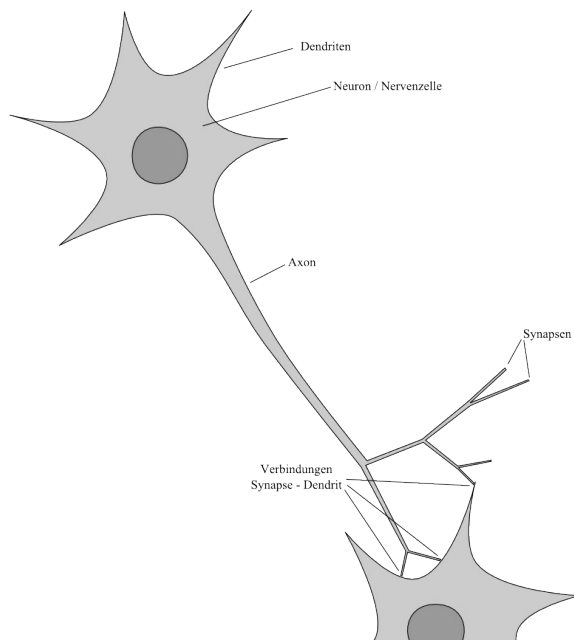


Abbildung 1: Modellhaftes Neuron inklusive Verbindung zu weiterem, benachbartem Neuron.

Das Neuron besteht aus der Soma, den Dendriten und dem Axon. Als Soma wird der Zellkörper ohne Fortsätze bezeichnet. Die Dendriten sind baumartige Fortsätze der Zelle welche synaptische Signale aufnehmen. Das Axon ist ein faserartiger Fortsatz und überträgt Signale elektrischer (Spannungsimpuls) und chemischer Natur. Spannungssignale gehen dabei lediglich von der Soma aus, chemische Stoffe werden sowohl von als auch zur Soma transportiert. Am Ende des Fortsatzes verzweigt sich dieser ebenfalls baumartig und schließt, nicht physisch berührend, an Dendrite an. Dendrit und Axon bilden zusammen eine Synapse, welche Signale chemisch über eine kleine Lücke zwischen den beiden überträgt. Das Axon kann zwischen 1 μm und bis zu 1 m lang sein, ersteres im Gehirn, letzteres im Rückenmark.

Neuronen sind von einer Zellmembran umgeben, die sie von ihrer Umgebung räumlich und elektrisch abgrenzen. An ihr liegt ein elektrisches Potential (Spannung), dem Membranpotential, an. Ein Neuron hat im Ruhezustand, zwischen dem Zellinneren und dem Zelläußern, stets eine negative Spannung.

Steigt das Potential (also der Betrag der Spannung) über eine Schwelle, so „feuert“ das Neuron, das heißt es erzeugt ein Aktionspotential, einen „Spike“. Dieser, vom Zellkörper generierte Spike, breitet sich als Spannungsimpuls entlang

des Axons aus und stößt an den Synapsen chemische Übertragungen an. Diese wirken auf die nachfolgenden, postsynaptischen Neuronen, je nach Synapse und Dendrit, membranpotentialsteigernd oder -schwächend. Man nennt die Stärkung auch Hyperpolarisation und die Schwächung Depolarisation. Siehe hierzu auch Abbildung 2, welche die Wirkung eingehender Signale auf das Membranpotential beispielhaft darstellt.

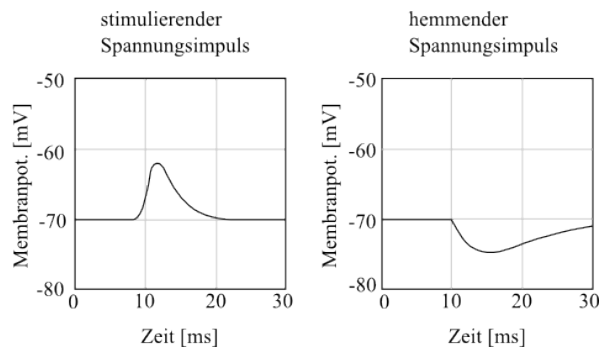


Abbildung 2: Impulsverlauf beim Eingehen von postsynaptischen Potentialen; de- bzw. hyperpolarisierend auf Membranpotential wirkend

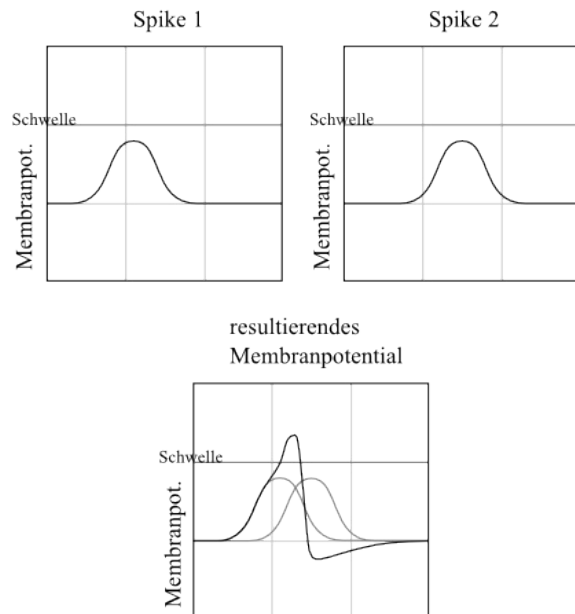


Abbildung 3: Veranschaulichung einer Schwellenüberschreitung durch Kombination zweier Spikes

Überschreitet das Membranpotential im nachfolgenden, postsynaptischen Neuron die Aktivierungsschwelle, so feuert auch dieses Neuron. Abbildung 3 zeigt eine solche Schwellenüberschreitung. Die einzelnen Spikes führen nicht zum Feuern. Treffen sie jedoch zeitlich nahe genug zusammen ein, so wird die

Schwelle überschritten und das Neuron feuert. Im Bild, wie auch in biologischen NN und je nach Modell auch künstlichen, ist das Neuron nach dem Feuern in einer kurzen Inaktivitätsphase, in welcher es nicht feuern kann (daher das starke Abfallen des resultierenden Membranpotentials trotz des noch eingehenden Spikes).

Die verschiedenen Dendrit-Synapsen haben unterschiedlich starken Einfluss auf die Membranpotentialänderung. Das heißt, sie sind gewichtet.

Die Neuronen handeln also autonom und reagieren lediglich auf ankommende Spikes. Gegebenenfalls leiten sie Spikes über das Axon und den diesem folgenden Synapsen weiter.

2.2 Historie künstlicher NN

Bereits 1943 stellten McCulloch und Pitts ein erstes Neuronen-Modell vor, welches Berechnungen durchführen konnte. Neuronen dieses Modells haben eine Binärschwelle, weshalb das Modell nur mit binären Signalen kommuniziert. Dies stellt eine starke Vereinfachung biologischer neuronaler Netze dar. Trotzdem können bereits mit diesen Netzen Berechnungen durchgeführt werden. Diese neuronalen Netze, welche mit binären Signalen kommunizieren, gehören der ersten Generation neuronaler Netze an.

Die Signale werden von den einzelnen Neuronen empfangen und gewichtet. Die Höhe der Gewichtung ist abhängig davon, von welcher eingehenden Verbindung die Signale empfangen werden. Die gewichteten Signale werden dann aufsummiert und anschließend wird für diesen aufsummierten Betrag das Ausgabesignal des Neurons mittels einer Aktivierungsfunktion berechnet. Bei den NN der ersten Generation ist das typischerweise eine Schwellwertfunktion (auch „Heaviside-“, Sprung-, oder Treppenfunktion genannt), siehe hierzu Abbildung 4.

Der aufsummierte Betrag ankommender Signale entspricht dem Aktivierungspotential von Neuronen. Der Zustand eines Neurons hängt also wesentlich von dessen Aktivierungsgrad ab. Je nach Definition des neuronalen Netzes kann die Aktivierungsenergie in einem Neuron mit der Zeit abfallen.

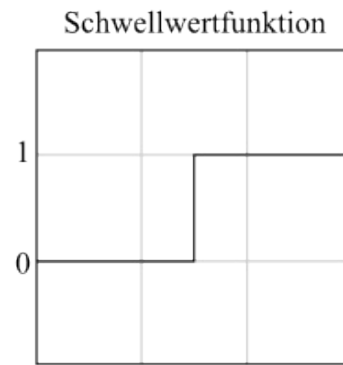


Abbildung 4: typische Schwellwertfunktion

Ein einfaches Beispiel für ein NN der ersten Generation ist das Perzeptron-Netzwerk. Abbildung 5 zeigt ein solches für das Beispiel einer AND-Berechnung zweier Eingabewerte.

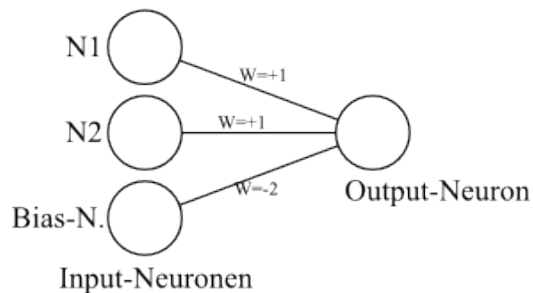


Abbildung 5: Einfaches Perzeptron Netzwerk für eine AND-Berechnung der Input-Werte

Im Bild ist, neben den beiden Eingabe-Neuronen N1 und N2, auch ein Bias-Neuron zu sehen. Bias-Neuronen sind konstant aktiv und negativ gewichtet, um einen Aktivierungsschwellenwert des nachfolgenden Neurons festzulegen. Eine Sonderbehandlung der Schwelle entfällt somit und das Bias-Neuron wird wie die anderen Neuronen mit eingerechnet. Die Höhe der Schwelle wird durch den Gewichtsbeitrag der gewichteten Verbindung des Bias-Neurons bestimmt.

Solche einfachen Perzeptron-Netzwerke mit zwei Schichten können nur lineare Probleme lösen. Es ist möglich sie auf mehrere Ebenen zu erweitern, wodurch sie auch komplexere Problemstellungen (als simples Beispiel etwa eine XOR-Funktion, welche nicht linear ist) lösen.

Auf der ersten Generation aufbauend, und unter Berücksichtigung weiterer biologisch-physikalischer Erkenntnisse aus der Erforschung realer neuronaler Netze, verfolgten die Modelle der zweiten Generation künstlicher NN einen

anderen Ansatz. Sie kamen in den 90er Jahren auf. Die Signale zwischen Neuronen wurden nicht mehr als Binärübertragung (Signal oder kein Signal) von Informationen gesehen, sondern vielmehr ging man davon aus, dass Informationen in der Anzahl an Signalen pro Sekunde, also in der Feuerrate der Impulse, kodiert sind. In den computerberechenbaren Modellen wurden nun nicht mehr Binärwerte verwendet, sondern reelle Zahlen die die Feuerrate repräsentieren. Berechnet werden diese, wie schon bei NN der ersten Generation, mittels der Aktivierungsfunktion, etwa einer Sigmoidfunktion („S-Funktion“).

2.3 Die dritte Generation

Wie [NAT00] erklärt, konnte aber durch weitere Forschungen nachgewiesen werden, dass Informationen in biologischen neuronalen Netzen weitaus schneller verarbeitet werden können, als dies mit einem künstlichen neuronalen Netz der zweiten Generation möglich ist. Dies ist etwa beim Verarbeiten visueller Reize und deren weiterer Verarbeitung beim Menschen der Fall. [NAT00] erläutert außerdem, dass nachgewiesen wurde, dass auch einzelne Spannungsimpulse für die Neuronen wichtige oder ausschlaggebende Informationen enthalten können. Es musste also ein neues oder erweitertes Modell gefunden werden, welches auch einzelne Neuronensignale beachtet und verarbeiten kann.

Diese neuen Erkenntnisse stehen jedoch nicht zwangsläufig im Gegensatz zu den bisherigen Modellen. Es gibt verschiedene Typen von biologischen neuronalen Netzen und je nach Typ scheinen, das zeigen Forschungsergebnisse, auch anders zu rechnen und zu kommunizieren, also anders zu funktionieren.

In neuronalen Netzen der dritten Generation werden Informationen in die Zeitabstände zwischen den Spikes, also zeitlich, kodiert. Ein künstliches neuronales Netz der dritten Generation nennt man auch *spiking neural network* („SNN“).

Die modellhafte Werteberechnung findet jeweils mit der Zeitdifferenz zwischen der Signalfeuerung und einem festen Zeitpunkt statt. Dieser feste Zeitpunkt kann ein definierter, voriger Impuls sein, oder ein von außen kommunizierter Zeitpunkt.

Der Output eines Spiking-Neuronen besteht, gegenüber vorigen Generationen, nicht mehr aus einem Wert, sondern vielmehr aus der Menge aller Zeitpunkte, zu denen ein Impuls abgefeuert wird.

Das einfachste Schema einer temporalen Kodierung analoger Größen ergibt sich über Formel 1.

$$x_{temporal} = T - c \cdot x$$

Formel 1

T ist eine Konstante, unabhängig von der analogen Größe. Sie gibt beispielsweise den Zeitpunkt eines bestimmten Stimulus an. Der temporale Wert x_t besteht dabei aus der Zeitdifferenz des reellen Wertes, welcher gewichtet ist.

Diese Formel kann sowohl zur Ent-/Kodierung des In- wie auch des Outputs verwendet werden.

Folgend wird ein konkretes mathematisches Modell für ein SNN vorgestellt.

Ein Neuron v feuert, wenn das Aktivierungspotential $P_v(t)$ eine bestimmte Schwelle erreicht.

Der Einfluss eines ankommenden Impulses auf das Neuron lässt sich mittels der *response*-Funktion berechnen, siehe Formel 2.

$$\epsilon_{v,u}(t-s)$$

Formel 2

Das dem Neuron v vorangestellte Neuron u generiert zur Zeit s einen Impuls.

Diese ankommende Information muss noch gewichtet werden. Dies wird mittels der nun zeitabhängigen Gewichtungsfunktion w , siehe Formel 3, umgesetzt.

$$w_{v,u}(t)$$

Formel 3

Trotz der Möglichkeit, das Gewicht der über diese Verbindung ankommenden Signale anpassen zu können, wird dies nur selten getan. Das Gewicht für eine Verbindung bleibt über die Zeit meist konstant, jedenfalls wenn das Netz nicht trainiert wird. Beim Trainieren der Netze lernen die NN meist durch anpassen der Gewichte. Dies hängt jedoch vom verwendeten Modell und Lernalgorithmus ab.

Aus der gewichteten Responsefunktion ergibt sich die Potentialänderung im Neuron, also den Einfluss des Signals auf das Membranpotential im Neuron. Zur Gewichtung werden die beiden Funktionen multipliziert.

$$\Delta_{\text{Potentialeinfluss auf } v} = w_{v,u} \cdot \epsilon_{v,u}(t-s)$$

Formel 4

Daraus ergibt sich für das Potential eines Neurons v als Berechnungsformel:

$$P_v(t) = \sum_u \sum_{\text{Feuerzeiten } s \text{ von } u} w_{v,u} \cdot \epsilon_{v,u}(t-s)$$

Formel 5

u ist die Menge präsynaptischer Neuronen, also solcher vorhergehender Neuronen, die mit v verbunden sind.

2.4 Entwurf von NN

Das Erstellen neuronaler Netze ist eine der Schwierigkeiten bei der Arbeit mit ihnen. Fasst man sie zu groß, das heißt, hat das Netz zu viele Neuronen und/oder Verbindungen, dann generalisiert das Netz nicht. Ist es dagegen zu klein, so sättigt das Netz und verlernt anfänglich gelerntes wieder.

Einen Ansatz ein Netz aufzubauen, oder zu optimieren, stellt [ADH09] vor. Der Algorithmus nimmt sich als Vorbild das menschliche Auge, welches Helligkeitseingabewerte in Spikes binarisiert, damit diese anschließend vom zentralen Nervensystem verarbeitet werden können. Das Kodierungsverfahren in Binärvektoren ist dabei verlustbehaftet, „verliert“ aber nur die unwichtigen, nicht differenzierenden Eingabeparameter. Der Algorithmus kann Eingabewerte vollautomatisch umwandeln. Er kann außerdem wichtige Datenunterschiede betonen und damit eine bessere Generalisierung begünstigen.

Der Algorithmus soll also die Weiterverarbeitung vereinfachen. Ein resultierendes Netz verarbeitet unter Umständen bestimmte Eingabeparameter gar nicht, weil diese den Datensatz nicht von anderen Datensätzen abgrenzen. [ADH09] liefert für 3 Beispiele auch die resultierenden Netze.

Eine andere Art neuronale Netze aufzubauen ist diese auf einer Grundform aufzubauen. Eine solche Grundform ist das kreisförmige Anordnen der Neuronen. Wie [ASC09] erläutert wurde

bereits in der Vergangenheit dargestellt, dass ein Verbinden der Neuronen zu den jeweils nächsten Nachbarn von Nachteil gegenüber einem vom Abstand abhängigen Zufall ist. Dies ist jedenfalls für (netz-)globale Berechnungen wie einen Assoziativspeicher, welcher eine Mustererkennung voraussetzt, der Fall. Eine Verbindung zu einem näheren Neuron sind damit wahrscheinlicher, jedoch nicht zwingend. Dies war trotz der höheren Laufzeiten durch längere Verbindungen, welche im Modell ebenfalls beachtet wurden, der Fall. Ein aus dieser Erkenntnis gefolgertes Modell ist das „small world“ – also „kleine Welt“ – Modell. Dieses verbindet die Neuronen zu den jeweils am nächsten liegenden Neuronen, fügt jedoch einige zufällige Querverstrebungen über den gesamten Ring ein.

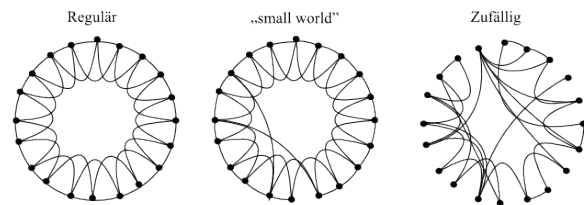


Abbildung 6: Ringnetze; nach rechts sind die Netze mit steigender Zufälligkeit aufgebaut

Zur Konstruktion eines „small world“-Netzes stellt [DUS98] eine Standardmethode vor, auf welche auch [ASC09] verweist. Dabei wird das Ringnetz zunächst regulär verbunden, jedes Neuron nur zu seinen näheren Nachbarn. Anschließend werden Verbindungen zu anderen, zufällig gewählten, Neuronen neu verbunden.

Als Lernalgorithmus kämen für ein solches Netz zunächst kovariante Gewichte, wie sie in einem standard Hopfield-Netzwerk verwendet werden, in Frage. Dieser Lernalgorithmus ist für zerstreute und nicht symmetrische Netze aber relativ ungeeignet. [ASC09] schlägt deshalb vor das Lernverfahren eines standard Perzeptron-Netzes zu verwenden. Hierbei starten alle Neuronenverbindungen mit einem Gewicht von 0, ankommende Signale werden also ignoriert. Die Gewichte werden dann anschließend in mehreren Durchläufen, mittels einer Trainingsmenge an Eingabesätzen, angepasst.

Unter anderem [ASC09] stellt fest: Performance-Vergleiche zwischen „small world“-Netzen und Hopfield-Netz-Assoziativspeichern haben gezeigt, dass „small world“-Netze mit

zunehmender Zufälligkeit der Verbindungen zwischen Neuronen gegenüber den Hopfield-Netzen mehr Kapazität besitzen. Zunehmende Kapazität heißt, sie können mehr Muster speichern.

Als weitere Erkenntnis lässt sich feststellen, dass eine Gauss-verteilte Zufälligkeit der Verbindungen eher zu größeren Kapazitäten führt, als reine „small world“ Netze mit linearer Zufälligkeit. Bezüglich der Ruhezeiten von Neuronen, also der Zeit in der ein Neuron nach dem Feuern eines Spikes nicht mehr feuern kann, lässt sich feststellen: (Höhere) Ruhezeiten haben relativ geringen (aber negativen) Einfluss auf die Kapazität.

2.5. Trainieren von NN

In den Abschnitten 2.3 Die dritte Generation und 2.5. Trainieren von NN wurde bereits kurz auf das Trainieren künstlicher NN eingegangen. In ersterem Abschnitt wurde das Lernen über Verbindungsgewichtsanpassung erwähnt, in letzterem wurde das Lernen in bestimmten, konkreten Modellen, vorgestellt. In diesem Abschnitt soll das Trainieren und Lernen nochmals vertieft und verallgemeinert werden.

Es wurde bereits erwähnt, dass das Lernen neuronaler Netze meist durch die Anpassung der verbindungsspezifischen Gewichte stattfindet. Die Netze bekommen hierfür bei ihrer Erzeugung eine feste Struktur, das heißt die Neuronen und ihre Verbindungen werden definiert. Dies ist vor allem in der ersten und zweiten Generation von NN der Fall. Mit einer Trainingsmenge an Datensätzen wird das Netz dann mit einem bestimmten Lernalgorithmus trainiert. Das Netz bekommt einen Datensatz als Eingabe und der Lernalgorithmus passt die Gewichte der Verbindungen an. Je nach Lernalgorithmus findet dies nur in einer Ebene, etwa der Ausgabebene, oder in mehreren Ebenen statt.

Mit der dritten Generation und neueren Forschungsergebnissen gibt es Modelle die NN nicht mehr nach einem festen Modell aus regelmäßigen Verbindungen aufbauen, sondern etwa alle Neuronen zufällig verbinden, oder in einem Kreis anordnen. In dieses Netz

Vor allem durch die zeitliche Komponente in der dritten Generation, im Zusammenhang mit Rückkopplung innerhalb von Netzen, wird das

interpretieren der Ausgaben eines Netzes immer wichtiger. Man geht hier in die Richtung das eigentliche Netz nicht mehr zu trainieren sondern nur noch eine Ausgabeschicht aus Neuronen zu trainieren, die lernen soll, wie sie aus den zahlreichen und zeitlich parallelen Informationen die richtigen zur Ausgabe auswählt. Mit zeitlich parallelen Informationen ist hier gemeint, dass Informationen über mehrere Eingaben von Datensätzen hinweg Informationen zu eben diesen mehreren Eingabedatensätzen haben. Dies entspricht auch der Funktionsweise biologischer NN. Am Beispiel des Sehens gibt es Forschungsergebnisse die zeigen, dass beim sequentiellen Betrachten von Ziffern auch beim Betrachten der dritten Ziffer noch Informationen zu den ersten beiden Ziffern im Gehirn zu finden sind (vgl. [VOG10]).

Das Lernen beschränkt sich also keineswegs nur auf das Anpassen von Gewichten, wie es in den ersten beiden Generationen durchgeführt wurde, sondern erlaubt viel Variation und Interpretation in den Theorien und Lernalgorithmen.

Häufig wird eine anfänglich vorhandene Menge an Datensätzen in Trainingsmenge und Testmenge geteilt um mittels der Testmenge nach dem Training prüfen zu können, wie gut das Netz klassifiziert, das heißt wie gut es gelernt hat. Dies ist vor allem bei der Evaluierung und Erforschung verschiedener Modelle und Lernalgorithmen der Fall.

3. Anwendungsgebiete

In diesem Kapitel wird zunächst auf die Anwendungsgebiete im Allgemeinen eingegangen. Anschließend werden konkrete Anwendungen beispielhaft vorgestellt.

Die Modelle neuronaler Netze der ersten und zweiten Generation sind bereits seit längerem im produktiven Einsatz und anerkannt. Entsprechende Entwicklungsbibliotheken und Programme sind ebenfalls verfügbar. Jene Modelle der dritten Generation nehmen erst seit den letzten Jahren an praktischer Bedeutung zu.

1997 wies Maass nach, dass Probleme und Lösungen, welche durch neuronale Netze der zweiten Generation repräsentiert werden können, auch in solchen der dritten Generation lösbar sind. Das heißt, beliebige Funktionen können

auch in neuronalen Netzen der dritten Generation modelliert, beziehungsweise gelöst werden. Weiterhin wurde nachgewiesen, dass bei bestimmten Problemstellungen wesentlich einfachere und effizientere (kleineres Netzwerk) Lösungen mit neuronalen Netzen der dritten Generation möglich sind, als mit der zweiten Generation. Dies bedeutet auch, dass neuronale Netze der dritten Generation für alle Anwendungsgebiete der ersten und zweiten Generation in Frage kommen.

Zu den Anwendungsgebieten neuronaler Netze gehören vor allem die Mustererkennung, Klassifizierung und Problemstellungen mit notwendiger Generalisierung. Die Mustererkennung und Klassifizierung kann dabei in Bildern, oder auch in anderen Daten durchgeführt werden. Erkannte Muster sollen dann gegebenenfalls klassifiziert werden. Zu Problemstellungen welche eine Generalisierung voraussetzen gehören beispielsweise solche, bei denen auf Basis einer Trainingsmenge (Beispieldaten) in anderen Fällen ähnlich geschlussfolgert werden soll. Vorher nicht bekannte Eingaben führen dann also zu einer Entscheidung aufgrund gelernter Daten. Dies gehört zum Bereich der künstlichen Intelligenz.

Bei der Lösungsfindung und Berechnung sind neuronale Netze außerdem sehr tolerant gegenüber Rauschen, eben wegen ihrer guten Generalisierung.

1997 wurde von Maass und Natschläger ein Modell für das assoziative Speichern von Information in neuronalen Netzen der dritten Generation vorgestellt. 1995 stellte Hopfield ein SNN-Modell zur Mustererkennung vor.

Aufgrund des andersartigen Modells zum Binär-Computer gibt es auch Ansätze, Hardware für eben diese SNNs zu optimieren beziehungsweise zu erstellen, wie etwa [NAT00] erläutert. Der EPSILON II-Chip beispielsweise arbeitet zur Kommunikation mit analogen Signalen, mit Spannungs-Impulsen, und den Zeiten der individuellen Impulsen.

3.1 konkrete Anwendungen

Die praktischen Anwendungsfelder rekurrenter, spikender Neuronen sind durch ihre komplexen Lernalgorithmen stark begrenzt. [RSN08] untersucht die Anwendung des Reservoir

Computing-Konzeptes auf rekurrente neuronale Netze mit dem Ziel der Effizienzsteigerung und Vereinfachung.

Beim Reservoir Computing wird ausschließlich mit den Eingaben und einer Ausgabebene des Netzes gearbeitet. Das Reservoir, also der Teil des Netzes zwischen Eingabeneuronen und Ausgabeneuronen, bleibt unbekannt, oder zumindest unangetastet. Das Reservoir ist meist ein zufälliges neuronales Netz, oder eines bei dem die Neuronen zufällig miteinander verbunden sind. Einfluss nimmt man auf das Ergebnis lediglich über die Verbindungen zu den Ausgabeneuronen, welche beeinflusst und geändert, also trainiert, werden können. Das zuvor erwähnte, sonst sehr komplexe Lernen rekurrenter NN, entfällt somit und es muss nur noch die Ausgabebene trainiert werden.

[RSN08] wendet dieses Konzept konkret auf Spracherkennung an. Dabei wird die Performanz, wie gut das Netz klassifiziert, evaluiert. Hierzu werden die Daten vor Eingabe in das Reservoir vorbereitet und die Ausgabeneuronen auf verschiedene Weisen an das Netz angebunden.

Weitere Anwendungen werden regelmäßig veröffentlicht. [VID09] stellt etwa vor, wie ein wachsendes, kompetitives neuronales Netz zur Objektüberwachung in Videosequenzen verwendet werden kann. [SRE08] stellt ein neuronales Netzwerk vor, welches aus Graustufenbildern und einem 3x3 Pixel Helligkeitsverteilungsblock eine 3D Szene wieder herstellt. Hierzu wird berechnet aus welcher Richtung das Licht kommt, woraufhin die Objekte ausgemacht und ihre dreidimensionale Form wiederhergestellt werden kann.

Zur Lösung des NP-vollständigen „Shortest Common Superstring“-Problems stellt [SCS09] ein neuronales Netz vor, welches nach experimentellen Versuchen schneller als andere bekannte, heuristische Methoden ist.

[SOL10] stellt ein NN der dritten Generation zur Lokalisation von Geräuschen vor. Ein Ansatz zur Geräuschlokalisation, wie er auch bei Lebewesen zu finden ist. Dabei wird das Geräusch an zwei Stellen empfangen (biol.: 2 Ohren) und die zeitlich verschieden ankommenden Signale werden im neuronalen Netz verarbeitet, woraus dann eine Position ausgelesen wird.

4. Zusammenfassung

Neuronale Netze der dritten Generation sind eine Weiterführung der Modelle der vorigen Generationen. Sie sind näher an biologische neuronale Netze angelehnt.

Aufgrund der Kodierung von Informationen in Zeitdifferenzen erfordert das Modell ein gewisses Umdenken. Es macht die ohnehin sehr undurchsichtigen Berechnungen in neuronalen Netzen nicht verständlicher. Eventuelles Vorwissen über das Problem kann auch nur über Trainingsdaten ein geübt, nicht aber vorgegeben werden.

Neben dem Umdenken beim Verstehen der Funktionsweisen ist auch die Implementierung und Umsetzung solcher Netze eine Herausforderung. Die massiv parallel arbeitenden Netze, die aus zahlreichen Neuronen bestehen, stehen dem Computer mit seiner zentralen Recheneinheit gegenüber. Hier dürfte der Wechsel zu Computern mit immer mehr CPU-Kernen bei der Berechnung von Vorteil sein. Darüber hinaus gibt es, wie in diesem Paper ebenfalls dargestellt wurde, Bestreben, die Strukturen und Verbindungen neuronaler Netze in Hardware zu gießen. Dies könnte einen weiteren Sprung bei der Verwendung von und Berechnung mit neuronalen Netzen bedeuten, vor allem aber eine höhere Effizienz der Hardware.

Neuronale Netze der dritten Generation kodieren Informationen nur anders, die dem Netz zugrunde liegenden Mechanismen bleiben gegenüber den vorigen Generationen jedoch gleich. Deshalb lassen sich sowohl neue Modelle (wie das Reservoir-Computing) auf die alten Generationen übertragen, wie auch umgekehrt Modelle und Lernregeln der vorigen Generationen auf die dritte Generation übertragen. Die neue Generation ist also keineswegs eine Revolution, sondern eine Evolution. Von neuen Entwicklungen in der einen Generation können, unter Umständen, auch die anderen Generationen profitieren.

Neuronale Netze aller Generationen sind, für passende Problemstellungen, ein sehr gutes Mittel zur Problemlösung. Sie generalisieren sehr gut, erkennen Muster und können auf verschiedene Weisen lernen. Ein paar Beispiele solcher Problemstellungen, wie Spracherkennung oder Videoanalyse, wurden in diesem Paper

ebenfalls vorgestellt.

Vorausblickend kann man davon ausgehen, dass die neuronalen Netze auch weiterhin verbessert und noch näher an biologische neuronale Netze angelehnt werden. Eine Berechnung ist noch in viel kleinerem Detail möglich, etwa indem man die Ionenkanäle in den Neuronen in das Modell mit einbezieht. Ob sich eine solche Verfeinerung in absehbarer oder auch fernerer Zukunft als Problemlösung eignet, muss sich erst noch zeigen. Die biologisch genauesten Modelle scheinen hierfür momentan noch zu komplex. Mit Neuronen in so einem Detailgrad zu rechnen scheint zu aufwändig. Für die interdisziplinäre Erforschung biologischer Netze und der Vorgänge in ihnen, bleiben künstliche Netze dieser Modelle jedoch unersetzlich.

Zusätzlich zur Annäherung an die biologischen Netze, was mit steigender Komplexität einher geht, gibt es auch weiterhin Bestreben, die neuen Modelle wiederum zu vereinfachen und performanter zu gestalten. In jüngster Vergangenheit wurde dies durch die Liquid State Machines und das Reservoir Computing erreicht. Ein Ende von neuen und interessanten Ansätzen und Modellen scheint nicht in Sicht.

5. Literatur

- [NAT00] Thomas Natschläger:
Netzwerke von „spiking“ Neuronen: Die dritte Generation von Modellen für Neuronale Netze
2000
Institute for Theoretical Computer Science, Technische Universität Graz
- [GDA09] Samanwoy Ghosh-Dastidar, Hojjat Adeli:
Third Generation Neural Networks: Spiking Neural Networks
Springer, 2009
- [KRE03] Jutta Kretzberg (Universität Oldenburg):
Informatik in den Neurowissenschaften FIFF Kommunikation 1/2003,
Themenheft „Bioinformatik“
- [SOL10] Brendan Glackin, Julie A. Wall, Thomas M. McGinnity*, Liam P. Maguire and Liam J. McDaid:
A spiking neural network model of the medial superior olive using spike timing dependent plasticity for sound localization
Frontiers in Computational Neuroscience

- , 2010
- [ASC09] Weiliang Chen, Reinoud Maex, Rod Adams, Volker Steuber, Lee Calcraft, Neil Davey:
Connection Strategies in Associative Memory Models with Spiking and Non-spiking Neurons
Springer, 2009
- [DUS98] Watts, D., Strogatz, S.:
Collective Dynamics of „small-world” networks
Nature 393, 1998
- [RSN08] Arfan Ghani, T. Martin McGinnity, Liam P. Maguire, and Jim Harkin:
Neuro-inspired Speech Recognition with Recurrent Spiking Neurons
Springer, 2008
- [ACS09] Pilar Caamaño, Jose Antonio Becerra, Francisco Bellas, Richard J. Duro:
Using Spiking Neural Networks for the Generation of Coordinated Action Sequences in Robots
Springer, 2009
- [VID09] J.M. Ortiz-de-Lazcano-Lobato, R. M. Luque, D. López-Rodríguez, E.J. Palomo:
Growing Competitive Network for Tracking Objects in Video Sequences
Springer, 2009
- [SRE08] David Elizondo, Shang-Ming Zhou, Charalambos Chrysostomou:
Surface Reconstruction Techniques Using Neural Networks to Recover Noisy 3D Scenes
Springer, 2008
- [SCS09] D. López-Rodríguez and E. Mérida-Casermeyro:
Shortest Common Superstring Problem with Discrete Neural Networks
Springer, 2009
- [ADH09] Adrian Horzyk:
Automatic Discriminative Lossy Binary Conversion of Redundant Real Training Data Inputs for Simplifying an Input Data Space and Data Representation
Springer, 2009
AGH University of Science and Technology
- [VOG10] Vorbild Gehirn,
Die Presse – forschung, 2010
- [EXP99] Fred Rieke, David Warland, Rob de Ruyter van Steveninck and William

Bialek:
Spikes – Exploring the Neural Code
MIT Press, 1999