Implementation Neuronaler Netze mittels Digit Online Algorithmen

Marc Haase, Andreas Wassatsch, Dirk Timmermann

1. Einleitung

Neuronale Netze finden ihre Anwendung bei der Implementierung von Steuer- und Regelprozessen deren Aufgabe bzw. Verhalten sich nur schwer durch einen Algorithmus darstellen, aber sehr gut an Hand von Verhaltensbeispielen beschreiben lässt. Neben der bezüglich der Speicherung der notwendigen Gewichte kritischen analogen Implementierungen finden auch digitale Realisierungen ihre Anwendung. Diese beschränken sich jedoch auf Grund der Komplexität der resultierenden Schaltung auf die Realisierung der für die Funktion des Netzes notwendigen Haupt-Komponenten, wie Verbindungsnetzwerk, der Eingangsbewertung und Stimuli-Summierung [1]. Ein weiteres Handicap der meist parallelen digitalen Realisierung ist der Aufwand für die Verbindung der einzelnen Netzelemente. Daher verfolgen wir einen seriellen Ansatz wodurch der Aufwand für die Verbindungen (interconnections) erheblich reduziert wird. Eine Verschachtelung der Teiloperationen ist nur mittels Most Signifikant Digit (MSD) first Algorithmen wie dem verwendeten Digit-Online Verfahren möglich. Hiermit kann zugleich auch eine Realisierung eines gesamten digitalen Neuronalen Netzes in Form eines eigenständigen universellen Neuro-Prozessors erreicht werden. Das Design des vorgestellten Prozessors ist bezüglich der Netzparameter der Feed-Forward Architektur parametrisierbar. Als Lernverfahren wurde On-Chip das Backpropagation Lernverfahren in seiner Offline-Variante implementiert.

In diesem Beitrag wird das implementierte Neuronale Netze kurz vorgestellt. Des weiteren wird die eingesetzte Digit Online Arithmetik erläutert und die Realisation des Neuronalen Netzes mittels dieser Arithmetik beschrieben. Die Architektur des entwickelten Neuro-Prozessors wird im Anschluss daran präsentiert.

2. Neuronale Netze

Neuronale Netze sind informationsverarbeitende Systeme, die aus einfachen Verarbeitungselementen, den Neuronen, aufgebaut sind. Sie sind Nachbildungen von biologischen Strukturen, wie sie sich zum Beispiel beim menschlichen Gehirn wiederfinden lassen. Dabei handelt es sich jedoch um eine starke Abstraktion vom biologischen Vorbild. Die einzelnen Verarbeitungselemente, die Neuronen, sind über gerichtete Verbindungen miteinander verbunden. Jede Verbindung ist durch einen Gewichtsfaktor in ihrer Stärke beeinflussbar. Die Gesamtheit der Neuronen steht für eine komplexe mathematische Funktion, die sich im Gegensatz zu regelbasierten Systemen nicht in Formeln ausdrücken lässt. Über ein Lernverfahren wird diese Funktion dem Netz eingeprägt.

Elementarer Bestandteil und Grundverarbeitungselement eines Neuronalen Netzes ist das Neuron. Es summiert die am Eingang präsentierten und gewichteten Aktivitäten auf und ermittelt über eine Ausgangsfunktion seine eigene Aktivierung. Die Weiterleitung der Aktivierungen erfolgt entweder nur in eine Richtung (Feed-Forward-Netz), oder auch zurück auf den Eingang des Neurons (Rückgekoppeltes Netz). Im folgenden wird auf die Implementierung einer Feed-Forward-Architektur eingegangen.

In Feed-Forward-Netzwerken sind die Neuronen einer Schicht mit allen Neuronen der folgenden Schicht verbunden. Die Anzahl der Verbindungen entspricht dem Produkt aus der Anzahl der Neuronen benachbarter Schichten. Bei der parallelen digitalen Implementierung dieses Netzes ist jede Verbindung entsprechend der gewählten Zahlenbasis der zu übertragenden Aktivierungen *n*-Bit breit. Die Anzahl der Verbindungen erhöht sich dadurch noch einmal um den Faktor *n* und führt zu einem vom Wertebereich abhängigen Verbindungsaufwand zwischen zwei Schichten von Neuronen. Diese Abhängigkeit von der Bitbreite *n* lässt sich durch Serialisierung der zu übertragenden Aktivierungen eliminieren. Die Anzahl der Verbindungen zwischen zwei Schichten von Neuronen reduziert sich dadurch auf ein Minimum [2].

Durch den Einsatz von serieller Arithmetik ist es möglich Daten, die in serieller Form vorliegen zu verarbeiten, ohne eine vorherige Parallelisierung vorzunehmen. Im folgenden Abschnitt wird diese Arithmetik vorgestellt.

3. Serielle Arithmetik

Klassische mathematische Algorithmen der Datentechnik verarbeiten Daten parallel. Vorteil dieser Verfahren ist die relativ geringe Latenzzeit auf Grund der gleichzeitig möglichen Berechnung von Teilaufgaben. Voraussetzung ist die Unabhängigkeit der Teilaufgaben voneinander, da sich ansonsten die Latenzzeit vergrößert. Deswegen ist dies nur bei einfachen Operationen möglich. Bei komplexeren Operationen stößt die parallele Verarbeitung an ihre Grenzen und man geht auf iterative oder serielle Datenverarbeitung über. Darüber hinaus sind parallele Algorithmen von der Datenbreite der Daten abhängig, und beanspruchen mit zunehmender Bitbreite auch mehr Chipfläche. Die Busstrukturen in diesen Systemen beanspruchen ebenfalls einen nicht unerheblichen Anteil an Chipfläche.

Serielle Berechnungsoperationen sind durch den im Vergleich zu einer parallelen Schaltungslösung äußerst geringen Schaltungsaufwand charakterisiert. Durch die zeitlich versetzte Abarbeitung von nahezu identischen Teilaufgaben ist eine gemeinsame Verwendung von Schaltungsstrukturen möglich. Ein weiterer Vorteil ist, dass durch effiziente Verkettung von Grundoperationen zu einer komplexen Funktion, im Vergleich zu einer parallelen Realisierung eine kürzere Latenzzeit erzielt werden kann und eine geringere Chipfläche beansprucht wird.

Ein Beispiel für serielle Datenverarbeitungsalgorithmen ist Digit Online. Die Daten werden hier mit dem Most Signifikant Digit (MSD) beginnend verarbeitet. Nach einem von der zu berechnenden Operation abhängigen Online-Delay wird das Ergebnis wieder seriell ausgegeben. Im Gegensatz zur Last Signifikant Digit First Methoden können bei MSDF auch kom-

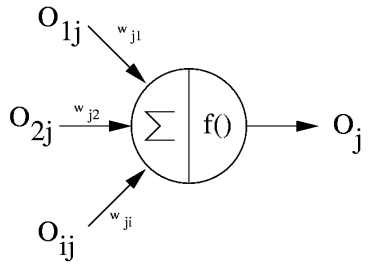

Abbildung 1 Schematischer Aufbau eines Neurons

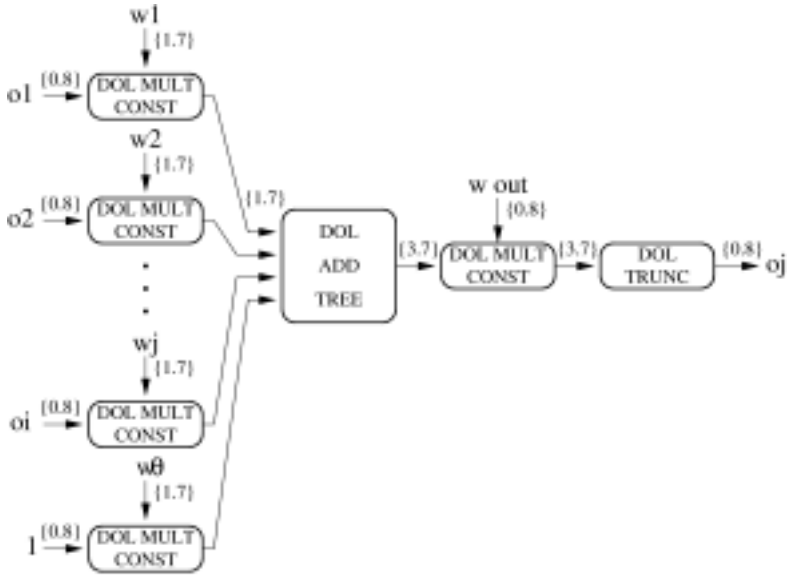

Abbildung 2 Realisierung des Neurons aus Digit Online Modulen

plexere Funktionen (Exponenzierung, Division) realisiert werden, die für die Implementierung des Neuronalen Netzes Voraussetzung sind.

4. Architektur des Neuro-Prozessors

Das implementierte Neuronale Signalverarbeitungssystem besteht aus einem zweischichtigen Neuronalen Netz, mit konfigurierbarer Anzahl von Neuronen, dem Backpropagation Lernverfahren, einem internen Gewichtsspeicher und einem Steuerwerk, das die selbstständige Durchführung des Lernverfahrens ermöglicht.

Für die Beschreibung des Systems wurde die Hardwarebeschreibungssprache VHDL verwendet [3]. Der Prozessor ist hinsichtlich Eingänge, Ausgänge, Anzahl der Neuronen, Bitbreite der Gewichte und Daten parametrisierbar. Dies wurde durch generische Beschreibung des Systems gewährleistet. Dadurch ist der Prozessor für unterschiedliche Netzwerkanforderungen, beschränkt auf Feed-Forward Architekturen, implementierbar. Der Aufbau des Prozessors ist modular. Dazu wurden die zu Grunde liegenden Digit Online Operationen mit einem zusätzlich erforderlichen Synchronisierungsalgorithmus [2] erweitert, der die Verkettung von einzelnen DOL Operationen ohne zusätzlich zu implementierende Steuerungssignale ermöglicht. Ein einkommender Datenstrom wird vom Synchronisierungsalgorithmus erkannt, initialisiert die DOL Operation und startet die Abarbeitung. Nach der Operation wird das DOL Modul zurückgesetzt.

Das Neuronale Netz als Teilkomponente des Neuro-Prozessors wurde aus einzelnen Neuronen (Abbildung 2) hierarchisch aufgebaut. Neben dem Neuronalen Netz wurde auch der Lernalgorithmus mittels DOL Modulen implementiert. Dieser lässt sich über Parameter an die Netzwerkstruktur generisch anpassen. Die Speicherung der Gewichte erfolgt in einem mit auf dem Prozessor integrierten Speicherbereich.

Auf Grund des hohen Simulationsaufwandes zur Verifikation des Systems wurde das Hardware-Emulationssystem APTIX MVP3 eingesetzt [4]. Das zu simulierende System wird hier auf frei programmierbare Hardwarebausteine (FPGA) abgebildet. Dadurch lässt sich die Simulation in Echtzeit durchführen und viel kürzere Simulationszeiten sind erreichbar.

Die Abbildung 3 zeigt den schematischen Aufbau des entwickelten Neuro-Prozessors [3]. Durch die Integration eines Steuerwerkes besitzt der Prozessor die Möglichkeit selbstständig eine gegebene Aufgabe mittels Lernvektoren, die über ein Interface vom externen Speicher eingelesen werden, zu erlernen und anschließend zu erfüllen. Damit der Prozessor in einer Umgebung mit paralleler Datenverarbeitung eingesetzt werden kann, findet die Parallel-Seriell- bzw. Seriell-Parallel-Wandlung im Prozessor statt. Dadurch sind zusätzliche Datenstromkonvertierungen außerhalb des Prozessors nicht erforderlich.

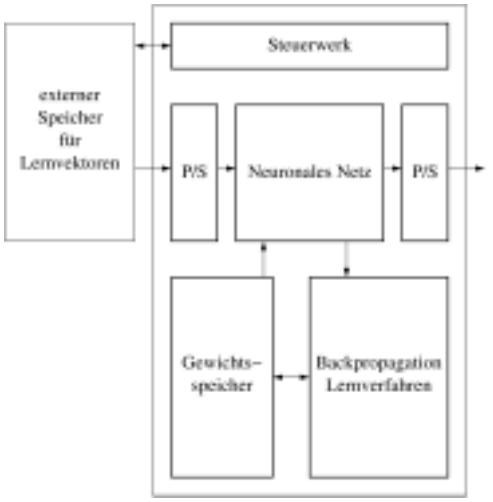

Abbildung 3 Schematischer Aufbau des Neuro-Prozessors

5. Zusammenfassung

Neuronale Netze lassen sich mit Hilfe von digitalen Signalverarbeitungsalgorithmen implementieren. Dabei konnte gezeigt werden, dass durch den Einsatz von seriellen Algorithmen der bei parallelen Implementierungen entstehende hohe Hardwareaufwand für Verbindungen zwischen benachbarten Neuronenschichten auf ein Minimum reduziert werden kann. Des weiteren lässt sich durch Koppelung von aufeinanderfolgenden Operationen eine bezüglich der Taktzyklen kürzere Latenzzeit erreichen. Der ausschließlich mittels seriellen Datenverarbeitungsalgorithmen implementierte Neuro-Prozessor ist auf Grund des integrierten Backpropagation-Lernverfahrens in der Lage, selbstständig eine Aufgabe zu lernen. Dadurch lässt sich der Prozessor als Stand-Alone Prozessor flexibel einsetzen.

6. Literatur

[1] U. Ramacher; Synapse- A Neurocomputer That Sythesizes Neural Algorithms on a Parallel Systolic Engine; Journal of Parallel and Distributed Computing, Vol. 14, p. 306-318;1992

[2] Wassatsch, A.; Haase, M.; Timmermann, D.; DOLFIN - Digit Online For Integrating Neural Networks; The IEEE International Symposium on Circuits and Systems (ISCAS´2000), Geneva, Switzerland, ISBN: 0-7803-5485-0, S. III-602 - 605, Mai 2000

[3] Marc Haase, Diplomarbeit: Untersuchung komplexer digitaler Signalverarbeitungsarchitekturen auf Eignung zur Abbildung auf eine sequentielle Zellbibliothek, Universität Rostock, 2000

[4] Wassatsch, A.; Haase, M.; Timmermann, D.; The DOLFIN Project: An application report on a consistent design and verification flow for a large digital neural network.; SNUGE´2000, Paris (France), S. A2.2.1-9, März 2000

7. Verfasser

Dipl.-Ing. Marc Haase, Dipl.-Ing. Andreas Wassatsch
Universität Rostock, Fachbereich Elektrotechnik und Informationstechnik
Institut für Angewandte Mikroelektronik und Datentechnik
Richard-Wagner Str. 31, 18119 Rostock-Warnemünde
e-mail: {marc.haase, andreas.wassatsch}@etechnik.uni-rostock.de