(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
07. Mai 2020 (07.05.2020)

WIPO | PCT

(10) Internationale Veröffentlichungsnummer
WO 2020/089287 A1

(51) Internationale Patentklassifikation:
G06N 3/08 (2006.01) *G06N 3/04* (2006.01)

(21) Internationales Aktenzeichen: PCT/EP2019/079645

(22) Internationales Anmeldedatum:
30. Oktober 2019 (30.10.2019)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2018 127 383.5
02. November 2018 (02.11.2018) DE

(71) Anmelder: **UNIVERSITÄT BREMEN** [DE/DE]; Bibliothekstraße 1, 28359 Bremen (DE).

(72) Erfinder: **PAWELZIK, Klaus**; Bei den drei Pfählen 9, 28205 Bremen (DE). **ROTERMUND, David**; Senator-Fritze-Str. 10, 28213 Bremen (DE).

(74) Anwalt: **V. BEZOLD & PARTNER PATENTANWÄLTE - PARTG MBB**; Akademiestraße 7, 80799 München (DE).

(81) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, RU, TJ, TM), europäisches (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Veröffentlicht:
— *mit internationalem Recherchenbericht (Artikel 21 Absatz 3)*

(54) Title: DATA-PROCESSING DEVICE HAVING AN ARTIFICIAL NEURAL NETWORK AND METHOD FOR PROCESSING DATA

(54) Bezeichnung: DATENVERARBEITUNGSVORRICHTUNG MIT EINEM KÜNSTLICHEN NEURONALEN NETZWERK UND VERFAHREN ZUR DATENVERARBEITUNG

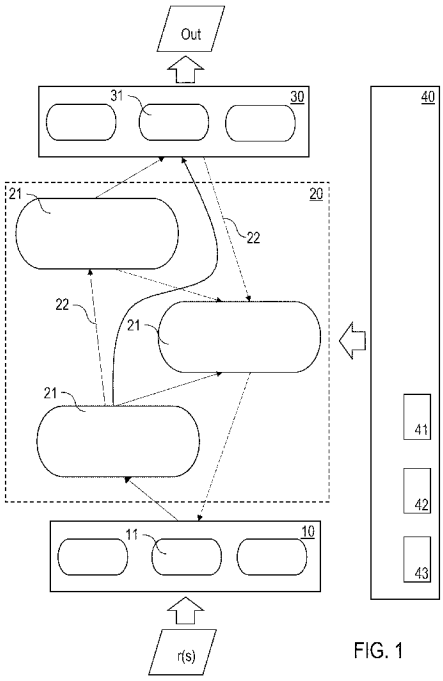


FIG. 1

(57) Abstract: The invention relates to a data-processing device (100) for mapping input data onto output data comprising: an input apparatus (10) for receiving the input data r(s), producing an input probability distribution p (s) of the input data and producing a stochastic representation of the input data as a sequence of input indices, each representing a stochastically determined position within the input probability distribution; at least one network (20) of at least two feature modules (21), which are coupled to one another by means of data connections (22), one of the feature modules being coupled to the input apparatus, each feature module comprising: state-variable memory locations each having weight storage locations; index storage locations; and a computing unit.

(57) Zusammenfassung: Eine Datenverarbeitungsvorrichtung (100) zur Abbildung von Eingabedaten auf Ausgabedaten umfasst eine (Eingabe-Einrichtung (10) zur Aufnahme der Eingabedaten r(s), Erzeugung einer Eingabe- Wahrscheinlichkeitsverteilung p (s) der Eingabedaten und Erzeugung einer stochastischen Repräsentation der Eingabedaten als Sequenz von Eingabe-Indizes, die jeweils eine stochastisch ermittelte Position innerhalb der Eingabe-Wahrscheinlichkeitsverteilung repräsentieren; mindestens ein Netzwerk (20) aus mindestens zwei (Merkmalsmodulen (21), die über Datenverbindungen (22) miteinander gekoppelt sind, wobei eines der Merkmalsmodule mit der Eingabe-Einrichtung gekoppelt ist, wobei jedes Merkmalsmodul Zustandsvariabien- Speicherplätze jeweils mit Gewichte-Speicherplätzen, Index- Speicherplätze und eine Recheneinheit umfasst.

WO 2020/089287 A1

Datenverarbeitungsvorrichtung mit einem künstlichen neuronalen Netzwerk und Verfahren zur Datenverarbeitung

Gebiet der Erfindung

Die Erfindung betrifft eine Datenverarbeitungsvorrichtung, die zur Abbildung von Eingabedaten auf Ausgabedaten konfiguriert ist, insbesondere eine Datenverarbeitungsvorrichtung mit einem künstlichen neuronalen Netzwerk. Des Weiteren betrifft die Erfindung ein Datenverarbeitungsverfahren, bei dem Eingabedaten auf Ausgabedaten abgebildet werden, insbesondere ein Datenverarbeitungsverfahren unter Verwendung eines künstlichen neuronalen Netzwerks. Anwendungen der Erfindung sind in allen Gebieten der Datenverarbeitung, insbesondere basierend auf maschinellem Lernen und/oder künstlicher Intelligenz, z. B. bei der Auswertung von Daten und/oder der datenbasierten Steuerung von Vorrichtungen oder Verfahren gegeben.

Technischer Hintergrund

In der vorliegenden Beschreibung wird auf den folgenden Stand der Technik Bezug genommen, der den technischen Hintergrund der Erfindung illustriert:

[1] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117;

[2] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386;

[3] Guo, Yanming, et al. "Deep learning for visual understanding: A review." *Neurocomputing* 187 (2016): 27-48;

[4] Azkarate Saiz, A. (2015). Deep learning review and its applications;

[5] Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters, 42, 11-24;

[6] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489;

[7] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529;

[8] Gatys, L. A., Ecker, A. S., & Bethge, M. (2016, June). Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (pp. 2414-2423). IEEE;

[9] WO 2017/021322A1;

[10] Lacey, G., Taylor, G. W., & Areibi, S. (2016). Deep learning on FPGAs: Past, present, and future. *arXiv preprint arXiv:1602.04283*;

[11] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533;

[12] US 2007/192267A1;

[13] US8219507B2;

[14] Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In *Neural networks: Tricks of the trade* (pp. 599-619). Springer, Berlin, Heidelberg;

[15] Salakhutdinov, R. and Hinton G. E. (2012) An efficient learning procedure for deep Boltzmann machines. Neural Comput. Aug; 24(8):1967-2006. doi: 10.1162/NECO_a_00311

[16] Salakhutdinov, R. (2015). Learning deep generative models. *Annual Review of Statistics and Its Application*, *2*, 361-385;

[17] Ernst, U., Rotermund, D., & Pawelzik, K. (2007). Efficient computation based on stochastic spikes. *Neural computation*, *19*(5), 1313-1343;

[18] Spanne, A., & Jörntell, H. (2015). Questioning the role of sparse coding in the brain. *Trends in neurosciences*, *38*(7), 417-427;

[19] Bruckstein, A. M., Elad, M., & Zibulevsky, M. (2008). On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Transactions on Information Theory*, *54*(11), 4813-4820;

[20] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980;

[21] Rolínek, M., & Martius, G. (2018). L4: Practical loss-based stepsize adaptation for deep learning. arXiv preprint arXiv:1802.05074; und

[22] Angeline, P. J., Saunders, G. M., & Pollack, J. B. (1994). An evolutionary algorithm that constructs recurrent neural networks. IEEE Transactions on Neural Networks, 5(1), 54-65).

Es ist allgemein bekannt, dass tiefe neuronale Netzwerke [1] eine wichtige Basis vieler Systeme mit Anwendungen künstlicher Intelligenz bilden. Tiefe neuronale Netzwerke bestehen aus einer

Eingabeschicht, einer Vielzahl von verborgenen Zwischenschichten und einer Ausgabeschicht. Die Schichten sind typischerweise miteinander über gewichtete Verbindungen gekoppelt. Meistens erfolgt die Verarbeitung von Daten gerichtet, indem Signale von der Eingabeschicht sukzessive über die Zwischenschichten bis zur Ausgangschicht propagiert werden (Vorwärtsrichtung). Während diese Netzwerke theoretisch beliebige Funktionen erfüllen können, ist die Anpassung von Verbindungsgewichten der Verbindungen zwischen den Schichten ("Lernen") aufgrund der erforderlichen Rechenleistung und/oder Hardware-Anforderungen eine der zentralen Herausforderungen herkömmlicher neuronaler Netzwerke.

Zum Lernen werden für tiefe neuronale Netzwerke bisher Algorithmen verwendet, die in der Ausgangsschicht auftretende Fehler in Rückwärtsrichtung über die Schichten propagieren (z. B. der so genannte "Backpropagation-Algorithmus") und so den Beitrag der Verbindungsgewichte auch in vorangehenden Schichten zum Fehler minimieren können (Lösung des "credit assignment problem"). Diese Algorithmen zeichnen sich dadurch aus, dass Fehlersignale modifiziert überall im Netzwerk zur Verfügung stehen müssen. Die Verbindungsgewichte müssen gemeinsam für das gesamte Netzwerk gelernt werden (nichtlokales Lernen). Dies wirkt sich nachteilig auf den Rechenaufwand und die Geschwindigkeit des Lernens aus und beschränkt die Zahl der in der Praxis realisierbaren verborgenen Schichten und die Geschwindigkeit der Anwendung des Netzwerks.

Ein weiterer Nachteil von herkömmlichen tiefen neuronalen Netzwerken besteht darin, dass meist nur dann nützliche Lösungen erreicht werden, wenn die Anzahl der freien Parameter im System stark beschränkt werden kann. Dies ist z. B. bei den "Deep Convolutional Networks" (DCN) der Fall, die sich in vielen Anwendungen bewährt haben ([2] bis [5]). Sie reichen von dem Spielen von Brett- und Computer-Spielen ([6], [7]) bis in die Kunst ([8], [9]). Weitere Anwendungen sind z. B. beim Betrieb autonom fahrender Fahrzeuge, bei der Bildverarbeitung, insbesondere Gesichtserkennung, oder Bildverarbeitung für medizinische Anwendungen (z. B. Tumor-Erkennung auf MRT/CT Aufnahmen), beim Erkennen von betrügerischen Bank-Transaktionen, bei der Analyse des Verhalten von Wählern und deren Manipulation (siehe z.B. https://www.nvidia.de/deep-learning-ai/) gegeben. Bisher wird versucht, das aufwendige Lernen herkömmlicher Netzwerke zu beschleunigen, indem sie als Hardware-Lösung implementiert werden (siehe z.B. [10]).

Neben den tiefen neuronalen Netzwerken gibt es weitere Typen von neuronalen Netzwerken, die allerdings weniger erfolgreich sind. Ein Beispiel sind rekurrente Netzwerke, bei denen die verborgene Schicht aus einer Ansammlung von Neuronen besteht, die im Gegensatz zu den typischen

vorwärts gerichteten Netzwerken auch untereinander verbunden sind. Das Lernen dieses Netzwerk-Typs ist notorisch schwierig [11]. Weitere Beispiele sind durch die so genannten Hierarchical Temporal Memory (HTM) Systeme [12, 13] gegeben.

Weitere Netzwerktypen sind rückgekoppelte Netzwerke vom Typ der so genannten Boltzmann Maschinen ([14], [15]) oder Netzwerke, die auf generativen Modellen [16] basieren. Zu Letzteren zählt das in [17] beschriebene vorwärts gerichtete Netzwerk, bei dem die Information vollständig mit stochastisch erzeugten Impulsen (oder: elektrische Aktionspotentiale, Indizes oder "Spikes" genannt) übertragen werden und alle Gewichte und Aktivitäten nur positive Werte annehmen dürfen.

Es erwies sich als vorteilhaft, dass das Netzwerk gemäß [17] rauschrobust ist und fehlende Eingabedaten ergänzen kann. Die Anwendbarkeit des Netzwerks blieb jedoch auf eine einzige verborgene Schicht oder die Kombination mit einem herkömmlichen hierarchischen neuronalen Netzwerk beschränkt, obwohl ein großes Interesse an einer Erweiterung auf mehrere verborgene Schichten bestand. Es war jedoch noch kein Verfahren bekannt, wie die für die Erweiterung auf mehrere verborgene Schichten notwendigen Gewichte gelernt werden können. Des Weiteren war das Netzwerk ein rein vorwärts gerichtetes Netzwerk. Im Ergebnis war das Netzwerk gemäß [17] insbesondere aufgrund seiner spärlichen Aktivität vorrangig von theoretischem Interesse, da sich Beziehungen zur so genannten Sparseness bei der Informationsverarbeitung im Gehirn [18] und zu "Compressed Sensing"-Verfahren ergaben [19]. Für praktische Anwendungen war das Netzwerk jedoch nur beschränkt geeignet.

Aufgabe der Erfindung

Die Aufgabe der Erfindung ist es, eine verbesserte Datenverarbeitungsvorrichtung und/oder ein verbessertes Datenverarbeitungsverfahren bereitzustellen, mit denen Nachteile herkömmlicher Techniken vermieden werden. Die Datenverarbeitung soll unter Verwendung eines Netzwerks erfolgen, dessen Eigenschaften für eine Abbildung von Eingabedaten auf Ausgabedaten trainierbar sind. Die Datenverarbeitungsvorrichtung und/oder das Datenverarbeitungsverfahren sollen insbesondere die Probleme herkömmlicher Netzwerke verringern, die beim Lernen und dem Rückwärtsfluss von Information auftreten. Das Netzwerk soll geeignet sein, eine Datenverarbeitung mit einer erhöhten Anzahl von Netzwerkknoten, insbesondere mit einer erhöhten Anzahl von Schichten, und/oder einer erhöhten Geschwindigkeit zu realisieren, insbesondere wenn Anwendungen mit erhöhter Komplexität gegeben sind. Des Weiteren soll das Netzwerk geeignet sein,

insbesondere für Anwendungen mit geringer Komplexität, eine Datenverarbeitung mit einer verringerten Anzahl von Netzwerkknoten, insbesondere mit einer verringerten Anzahl von Schichten, und/oder einer erhöhten Geschwindigkeit ohne Beeinträchtigung der Funktion des Netzwerks zu realisieren. Das Netzwerk soll ferner schaltungstechnisch vereinfacht, insbesondere mit einem verringerten Aufwand an Hardware-Komponenten realisierbar sein.

Zusammenfassung der Erfindung

Diese Aufgaben werden durch eine Datenverarbeitungsvorrichtung und/oder ein Datenverarbeitungsverfahren mit den Merkmalen der unabhängigen Ansprüche gelöst. Vorteilhafte Ausführungsformen und Anwendungen der Erfindung ergeben sich aus den abhängigen Ansprüchen.

Gemäß einem ersten allgemeinen Gesichtspunkt der Erfindung wird die o. g. Aufgabe durch eine Datenverarbeitungsvorrichtung (oder: Neuronales Netzwerk zur Informationsverarbeitung) gelöst, die zur Abbildung von Eingabedaten auf Ausgabedaten konfiguriert ist. Mit Eingabedaten r(s) werden allgemein vorgegebene Daten bezeichnet, die Informationen repräsentieren und mit der Datenverarbeitungsvorrichtung verarbeitet werden sollen. Die Eingabedaten sind die Daten, aus denen die Ausgabedaten ermittelt werden sollen. Die Eingabedaten liegen z. B. als Ergebnis von Messungen, Beobachtungen, Informationssammlungen oder numerischen Verfahren vor. Typischerweise umfassen die Eingabedaten numerische Werte r, die als ein- oder mehrdimensionale Anordnung (Positionen s) an einem Eingang der Datenverarbeitungsvorrichtung aufgenommen werden. Die Informationen sind in den Eingabedaten als Muster enthalten, die z. B. quantitative Werte der Eingabedaten und/oder eine Eigenschaft der Anordnung der Eingabedaten umfassen. Mit Ausgabedaten werden allgemein Daten bezeichnet, die Informationen repräsentieren und mit der Datenverarbeitungsvorrichtung aus den Eingabedaten erzeugt werden sollen. Mit Datenverarbeitung wird die maschinengestützte (computergestützte) Erzeugung der Ausgabedaten aus den Eingabedaten bezeichnet. Die Datenverarbeitung schließt in Abhängigkeit von der konkreten Anwendung der Erfindung insbesondere eine Analyse der Eingabedaten, eine Mustererkennung in den Eingabedaten, eine Merkmalserkennung in den Eingabedaten, eine Quantifizierung von Merkmalen, welche durch die Eingabedaten repräsentiert werden, und/oder eine Prognose der zukünftigen Veränderung des Systems, welches die Eingabedaten erzeugt, ein. Allgemein werden die Ausgabedaten durch eine vielparametrige Funktion, welche durch die Datenverarbeitungsvorrichtung, insbesondere ein in dieser enthaltenes Netzwerk, repräsentiert wird, aus den vorgegebenen Eingabedaten ermittelt oder diesen zugeordnet (Abbildung von Eingabedaten auf Ausgabe-

daten). Die vielparametrige Funktion wird in einem Lernmodus der Datenverarbeitungsvorrichtung mit bekannten Eingabedaten-Ausgabedaten-Paaren ermittelt und in einem Anwendungsmodus der Datenverarbeitungsvorrichtung auf vorgegebene Eingabedaten angewendet, um zugehörige Ausgabedaten zu ermitteln.

Die Datenverarbeitungsvorrichtung umfasst eine Eingabe-Einrichtung, die zur Aufnahme der Eingabedaten r(s), Erzeugung einer Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten und Erzeugung einer stochastischen Repräsentation der Eingabedaten in Gestalt einer Sequenz von Eingabe-Indizes eingerichtet ist. Jeder Eingabe-Index (oder: Spike) repräsentiert eine stochastisch ermittelte Position innerhalb der Eingabe-Wahrscheinlichkeitsverteilung p*(s). Die Eingabe-Indizes sind beispielsweise die stochastisch ermittelten Positionen an sich oder davon abgeleitete Zahlen.

Die Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten umfasst eine Folge von Wahrscheinlichkeiten der einzelnen Eingabedaten (z. B. berechnet als die Werte der Eingabedaten, jeweils dividiert durch die Summe der Eingabedaten). Die Sequenz von Eingabe-Indizes wird durch stochastische Auswahl von Werten aus der Eingabe-Wahrscheinlichkeitsverteilung, z. B. die Vorgabe einer Sequenz von Zufallszahlen zwischen 0 und 1, z. B. durch einen Zufallsgenerator, und die Ermittlung der Positionen in der Eingabe-Wahrscheinlichkeitsverteilung, deren summierte Wahrscheinlichkeiten aller vorherigen Positionen an die jeweilige Zufallszahl angepasst ist, gebildet. Die stochastische Auswahl wird im Folgenden auch als "ziehen" der Indizes bezeichnet.

Im Ergebnis erzeugt die Eingabe-Einrichtung die Sequenz der Eingabe-Indizes als Folge von Zahlen, welche eine stochastische Repräsentation der in den Eingabedaten enthaltenen Informationen darstellt. Die Eingabe-Einrichtung umfasst z. B. eine Schaltung (Eingabe-Schaltung oder Eingabe-Schaltkreis) mit Speichern und einer Recheneinheit, die für die Erzeugung der Sequenz von Eingabe-Indizes eingerichtet ist. Des Weiteren kann die Recheneinheit Komponenten enthalten, die für die Steuerung von Speichervorgängen in der Eingabe-Einrichtung und/oder des Sendens der Eingabe-Indizes eingerichtet sind.

Die Datenverarbeitungsvorrichtung umfasst des Weiteren mindestens ein Netzwerk aus mindestens zwei Merkmalsmodulen, die über Datenverbindungen miteinander gekoppelt sind, wobei mindestens eines der Merkmalsmodule über eine Datenverbindung mit der Eingabe-Einrichtung gekoppelt ist. Die Eingabe-Einrichtung ist zum Senden der Sequenz von Eingabe-Indizes an das mindestens eine mit der Eingabe-Einrichtung gekoppelte Merkmalsmodul eingerichtet. Jedes

Merkmalsmodul (auch als Normalisierungseinheit bezeichnet) ist zum Senden einer Sequenz von Modul-Indizes an die mit diesem Merkmalsmodul gekoppelten Merkmalsmodule eingerichtet. Das Senden der Sequenz von Eingabe- oder Modul-Indizes umfasst eine Übertragung der Indizes über die Datenverbindung. Welche Merkmalsmodule miteinander gekoppelt sind, d. h. welche Datenverbindungen bestehen, wird durch die Konfiguration des Netzwerks bestimmt. Die Datenverbindungen werden, insbesondere in Abhängigkeit von der konkreten Anwendung der Datenverarbeitungsvorrichtung, z. B. basierend auf Erfahrungswerten vorgegeben, und sie können bei Betrieb der Datenverarbeitungsvorrichtung modifiziert werden. Jedes Merkmalsmodul umfasst, wie im Folgenden erläutert wird, eine Schaltung (Merkmalsmodul-Schaltung oder Merkmalsmodul-Schaltkreis) mit Speichern und einer Recheneinheit. Die Zahl der erfindungsgemäß verwendeten Merkmalsmodule beträgt zusätzlich zu den Eingabe- und Ausgabe-Einrichtungen mindestens zwei, kann aber für praktische Anwendungen erheblich größer sein.

Die Datenverbindungen von Merkmalsmodulen mit anderen Merkmalsmodulen und/oder der Eingabe-Einrichtung umfassen vorzugsweise Verbindungen über einen Datenbus oder einen drahtlosen Übertragungskanal. Die Verbindungen werden z. B. Listen-basiert implementiert. Im Rahmen einer Netzwerkkonfiguration werden Listen gebildet, die festlegen, von welchen Merkmalsmodulen und/oder der Eingabe-Einrichtung an welche anderen Merkmalsmodule Eingabe- oder Modul-Indizes übertragen werden. Die zwischen den Merkmalsmodulen übertragenen Informationen zur erfindungsgemäßen Datenverarbeitungsvorrichtung umfassen vorzugsweise ausschließlich die Eingabe- und Modul-Indizes. Die Merkmalsmodule bilden Knoten des Netzwerks, an denen die durch die Eingabe- und Modul-Indizes repräsentierten Informationen verarbeitet werden. Obwohl das Netzwerk keine einzelnen Elemente enthält, die Neuronen mit einer Konfiguration wie bei einem herkömmlichen neuronalen Netzwerk sind, wird das Netzwerk aufgrund seiner Funktion hier auch als neuronales Netzwerk bezeichnet.

Das Netzwerk wird allgemein aus den Merkmalsmodulen gebildet, deren Anzahl mindestens 2 beträgt, bei praktischen Anwendungen jedoch vorzugsweise im Bereich von 10 bis 10000 gewählt ist, aber auch darüber liegen kann. Anders als bei herkömmlichen geschichteten Netzwerken sind die Merkmalsmodule im Netzwerk unabhängig vom Abstand von den Eingabe- und Ausgabe-Einrichtungen und der Reihenfolge in Vorwärts- oder Rückwärtsrichtung frei kombinierbar, d. h. das Netzwerk hat nicht notwendigerweise eine Schichtstruktur. Es besteht jedoch die Möglichkeit, Merkmalsmodule zu Schichten zusammenzufassen, um die Architektur eines geschichteten Netzwerks nachzubilden. Wenn die erfindungsgemäße Datenverarbeitungsvorrichtung ein geschichte-

tes Netzwerk mit mehreren Merkmalsmodulen pro Schicht aufweist, unterscheidet sich dieses jedoch von den herkömmlichen Netzwerken, die jeweils eine durchgängige Schicht aus einer Vielzahl von Neuronen haben.

Jedes Merkmalsmodul enthält eine Vielzahl von Speicherplätzen, von denen eine Gruppe von Speicherplätzen als Zustandsvariablen-Speicherplätze und eine weitere Gruppe von Speicherplätzen als Index-Speicherplätze bezeichnet werden. Die Zustandsvariablen-Speicherplätze speichern normierte latente Zustandsvariablen h_i. Jedem Zustandsvariablen-Speicherplatz ist eine Vielzahl von Gewichte-Speicherplätzen zugeordnet. Die Gewichte-Speicherplätze speichern Gewichte p(s|i), die den latenten Zustandsvariablen h_i zugeordnet sind. Die Index-Speicherplätze jedes Merkmalsmoduls speichern empfangene Eingabe- oder Modul-Indizes oder von einer Ausgabe-Einrichtung empfangene Ausgabe-Indizes.

Die Merkmalsmodule sind jeweils zum Senden (Übertragen) einer zugehörigen Sequenz von Modul-Indizes an die mit dem jeweiligen Merkmalsmodul gekoppelten anderen Merkmalsmodule eingerichtet. Jeder Modul-Index repräsentiert, basierend auf einer Merkmalsmodul-Wahrscheinlichkeitsverteilung h_i des jeweils sendenden Merkmalsmoduls, eine stochastisch ermittelte Position eines Zustandsvariablen-Speicherplatzes. Die Modul-Indizes werden vorzugsweise erzeugt, wie oben für die Eingabe-Indizes und mit weiteren Einzelheiten unten unter Bezug auf die Figuren erläutert ist. Vorzugsweise umfassen die Modul-Indizes die stochastisch ermittelte Folge von Position der Zustandsvariablen-Speicherplätze.

Jedes Merkmalsmodul enthält des Weiteren die Recheneinheit, die zur Optimierung von den Zustandsvariablen h_i und optional von deren Gewichten p(s|i) in Abhängigkeit von Eingabe- oder Modul-Indizes konfiguriert ist. Des Weiteren umfasst die Recheneinheit Komponenten, die für die Steuerung der Speichervorgänge an den Speicherplätzen des jeweiligen Merkmalsmoduls, die Berechnung der Modul-Indizes, und/oder die Steuerung des Sendens und Empfanges von Eingabe- und/oder Modul-Indizes eingerichtet sind.

Im Einzelnen ist die Recheneinheit jedes Merkmalsmoduls dazu eingerichtet, die latenten Zustandsvariablen h_i und die zugeordneten Gewichte p(s|i), die in dem Merkmalsmodul gespeichert sind, durch Anwendung einer numerischen Optimierung basierend auf den Werten in den Index-Speicherplätzen schrittweise so zu verändern, dass für alle direkt miteinander gekoppelten Merkmalsmodule wechselweise die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des einen Merkmalsmoduls an die latenten Zustandsvariablen h_s des jeweils anderen Merkmalsmoduls

angepasst ist und die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des mindestens einen mit der Eingabe-Einrichtung gekoppelten Merkmalsmoduls an die Eingabe-Wahrscheinlichkeitsverteilung p*(s) angepasst ist.

Die Datenverarbeitungsvorrichtung umfasst des Weiteren eine Ausgabe-Einrichtung, die mit mindestens einem Merkmalsmodul des Netzwerks über eine Datenverbindung gekoppelt und zur Bereitstellung der Ausgabedaten eingerichtet ist. Die Ausgabe-Einrichtung umfasst wiederum z. B. eine Schaltung (Ausgabe-Schaltung oder Ausgabe-Schaltkreis) mit Speichern und einer Recheneinheit, die für die Erzeugung der Ausgabedaten basierend auf der Sequenz von empfangenen Modul-Indizes eingerichtet ist. Des Weiteren kann die Recheneinheit Komponenten enthalten, die für die Steuerung von Speichervorgängen in der Ausgabe-Einrichtung und/oder der Verarbeitung der Modul-Indizes und optional für die Erzeugung und rückwärts gerichtete Übertragung von Modul-Indizes (so genannte Ausgabe-Indizes) an die mit der Ausgabe-Einrichtung gekoppelten Merkmalsmodule eingerichtet sind.

Die Architektur des Netzwerkes ist dadurch bestimmt, welche der Eingabe-Einrichtung, der Ausgabe-Einrichtung und der Merkmalsmodule als Quellen für Indizes für das empfangende Merkmalsmodul bzw. die Ausgabe-Einrichtung dienen, wo dann die empfangenen Indizes für die Aktualisierung der latenten Zustandsvariablen und optional für das Lernen der Gewichte verwendet werden. Die Gewichte eines Moduls beschreiben in lokaler Weise, wie das empfangende Modul seine latenten Zustandsvariablen, gegeben einen empfangenen Index und den unten beschriebenen Gleichungen (insbesondere Regeln zur Aktualisierung der latenten Zustandsvariablen), zu aktualisieren hat. Die Gewichte in einem Modul entsprechen einer zweidimensionalen Matrix an Speicherplätzen mit positiven Werten, wo diese Werte im reellen Wertebereich zwischen 0 und 1 aufweisen.

Jedem Zustandsvariablen-Speicherplatz für die latenten Zustandsvariablen sind auf den zugehörigen Gewichte-Speicherplätzen die Gewichte zugeordnet, die als ein Vektor an Gewichten betrachtet werden können. Die Länge des Vektors entspricht der Summe über die Anzahl der Speicherplätze für latente Zustandsvariablen über alle sendenden Komponenten, umfassend die Eingabe-Einrichtung, die Ausgabe-Einrichtung und die Merkmalsmodule, auf die, entsprechend der Architektur des Netzwerks, dieses empfangende Modul reagieren soll. In einer optionalen Ausführungsform ist die Summe über die Speicherwerte des gesamten Vektors auf einen festen Wert normiert (z.B. den Wert 1). Alternativ kann es vorteilhaft sein, dass die Summe über die Speicherwerte von Abschnitten dieses Gewichtsvektors, insbesondere diejenigen, welche gemeinsam zu

einem sendenden Eingangs-, Ausgangs-, und Merkmalsmodul gehören, auf einen festen Wert normiert (z.B. den Wert 1) sind.

Die Gewichte stellen ein Maß für die Relevanz der Indizes zu den dazugehörigen Speicherplätzen der sendenden Module bzw. die Relevanz des kombinierten Auftretens dieser Indizes dar. Werte von 0 in dem Gewichtsvektor entsprechen einem Ignorieren des entsprechenden Speicherplatzes des Index-sendenden Moduls. Die Gewichte enthalten die, beim Lernen aus dem präsentierten Ensemble an Mustern aus den Übungsdaten (oder aus anderen Quellen wie z.B. bekannte mathematische oder biologische Zusammenhänge), extrahierten funktionalen Zusammenhänge, welche zum Lösen der Datenverarbeitungsaufgabe verwendet werden.

Vorteilhafterweise kann die Gewichtsmatrix Templates (Schablonen) und/oder eine Sammlung an Mischungen von Kombinationen an Merkmalen, die für die Lösung der Datenverarbeitung relevant sind, umfassen. Ein Beispiel für das Entstehen von Templates ist es, wenn besonders informative Muster aus den Übungsdaten direkt in die Gewichte abgebildet werden. Im Fall eines Netzwerkes zur Erkennung von handgeschriebenen Ziffern könnten dies beispielsweise verschiedene prägnante Ausführungen einer Ziffer, z. B. der Ziffer 2, sein. Die dazugehörige latente Zustandsvariable würde dann, im Laufe der Verarbeitung der eingehenden Indizes, besonders hohe Werte erhalten, wenn im Anwendungsmodus diese spezielle Ausführung der Ziffer 2 (oder mit einer leichten Variation) gezeigt werden würde. Im Beispiel einer Berechnung einer Booleschen Funktion könnte dies der Übernahme der Funktionstabelle in die Gewichte entsprechen.

Gemäß einem zweiten allgemeinen Gesichtspunkt der Erfindung wird die o. g. Aufgabe durch ein Datenverarbeitungsverfahren gelöst, das zur Abbildung von Eingabedaten auf Ausgabedaten konfiguriert ist. Vorzugsweise wird das Datenverarbeitungsverfahren mit der Datenverarbeitungsvorrichtung gemäß dem obigen ersten allgemeinen Gesichtspunkt der Erfindung ausgeführt.

Das erfindungsgemäße Datenverarbeitungsverfahren umfasst die folgenden Verfahrensschritte. Zunächst erfolgen die Aufnahme der Eingabedaten r(s), die Erzeugung einer Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten und die Erzeugung einer stochastischen Repräsentation der Eingabedaten in Gestalt einer Sequenz von Eingabe-Indizes mit einer Eingabe-Einrichtung. Jeder Eingabe-Index ist eine Zahl, mit der eine stochastisch ermittelte Position innerhalb der Eingabe-Wahrscheinlichkeitsverteilung p*(s) bezeichnet wird.

Des Weiteren wird mindestens ein Netzwerk aus mindestens zwei Merkmalsmodulen betrieben, die über Datenverbindungen miteinander gekoppelt sind, wobei mindestens eines der Merkmalsmodule über eine Datenverbindung mit der Eingabe-Einrichtung gekoppelt ist und jedes Merkmalsmodul eine Vielzahl von Zustandsvariablen-Speicherplätzen, denen jeweils eine Vielzahl von Gewichte-Speicherplätzen zugeordnet ist, eine Vielzahl von Index-Speicherplätzen und eine Recheneinheit umfasst, wobei die Zustandsvariablen-Speicherplätze normierte latente Zustandsvariablen h_i und die Gewichte-Speicherplätze Gewichte p(s|i) speichern, die den latenten Zustandsvariablen h_i zugeordnet sind. Die Eingabe-Einrichtung sendet die Sequenz von Eingabe-Indizes an das mindestens eine mit der Eingabe-Einrichtung gekoppelte Merkmalsmodul. Jedes Merkmalsmodul sendet eine Sequenz von Modul-Indizes an die mit diesem Merkmalsmodul gekoppelten Merkmalsmodule, wobei jeder Modul-Index basierend auf einer Merkmalsmodul-Wahrscheinlichkeitsverteilung h_i des jeweils sendenden Merkmalsmoduls eine stochastisch ermittelte Position eines Zustandsvariablen-Speicherplatzes ist. Die Index-Speicherplätze jedes Merkmalsmoduls speichern empfangene Eingabe- oder Modul-Indizes. Die Recheneinheit jedes Merkmalsmoduls verändert schrittweise die latenten Zustandsvariablen h_i und die zugeordneten Gewichte p(s|i), die in dem Merkmalsmodul gespeichert sind, durch Anwendung einer numerischen Optimierung basierend auf den Werten in den Index-Speicherplätzen, so dass für alle direkt miteinander gekoppelten Merkmalsmodule wechselweise die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des einen Merkmalsmoduls an die latenten Zustandsvariablen $h_s$des jeweils anderen Merkmalsmoduls angepasst ist und die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des mindestens einen, mit der Eingabe-Einrichtung gekoppelten Merkmalsmoduls an die Eingabe-Wahrscheinlichkeitsverteilung p*(s) angepasst ist.

Schließlich werden die Ausgabedaten mit einer Ausgabe-Einrichtung ausgegeben, die mit mindestens einem Merkmalsmodul des Netzwerks über eine Datenverbindung gekoppelt ist.

Weitere unabhängige Gegenstände der Erfindung sind ein Computerprogrammprodukt, das auf einem Computer-lesbaren Speichermedium gespeichert und zur Ausführung des Verfahrens gemäß dem zweiten allgemeinen Gesichtspunkt der Erfindung eingerichtet ist, und ein Computer-lesbares Speichermedium, auf dem das Computerprogrammprodukt zur Ausführung des Verfahrens gemäß dem zweiten allgemeinen Gesichtspunkt der Erfindung gespeichert ist.

Ein Hauptvorteil der Erfindung besteht in der Bereitstellung einer universell anwendbaren Netzwerkarchitektur, die eine Ermittlung der Ausgabedaten aus den jeweils gegebenen Eingabedaten

auf der Grundlage der lokal in den Merkmalsmodulen optimierten Gewichte und latenten Zustandsvariablen und der gegenseitigen Anpassung der Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des einen Merkmalsmoduls an die latenten Zustandsvariablen h_s des jeweils anderen Merkmalsmoduls von unmittelbar miteinander verbundenen Merkmalsmodulen ermöglicht. Die Verwendung der Merkmalsmodule für die Konstruktion von Netzwerken mit frei wählbarer Architektur schließt insbesondere Netzwerke vom Typ eines DCNs ein, deren Anwendung von großem technologischem Interesse ist.

Die vorzugsweise ausschließlich lokale Ermittlung der Gewichte und latenten Zustandsvariablen jeweils in den Merkmalsmodulen erlaubt eine parallele Arbeit aller Recheneinheiten, wodurch sowohl der Lernmodus als auch der Anwendungsmodus beschleunigt werden. Die Datenverarbeitung erfolgt mit einer substantiell erhöhten Geschwindigkeit, die z. B. bei einer gegebenen Aufgabe zur Verarbeitung von Eingangsdaten gegenüber der Anwendung herkömmlicher Techniken um den Faktor 10.000 oder mehr erhöht sein kann. Eine derartige Beschleunigung ist durch herkömmliche Techniken, etwa durch Parallelisierung von Schaltungen zur Datenverarbeitung, nicht oder nur unter einem unannehmbar hohen Kosten- und Energieaufwand zu erreichen. Die gegenseitige Anpassung unmittelbar miteinander gekoppelter Merkmalsmodule und die Transitivität der Anpassung miteinander gekoppelter Merkmalsmodule erlaubt es, ein rekurrentes Netzwerk mit einer Vielzahl von Merkmalsmodulen, insbesondere ein geschichtetes rekurrentes Netzwerk mit vielen verborgenen Schichten, aufzubauen, in dem Information sowohl vorwärts als auch rückwärts fließen kann.

Vorteilhafterweise werden Informationen zwischen den Merkmalsmodulen ausschließlich in Gestalt der Eingabe- oder Modulindizes (und optional Ausgabe-Indizes), d. h. in Gestalt von stochastischen Zahlenfolgen, ausgetauscht, wodurch die Netzwerk-Architektur erheblich vereinfacht wird. Ein weiterer Vorteil der Erfindung ergibt sich aus der intrinsischen Stochastik, die erlaubt, dass das Netzwerk mit Zahlenwerten arbeiten kann, welche eine geringe Auflösung (e.g. Fixpunktzahlen mit wenigen Bits) aufweisen. Dies ermöglicht es, Merkmalsmodule zu verwenden, die jeweils für sich nur einfache Berechnungen ausführen. So kann jedes Merkmalsmodul mit seiner eigenen Recheneinheit versehen sein. Mit anderen Worten, man kann einfache Recheneinheiten verwenden (insbesondere mit Zustandsvariablen, die eine geringe Auflösung haben), die eine massive Parallelisierung der Datenverarbeitung ermöglichen. Des Weiteren ermöglicht die erfindungsgemäß vorgesehene Verarbeitung stochastischer Informationen im System auch eine inhärente Beseitigung oder Unterdrückung von Störungen, wie z. B. Rauschen, Überlagerungen u. ä..

Die Robustheit der Datenverarbeitung gegenüber Störungen steigt. Bei der Verarbeitung der Eingabe- und/oder Modulindizes werden wahrscheinlichere und spärlichere (englisch: 'sparse') wie beim 'importance sampling' Repräsentationen und Lösungen erreicht. Weniger relevante Informationen werden unterdrückt. Dieser Vorteil wird insbesondere durch Rückkopplungen zwischen den Merkmalsmodulen erzielt, die z. B. in [17] nicht vorgesehen waren.

Praktische Tests der Erfindung wurden bereits bei der Bildbearbeitung, insbesondere Zifferner-kennung erfolgreich realisiert. Beispielsweise umfasst ein Ziffererkennungsnetzwerk über 800 Merkmalsmodule, die in fünf verborgenen Schichten angeordnet sind. Da die Merkmalsmodule gleichzeitig und unabhängig voneinander berechnet werden können und zwischen den Merkmalsmodulen relativ wenige Daten ausgetauscht werden (nur die Indizes), wird das Skalierungsproblem herkömmlicher Netzwerke überwunden.

Die erfindungsgemäße Datenverarbeitungsvorrichtung kann lokal an einem System, z. B. Gerät, betrieben werden, an dem die zu verarbeitenden Eingabedaten vorliegen oder erzeugt, z. B. gemessen, erfasst, beobachtet, numerisch generiert oder gesammelt, werden. Alternativ oder zusätzlich kann die erfindungsgemäße Datenverarbeitungsvorrichtung in einem Netzwerk betrieben werden, in dem die zu verarbeitenden Eingabedaten von einem lokal betriebenen System an einen Server, z. B. einen zentralen Netzwerk-Server übertragen werden, in dem das Datenverarbeitungsverfahren ausgeführt wird.

Gemäß einer bevorzugten Ausführungsform der Erfindung umfasst die Eingabe-Einrichtung mindestens ein Eingabe-Merkmalsmodul, das wie die übrigen Merkmalsmodule mit einer Vielzahl von Zustandsvariablen-Speicherplätzen und einer Recheneinheit, vorzugsweise jedoch ohne Index-Speicherplätze aufgebaut ist. Die Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten wird mit der Recheneinheit berechnet und auf den Zustandsvariablen-Speicherplätzen gespeichert. Die Verwendung des Eingabe-Merkmalsmoduls als Eingabe-Einrichtung hat den Vorteil, dass bei der Gestaltung der Datenverarbeitungsvorrichtung eines der Merkmalsmodule frei für die Eingabe von Daten ausgewählt werden kann. Die Eingabe-Einrichtung kann als Teil des Netzwerks aus Merkmalsmodulen betrachtet werden. Im Unterschied zu den übrigen Merkmalsmodulen ist im Eingabe-Merkmalsmodul das Rechenwerk zur Aktualisierung der latenten Zustandsvariablen und Gewichte entfernt oder deaktiviert. Im Eingabe-Merkmalsmodul werden keine Gewichte berechnet und die Wahrscheinlichkeit in dem Eingabe-Merkmalsmodul, aus der die Indizes generiert werden, wird von außen vorgegeben und ändert sich nicht, weil ein Eingabe-Merkmalsmodul nicht auf eingehende Indizes reagiert.

Vorzugsweise ist die erfindungsgemäße Datenverarbeitungsvorrichtung zum Betrieb in einem Lernmodus oder in einem Anwendungsmodus eingerichtet. Vorteilhafterweise erfordert der Wechsel zwischen den Modi keine schaltungstechnischen Vorkehrungen. Im Lernmodus (Training des Netzwerks) ist die Eingabe-Einrichtung zur Aufnahme vorbekannter Lern-Eingabedaten, die Ausgabe-Einrichtung zur Aufnahme vorbekannter Lern-Ausgabedaten, die den Lern-Eingabedaten zugeordnet sind, und das Netzwerk aus Merkmalsmodulen für das Aktualisieren der latenten Zustandsvariablen h_i und das Lernen der zugeordneten Gewichte p(s|i), mit denen die Lern-Eingabedaten auf die Lern-Ausgabedaten abgebildet werden, eingerichtet. Im Lernmodus werden die Eingabe-Indizes und die Modul-Indizes gleichzeitig oder in einer vorbestimmten Reihenfolge gesendet, um die vielparametrige Funktion zu ermitteln, welche durch das Netzwerk repräsentiert wird und die Eingabedaten auf die Ausgabedaten abbildet. Das Netzwerk lernt durch Beispiele (Input – Output – Paare) seine Funktion, so dass es im Anwendungsmodus bei der Präsentation von neuen Eingabedaten die korrekten (bisher unbekannten) Ausgabedaten berechnet. Im Lernmodus ist die Recheneinheit zur Aktualisierung der latenten Zustandsvariablen deaktiviert, weil die Wahrscheinlichkeitsverteilung zum Generieren der (Ausgabe-)Modul-Indizes von außen (Trainings-Ausgabedaten) vorgegeben wird und sich nicht durch eingehende Modul-Indizes ändern soll.

Im Anwendungsmodus ist die Eingabe-Einrichtung zur Aufnahme von Anwendungs-Eingabedaten und die Ausgabe-Einrichtung zur Ausgabe von Anwendungs-Ausgabedaten eingerichtet, die den Anwendungs-Eingabedaten zugeordnet und durch die Merkmalsmodule des Netzwerks mit den im Lernmodus aktualisierten latenten Zustandsvariablen h_i und zugeordneten gelernten Gewichten p(s|i) ermittelt sind.

Der Anwendungsmodus kann mit dem Lernmodus kombiniert werden, indem während der Anwendung der Datenverarbeitung das Netzwerk auf aktuelle Umgebungsbedingungen eingestellt wird (Nachlernen). Beim Nachlernen werden zusätzlich Paare von Eingabedaten und Ausgabedaten in das Netzwerk eingegeben und in die Optimierung der latenten Zustandsvariablen und Gewichte einbezogen.

Gemäß einer weiteren Variante der Erfindung kann die Eingabe-Einrichtung optional zusätzlich mindestens ein Lerneingabe-Merkmalsmodul umfassen, das im Lernmodus mit dem mindestens Netzwerk aus mindestens zwei Merkmalsmodulen gekoppelt und im Anwendungsmodus von dem mindestens Netzwerk getrennt ist. Vorteilhafterweise können damit im Lernmodus zusätzliche

Lern-Eingabedaten bereitgestellt werden, die das Lernen beschleunigen und/oder das Lernergebnis verbessern (erhöhte Zuverlässigkeit des Netzwerks im Anwendungsmodus).

Ein weiterer wichtiger Vorteil der Erfindung besteht darin, dass die verarbeiteten Daten Informationen verschiedener Kategorien tragen können. Die Eingabedaten und/oder Ausgabedaten können z. B. Informationen von Bildmerkmalen, Audiomerkmalen, Sprachmerkmalen, Sensormerkmalen, Signalmerkmalen, ökonomische Zeitreihen, Verhaltensdaten, Zustandsmerkmalen eines biologischen Organismus, Zustandsmerkmalen eines natürlichen Systems, wie z. B. Wetterdaten, physikalischen Größen, chemischen Parametern und/oder Zustandsmerkmale einer technischen Einrichtung umfassen. Die Kategorien der Eingabedaten und der Ausgabedaten können gleich oder verschieden sein. Beispielsweise können eingabeseitige Sprachmerkmalen ausgabeseitig Bildmerkmalen oder eingabeseitige Bildmerkmale ausgabeseitig diagnostischen Informationen zugeordnet werden. Vorteilhafterweise können die latenten Zustandsvariablen h_i Merkmale der Eingabedaten repräsentieren. Die Zuordnung zu Merkmalen der Eingabedaten erfolgt automatisch im Lernmodus. Alternativ kann es sein, dass latente Zustandsvariablen h_i nicht bestimmten, benennbaren Merkmalen der Eingabedaten zuordenbar sind. Des Weiteren können die latenten Zustandsvariablen h_i verschiedener Merkmalsmodule verschiedene Klassen von Merkmalen der Eingabedaten repräsentieren. Beispielsweise können die latenten Zustandsvariablen h_i verschiedener Merkmalsmodule bei der Verarbeitung von Bilddaten verschiedene Bildmerkmale, wie z. B. bestimmte geometrische Muster, Farben oder auch Bildbestandteile oder bei der Verarbeitung von Wetterdaten verschiedene Wettermerkmale repräsentieren.

Gemäß einer weiteren bevorzugten Ausführungsform der Erfindung kann die Datenverarbeitungsvorrichtung mit einer Konfigurationseinrichtung ausgestattet sein, mit der eine Netzwerkkonfiguration einstellbar ist. Die Konfigurationseinrichtung kann z. B. Teil einer Steuereinrichtung der Datenverarbeitungsvorrichtung und über eine Eingabeschnittstelle durch einen Nutzer, z. B. auf der Basis von Erfahrungswerten, bedienbar und/oder für eine automatische Konfiguration ausgelegt sein. Die Netzwerkkonfiguration umfasst die Anzahl der Merkmalsmodule, die Anzahl von Zustandsvariablen-Speicherplätzen jedes Merkmalsmoduls, die Anzahl von Gewichte-Speicherplätzen, die jeweils einem Zustandsvariablen-Speicherplatz in einem Merkmalsmodul zugeordnet sind, die Anzahl von Index-Speicherplätzen jedes Merkmalsmoduls und/oder die Datenverbindungen zwischen Merkmalsmodulen.

Die Konfigurationseinrichtung kann vorteilhafterweise für eine iterative Einstellung der Netzwerkkonfiguration eingerichtet sein, bei der die Anzahl der Merkmalsmodule und die Anzahl der Zustandsvariablen-Speicherplätze jedes Merkmalsmoduls manuell oder programmgesteuert schrittweise erhöht werden, bis die Leistung der Abbildung der Eingabedaten auf die Ausgabedaten einem vorbestimmten Betriebskriterium entspricht, wobei beispielsweise evolutionäre Algorithmen (siehe [22]) angewendet werden.

Alternativ oder zusätzlich kann die Datenverarbeitungsvorrichtung mindestens ein Reserve-Merkmalsmodul enthalten, das zu Beginn des Betriebs der Datenverarbeitungsvorrichtung oder auch nach Durchlaufen des Lernmodus keine Datenverbindung mit dem Netzwerk aus Merkmalsmodulen aufweist. Die Konfigurationseinrichtung der Datenverarbeitungsvorrichtung ist in diesem Fall zur bedarfsweisen Herstellung mindestens einer Datenverbindung des mindestens einen Reserve-Merkmalsmoduls mit dem Netzwerk aus Merkmalsmodulen eingerichtet. Vorteilhafterweise liefert das mindestens eine Reserve-Merkmalsmodul eine hohe Flexibilität der Datenverarbeitungsvorrichtung in Bezug auf die Anpassung an geänderte Anwendungsbedingungen. Außerdem erlaubt dies auch, Ausfälle durch Fertigungsdefekte auf integrierten Schaltungen auszugleichen und dadurch die Ausbeute zu erhöhen.

Gemäß einer weiteren bevorzugten Ausführungsform der Erfindung kann das Netzwerk aus Merkmalsmodulen ein geschichtetes Netzwerk, insbesondere Deep Convolutional Network (DCN), mit mehreren Schichten aus Merkmalsmodulen umfassen, bei dem Merkmalsmodule innerhalb einer Schicht gemeinsame Gewichte haben, die den latenten Zustandsvariablen h_i zugeordnet sind. Der besondere Vorteil dieser Ausführungsform besteht darin, dass an sich bekannte Architekturen geschichteter Netzwerke nachgebildet werden können, wobei jedoch die Anwendung des Netzwerks erheblich beschleunigt ist. DCNe haben den Vorteil, dass sich viele Merkmalsmodule dieselben Gewichte teilen, was das Lernen vereinfacht und oft die Leistungsfähigkeit des Netzwerkes verbessert.

Das geschichtete Netzwerk kann insbesondere für einen schichtbasierten Lernmodus eingerichtet sein, wobei die Schichten von der Eingabe-Einrichtung beginnend für ein Aktualisieren der latenten Zustandsvariablen h_i und ein Lernen der zugeordneten Gewichte p(s|i) derart eingerichtet ist, dass die Lern-Eingabedaten schichtweise auf die Lern-Ausgabedaten abgebildet werden.

Alternativ kann das geschichtete Netzwerk für einen unüberwachten, schichtbasierten Lernmodus eingerichtet sein, wobei die Schichten von der Eingabe-Einrichtung beginnend bis zu einer vorletzten Schicht vor der Ausgabe-Einrichtung aufeinanderfolgend für eine Ermittlung der latenten Zustandsvariablen h_i und der zugeordneten Gewichte p(s|i) ohne Berücksichtigung der Lern-Ausgabedaten eingerichtet sind (unüberwachtes Vorlernen) und erst eine letzte Schicht vor der Ausgabe-Einrichtung für eine Ermittlung der latenten Zustandsvariablen h_i und der zugeordneten Gewichte p(s|i) mit einer Berücksichtigung der Lern-Ausgabedaten eingerichtet ist.

Beim unüberwachten Vorlernen werden die Eingabedaten an der Eingabe-Einrichtung eingegeben, aber nur die erste verborgene Schicht des Netzwerkes ist zunächst mit der Eingabe-Einrichtung gekoppelt. In diesem Zustand wird der erste Gewichtssatz gelernt, welcher der ersten verborgenen Schicht zugeordnet ist. Nach einer ausreichenden Menge an Lernschritten werden die Gewichte dieser Schicht festgehalten. Anschließend werden die Gewichte der nächsten verborgenen Schicht gelernt. Erst bei der letzten verborgenen Schicht werden die Ausgabedaten beim Lernen berücksichtigt.

Alternativ kann beim Lernen zunächst das Netzwerk für eine Vorlaufzeit arbeiten, ohne dass die Lern-Eingabedaten und Lern-Ausgabedaten eingegeben werden. Des Weiteren kann es von Vorteil sein, die Gewichte, die ein Merkmalsmodul mit einem anderen verbinden, mit denen in der Gegenrichtung in ihrer Struktur gleichzusetzen. Wegen der notwenigen Normierung unterscheiden sich die Werte von der reinen Transposition der entsprechenden Matrizen.

Alternativ kann das geschichtete Netzwerk für einen überwachten, schichtbasierten Lernmodus eingerichtet sein, bei dem basierend auf dem aktuellen Lernfehler die Gewichte p(s|i) ausgewählter Module reinitialisiert werden.

Ein weiterer Vorteil der Erfindung ist es, dass verschiedene numerische Maße für die numerische Optimierung in den Merkmalsmodulen verfügbar sind. Gemäß bevorzugten Varianten der Erfindung kann die Recheneinheit jedes Merkmalsmodul dazu eingerichtet sein, die numerische Optimierung basierend auf einem probabilistischen Ziel-Maß, insbesondere der Log-Likelihood, der Kullbach-Leibler-Divergenz oder Kombinationen von Vektornormen, auszuführen. Das numerische Maß kann in Abhängigkeit von der Anwendung der Erfindung für eine schnelle Optimierung beim Lernen gewählt werden. Die parallele und asynchrone Aktualisierung sowohl der latenten Zustandsvariablen als auch der Gewichte beim Lernen stellt einen wichtigen Vorteil der Erfindung

dar. Sie folgt aus der Optimierung einer Zielfunktion, die bisher nicht für das Lernen von Netzwerken aus Merkmalsmodulen verwendet wurde (z. B. Log-Likelihood-Maß). Die Anwendung von Ziel-Maßen hat ferner den Vorteil, dass diese Abbruchkriterien für das Lernen ergeben. Außerdem wird dabei die Anzahl der nötigen Iterationen bei der Aktualisierung der latenten Variablen festgestellt.

Gemäß einer weiteren bevorzugten Ausführungsform der Erfindung kann die Ausgabe-Einrichtung der Datenverarbeitungsvorrichtung mindestens ein Ausgabe-Merkmalsmodul umfassen, das wie die übrigen Merkmalsmodule mit einer Vielzahl von Zustandsvariablen-Speicherplätzen, einer Vielzahl von Index-Speicherplätzen und einer Recheneinheit aufgebaut ist, wobei die latenten Zustandsvariablen des mindestens einen Ausgabe-Merkmalsmoduls oder davon abgeleitete Größen als Ausgabedaten ausgegeben werden. Zusätzlich enthält das Ausgabe-Merkmalsmodul einen Schaltungsteil zur Extraktion eines Ergebnisses (Ausgabe-Daten) aus den latenten Zustandsvariablen. Hierzu ist z. B. eine Schaltung vorgesehen, die auf den Speicherplätzen der latenten Zustandsvariablen eine Ermittlung der Position des Speicherplatzes, welcher den maximalen h_iWert beinhaltet, ausführt (argmax-Schaltung). Das Ausgabe-Merkmalsmodul kann wie die übrigen Merkmalsmodule Modul-Indizes in Rückwärtsrichtung an die mit dem Ausgabe-Merkmalsmodul gekoppelten Merkmalsmodule senden (Übertragung von Ausgabe-Indizes). Die Ausgabe-Indizes werden vorzugsweise so erzeugt, wie oben für die Eingabe-Indizes und mit weiteren Einzelheiten unten unter Bezug auf die Figuren erläutert ist. Die Verwendung des Ausgabe-Merkmalsmoduls als Ausgabe-Einrichtung hat den Vorteil, dass bei der Gestaltung der Datenverarbeitungsvorrichtung eines der Merkmalsmodule frei für die Ausgabe von Daten ausgewählt werden kann. Die Ausgabe-Einrichtung kann als Teil des Netzwerks aus Merkmalsmodulen betrachtet werden.

Alternativ oder zusätzlich kann die Ausgabe-Einrichtung einen Klassifikator, insbesondere basierend auf Perzeptrons, oder einer Support-Vektor-Maschine, umfassen, der auf der Basis der latenten Zustandsvariablen von Merkmalsmodulen operiert, mit denen die Ausgabe-Einrichtung gekoppelt ist.

Die Datenverarbeitungsvorrichtung kann durch eine integrierte Schaltung, optional gekoppelt mit einer Steuereinrichtung, z. B. einem Steuercomputer, realisiert werden. Es kann z. B. ein herkömmlicher Universal-Prozessor mit z. B. 8 Kernen verwendet werden um 8 Merkmalsmodule gleichzeitig bereitzustellen. Wegen der hohen Zahl der erfindungsgemäß verwendeten Merkmalsmodule (z. B. 100 bis 1000) hat die integrierte Schaltung jedoch vorzugsweise eine Prozessorarchi-

tektur, welche die Verfahren zur Verarbeitung der Daten optimal (d.h. insbesondere mit minimaler Anzahl von Speicherzugriffen) umsetzt. Dies wird durch die Einfachheit der auszuführenden mathematischen Einzelschritte des Datenverarbeitungsverfahrens ermöglicht, die lediglich Addition, Multiplikation und Division von positiven Zahlen umfassen. Vorteilhafterweise können dann hunderte Merkmalsmodule von einem einzigen speziell angepassten Prozessor verarbeitet werden. Eine Vielzahl von Merkmalsmodulen kann in einer einzigen integrierten Schaltung, insbesondere auf einem Chip, realisiert werden.

Die Datenverarbeitungsvorrichtung kann ein einziges Netzwerk aus Merkmalsmodulen beinhalten. Gemäß einer vorteilhaften Ausführungsform der Erfindung umfasst die Datenverarbeitungsvorrichtung ein modulares System mit mindestens zwei miteinander verbundenen, integrierten Schaltungen, die jeweils ein Netzwerk aus Merkmalsmodulen aufweisen. Die mindesten zwei Netzwerke können mit einer einzigen Eingabe-Einrichtung oder mit mehreren Eingabe-Einrichtungen und/oder mit einer einzigen Ausgabe-Einrichtung oder mit mehreren Ausgabe-Einrichtungen verbunden sein. In diesem Fall können die Flexibilität bei der Anpassung der Datenverarbeitungsvorrichtung an eine konkrete Aufgabe verbessert, die Komplexität der Anwendung erhöht und/oder die Ausfallsicherheit verbessert werden.

Zusammengefasst bietet die Erfindung insbesondere die folgenden Vorteile. Das Netzwerk ist im Vergleich mit herkömmlichen Netzwerken mit einer erhöhten Anzahl von Merkmalsmodulen, insbesondere Schichten aus Merkmalsmodulen realisierbar. Die erfindungsgemäße Datenverarbeitung ist hochgradig parallelisierbar und in größeren Netzwerken bei Real-Time Anwendungen einsetzbar. Das Skalierungsproblem herkömmlicher Netzwerke wird durch die Lokalität, die damit verbundene Parallelisierung sowie den reduzierten Datenaustausch zwischen Modulen überwunden. Das Lernen und die Anwendung der Datenverarbeitung sind, insbesondere mit angepasster Hardware (z. B. FPGA) , 1000 bis 10000 mal schneller realisierbar, so dass insbesondere bei closed-loop-Anwendungen (z.B. in der Robotik) die jeweilige spezielle Aufgabe effizient schnell erlernt werden kann oder eine vorgelernte Gewichtsstruktur individuell und möglichst schnell angepasst werden kann.

Kurzbeschreibung der Zeichnungen

Weitere Einzelheiten und Vorteile der Erfindung werden im Folgenden mit Bezug auf die beigefügten Zeichnungen beschrieben. Es zeigen

Figur 1: eine Übersichtsdarstellung einer Datenverarbeitungsvorrichtung mit Merkmalen bevorzugter Ausführungsformen der Erfindung;

Figur 2: Einzelheiten eines Merkmalsmoduls einer Datenverarbeitungsvorrichtung gemäß der Erfindung;

Figur 3: eine Darstellung des Informationsaustauschs innerhalb einer Datenverarbeitungsvorrichtung gemäß der Erfindung;

Figur 4: eine Darstellung des Anwendungs- und des Lernmodus der Datenverarbeitungsvorrichtung gemäß der Erfindung;

Figur 5: eine Abwandlung des Lernmodus gemäß Figur 4;

Figur 6: eine beispielhafte Illustration einer Anwendung der Datenverarbeitungsvorrichtung gemäß der Erfindung;

Figur 7: eine beispielhafte Illustration einer Anwendung der Datenverarbeitungsvorrichtung gemäß der Erfindung bei der Bildung eines DCN; und

Figuren 8 bis 26: Einzelheiten einer Hardware-Konfiguration einer Datenverarbeitungsvorrichtung gemäß der Erfindung.

Bevorzugte Ausführungsformen der Erfindung

Einzelheiten bevorzugter Ausführungsformen der Erfindung werden im Folgenden beispielhaft in Bezug auf die Konfiguration der Datenverarbeitungsvorrichtung, Beispiele von Anwendungen, die schaltungstechnische Ausführung der Datenverarbeitungsvorrichtung und die mathematischen Grundlagen der Datenverarbeitung beschrieben. Es wird betont, dass die Umsetzung der Erfindung in der Praxis nicht auf die beschriebenen Ausführungsformen, insbesondere die Anwendungsbeispiele und die schaltungstechnischen Details beschränkt ist. Vielmehr können alle Anwendungen, die von herkömmlichen neuronalen Netzwerken bekannt sind, und darüber hinaus komplexere Datenverarbeitungsaufgaben abgearbeitet werden, und Abwandlungen der Konfiguration und Schaltungstechnik vorgesehen sein.

Das Verfahren der Bildung eines einzelnen Merkmalsmoduls mit internen latenten Zustandsvariablen sowie die Erzeugung von Modul-Indizes mit einem einzigen Merkmalsmodul sind an sich bereits aus [17] bekannt. Die Veröffentlichung [17] wird durch Bezugnahme ausdrücklich in die vorliegende Offenbarung aufgenommen, so dass der Fachmann bei der Konfiguration des Netzwerks der erfindungsgemäßen Datenverarbeitungseinrichtung ergänzende Informationen aus [17] erhält, welche insbesondere die mathematischen Grundlagen der Bildung der latenten Zustandsvariablen und der Erzeugung der Modul-Indizes (in [17] als "Spikes" bezeichnet) betreffen.

Konfiguration der Datenverarbeitungsvorrichtung

Figur 1 zeigt eine Übersichtsdarstellung der Datenverarbeitungsvorrichtung 100 mit einer Eingabe-Einrichtung 10 zur Aufnahme von Eingabedaten r(s), einem Netzwerk 20 mit Merkmalsmodulen 21, die über Datenverbindungen 22 miteinander und/oder mit der Eingabe-Einrichtung 10 gekoppelt sind, einer Ausgabe-Einrichtung 30 zur Ausgabe der Ausgabedaten Out und einer Steuereinrichtung 40. Die Eingabe-Einrichtung 10 und die Ausgabe-Einrichtung 30 können jeweils mindestens ein Modul 11, 31 umfassen, das wie die Merkmalsmodule 21 aufgebaut ist. Wenn mehrere Module 11 oder 31 vorgesehen sind, können diese jeweils untereinander und mit Merkmalsmodulen 21 oder jeweils ausschließlich mit Merkmalsmodulen 21 verbunden sein. Des Weiteren sind die Eingabe- und Ausgabe-Einrichtungen 10, 30 mit Daten-Schnittstellen (Ein- und Auslesemechanismen) versehen (nicht dargestellt).

Die Datenverbindungen 22 bezeichnen die software- oder hardware-basierte Kopplung von Merkmalsmodulen zur Weitergabe von Eingabe- oder Modul-Indizes. Die Steuereinrichtung 40 kann z. B. einen Computer umfassen, der die Komponenten 10 bis 30 steuert und insbesondere eine Konfigurationseinrichtung 41, mit der eine Netzwerkkonfiguration und insbesondere die Datenverbindungen 22 einstellbar sind, eine Bedieneinrichtung 42 und/oder eine Anzeigeeinrichtung 43 enthalten. Einzelheiten der Komponenten 10 bis 30 werden unten unter Bezug auf die Figuren 8 bis 26 beschrieben. Die Funktion der Komponenten 10 bis 30 wird unten im Abschnitt "Mathematische Grundlagen der Datenverarbeitung" beschrieben.

Die Datenverarbeitungsvorrichtung 100 kann durch einen Computer, auf dem eine Software zur Ausführung des erfindungsgemäßen Verfahrens läuft, oder mindestens eine mit den genannten Komponenten konfigurierte Schaltung, insbesondere mindestens eine integrierte Schaltung (Netzwerk-Chip, z. B. ASIC-Chip) oder eine Kombination aus beiden realisiert werden. Es kann ferner ein

modulares Hardware-System mit mindestens zwei integrierten Schaltungen vorgesehen sein, die jeweils ein oder mehrere Netzwerk(e) 20 aus Merkmalsmodulen 21 umfassen.

Der Aufbau eines Merkmalsmoduls 21 ist schematisch in Figur 2 gezeigt, während die konkrete schaltungstechnische Umsetzung von Merkmalsmodulen und deren Verbindung in den Figur 8 bis 26 beispielhaft gezeigt ist. Jedes Merkmalsmodul 21 umfasst eine Vielzahl von Speicherplätzen 23 bis 25 und eine Recheneinheit 26. Die Speicherplätze 23 bis 25 umfassen Zustandsvariablen-Speicherplätze 23, denen jeweils eine Gruppe von Gewichte-Speicherplätze 24 zugeordnet sind, und Index-Speicherplätzen 25, deren Funktionen unten erläutert werden. Es sind N_H Zustandsvariablen-Speicherplätze 23 für latente Zustandsvariablen vorgesehen. Bei typischen Anwendungen ist N_H im Bereich von 4 bis 1024 Aus den Zustandsvariablen-Speicherplätze 23 wird in jedem Verarbeitungsschritt ein Modul-Index $i_t$stochastisch erzeugt und an alle mit diesem Merkmalsmodul 21 verbundenen anderen Merkmalsmodule gesendet. Diese anderen Merkmalsmodule (ggf. einschließlich die Eingabe-Einrichtung 10) schicken die in diesen stochastisch erzeugten Eingabe- oder Modul-Indizes s_t^k an dieses Merkmalsmodul 21. Die eingehenden Indizes (hier $s_t^1, s_t^2, s_t^3, \ldots$) werden in den Index-Speicherplätzen 25 im empfangenden Merkmalsmodul 21 gespeichert. Wenn genügend Indizes gesammelt wurden, werden die latenten Zustandsvariablen dieses Merkmalsmoduls auf der Basis ihrer vorigen Werte, dieser gespeicherten Indizes s_t^kund optional den dazugehörigen Gewichten $p(s|i)$ aktualisiert. Die aktualisierten Werte in den latenten Zustandsvariablen werden dann wieder verwendet um neue Indizes zu generieren, die an die verbundenen Merkmalsmodule gesendet werden. Die eingehenden Indizes könnten im Rahmen eines Online-Lern-Verfahrens auch dazu verwendet werden, die Gewichte selber zu aktualisieren.

Figur 3 zeigt weitere Einzelheiten des Flusses der Indizes am Beispiel eines einfachen Netzwerks 20, das in diesem Fall ein Eingabe-Modul 11 der Eingabe-Einrichtung (EE), zwei Merkmalsmodule 21 (MM1, MM2) und einem Ausgabe-Modul 31 der Ausgabe-Einrichtung (AE) umfasst. In dem Eingabe-Modul 11 wird im Laufe des Betriebes des Netzwerks 20 aus $p^*(s)$ eine Sequenz von Eingabe-Indizes $s_1, s_2, s_3, s_4, \ldots$ stochastisch erzeugt („gezogen"). Auf Grund der in diesem Beispiel gewählten Architektur werden diese Eingabe-Indizes dem ersten Merkmalsmodul MM1 mitgeteilt. Das Merkmalsmodul MM1 wiederum erzeugt aus dem jeweiligen Zustand seiner latenten Zustandsvariablen $h_{MM1}(i)$ Modul-Indizes $i_1, i_2, i_3, i_4, \ldots$ und sendet diese entsprechend der hier gewählten Architektur an das Merkmalsmodul MM2. Merkmalsmodul MM2 generiert die Indizes $j_1, j_2, j_3, j_4, \ldots$ aus $h_{MM2}(j)$ und sendet die Indizes sowohl in Rückwärtsrichtung an Merkmalsmodul MM1 als auch an das Ausgabe-Modul 31 der Ausgabe-Einrichtung AE. Das Ausgabe-Modul 31 wiederum sendet $k_1, k_2, k_3, k_4, \ldots$ an das Merkmalsmodul MM2. Diese Folge an Indizes wird aus

dem jeweils aktuellen Stand von $h_{AE}(k)$ erzeugt. Jeder eingehende Index führt zu einer Aktualisierung der latenten Zustandsvariablen des empfangenden Merkmalsmoduls, was die Änderung der Wahrscheinlichkeitsverteilung für die Erzeugung der Indizes in jedem Merkmalsmodul zur Folge hat. Dies gilt nicht für $p^*(s)$, da diese Wahrscheinlichkeitsverteilung nur von den angelegten Eingabedaten abhängt und von der Eingabe-Einrichtung keine Indizes verarbeitet werden. Die Aktualisierung basiert auf den unten beispielhaft beschriebenen mathematischen Operationen.

Figur 4 zeigt den Fluss der Eingabe- und/oder Modul-Indizes in einem Anwendungsmodus und einem Lernmodus des Netzwerks 20. Typischerweise wird sowohl im Anwendungsmodus wie auch im Lernmodus ein Muster von Eingabedaten $r(s)$ an die Eingabe-Einrichtung 10 (EE) angelegt. Diese rechnet die Eingabedaten in die Wahrscheinlichkeitsverteilung $p^*(s) = \frac{r(s)}{\sum_k r(s)}$ um, aus der dann Indizes gezogen werden. Diese Wahrscheinlichkeitsverteilung wird über die Verarbeitung dieser vorgegeben Eingabedaten festgehalten. Im Anwendungsmodus führt dieser Satz an Eingabedaten, über die gegenseitige Aktualisierung der latenten Zustandsvariablen durch die ausgetauschten Indizes, zu einer Wahrscheinlichkeitsverteilung in der Ausgabe-Einrichtung 30 (AE). Diese Wahrscheinlichkeitsverteilung wird dann entsprechend ausgewertet und in ein Ergebnis der Verarbeitung übersetzt (Ausgabedaten OUT). Im Lernmodus wird typischerweise ein zum jeweils angelegten Eingabe-Muster von Eingabedaten gehöriges vorbekanntes Ausgabe-Muster von Ausgabedaten an die Ausgabe-Einrichtung 30 angelegt. Dies führt zu einer (für das jeweilige Muster festen) Wahrscheinlichkeitsverteilung, die dazu führt, dass die Ausgabe-Einrichtung 30 Indizes an andere Merkmalsmodule 21 sendet. Die Merkmalsmodule 21, die diese Indizes von der Ausgabe-Einrichtung 30 empfangen, lernen in ihren Gewichten den Zusammenhang (oder Teile davon) zwischen den Eingabe- und Ausgabedaten. Im Anwendungsmodus, in dem neue Eingabedaten zugeführt werden, ergänzt das Netzwerk die nun nicht mehr anliegenden Ausgabedaten.

Im Lernmodus kann, wie in Figur 5 dargestellt ist, die Eingabe-Einrichtung 10 zusätzlich mindestens ein Lerneingabe-Merkmalsmodul 12 aufweisen, das mit dem Netzwerk 20 aus Merkmalsmodulen 21 gekoppelt ist (Lernmodus mit Seiteneinkopplung). Im Anwendungsmodus ist das Lerneingabe-Merkmalsmodul 12 von dem mindestens Netzwerk 20 getrennt. An dem zusätzlichen Lerneingabe-Merkmalsmodul 12 werden die zu lernenden Ausgabedaten zugeführt und beim Aktualisieren der Zustandsvariablen und Lernen der Gewichte in den Merkmalsmodulen 21 berücksichtigt. Dadurch wird die Wahrscheinlichkeitsverteilung in der Ausgabe-Einrichtung 30 nicht mit der gewünschten Wahrscheinlichkeitsverteilung festgehalten, sondern sie verhält sich so wie im Anwendungsmodus. Vorteilhafterweise entspricht die Dynamik der latenten Zustandsvariablen in

der Ausgabe-Einrichtung 30 und dem Merkmalsmodulen MM2 so mehr dem gewünschten Zielzustand, da die latenten Zustandsvariablen, im Vergleich zu der Variante, bei der die Ausgabedaten an der Ausgabe-Einrichtung 30 beim Lernen festgehalten werden, freier variieren können.

Die Figuren 6a bis 6c zeigen die Struktur von Netzwerken, welche die Boolesche Paritäts-Funktion berechnen (Figuren 6a und 6b für 4 bit und Figur 6c für 8 bit). Wiederum sind beispielhaft die Eingabe-Einrichtung 10, mehrere Merkmalsmodule 21, die gruppenweise zu Schichten 50, 51, 52 zusammengefasst sind, und die Ausgabe-Einrichtung 30 gezeigt.

Figur 6a zeigt das Boolesches Parität-Netzwerk für 4 Bit. Jeweils vier Eingabe-Module 11 der Eingabe-Einrichtung 10 mit jeweils zwei Speicherplätzen beschreiben die Eingabedaten. Entweder ist der Speicherplatz für den Bit-Wert 0 oder der Speicherplatz für den Bit-Wert 1 aktiv. Die von diesen vier Eingabe-Modulen 11 erzeugten Eingabe-Indizes werden an die erste Schicht 50 weitergeleitet, wo zwei Merkmalsmodule 21 mit jeweils 4 Speicherplätzen für latente Zustandsvariablen vorhanden sind. Diese Merkmalsmodule 21 der ersten Schicht 50 senden Modul-Indizes an das Merkmalsmodul 21 in der zweiten Schicht 51 sowie empfangen Modul-Indizes aus der zweiten Schicht 51. Die zweite Schicht 51 besteht aus einem Merkmalsmodul 21 mit vier Speicherplätzen für latente Zustandsvariablen. Diese Schicht 51 sendet und empfängt Modul-Indizes an die Ausgabe-Einrichtung 30 bzw. von den Merkmalsmodulen 21 in der ersten Schicht 50. Die Ausgabe-Einrichtung 30 erhält Modul-Indizes von dem Merkmalsmodul 21 der zweiten Schicht 51. Es wird dann ausgewertet, welcher der beiden Speicherplätze der Ausgabe-Einrichtung 30 für den Bit-Wert 0 oder 1 den höheren Wert hat und als Ergebnis der Berechnung ausgegeben. Die Gewichte können gelernt werden. In Figur 6a ist ein Satz an Gewichten durch Linien dargestellt, der die Aufgabe löst. Dabei sind Gewichte mit einem Wert 0 nicht eingezeichnet. Die gezeigten Gewichte in einer Schicht haben alle die gleichen Werte, wobei der Wert von der Normalisierung der Gewichte bestimmt ist. Figur 6b zeigt, wie die Indizes zwischen den Merkmalsmodulen 21 in Figur 6a ausgetauscht werden. Figur 6c illustriert eine analoge Struktur und der Informationsfluss für die 8 Bit-Paritätsfunktion.

Ein Beispiel eines Deep-Convolution-Netzwerks für die Klassifikation handgeschriebener Ziffern, das durch die erfindungsgemäße Datenverarbeitungseinrichtung realisiert wird, ist schematisch in Figur 7 illustriert. Bei diesem sind die Merkmalsmodule 21 zu Schichten 53 bis 57 wie folgt zusammengefasst. Der Block A zeigt einen Satz von Eingabe-Merkmalsmodulen der Eingabe-Einrichtung 10 (28 x 28) für Bilder mit 28 x 28 Pixeln. Jedes Eingabe-Merkmalsmodul 21 enthält zwei Zustandsvariablen. Dies realisiert eine vereinfachte Version der aus der Biologie bekannten „On/Off center

cells“. Dieser Satz an Eingabe-Merkmalsmodulen sendet Eingabe-Indizes an die Sammlung von Merkmalsmodulen (Schicht 53). Block B illustriert die Faltungsschicht 53 mit 24 x 24 Merkmalsmodulen mit jeweils 32 latenten Zustandsvariablen. Jedes Merkmalsmodul verarbeitet Modul-Indizes von 5 x 5 Blöcken von den vorgeschalteten Eingabe- Merkmalsmodulen und Indizes von den nachgeschalteten Merkmalsmodulen mit einer zusammenfassenden Eigenschaft in Schicht 54. Block C illustriert die Schicht 54 aus 12 x 12 Merkmalsmodulen mit jeweils 32 latenten Zustandsvariablen. Jedes Merkmalsmodul in dieser Schicht 54 erhält die Modul-Indizes eines 2 x 2 großen und nicht überlappenden Bereiches in der Schicht 53. Die Gewichte zwischen den Schichten 53 und 54 werden nicht gelernt, sondern werden vorkonfiguriert (z. B. manuell vorgegeben), so dass ein Wettbewerb zwischen den 32 Merkmalen der vorigen Schicht entsteht. Die Merkmalsmodule der Schicht 54 verarbeiten auch die Modul-Indizes, welche von den Merkmalsmodulen aus der Faltungsschicht 55 stammen und mit dem jeweiligen Einzugsgebiet der Merkmalsmodule überlappen. Wenn man hierfür gemeinsame Gewichte verwenden möchte, kann hier die so genannte „Transposed convolution“ (auch „fractionally strided convolutions“) angewendet werden. Block D illustriert die 5 x 5 Faltungsschicht 55 mit 8 x 8 Merkmalsmodulen. Ähnlich zu den Merkmalsmodulen in der Schicht 53, aber mit 64 Speicherplätzen für die latenten Zustandsvariablen in den jeweiligen Merkmalsmodulen. Block E ist eine Schicht 56 wie die Zusammenfassungs-Schicht 54, aber nur mit 2 x 2 Merkmalsmodulen mit jeweils 64 Speicherplätzen für die latenten Zustandsvariablen. Block F zeigt ein vollverbundenes Merkmalsmodul 57. Dieses enthält 1024 Speicherplätze für latente Zustandsvariablen und ist vollständig mit der Zusammenfassungs-Schicht 56 und Ausgabe-Einrichtung 30 vernetzt. Diese ist in Block G mit 10 Speicherplätzen für die latenten Zustandsvariablen gezeigt, welche die 10 möglichen Ziffern repräsentieren. Zum Bestimmen der erkannten Ziffer wird die Position des Speicherplatzes mit dem höchsten Wert in der Verteilung der latenten Zustandsvariablen errechnet und als Ausgabedaten verwendet.

Hardware-Konfiguration einer Datenverarbeitungsvorrichtung

Die Figuren 8 bis 26 zeigen Einzelheiten einer Hardware-Konfiguration einer Datenverarbeitungsvorrichtung gemäß der Erfindung (insbesondere gemäß den Darstellungen in den Figuren 1 bis 7), wie sie durch eine oder mehrere integrierte Schaltung(en) realisiert werden kann.

Die Figuren 8 bis 13 zeigen zunächst Grundbausteine, die in der Datenverarbeitungsvorrichtung verwendet werden. Grundbaustein * in der integrierten Schaltung gemäß Figur 8 ist für die Multiplikation zweier Zahlen in Pipeline-Bauweise konfiguriert. In jedem Zeitschritt werden ein Werte-

paar von Eingabevariablen (A, B) sowie ein Ordnungsparameter i eingegeben. Nach einer Verzögerung durch die Ausführung der Operation verlässt das Ergebnis der Multiplikation und der Ordnungsparameter i, der das A & B Paar bei seiner Verarbeitung begleitet, den Grundbaustein *. Dieser Grundbaustein wird in den folgenden Figuren durch das Zeichen * repräsentiert.

Grundbaustein + in der integrierten Schaltung gemäß Figur 9 ist für die Addition zweier Zahlen in Pipeline-Bauweise konfiguriert. In jedem Zeitschritt werden ein Wertepaar von Eingabevariablen (A, B) und ein Ordnungsparameter i eingegeben. Nach einer Verzögerung durch die Ausführung der Operation verlässt das Ergebnis der Addition und der Ordnungsparameter i, der das A & B Paar bei seiner Verarbeitung begleitet, den Grundbaustein. Dieser Grundbaustein wird in den folgenden Figuren durch das Zeichen + repräsentiert.

Figur 10 zeigt den Grundbaustein / in der integrierten Schaltung. In diesem werden zunächst ein Wert A und der Wert 1 addiert. Aus A+1 wird in sequentieller Weise 1/(A+1) ausgerechnet. Nach der Beendigung dieser Operation wird B mit 1/(A+1) durch eine Multiplikationspipeline multipliziert. A ändert sich deutlich seltener als B. i ist ein Ordnungsparameter zu B, der das B während seiner Verarbeitung begleitet. Dieser Grundbaustein wird in den folgenden Figuren durch das Zeichen / repräsentiert.

In Figur 11 ist der Grundbaustein MEM in der integrierten Schaltung gezeigt, der Speicherplätze in einem Merkmalsmodul mit Speicherplätzen 23, 24 für die latenten Zustandsvariablen $h(i)$ und den dazugehörigen Gewichtswerte $p(s|i)$ beinhaltet (siehe Figur 2). Das Modul MEM hat die drei Funktionen Lesen (Figur 11a), Schreiben der latenten Variablen (Figur 11b) und Schreiben der Gewichte (Figur 11c). Auf der Basis eines Paares (s,i) können gemäß Figur 11A die beiden Arten von Speicherplätzen 23, 24 ausgelesen werden. Alternativ kann z.B. bei Anliegen eines neuen Eingabe-Musters die Ausgabe von $h(i)$ durch einen Initialwert (hier $1/N_H$, wobei N_H der Anzahl der verwendeten Speicherplätze für die latenten Zustandsvariablen entspricht) ersetzt werden, was das zusätzliche initiales Überschreiben der Speicherplätze der latenten Zustandsvariablen einspart. Um den richtigen Speicherplatz für $p(s|i)$ zu finden, wird ein Speicherplatzpositionsrechner verwendet, der Teil der Recheneinheit 26 (siehe Figur 2) ist. Als Reaktion auf ein Paar (s,i), liefert der Grundbaustein MEM die dazugehörigen $h(i)$, $p(s|i)$ und den dazugehörigen Ordnungsparameter i. Figur 11b zeigt die Konfiguration für das Schreiben in die Speicherplätze $h(i)$ 23 und Figur 11c zeigt die Konfiguration für das Schreiben der Gewichte $p(s|i)$. Dieser Grundbaustein wird in den folgenden Figuren durch das Zeichen MEM repräsentiert.

Figur 12 illustriert den Grundbaustein MULTI-NORM in der integrierten Schaltung. Dieser Baustein erfüllt mehrere Aufgaben: 1.) eine Kombination $h(i)$, $p(s|i)$ und i wird durch eine Multiplikationspipeline gemäß Figur 8 in $h(i)p(s|i)$ umgerechnet und von dem dazugehörigen Ordnungsparameter i begleitet. 2.) Das Ergebnis der Multiplikationspipeline gemäß Figur 8 wird von einem Sub-Baustein erfasst, und daraus wird die kumulative Summe über alle $h(i)p(s|i)$ gebildet. Wenn alle i Tupel abgearbeitet wurden, wird $X/\sum_i h(i)p(s|i)$ ausgerechnet. Wenn parallel zur Aktualisierung der latenten Zustandsvariablen $h(i)$ auch ein Lernen der dazugehörenden Gewichte durchgeführt werden soll, ist es effizient, parallel $\gamma(t)/\sum_i h(i)p(s|i)$ auszurechnen, wobei $\gamma(t)$ der aktuellen Lernrate entspricht. Nach dem Ende der Division bleibt das Ergebnis erhalten, auch wenn weitere Berechnungen in der Multiplikationspipeline durchgeführt werden, bis das Ergebnis durch einen Reset gelöscht wird. Dieser Grundbaustein wird in den folgenden Figuren durch das Zeichen MULTI-NORM repräsentiert.

Schließlich zeigt Figur 13 den Grundbaustein INDEX GENERATOR in der integrierten Schaltung. Die Aufgabe dieses Bausteins ist es, die Werte der latenten Zustandsvariablen eines Moduls (Merkmalsmodul, Eingabe-Einrichtung oder Ausgabe-Einrichtung) und einer Zufallszahl R_1 in entsprechende Indizes umzurechnen. Für die Berechnung jedes neuen Index-Wertes der Eingabe- oder Modul-Indizes wird die Anzahl der aktiven Speicherplätze der latenten Zustandsvariablen N_H gesetzt und eine Zufallszahl R_1 zur Verfügung gestellt (z.B. eine 32 Bit Zufallszahl aus einem Mersenne Twister MT 19937). Es werden nun Paare an latenten Zustandsvariablen $h(i)$ und deren Ordnungsparameter i an den INDEX GENERATOR übergeben. Dieser Grundbaustein konvertiert $h(i)$ in das Zahlenformat der Zufallszahl (z.B. von Fließkommazahl in Fixpunktzahl) und summiert die gezeigten $h(i)$ schrittweise zu $\sigma(i)$ kumulativ auf. Wenn $\sigma(i)$ größer oder gleich R_1 wird, wird dieser i Wert gespeichert und als Index-Wert weitergegeben. Sollte i den Wert N_H annehmen, ohne dass $\sigma(i)$ größer oder gleich R_1 wird, wird N_H als neuer Index-Wert weitergegeben.

Ein Standard-Vorgehen beim Lernen von Gewichten in herkömmlichen Netzwerken wird als "Abkühlen" bezeichnet. Durch dieses Verfahren soll vermieden werden, dass die Gewichte in lokalen Minima der Zielfunktion (hier: das probabilistische Ziel-Maß) hängen bleiben. Zum optional vorgesehenen "Abkühlen" in der erfindungsgemäßen Datenverarbeitungsvorrichtung wird die Erzeugung der Indizes modifiziert. Anstatt direkt aus den internen Zustandsvariablen die Indizes stochastisch zu ziehen, wird eine Kopie der internen Zustandsvariablen erzeugt, auf die ein Offset addiert wird. Dies wird dann wieder normalisiert und zum Ziehen der Indizes verwendet. Über die Lernschritte wird die Größe des Offsets immer weiter verringert.

Figur 14 zeigt beispielhaft einen Rechenblock der Recheneinheit jedes Merkmalsmoduls, der zum Erzeugen von Indizes mit dem Verfahren zum "Abkühlen" der Eingabe-Wahrscheinlichkeit $p^*(s)$ bzw. Merkmalsmodul-Wahrscheinlichkeit $h(i)$ beim Lernen konfiguriert ist. Bei einigen Lern-Problemen ist es bevorzugt, die Wahrscheinlichkeit, aus der die Indizes erzeugt werden, nach der Gleichung $h^{neu}(i) = \alpha(t)/N_H + (1 - \alpha(t))h(i)$ bzw. $p^{*,neu}(i) = \alpha(t)/N_S + \big(1 - \alpha(t)\big)p^*(s)$ zu glätten. $\alpha(t)$ wird dann über das Lernen immer weiter reduziert. Eine effiziente Art, aus diesen modifizierten Wahrscheinlichkeitsverteilungen Indizes zu erzeugen, ist es, einen doppel-stochastischen Prozess zu verwenden. Einerseits werden auf der Basis einer Zufallszahl R_1 zwei Indizes gezogen, die einen Index aus der Verteilung der latenten Zustandsvariablen $h(i)$ (bzw. $p^*(s)$) und einen Index aus der Gleichverteilung $1/N_H$ bzw. $1/N_S$ (N_H bzw. N_S entsprechen der Anzahl der jeweils aktiven Speicherplätze) umfassen. Mit der Hilfe einer zweiten Zufallszahl R_2 wird auf Basis der Wahrscheinlichkeit $\alpha(t)$ bestimmt, welcher der beiden aus der Zufallszahl R_1 bestimmten Indizes als Ergebnis verwendet wird. Dieser Rechenblock besteht aus drei Grundbausteinen INDEX GENERATOR.

Figur 15 zeigt einen Rechenblock der Recheneinheit jedes Merkmalsmoduls, der zum Erzeugen eines normalisierten Offsets auf den Gewichten beim Lernen konfiguriert ist. Es kann von Vorteil sein, dafür zu sorgen, dass die Gewichte $p(s|i)$ beim Lernen nicht den Wert einer perfekten Null annehmen. Dies kann durch die Addition eines Offsets erfolgen. Dabei wird aber dafür gesorgt, dass trotzdem die Normierung $\sum_s p^{Neu}(s|i) = 1$ erhalten beibt. Um dies zu erfüllen, kann dieser Rechenblock an den Ausgang des Grundbaustein MEM gekoppelt werden (wird in den folgenden Zeichnungen nicht gezeigt, um die Übersichtlichkeit der Zeichnungen zu erhalten). Zunächst wird dem Rechenblock gemäß Figur 15 zur Vorbereitung der Wert des Offsets φ und $1/N_S$ mitgeteilt, wobei N_S der Anzahl der möglichen s in $p(s|i)$ entspricht. Mit diesen Informationen wird $\frac{\varphi}{N_S}$ berechnet. Mit Hilfe der Grundbausteine + und / wird der normalisierte Offset auf den Gewichten in einer Pipeline-weise ausgeführt.

Figur 16 zeigt einen Rechenblock der Recheneinheit jedes Merkmalsmoduls, der zum Berechnen einer Wahrscheinlichkeitsverteilung $\hat{p}(s)$ aus beobachteten Indizes konfiguriert ist. Die beobachteten Index-Werte s werden verwendet, um die Anzahl der bisher beobachteten Indizes (mit diesem Index-Wert s) $c(s)$ aus ihrem Speicherplatz zu lesen, anschließend um den Wert 1 zu erhöhen und wieder in den Speicherplatz $c(s)$ zu schreiben. Parallel dazu wird immer der Speicherwert c_{All} gelesen und um den Wert 1 erhöht und wieder in den Speicher geschrieben. Wenn die

Wahrscheinlichkeitsverteilung benötigt und abgerufen wird, wird C_{All} in ein passendes Zahlenformat überführt und $\frac{1}{C_{All}}$ wird berechnet. Dann wird ein Index-Wert s angelegt, der dazu führt, dass der Speicherplatz $c(s)$ gelesen wird, in ein passendes Zahlenformat überführt wird, und dann durch eine Multiplikationspipeline mit $\frac{1}{C_{All}}$ multipliziert wird. Das Ergebnis dieser Berechnung verlässt dann diesen Rechenblock gemäß Figur 16 und wird in den Figuren 22 und 23 als Eingabe verwendet.

Die Phasen der Dynamik der latenten Zustandsvariablen ist in den Figuren 17 und 18 illustriert: Figur 17 zeigt Phase 1 der Aktualisierung der latenten Zustandsvariablen $h(i)$ auf der Basis des Index-Werts s. Der Index-Wert s wird von außen vorgegeben. Alle i aus $\{1, \cdots, N_H\}$ werden nacheinander an den Grundbaustein MEM angelegt. Alle N_HSpeicherplätze der latenten Zustandsvariablen $h(i)$und den dazugehörigen N_HGewichtswerte $p(s|i)$ werden abgerufen. Optional können diese Werte durch den Rechenblock „normalisierten Offset“ geleitet werden. Danach enden die Werte in dem Grundbaustein MULTI-NORM. Nach Durchlaufen aller N_H Wertepaare erhält man $\epsilon(t)/\sum_i h(i)p(s|i)$, was die Grundlage für Phase 2 bildet.

Figur 18 zeigt Phase 2 der Aktualisierung der latenten Zustandsvariablen $h(i)$ auf der Basis des Index-Werts s. Der Index-Wert s wird von außen vorgegeben. Alle i aus $\{1, \cdots, N_H\}$ werden nacheinander an den Grundbaustein MEM angelegt. Alle N_HSpeicherplätze der latenten Zustandsvariablen $h(i)$und den dazugehörigen N_HGewichtswerte $p(s|i)$ werden abgerufen. Optional können diese Werte durch den Rechenblock „normalisierten Offset“ geleitet werden. Der Grundbaustein MULTI-NORM liefert $h(i)p(s|i)$ und hält noch $\epsilon(t)/\sum_i h(i)p(s|i)$ aus Phase 1 vor. Mit der Hilfe einer Serie von Grundbaustein (*, + und \) werden die latenten Zustandsvariablen $h(i)$ entsprechend aktualisiert. Die neuen Werte werden einerseits in die dazugehörigen Speicherplätze in MEM für die latenten Zustandsvariablen geschrieben, andererseits beobachtet der Grundbaustein INDEX GENERATOR (bzw. der Rechenblock zum Erzeugen von Indizes mit „Abkühlen“) parallel die neuen $h(i)$ Werte, um einen neuen Index für dieses Merkmalmodul (oder dieser Ausgabe-Einheit) zu berechnen.

Die Phasen des Online-Lernens der Gewichte $p(s|i)$ ist in den Figuren 19 bis 21 illustriert: Figur 19 zeigt Phase 1 des Online-Lernens der Gewichte $p(s|i)$. Parallel zur Phase 1 der Aktualisierung der latenten Zustandsvariablen gemäß Figur 17 wird $\epsilon(t)/\sum_i h(i)p(s|i)$ berechnet. Figur 20 zeigt Phase 2 des Online-Lernens der Gewichte $p(s|i)$. Parallel zur Phase 2 der Aktualisierung der latenten Zustandsvariablen $h(i)$ gemäß Figur 18 werden mit der Hilfe eines weiteren Grundbausteins *

die Werte $\gamma(t)h(i)p(s|i)/\sum_i h(i)p(s|i)$ ausgerechnet und in den Speicherplätzen einer Hilfsvariablen $U(i)$ gespeichert. Figur 21 zeigt Phase 3 des Online-Lernens der Gewichte $p(s|i)$. Der externe Index, der zur Aktualisierung der latenten Zustandsvariablen genutzt wurde, wird im Folgenden als s_t weiter bereitgehalten. Der Speicherplatz i der Hilfsvariablen $U(i)$ wird gelesen, und ein Grundbaustein \ wird mit diesem Wert bestückt. Dann werden alle N_SWerte für s durchlaufen. Im Fall, dass $s == s_t$, wird $U(i)$ zu $p(s|i)$ addiert. Der Grundbaustein \ multipliziert dann $p(s|i)$ bzw. $p(s|i) + U(i)$ mit $\frac{1}{1+U(i)}$. Das Ergebnis dieser Berechnung wird dann in die entsprechenden Speicherplätze $p(s|i)$ in MEM geschrieben.

Das Lernens der Gewichte $p(s|i)$ auf Basis einer Sammlung von Mustern ist in den Figuren 22 bis 23 illustriert. Figur 22 zeigt Phase 1 des Lernens der Gewichte $p(s|i)$ auf Basis der Sammlung von Mustern. Nachdem für ein Eingabe-Muster von Eingabedaten ausreichend Indizes zur Aktualisierung der latenten Zustandsvariablen verwendet wurden und mit dem Rechenblock zum Berechnen einer Wahrscheinlichkeitsverteilung $\hat{p}(s)$ aus beobachteten Indizes ausgewertet wurden, wird über alle der N_H aktiven Speicherplätze der latenten Zustandsvariablen $\frac{\hat{p}(s)}{\sum_i h(i)p(s|i)}$ ausgerechnet. Figur 23 zeigt Phase 2 des Lernens der Gewichte $p(s|i)$ auf Basis einer Sammlung von Mustern. Mit Hilfe des Grundbausteins * wird $\frac{\hat{p}(s)}{\sum_i h(i)p(s|i)}$ mit $p(s|i)h(i)$ verrechnet. Das Ergebnis $\frac{\hat{p}(s)p(s|i)h(i)}{\sum_i h(i)p(s|i)}$ für dieses Muster wird zu der Hilfsvariable $W(s|i)$ addiert, in die schon die Informationen von den vorangegangen Mustern akkumuliert wurden. Nachdem alle Muster abgearbeitet wurden, muss $W(s|i)$ mit $p(s|i)$ verrechnet werden. Dies kann auch auf der integrierten Schaltung erfolgen. Alternativ kann $W(s|i)$ und $p(s|i)$ an eine CPU übergeben werden, die dann die Aktualisierung von $p(s|i)$ durchführt. Dies erlaubt den flexiblen Einsatz von komplexeren Aktualisierungsverfahren z.B. basierend auf ADAM oder L4.

Die Organisation der Kommunikation der Merkmalsmodule (einschließlich Eingabe-Einrichtungen und/oder Ausgabe-Einrichtungen) durch gemeinsame Datenbusse A, B ist in Figur 24 illustriert. Alternativ kann das die Datenverarbeitungsvorrichtung mit einem einzigen Datenbus konfiguriert sein. Alle oder eine Untermenge der Merkmalsmodule sind an einen gemeinsamen Datenbus angebunden. Jedes Modul, das mit einem der Datenbusse A, B verbunden ist, hat eine eindeutige Nummer auf jedem dieser Datenbusse A, B. Diese Nummer kann sich je nach Datenbus unterscheiden. Eine Zähler-Kontrolleinheit beobachtet, ob alle Merkmalsmodule mit der Abarbeitung ihrer Index-Speicher fertig sind. Sind alle Merkmalsmodule entsprechend fertig, ruft die Zähler-Kontrolleinheit alle an ihrem Datenbus vorhandenen Merkmalsmodule durch Veröffentlichung

derer eindeutigen Nummern einmal auf. In einem definierten zeitlichen Abstand (der für alle Merkmalsmodule, Eingabe- und Ausgabemodule gleich ist) reagieren die aufgerufenen Merkmalsmodule auf die Nennung ihrer Nummer. Als Reaktion wird, falls dieses aufgerufene Merkmalsmodule Informationen besitzt, welche den anderen Merkmalsmodule mitgeteilt werden muss, die eigene Merkmalsmodul-Nummer und der generierte Index-Wert über den gemeinsamen Datenbus an alle anderen angeschlossenen Merkmalsmodule übermittelt. Es kann von Vorteil sein, mehrerer dieser Datenbusse zu betreiben, und die Datenbusse können sich über mehrere integrierte Schaltungen erstrecken. Die Zähler-Kontrolleinheit wird wiederum durch eine übergeordnete Kontrolleinheit (Steuereinrichtung 40, siehe Figuren 1 und 26) gesteuert. Haben alle an einem Datenbus angeschlossenen Merkmalsmodule ihre Verarbeitung beendet, teilt dies die Zähler-Kontrolleinheit der übergeordneten Kontrolleinheit mit und wartet auf einen Befehl, der die nächste Runde an Datenaustausch und Abarbeitung der errechneten Index-Werte anstößt.

Figur 25 illustriert einen gemäß einer vorteilhaften Variante der Erfindung vorgesehenen Modul-Filter innerhalb eines Merkmalmoduls und der Ausgabe-Einrichtung. Alle Merkmalsmodule an dem gemeinsamen Datenbus teilen ihre Information über ihre Indizes allen anderen angeschlossenen Merkmalsmodulen mit. Zur Realisierung einer geeigneten Architektur kann es aber von Vorteil sein, wenn (während der Laufzeit) definierbar ist, auf welche anderen Merkmalsmodule jedes Merkmalsmodul reagieren soll. Hierzu hat jedes Merkmalsmodul für jeden der Datenbusse, an denen dieses angeschlossen ist, eine Liste. Nur wenn die eindeutige Nummer eines empfangenden Merkmalsmoduls in dieser Liste vorhanden ist, wird der eingehende Index der sendenden Gruppe in die Index-Speicherliste aufgenommen. Die Liste enthält noch weitere Informationen für jedes der zu beobachtenden Merkmalsmodule, wie z.B. einen Index-Offset s_{Offset} , welcher von der eingehenden Modul-Nummer abhängt, damit zwischen dem empfangenen Index-Wert s_{in} und den Speicherplätzen für die Gewichte $p(s|i)$ eine Zuordnung stattfinden kann. So kann z.B eine Umrechnung des eingegangenen Index-Wertes s_{in} auf die Gewichtsmatrix in folgender Weise erfolgen: $p(s = s_{in} + s_{Offset}\,|i)$. Auch kann die Liste individuelle ϵ und Lernraten γ für jedes sendende Modul beinhalten. Wenn die Zähler-Kontrolleinheit dem Modul mitteilt, dass alle Indizes ausgetauscht sind, arbeitet das Modul die in der Index-Speicherliste gesammelten Indizes ab. Abgearbeitete Indizes werden aus der Liste entfernt. Ist die Liste leer, wird der letzte bei der Aktualisierung erzeugte eigene Index-Wert in einen Speicher kopiert. Der Zähler-Kontrolleinheit wird mitgeteilt, dass die Liste abgearbeitet ist. Kommt nun ein neuer Aufruf der Zähler-Kontrolleinheit, werden die eindeutige Nummer des Moduls und der gespeicherte eigene Index über den Datenbus entsprechend an alle angeschlossenen Module veröffentlicht.

Figur 26 illustriert Einzelheiten eines Datenbusses für Steuerinformationen. Es ist für viele Anwendungen von Vorteil, Informationen an die Eingabe-Einrichtung 10, Merkmalsmodule 21, Ausgabe-Einrichtung 30 und diverse Kontrolleinheiten (40) des Netzwerkes 20 von außen zuzuführen oder Daten über den aktuellen Zustand aus dem Netzwerk auszulesen (siehe Figur 1). Des Weiteren sollen die ausgehenden Datenflüsse der verschiedenen Merkmalsmodule zusammengeführt werden. Dies geschieht vorzugsweise über einen gemeinsamen Datenbus mit einer „Arbiter"-Weiche. Auf der Seite der eingehenden Daten in das Netzwerk ist es effizienter, wenn Information über Parameter oder andere Befehle, die immer nach einer bestimmten Anzahl an Indizes auftreten (z.B. Wechsel der Lernrate nach einer vorgegebenen Anzahl an Aktualisierungen der latenten Zustandsvariablen), nicht zur richtigen Zeit von außen zugeführt werden müssen. Dies kann zu nicht gewünschten Wartezeiten bis zum Eintreffen der notwendigen Information über den Wechsel der Parameter führen. Um dies zu vermeiden, kann ein internes Nachrichten-Speicher-System 44 verwendet werden. Es erfasst die Anzahl der verarbeiteten Eingabe- oder Modul-Indizes und abhängig davon sendet es dort abgelegte Nachrichten an die Merkmalsmodule. Ein Zufallsgenerator versorgt in diesem Fall alle Merkmalsmodule mit neuen Zufallszahlen und ein „Endlicher Automat" (Finite State Machine) sorgt für den geordneten Ablauf und die autonome Abarbeitung einer von außen vorgegeben Anzahl an Aktualisierungen der latenten Zustandsvariablen und dem Einsetzen eines optionalen online Lernens.

Mathematische Grundlagen der Datenverarbeitung

Im Folgenden werden mathematische Gleichungen für das Beispiel eines Netzwerkes mit vier Merkmalsmodulen (Eingabe-Einrichtung, Ausgabe-Einrichtung und zwei Merkmalsmodule, z. B. gemäß Figur 3) präsentiert. Diese lassen sich in entsprechender Weise auf Netzwerke mit mehr Merkmalsmodulen anwenden und verallgemeinern. Bei der Anwendung numerischer Verfahren auftretende Parameter werden, soweit sie sich nicht aus den Daten ergeben, empirisch bestimmt.

Aktualisieren der latenten Zustandsvariablen und Lernen der Gewichte

Im Allgemeinen wird beim Aktualisieren der latenten Zustandsvariablen und Lernen der Gewichte ein Abstandsmaß zwischen der Wahrscheinlichkeit $\hat{p}(s^v)$, welche zur Erzeugung des Index s^v verwendet wird, und der Rekonstruktion $p(s^v|h_\zeta)$ basierend auf der latenten Variable h_ζ betrachtet und optimiert. Ein Beispiel ist:

$$Abstand = -\sum_{\mu}\sum_{\upsilon}\sum_{\zeta}\psi_{\upsilon,\zeta}\sum_{s^{\upsilon}}\hat{p}(s^{\upsilon})\log\left(\sum_{i^{\zeta}}w_{\upsilon\to\zeta}(s^{\upsilon}|i^{\zeta})\,h_{\zeta,\mu}(i^{\zeta})\right)$$

wobei μ einen Satz an Eingabedaten (Muster) durchnummeriert (z.B. Bilder, Zeitreihen), υ alle Merkmalsmodule sowie Eingabe- und Ausgabemodule durchnummeriert, ζ alle Merkmalsmodule und Ausgabemodule durchläuft, wobei $\zeta \neq \upsilon$ gilt. s^{υ} und i^{ζ} entspricht den Speicherplätzen für latente Variablen bzw. der Eingabedatenwahrscheinlichkeiten. $h_{\zeta,\mu}(i^{\zeta})$ mit $0 \leq h_{\zeta,\mu}(i^{\zeta}) \leq 1$ und $\sum_{i^{\zeta}} h_{\zeta,\mu}(i^{\zeta}) = 1$ für die latenten Variablen und $w_{\upsilon\to\zeta}(s^{\upsilon}|i^{\zeta})$ für die zugeordneten Gewichte mit $0 \leq w_{\upsilon\to\zeta}(s^{\upsilon}|i^{\zeta}) \leq 1$ und $\sum_{s^{\upsilon}} w_{\upsilon\to\zeta}(s^{\upsilon}|i^{\zeta}) = 1$. $\psi_{\upsilon,\zeta}$ kann die Werte 0 oder 1 annehmen und repräsentiert die Architektur des Netzwerkes.

Aus dem Abstand kann nun eine Regel zur Aktualisierung der latenten Zustandsvariablen mit der Hilfe eines Gradientenverfahrens errechnet werden. In diesem Fall ist es z.B.:

$$h^{neu}{}_{\zeta\prime,\mu}(i^{\zeta\prime}) = h_{\zeta\prime,\mu}(i^{\zeta\prime}) - \epsilon_{\upsilon,\zeta\prime}\,h_{\zeta\prime,\mu}(i^{\zeta\prime})\frac{\partial Abstand}{\partial h_{\zeta\prime,\mu}(i^{\zeta\prime})}$$

mit $\epsilon_{\upsilon,\zeta\prime}$ als einem, in Abhängigkeit von der Anzahl der zu verarbeitenden Indizes veränderlichen Parameter. Um die zu verarbeitende Information auf die aktuell beobachteten eingehenden und verarbeiteten Indizes s_t^{ν} zu beschränken, wird $\hat{p}(s^{\upsilon})$ durch $\delta_{s_t^{\nu},s_t}$ ersetzt, wobei $\delta_{i,j}$ das Kronecker-Delta ist. Zur Erhaltung der Normierung muss

$$h^{neu,norm}{}_{\zeta\prime,\mu}(i^{\zeta\prime}) = h^{neu}{}_{\zeta\prime,\mu}(i^{\zeta\prime})/\sum_{i^{\zeta}} h^{neu}{}_{\zeta\prime,\mu}(i^{\zeta})$$

impliziert oder expliziert durchgeführt werden, um den aktualisierten Wert für die latente Variable zu erhalten.

Das Lernen der Gewichte basiert ebenfalls auf Gradientenverfahren. Dabei wird ein multiplikatives Lernverfahren durch

$$w^{neu}_{\upsilon'\to\zeta'}(s^{\upsilon'}|i^{\zeta'}) = w_{\upsilon\prime\to\zeta\prime}(s^{\upsilon\prime}|i^{\zeta\prime}) - \gamma_{\upsilon\prime,\zeta\prime}\,w_{\upsilon\prime\to\zeta\prime}(s^{\upsilon\prime}|i^{\zeta\prime})\frac{\partial Abstand}{\partial w_{\upsilon\prime\to\zeta\prime}(s^{\upsilon\prime}|i^{\zeta\prime})}$$

und ein additives Lernverfahren durch

$$w^{neu}_{\upsilon'\to\zeta'}(s^{\upsilon'}|i^{\zeta'}) = w_{\upsilon\prime\to\zeta\prime}(s^{\upsilon\prime}|i^{\zeta\prime}) - \gamma_{\upsilon\prime,\zeta\prime}\frac{\partial Abstand}{\partial w_{\upsilon\prime\to\zeta\prime}(s^{\upsilon\prime}|i^{\zeta\prime})}$$

dargestellt, wobei $\gamma_{\upsilon\prime,\zeta\prime}$ einer Lernrate entspricht, die sich über die Lernschritte verändern kann. Es wird die Normierung der Gewichte sichergestellt, so dass

$$w^{neu,norm}_{\upsilon'\to\zeta'}(s^{\upsilon'}|i^{\zeta'}) = w^{neu}_{\upsilon'\to\zeta'}(s^{\upsilon'}|i^{\zeta'})/\sum_{s^{\upsilon}} w^{neu}_{\upsilon'\to\zeta'}(s^{\upsilon}|i^{\zeta'})$$

impliziert oder expliziert durchgeführt wird. Diese Lernregeln können nun als Batch-Lernregel angewendet werden, wobei die gesammelten Informationen über mehr als das aktuelle Muster μ für eine Aktualisierung der Gewichte verwendet werden, oder als Online-Lernregel, bei der nur der aktuelle Index für eine Aktualisierung der Gewichte verwendet wird.

Beispiel-Netzwerk mit zwei Merkmalsmodulen

Die folgenden Gleichungen werden für ein Beispiel-Netzwerk mit einer Eingabe-Einrichtung 10 (EE) X (mit einer Eingabe-Wahrscheinlichkeitsverteilung $p^*(s), s = 1, \dots, N_S$) und einer Ausgabe-Einrichtung 30 (AE) Y (mit den latenten Zustandsvariablen $h_{y,\mu}(i), i = 1, \dots, N_y$) beschrieben. Zwischen diesen beiden Einrichtungen befinden sich zwei Merkmalsmodule MM1 (mit den latenten Zustandsvariablen $h_{1,\mu}(i), i = 1, \dots, N_{H1}$) und Merkmalsmodule MM2 (mit den latenten Zustandsvariablen $h_{2,\mu}(i), i = 1, \dots, N_{H2}$), siehe Figur 3. Dabei nummeriert μ das aktuelle Muster $r_\mu(s)$, welches gerade verarbeitet oder gelernt wird. In diesem Beispiel Netzwerk ist MM1 mit der Eingabe-Einrichtung 10 EE X verbunden und MM2 ist mit der Ausgabe-Einrichtung 30 AE Y verbunden. Des Weiteren sind MM1 und MM2 miteinander verbunden. Es gilt

$\sum_{s=1}^{N_S} p_\mu^*(s) = 1$, $\sum_{i=1}^{N_Y} h_{y,\mu}(i) = 1$, $\sum_{i=1}^{N_{H1}} h_{1,\mu}(i) = 1$ und $\sum_{i=1}^{N_{H2}} h_{2,\mu}(i) = 1$

sowie

$0 \leq p_\mu{}^*(s) \leq 1$, $0 \leq h_{y,\mu}(i) \leq 1$, $0 \leq h_{1,\mu}(i) \leq 1$und $0 \leq h_{2,\mu}(i) \leq 1$,

da es sich bei diesen Größen um Wahrscheinlichkeiten handelt. Diese Wahrscheinlichkeiten werden dazu verwendet, stochastisch Indizes zu erzeugen, die zum Austausch von Informationen zwischen den EEs, EAs und MMs verwendet werden. Indizes erzeugt aus $p_\mu^*(s)$ werden mit s^x benannt. Indizes aus $h_{y,\mu}(i)$ mit s^y, aus $h_{1,\mu}(i)$ mit s^1 und aus $h_{2,\mu}(i)$ mit s^2.

Die Gewichte des Netzwerkes werden auf einem Ensemble von M Mustern ($\mu = 1, \dots, M$) von Eingabedaten $r_\mu(s) \geq 0$ gelernt, die z. B. Bilder kodieren können. Im Folgenden werden die Gewichte mit $w_{a\to b}$ bezeichnet. Dies entspricht einem Gewichtssatz, bei dem Information von Merkmalsmodul a (dieses Modul erzeugt den Index) nach Merkmalsmodul b fließt. In diesem einfachen Beispiel gibt es die folgenden fünf Gewichtssätze:

$w_{X\to 1}(k|l)$ mit einer Dimension von $N_S \times N_{H1}$ und $\sum_k w_{X\to 1}(k|l) = 1, 0 \leq w_{X\to 1}(k|l) \leq 1$

$w_{1\to 2}(k|l)$ mit einer Dimension von $N_{H1} \times N_{H2}$ und $\sum_k w_{1\to 2}(k|l) = 1, 0 \leq w_{1\to 2}(k|l) \leq 1$

$w_{2\to y}(k|l)$ mit einer Dimension von $N_{H2} \times N_Y$ und $\sum_k w_{2\to y}(k|l) = 1, 0 \leq w_{2\to y}(k|l) \leq 1$

$w_{y\to 2}(k|l)$ mit einer Dimension von $N_Y \times N_{H2}$ und $\sum_k w_{y\to 2}(k|l) = 1, 0 \leq w_{y\to 2}(k|l) \leq 1$

$w_{2\to 1}(k|l)$ mit einer Dimension von $N_{H2} \times N_{H1}$ und $\sum_k w_{2\to 1}(k|l) = 1, 0 \leq w_{2\to 1}(k|l) \leq 1$

Basierend auf den Gewichten und den latenten Zustandsvariablen werden folgende fünf „Rekonstruktionen“ der erzeugenden Wahrscheinlichkeitsverteilung auf der Basis der latenten Zustandsvariablen der Module MM1, MM2 und AE und den jeweilig dazugehörenden Gewichten definiert:

$$p(s^x|h_{1,\mu}) = \sum_i h_{1,\mu}(i) w_{X\to 1}(s^x|i)$$

$$p(s^1|h_{2,\mu}) = \sum_i h_{2,\mu}(i) w_{1\to 2}(s^1|i)$$

$$p(s^2|h_{y,\mu}) = \sum_i h_{y,\mu}(i) w_{2\to y}(s^2|i)$$

$$p(s^2|h_{1,\mu}) = \sum_i h_{1,\mu}(i) w_{2\to 1}(s^2|i)$$

$$p(s^y|h_{2,\mu}) = \sum_i h_{2,\mu}(i) w_{y\to 2}(s^y|i)$$

Ziel ist es, einen Abstand der Rekonstruktionen und deren dazugehörigen erzeugenden Wahrscheinlichkeitsverteilungen zu minimieren, wobei latente Zustandsvariablen in mehreren Rekonstruktionen gleichzeitig vorkommen können. Dies koppelt die Rekonstruktionen aneinander.
Für dieses Beispiel wird der Abstand nach der folgenden Gleichung (ähnlich einer Likelihood oder Kullback-Leibler-Divergenz) berechnet:

$$\begin{aligned} Abstand = -\sum_\mu \Bigg(& \alpha_1 \sum_a \hat{p}_{x,\mu}(a) \log\left(\sum_i w_{x\to 1}(a|i) h_{1,\mu}(i)\right) \\ & + \beta_1 \sum_b \hat{p}_{h_2,\mu}(b) \log\left(\sum_i w_{2\to 1}(b|i) h_{1,\mu}(i)\right) \\ & + \alpha_2 \sum_c \hat{p}_{h_1,\mu}(c) \log\left(\sum_i w_{1\to 2}(c|i) h_{2,\mu}(i)\right) \\ & + \beta_2 \sum_d \hat{p}_{h_y,\mu}(d) \log\left(\sum_i w_{y\to 2}(d|i) h_{2,\mu}(i)\right) \\ & + \alpha_y \sum_e \hat{p}_{h_2,\mu}(e) \log\left(\sum_i w_{2\to y}(e|i) h_{y,\mu}(i)\right)\Bigg) \end{aligned}$$

Dabei entspricht $\hat{p}_{x,\mu}(a)$ der Eingabe-Wahrscheinlichkeit der Eingabe-Einrichtung. Für die anderen drei Wahrscheinlichkeiten $\hat{p}_{h_1,\mu}(i)$, $\hat{p}_{h_2,\mu}(i)$ und $\hat{p}_{h_y,\mu}(i)$ gilt, dass diese aus den „alten“ (vom letzten Zeitschritt) latenten Zustandsvariablen bestehen, aus denen die aktuell zu bearbeitenden Indizes erzeugt wurden. Vor jeder Aktualisierung der latenten Zustandsvariablen findet folgende Zuweisung statt:

$h_{1,\mu}(i) \to \hat{p}_{h_1,\mu}(i)$

$h_{2,\mu}(i) \rightarrow \hat{p}_{h_2,\mu}(i)$

$h_{y,\mu}(i) \rightarrow \hat{p}_{h_y,\mu}(i)$

Dabei sind die α's und β's Konstanten.

Aus dem Abstand zwischen den Rekonstruktionen und den erzeugenden Wahrscheinlichkeiten wird ein Gradienten zur Optimierung für die latenten Zustandsvariablen berechnet:

$$-\frac{\partial Abstand}{\partial h_{1,\mu}(j)} = \alpha_1 \sum_a \frac{\hat{p}_{x,\mu}(a)\, w_{x\to 1}(a|j)}{\sum_i w_{x\to 1}(a|i) h_{1,\mu}(i)} + \beta_1 \sum_b \frac{\hat{p}_{h_2,\mu}(b)\, w_{2\to 1}(b|j)}{\sum_i w_{2\to 1}(b|i) h_{1,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial h_{2,\mu}(j)} = \alpha_2 \sum_c \frac{\hat{p}_{h_1,\mu}(c)\, w_{1\to 2}(c|j)}{\sum_i w_{1\to 2}(c|i) h_{2,\mu}(i)} + \beta_3 \sum_d \frac{\hat{p}_{h_y,\mu}(d)\, w_{y\to 2}(d|j)}{\sum_i w_{y\to 2}(d|i) h_{2,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial h_{y,\mu}(j)} = \alpha_y \sum_e \frac{\hat{p}_{h_2,\mu}(e)\, w_{2\to y}(e|j)}{\sum_i w_{2\to y}(e|i) h_{y,\mu}(i)}$$

Jetzt werden die $\hat{p}$ durch die in einem Zeitschritt tatsächlich beobachteten Indizes ersetzt, was einer Aktualisierung mit jedem einzelnen beobachten Index s_t entspricht:

$$-\frac{\partial Abstand}{\partial h_{1,\mu}(j)} = \alpha_1 \frac{w_{x\to 1}(s_t^x|j)}{\sum_i w_{x\to 1}(s_t^x|i) h_{1,\mu}(i)} + \beta_1 \frac{w_{2\to 1}(s_t^2|j)}{\sum_i w_{2\to 1}(s_t^2|i) h_{1,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial h_{2,\mu}(j)} = \alpha_2 \frac{w_{1\to 2}(s_t^1|j)}{\sum_i w_{1\to 2}(s_t^1|i) h_{2,\mu}(i)} + \beta_2 \frac{w_{y\to 2}(s_t^y|j)}{\sum_i w_{y\to 2}(s_t^y|i) h_{2,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial h_{y,\mu}(j)} = \alpha_y \frac{w_{2\to y}(s_t^2|j)}{\sum_i w_{2\to y}(s_t^2|i) h_{y,\mu}(i)}$$

Daraus folgen Aktualisierungsregeln für eine nicht normierte Version latenten Zustandsvariablen:

$$h_{1,\mu}^{Neu}(j) = h_{1,\mu}(j) + \alpha_1 \frac{w_{x\to 1}(s_t^x|j)\, h_{1,\mu}(j)}{\sum_i w_{x\to 1}(s_t^x|i) h_{1,\mu}(i)} + \beta_1 \frac{w_{2\to 1}(s_t^2|j)\, h_{1,\mu}(j)}{\sum_i w_{2\to 1}(s_t^2|i) h_{1,\mu}(i)}$$

$$h_{2,\mu}^{Neu}(j) = h_{2,\mu}(j) + \alpha_2 \frac{w_{1\to 2}(s_t^1|j) h_{2,\mu}(j)}{\sum_i w_{1\to 2}(s_t^1|i) h_{2,\mu}(i)} + \beta_2 \frac{w_{y\to 2}(s_t^y|j) h_{2,\mu}(j)}{\sum_i w_{y\to 2}(s_t^y|i) h_{2,\mu}(i)}$$

$$h_{y,\mu}^{Neu}(j) = h_{y,\mu}(j) + \alpha_y \frac{w_{2\to y}(s_t^2|i)\, h_{y,\mu}(j)}{\sum_i w_{2\to y}(s_t^2|i) h_{y,\mu}(i)}$$

Diese Zwischenform der latenten Zustandsvariablen $h^{Neu}(j)$ wird dann noch normalisiert $h^{Neu,Norm}(j) = h^{Neu}(j) / \sum_i h^{Neu}(i)$.

Dies ist vorgesehen, um die Wahrscheinlichkeitseigenschaften der latenten Zustandsvariablen zu erhalten, die für die stochastische Erzeugung der Indizes benötigt wird.

Es ist ersichtlich, dass jedes $h^{Neu,Norm}(j)$ nur von den eingehenden Indizes abhängt, was die internen Berechnungen jedes Merkmalsmoduls von den anderen Merkmalsmodulen vollständig isoliert. Die Merkmalsmodule gehen in einen angepassten Zustand relativ zu den Merkmalsmodulen von denen diese Indizes zugesendet bekommen.

In der Anwendung hat sich gezeigt, dass die einzelnen Indizes nacheinander abgearbeitet werden können, was zu einer deutlich vereinfachten Berechnung führt:

$$h_{1,\mu}^{Neu,Norm}(j) = \frac{\left(h_{1,\mu}(j)+\alpha_1 \frac{w_{x\to1}(s_t^x|j)\,h_{1,\mu}(j)}{\sum_i w_{x\to1}(s_t^x|i)h_{1,\mu}(i)}\right)}{1+\alpha_1}$$

$$h_{1,\mu}^{Neu,Norm}(j) = \frac{\left(h_{1,\mu}(j)+\beta_1 \frac{w_{2\to1}(s_t^2|j)\,h_{1,\mu}(j)}{\sum_i w_{2\to1}(s_t^2|i)h_{1,\mu}(i)}\right)}{1+\beta_1}$$

$$h_{2,\mu}^{Neu,Norm}(j) = \frac{\left(h_{2,\mu}(j) + \alpha_2 \frac{w_{1\to2}(s_t^1|j)h_{2,\mu}(j)}{\sum_i w_{1\to2}(s_t^1|i)h_{2,\mu}(i)}\right)}{1+\alpha_2}$$

$$h_{2,\mu}^{Neu,Norm}(j) = \frac{\left(h_{2,\mu}(j) + \beta_2 \frac{w_{y\to2}(s_t^y|j)h_{2,\mu}(j)}{\sum_i w_{y\to2}(s_t^y|i)h_{2,\mu}(i)}\right)}{1+\beta_2}$$

$$h_{y,\mu}^{Neu,Norm}(j) = \frac{\left(h_{y,\mu}(j) + \alpha_y \frac{w_{2\to y}(s_t^2|i)\,h_{y,\mu}(j)}{\sum_i w_{2\to y}(s_t^2|i)h_{y,\mu}(i)}\right)}{1+\alpha_y}$$

Eine ähnliche Gradienten-Berechnung kann nun auch für die fünf Gewichtssätze durchgeführt werden:

$$-\frac{\partial Abstand}{\partial w_{x\to1}(a|j)} = \alpha_1 \sum_\mu \frac{\hat{p}_{x,\mu}(a)\,w_{x\to1}(a|j)}{\sum_i w_{x\to1}(a|i)h_{1,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial w_{1\to2}(a|j)} = \alpha_2 \sum_\mu \frac{\hat{p}_{h_1,\mu}(a)\,w_{1\to2}(a|j)}{\sum_i w_{1\to2}(a|i)h_{2,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial w_{2\to y}(a|j)} = \alpha_y \sum_\mu \frac{\hat{p}_{h_2,\mu}(a)\,w_{2\to y}(a|j)}{\sum_i w_{2\to y}(a|i)h_{y,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial w_{y\to2}(a|j)} = \beta_3 \sum_\mu \frac{\hat{p}_{h_y,\mu}(a)\,w_{y\to2}(a|j)}{\sum_i w_{y\to2}(a|i)h_{2,\mu}(i)}$$

$$-\frac{\partial Abstand}{\partial w_{2\to1}(a|j)} = \beta_1 \sum_\mu \frac{\hat{p}_{h_2,\mu}(a)\,w_{2\to1}(a|j)}{\sum_i w_{2\to1}(a|i)h_{1,\mu}(i)}$$

Hieraus kann man verschiedene Lernregeln herleiten. Zunächst zwei Online-Lernregeln (also Lernen mit jeweils einem Index).

Additive Online-Lernregel:

$$w_{x\to1}^{Neu}(s_t^x|j) = w_{x\to1}(s_t^x|j) + \gamma_1 \frac{h_{1,\mu}(j)}{\sum_i w_{x\to1}(s_t^x|i)h_{1,\mu}(i)}$$

$$w_{1\to 2}^{Neu}(s_t^1|j) = w_{1\to 2}(s_t^1|j) + \gamma_2 \frac{h_{2,\mu}(j)}{\sum_i w_{1\to 2}(s_t^1|i) h_{2,\mu}(i)}$$

$$w_{2\to y}^{Neu}(s_t^2|j) = w_{2\to y}(s_t^2|j) + \gamma_3 \frac{h_{y,\mu}(j)}{\sum_i w_{2\to y}(s_t^2|i) h_{y,\mu}(i)}$$

$$w_{y\to 2}^{Neu}(s_t^y|j) = w_{y\to 2}(s_t^y|j) + \gamma_4 \frac{h_{2,\mu}(j)}{\sum_i w_{y\to 2}(s_t^y|i) h_{2,\mu}(i)}$$

$$w_{2\to 1}^{Neu}(s_t^2|j) = w_{2\to 1}(s_t^2|j) + \gamma_5 \frac{h_{1,\mu}(j)}{\sum_i w_{2\to 1}(s_t^2|i) h_{1,\mu}(i)}$$

Multiplikative Online-Lernregel:

$$w_{x\to 1}^{Neu}(s_t^x|j) = w_{x\to 1}(s_t^x|j) + \gamma_1 \frac{w_{x\to 1}(s_t^x|j)\, h_{1,\mu}(j)}{\sum_i w_{x\to 1}(s_t^x|i) h_{1,\mu}(i)}$$

$$w_{1\to 2}^{Neu}(s_t^1|j) = w_{1\to 2}(s_t^1|j) + \gamma_2 \frac{w_{1\to 2}(s_t^1|j)\; h_{2,\mu}(j)}{\sum_i w_{1\to 2}(s_t^1|i) h_{2,\mu}(i)}$$

$$w_{2\to y}^{Neu}(s_t^2|j) = w_{2\to y}(s_t^2|j) + \gamma_3 \frac{w_{2\to y}(s_t^2|j)\; h_{y,\mu}(j)}{\sum_i w_{2\to y}(s_t^2|i) h_{y,\mu}(i)}$$

$$w_{y\to 2}^{Neu}(s_t^y|j) = w_{y\to 2}(s_t^y|j) + \gamma_4 \frac{w_{y\to 2}(s_t^y|j)\; h_{2,\mu}(j)}{\sum_i w_{y\to 2}(s_t^y|i) h_{2,\mu}(i)}$$

$$w_{2\to 1}^{Neu}(s_t^2|j) = w_{2\to 1}(s_t^2|j) + \gamma_5 \frac{w_{2\to 1}(s_t^2|j)\; h_{1,\mu}(j)}{\sum_i w_{2\to 1}(s_t^2|i) h_{1,\mu}(i)}$$

Nach jedem Lernschritt (Multiplikative oder Additive Lernregel) werden die Gewichte normiert:

$$w_{x\to 1}^{Neu,Norm}(s|j) = \frac{w_{x\to 1}^{Neu}(s|j)}{\sum_r w_{x\to 1}^{Neu}(r|j)}$$

$$w_{1\to 2}^{Neu,Norm}(s|j) = \frac{w_{1\to 2}^{Neu}(s|j)}{\sum_r w_{1\to 2}^{Neu}(r|j)}$$

$$w_{2\to y}^{Neu,Norm}(s|j) = \frac{w_{2\to y}^{Neu}(s|j)}{\sum_r w_{2\to y}^{Neu}(r|j)}$$

$$w_{y\to 2}^{Neu,Norm}(s|j) = \frac{w_{y\to 2}^{Neu}(s|j)}{\sum_r w_{y\to 2}^{Neu}(r|j)}$$

$$w_{2\to 1}^{Neu,Norm}(s|j) = \frac{w_{2\to 1}^{Neu}(s|j)}{\sum_r w_{2\to 1}^{Neu}(r|j)}$$

Diese Lernregeln haben unterschiedliche Eigenschaften, wobei vorzugsweise getestet wird, welche am besten bei welchem Problem funktioniert. Beide Online-Lernregeln wurden erfolgreich zum Lernen von 4Bit Booleschen Parity-Funktionen angewendet.

Außerdem kann man Batch-Lernregeln (ein Ensemble an Mustern und mehrere Indizes pro Muster) mit Hilfe des Gradienten $-\frac{\partial Abstand}{\partial w_{x\to 1}(a|j)}$ aufstellen, wie z.B. als multiplikative Batch-Lernregel:

$$w_{x\to 1}^{Neu}(s|j) = \eta_1 w_{x\to 1}(s|j) + \phi_1 \sum_{\mu} \frac{\hat{p}_{x,\mu}(s)\, w_{x\to 1}(s|j) h_{1,\mu}(j)}{\sum_i w_{x\to 1}(s|i) h_{1,\mu}(i)}$$

$$w_{1\to 2}^{Neu}(s|j) = \eta_2\, w_{1\to 2}(s|j) + \phi_2 \sum_{\mu} \frac{\hat{p}_{h_1,\mu}(s)\, w_{1\to 2}(s|j)\, h_{2,\mu}(j)}{\sum_i w_{1\to 2}(s|i) h_{2,\mu}(i)}$$

$$w_{2\to y}^{Neu}(s|j) = \eta_3\, w_{2\to y}(s|j) + \phi_3 \sum_{\mu} \frac{\hat{p}_{h_2,\mu}(s)\, w_{2\to y}(s|j) h_{y,\mu}(i)}{\sum_i w_{2\to y}(s|i) h_{y,\mu}(i)}$$

$$w_{y\to 2}^{Neu}(s|j) = \eta_4\, w_{y\to 2}(s|j) + \phi_4 \sum_{\mu} \frac{\hat{p}_{h_y,\mu}(s)\, w_{y\to 2}(s|j)\, h_{2,\mu}(i)}{\sum_i w_{y\to 2}(s|i) h_{2,\mu}(i)}$$

$$w_{2\to 1}^{Neu}(s|j) = \eta_5\, w_{2\to 1}(s|j) + \phi_5 \sum_{\mu} \frac{\hat{p}_{h_2,\mu}(s)\, w_{2\to 1}(s|j) h_{1,\mu}(i)}{\sum_i w_{2\to 1}(s|i) h_{1,\mu}(i)}$$

wobei die η's und ϕ's Lernraten entsprechen und wieder, nach jedem Lernschritt, eine Normalisierung erfolgen muss:

$$w_{x\to 1}^{Neu,Norm}(s|j) = \frac{w_{x\to 1}^{Neu}(s|j)}{\sum_r w_{x\to 1}^{Neu}(r|j)}$$

$$w_{1\to 2}^{Neu,Norm}(s|j) = \frac{w_{1\to 2}^{Neu}(s|j)}{\sum_r w_{1\to 2}^{Neu}(r|j)}$$

$$w_{2\to y}^{Neu,Norm}(s|j) = \frac{w_{2\to y}^{Neu}(s|j)}{\sum_r w_{2\to y}^{Neu}(r|j)}$$

$$w_{y\to 2}^{Neu,Norm}(s|j) = \frac{w_{y\to 2}^{Neu}(s|j)}{\sum_r w_{y\to 2}^{Neu}(r|j)}$$

$$w_{2\to 1}^{Neu,Norm}(s|j) = \frac{w_{2\to 1}^{Neu}(s|j)}{\sum_r w_{2\to 1}^{Neu}(r|j)}$$

Anstatt der hier gezeigten multiplikativen Batch-Lernregen kann man analog auch Lernregeln basierend auf moderneren Lernverfahren wie Adam [20] oder L4 [21] aufsetzen.

Das hier gezeigte Batch-Lernen hat sich bei Deep-Convolutional Netzwerken als besonders nützlich erwiesen. Z.B. konnte von den Erfindern erfolgreich ein Deep-Convolutional Netzwerk zur Erkennung von handgeschriebenen Ziffern trainiert (gelernt) werden. Dabei wurde die hier gezeigte multiplikative Batch-Lernregel erfolgreich angewandt.

Es hat sich auch als vorteilhaft erwiesen, das Netzwerk schichtweise vorzulernen, wobei der Zustand der vorangegangenen Schichten zunächst unverändert gehalten wird und erst danach unter der vollen Dynamik zu Ende zu lernen.

Um aus der latenten Zustandsvariable $h_{y,\mu}(i)$ eine Klassifizierung zu berechnen gibt es mehrere Möglichkeiten. Zum Beispiel kann man

$$Klasse = argmax_i\, h_{y,\mu}(i)$$

verwenden, wobei der Zusammenhang zwischen Klasse und i während des Lernens des Netwerks festgelegt wurde. Oder analog kann man die von der Ausgabe-Einrichtung erzeugten Indizes nach ihrer Quelle sortiert zählen, und anschließend die Quelle mit der maximalen Aktivität als Klasse wählen. In beiden Varianten kann es von Vorteil sein, nach dem Einschwingen der Dynamik der latenten Zustandsvariablen, Fluktuationen herauszumittlen. Es können auch komplizierte Verfahren (z.B. Klassifikatoren wie Nearest-Neighbor, Support-Vector Maschinen oder Perceptron-Netzwerke) aus dem Maschinenlernen Bereich zum Einsatz kommen.

Die in der vorstehenden Beschreibung, den Zeichnungen und den Ansprüchen offenbarten Merkmale der Erfindung können sowohl einzeln als auch in Kombination oder Unterkombination für die Verwirklichung der Erfindung in ihren verschiedenen Ausgestaltungen von Bedeutung sein. Die Erfindung ist nicht auf die oben beschriebenen bevorzugten Ausführungsformen beschränkt. Vielmehr ist eine Vielzahl von Varianten und Abwandlungen möglich, die ebenfalls von dem Erfindungsgedanken Gebrauch machen und deshalb ebenfalls in den Schutzbereich fallen. Darüber hinaus beansprucht die Erfindung auch Schutz für den Gegenstand und die Merkmale der Unteransprüche unabhängig von den in Bezug genommenen Merkmalen und Ansprüchen.

Ansprüche

1. Datenverarbeitungsvorrichtung (100), die zur Abbildung von Eingabedaten r(s) auf Ausgabedaten konfiguriert ist, umfassend

(a) eine Eingabe-Einrichtung (10), die zur Aufnahme der Eingabedaten r(s), Erzeugung einer Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten und Erzeugung einer stochastischen Repräsentation der Eingabedaten in Gestalt einer Sequenz von Eingabe-Indizes eingerichtet ist, wobei jeder Eingabe-Index eine stochastisch ermittelte Position innerhalb der Eingabe-Wahrscheinlichkeitsverteilung p*(s) repräsentiert,

(b) mindestens ein Netzwerk (20) aus mindestens zwei Merkmalsmodulen (21), die über Datenverbindungen (22) miteinander gekoppelt sind, wobei mindestens eines der Merkmalsmodule (21) über eine Datenverbindung (22) mit der Eingabe-Einrichtung (10) gekoppelt ist, wobei

(b1) jedes Merkmalsmodul (21) eine Vielzahl von Zustandsvariablen-Speicherplätzen (23), denen jeweils eine Vielzahl von Gewichte-Speicherplätzen (24) zugeordnet ist, eine Vielzahl von Index-Speicherplätzen (25) und eine Recheneinheit (26) umfasst,

(b2) die Zustandsvariablen-Speicherplätze (23) zur Speicherung von normierten latenten Zustandsvariablen h_i und die Gewichte-Speicherplätze (24) zur Speicherung von Gewichten p(s|i), die den latenten Zustandsvariablen h_i zugeordnet sind, eingerichtet sind,

(b3) die Eingabe-Einrichtung (10) zum Senden der Sequenz von Eingabe-Indizes an das mindestens eine mit der Eingabe-Einrichtung (10) gekoppelte Merkmalsmodul (21) eingerichtet ist,

(b4) jedes Merkmalsmodul (21) zum Senden einer Sequenz von Modul-Indizes an die mit diesem Merkmalsmodul (21) gekoppelten Merkmalsmodule (21) eingerichtet ist, wobei jeder Modul-Index basierend auf einer Merkmalsmodul-Wahrscheinlichkeitsverteilung h_i des jeweils sendenden Merkmalsmoduls eine stochastisch ermittelte Position eines Zustandsvariablen-Speicherplatzes ist,

(b5) die Index-Speicherplätze (25) jedes Merkmalsmoduls zur Speicherung von empfangenen Eingabe- oder Modul-Indizes eingerichtet sind, und

(b6) die Recheneinheit (26) jedes Merkmalsmoduls dazu eingerichtet ist, die latenten Zustandsvariablen h_i und die zugeordneten Gewichte p(s|i), die in dem Merkmalsmodul (21) gespeichert sind, durch Anwendung einer numerischen Optimierung basierend auf den Werten in den Index-Speicherplätzen (25) schrittweise so zu verändern, dass für alle direkt miteinander gekoppelten Merkmalsmodule (21) wechselweise die Merkmalsmodul-

Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des einen Merkmalsmoduls an die latenten Zustandsvariablen h_s des jeweils anderen Merkmalsmoduls angepasst ist und die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des mindestens einen, mit der Eingabe-Einrichtung (10) gekoppelten Merkmalsmoduls an die Eingabe-Wahrscheinlichkeitsverteilung p*(s) angepasst ist, und

(c) eine Ausgabe-Einrichtung (30), die mit mindestens einem Merkmalsmodul (21) des Netzwerks (20) über eine Datenverbindung (22) gekoppelt und zur Bereitstellung der Ausgabedaten eingerichtet ist.

2. Datenverarbeitungsvorrichtung gemäß Anspruch 1, bei der
- die Eingabe-Einrichtung (10) mindestens ein Eingabe-Merkmalsmodul (21) umfasst, das wie die übrigen Merkmalsmodule (21) mit einer Vielzahl von Zustandsvariablen-Speicherplätzen (23) und einer Recheneinheit (26) aufgebaut ist, wobei die Recheneinheit (26) zur Erzeugung der Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten und die Zustandsvariablen-Speicherplätze (23) zur Speicherung der Eingabe-Wahrscheinlichkeitsverteilung p*(s) eingerichtet sind.

3. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, die für einen Lernmodus und für einen Anwendungsmodus eingerichtet ist, wobei
- im Lernmodus die Eingabe-Einrichtung (10) zur Aufnahme vorbekannter Lern-Eingabedaten, die Ausgabe-Einrichtung (30) zur Aufnahme vorbekannter Lern-Ausgabedaten, die den Lern-Eingabedaten zugeordnet sind, und das Netzwerk (20) aus Merkmalsmodulen (21) für das Aktualisieren der latenten Zustandsvariablen h_i und das Lernen der zugeordneten Gewichte p(s|i), mit denen die Lern-Eingabedaten auf die Lern-Ausgabedaten abgebildet werden, eingerichtet sind, und
- im Anwendungsmodus die Eingabe-Einrichtung (10) zur Aufnahme von Anwendungs-Eingabedaten und die Ausgabe-Einrichtung (30) zur Ausgabe von Anwendungs-Ausgabedaten, die den Anwendungs-Eingabedaten zugeordnet und durch die Merkmalsmodule (21) des Netzwerks (20) mit den aktualisierten latenten Zustandsvariablen h_i und zugeordneten gelernten Gewichten p(s|i) ermittelt sind, eingerichtet sind.

4. Datenverarbeitungsvorrichtung gemäß den Ansprüchen 2 und 3, bei der
- die Eingabe-Einrichtung (10) zusätzlich mindestens ein Lerneingabe-Merkmalsmodul (12) umfasst, das im Lernmodus mit dem mindestens Netzwerk (20) aus mindestens zwei Merkmalsmodulen (21) gekoppelt und im Anwendungsmodus von dem mindestens Netzwerk (20) getrennt ist.

5. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, bei der
- die Merkmale der Eingabedaten r(s) Bildmerkmale, Audiomerkmale, Sprachmerkmale, Sensormerkmale, Signalmerkmale, ökonomische Zeitreihen, Verhaltensdaten, Zustandsmerkmale eines biologischen Organismus, Zustandsmerkmale eines natürlichen Systems, physikalische Größen, chemische Parameter und/oder Zustandsmerkmale einer technischen Einrichtung umfassen.

6. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, umfassend
- eine Konfigurationseinrichtung (41), mit der eine Netzwerkkonfiguration einstellbar ist, welche die Anzahl der Merkmalsmodule, die Anzahl von Zustandsvariablen-Speicherplätzen (23) jedes Merkmalsmoduls, die Anzahl von Gewichte-Speicherplätzen (24), die jeweils einem Zustandsvariablen-Speicherplatz zugeordnet sind, die Anzahl von Index-Speicherplätzen (25) jedes Merkmalsmoduls und/oder die Datenverbindungen (22) zwischen Merkmalsmodulen (21) umfasst.

7. Datenverarbeitungsvorrichtung gemäß Anspruch 6, umfassend
- die Konfigurationseinrichtung (41) für eine iterative Einstellung der Netzwerkkonfiguration eingerichtet ist, bei der die Anzahl der Merkmalsmodule (21) und die Anzahl der Zustandsvariablen-Speicherplätze (23) jedes Merkmalsmoduls schrittweise erhöht werden, bis die Leistung der Abbildung der Eingabedaten auf die Ausgabedaten einem vorbestimmten Betriebskriterium entspricht.

8. Datenverarbeitungsvorrichtung gemäß Anspruch 6 oder 7, bei der
- mindestens ein Reserve-Merkmalsmodul (21) vorgesehen ist, das keine Datenverbindung (22) mit dem Netzwerk (20) aus Merkmalsmodulen (21) aufweist, und
- die Konfigurationseinrichtung (41) zur Herstellung mindestens einer Datenverbindung (22) des mindestens einen Reserve-Merkmalsmoduls mit dem Netzwerk (20) aus Merkmalsmodulen (21) eingerichtet ist.

9. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, bei der
- das Netzwerk (20) aus Merkmalsmodulen (21) ein geschichtetes Netzwerk (20), insbesondere Deep Convolutional Network (DCN), mit mehreren Schichten (50 - 57) aus Merkmalsmodulen (21) umfasst, bei dem Merkmalsmodule (21) innerhalb einer Schicht gemeinsame Gewichte haben, die den latenten Zustandsvariablen h_i zugeordnet sind.

10. Datenverarbeitungsvorrichtung gemäß Anspruch 9, bei der

- das geschichtete Netzwerk (20) für einen schichtbasierten Lernmodus eingerichtet ist, wobei die Schichten von der Eingabe-Einrichtung beginnend für ein Aktualisieren der latenten Zustandsvariablen h_i und ein Lernen der zugeordneten Gewichte $p(s|i)$ derart eingerichtet ist, dass die Lern-Eingabedaten auf die Lern-Ausgabedaten abgebildet werden.

11. Datenverarbeitungsvorrichtung gemäß Anspruch 9, bei der

- das geschichtete Netzwerk (20) für einen unüberwachten, schichtbasierten Lernmodus eingerichtet ist, wobei die Schichten von der Eingabe-Einrichtung beginnend bis zu einer vorletzten Schicht vor der Ausgabe-Einrichtung aufeinanderfolgend für eine Ermittlung der latenten Zustandsvariablen h_i und der zugeordneten Gewichte $p(s|i)$ ohne Berücksichtigung der Lern-Ausgabedaten eingerichtet sind und eine letzte Schicht vor der Ausgabe-Einrichtung für eine Ermittlung der latenten Zustandsvariablen h_i und der zugeordneten Gewichte $p(s|i)$ mit einer Berücksichtigung der Lern-Ausgabedaten eingerichtet ist.

12. Datenverarbeitungsvorrichtung gemäß Anspruch 9, bei der

- das geschichtete Netzwerk (20) für einen überwachten, schichtbasierten Lernmodus eingerichtet ist, bei dem basierend auf dem aktuellen Lernfehler die Gewichte $p(s|i)$ ausgewählter Module reinitialisiert werden.

13. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, bei der

- die Recheneinheit (26) jedes Merkmalsmodul (21) dazu eingerichtet ist, die numerische Optimierung basierend auf einem probabilistischen Ziel-Maß, insbesondere der Log-Likelihood, der Kullbach-Leibler-Divergenz oder Kombinationen von Vektornormen, auszuführen.

14. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, bei der

- die Ausgabe-Einrichtung (30) mindestens ein Ausgabe-Merkmalsmodul (21) umfasst, das wie die übrigen Merkmalsmodule (21) mit einer Vielzahl von Zustandsvariablen-Speicherplätzen (23), einer Vielzahl von Index-Speicherplätzen (25) und einer Recheneinheit (26) aufgebaut ist, wobei die latenten Zustandsvariablen des mindestens einen Merkmalsmoduls oder davon abgeleitete Größen als Ausgabedaten ausgegeben werden.

15. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, bei der
- die Ausgabe-Einrichtung (30) einen Klassifikator, insbesondere basierend auf Perzeptrons oder einer Support-Vector-Maschine, umfasst, der auf der Basis der latenten Zustandsvariablen von Merkmalsmodulen (21) operiert, mit denen die Ausgabe-Einrichtung (30) gekoppelt ist.

16. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, die umfasst
- ein modulares System mit mindestens zwei integrierten Schaltungen, die jeweils ein Netzwerk (20) aus Merkmalsmodulen (21) umfassen.

17. Datenverarbeitungsvorrichtung gemäß einem der vorherigen Ansprüche, bei der
- die Eingabedaten als Ergebnis von Messungen, Beobachtungen, Informationssammlungen oder numerischen Verfahren vorliegen, und
- die Datenverarbeitungsvorrichtung konfiguriert ist, die Ausgabedaten aus den Eingabedaten zu ermitteln.

18. Datenverarbeitungsverfahren, das zur Abbildung von Eingabedaten r(s) auf Ausgabedaten konfiguriert ist, umfassend die Schritte
(S1) Aufnahme der Eingabedaten r(s), Erzeugung einer Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten und Erzeugung einer stochastischen Repräsentation der Eingabedaten in Gestalt einer Sequenz von Eingabe-Indizes eingerichtet ist, mit einer Eingabe-Einrichtung (10), wobei jeder Eingabe-Index eine stochastisch ermittelte Position innerhalb der Eingabe-Wahrscheinlichkeitsverteilung p*(s) repräsentiert,
(S2) Betreiben mindestens eines Netzwerks (20) aus mindestens zwei Merkmalsmodulen (21), die über Datenverbindungen (22) miteinander gekoppelt sind, wobei mindestens eines der Merkmalsmodule (21) über eine Datenverbindung (22) mit der Eingabe-Einrichtung (10) gekoppelt ist, wobei

jedes Merkmalsmodul (21) eine Vielzahl von Zustandsvariablen-Speicherplätzen (23), denen jeweils eine Vielzahl von Gewichte-Speicherplätzen (24) zugeordnet ist, eine Vielzahl von Index-Speicherplätzen (25) und eine Recheneinheit (26) umfasst,

die Zustandsvariablen-Speicherplätze (23) normierte latente Zustandsvariablen h_i und die Gewichte-Speicherplätze (24) Gewichte $p(s|i)$ speichern, die den latenten Zustandsvariablen h_i zugeordnet sind,

(S2.1) die Eingabe-Einrichtung (10) die Sequenz von Eingabe-Indizes an das mindestens eine mit der Eingabe-Einrichtung (10) gekoppelte Merkmalsmodul (21) sendet,

(S2.2) jedes Merkmalsmodul (21) eine Sequenz von Modul-Indizes an die mit diesem Merkmalsmodul (21) gekoppelten Merkmalsmodule (21) sendet, wobei jeder Modul-Index basierend auf einer Merkmalsmodul-Wahrscheinlichkeitsverteilung h_i des jeweils sendenden Merkmalsmoduls eine stochastisch ermittelte Position eines Zustandsvariablen-Speicherplatzes repräsentiert,

(S2.3) die Index-Speicherplätze (25) jedes Merkmalsmoduls empfangene Eingabe- oder Modul-Indizes speichern, und

(S2.4) die Recheneinheit (26) jedes Merkmalsmoduls die latenten Zustandsvariablen h_i und die zugeordneten Gewichte p(s|i), die in dem Merkmalsmodul (21) gespeichert sind, durch Anwendung einer numerischen Optimierung basierend auf den Werten in den Index-Speicherplätzen (25) schrittweise verändert, dass für alle direkt miteinander gekoppelten Merkmalsmodule (21) wechselweise die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des einen Merkmalsmoduls an die latenten Zustandsvariablen h_s des jeweils anderen Merkmalsmoduls angepasst ist und die Merkmalsmodul-Wahrscheinlichkeitsverteilung $\sum_i p(s|i)h_i$ des mindestens einen, mit der Eingabe-Einrichtung (10) gekoppelten Merkmalsmoduls an die Eingabe-Wahrscheinlichkeitsverteilung p*(s) angepasst ist, und

- (S3) Bereitstellung der Ausgabedaten mit einer Ausgabe-Einrichtung (30), die mit mindestens einem Merkmalsmodul (21) des Netzwerks (20) über eine Datenverbindung (22) gekoppelt ist.

19. Datenverarbeitungsverfahren gemäß Anspruch 18, bei dem

- die Eingabe-Einrichtung (10) mindestens ein Eingabe-Merkmalsmodul (21) umfasst, das wie die übrigen Merkmalsmodule (21) mit einer Vielzahl von Zustandsvariablen-Speicherplätzen (23) und einer Recheneinheit (26) aufgebaut ist, und
- die Recheneinheit (26) die Eingabe-Wahrscheinlichkeitsverteilung p*(s) der Eingabedaten erzeugt und die Zustandsvariablen-Speicherplätze (23) die Eingabe-Wahrscheinlichkeitsverteilung p*(s) speichern.

20. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 19, bei dem ein Lernmodus oder ein Anwendungsmodus ausgeführt wird, wobei

- im Lernmodus die Eingabe-Einrichtung (10) vorbekannte Lern-Eingabedaten aufnimmt, die Ausgabe-Einrichtung (30) vorbekannter Lern-Ausgabedaten aufnimmt, die den Lern-Eingabedaten zugeordnet sind, und das Netzwerk (20) aus Merkmalsmodulen (21) die latenten Zustandsvariablen h_i aktualisiert und die zugeordneten Gewichte p(s|i) lernt, mit denen die Lern-Eingabedaten auf die Lern-Ausgabedaten abgebildet werden, und

- im Anwendungsmodus die Eingabe-Einrichtung (10) Anwendungs-Eingabedaten aufnimmt und die Ausgabe-Einrichtung (30) Anwendungs-Ausgabedaten ausgibt, die den Anwendungs-Eingabedaten zugeordnet und durch die Merkmalsmodule (21) des Netzwerks (20) mit den aktualisierten latenten Zustandsvariablen h_i und zugeordneten gelernten Gewichten $p(s|i)$ ermittelt sind.

21. Datenverarbeitungsverfahren gemäß Anspruch 20, bei dem
- die Eingabe-Einrichtung (10) zusätzlich mindestens ein Lerneingabe-Merkmalsmodul (21) umfasst, das im Lernmodus mit dem mindestens Netzwerk (20) aus mindestens zwei Merkmalsmodulen (21) gekoppelt und im Anwendungsmodus von dem mindestens Netzwerk (20) getrennt ist.

22. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 21, bei dem
- die Merkmale der Eingabedaten r(s) Bildmerkmale, Audiomerkmale, Sprachmerkmale, Sensormerkmale, Signalmerkmale, ökonomische Zeitreihen, Verhaltensdaten, Zustandsmerkmale eines biologischen Organismus, physikalische Größen, chemische Parameter und/oder Zustandsmerkmale einer technischen Einrichtung umfassen.

23. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 22, mit dem Konfigurationsschritt
- Einstellung einer Netzwerkkonfiguration mit einer Konfigurationseinrichtung (41), welche die Anzahl der Merkmalsmodule, die Anzahl von Zustandsvariablen-Speicherplätzen (23) jedes Merkmalsmoduls, die Anzahl von Gewichte-Speicherplätzen (24), die jeweils einem Zustandsvariablen-Speicherplatz zugeordnet sind, die Anzahl von Index-Speicherplätzen (25) jedes Merkmalsmoduls und/oder die Datenverbindungen (22) zwischen Merkmalsmodulen (21) umfasst.

24. Datenverarbeitungsverfahren gemäß Anspruch 23, bei dem
- die Einstellung der Netzwerkkonfiguration iterativ erfolgt, wobei bei die Anzahl der Merkmalsmodule (21) und die Anzahl der Zustandsvariablen-Speicherplätze (23) jedes Merkmalsmoduls schrittweise erhöht werden, bis die Leistung der Abbildung der Eingabedaten auf die Ausgabedaten einem vorbestimmten Betriebskriterium entspricht.

25. Datenverarbeitungsverfahren gemäß Anspruch 23 oder 24, bei dem
- der Konfigurationsschritt während der Abbildung der Eingabedaten r(s) auf die Ausgabedaten erfolgt und die Herstellung mindestens einer Datenverbindung (22) von mindestens einem Reserve-Merkmalsmodul (21) mit dem Netzwerk (20) aus Merkmalsmodulen (21) umfasst.

26. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 25, bei dem
- das Netzwerk (20) aus Merkmalsmodulen (21) ein geschichtetes Netzwerk (20), insbesondere Deep Convolutional Network (DCN), mit mehreren Schichten aus Merkmalsmodulen (21) umfasst, bei dem Merkmalsmodule (21) innerhalb einer Schicht gemeinsame Gewichte haben, die den latenten Zustandsvariablen h_i zugeordnet sind.

27. Datenverarbeitungsverfahren gemäß Anspruch 26, bei dem
- das geschichtete Netzwerk (20) einen schichtbasierten Lernmodus ausführt, wobei die Schichten von der Eingabe-Einrichtung beginnend die latenten Zustandsvariablen h_i aktualisieren und die zugeordneten Gewichte $p(s|i)$ lernen, so dass die Lern-Eingabedaten auf die Lern-Ausgabedaten abgebildet werden.

28. Datenverarbeitungsverfahren gemäß Anspruch 26, bei dem
- das geschichtete Netzwerk (20) für einen unüberwachten, schichtbasierten Lernmodus ausführt, wobei die Schichten von der Eingabe-Einrichtung beginnend bis zu einer vorletzten Schicht vor der Ausgabe-Einrichtung aufeinanderfolgend unabhängig von den Lern-Ausgabedaten die latenten Zustandsvariablen h_i aktualisieren und die zugeordneten Gewichte $p(s|i)$ lernen und eine letzte Schicht vor der Ausgabe-Einrichtung unter Berücksichtigung der Lern-Ausgabedaten die latenten Zustandsvariablen h_i aktualisiert und die zugeordneten Gewichte $p(s|i)$ lernt.

29. Datenverarbeitungsverfahren gemäß Anspruch 26, bei dem
- das geschichtete Netzwerk (20) einen überwachten, schichtbasierten Lernmodus ausführt, bei dem basierend auf dem aktuellen Lernfehler die Gewichte $p(s|i)$ ausgewählter Module reinitialisiert werden.

30. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 29, bei dem
- die Recheneinheit (26) jedes Merkmalsmodul (21) die numerische Optimierung basierend auf einem probabilistischen Ziel-Maß, insbesondere der Log-Likelihood, der Kullbach-Leibler-Divergenz oder Kombinationen von Vektornormen, ausführt.

31. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 30, bei dem
- die Ausgabe-Einrichtung (30) mindestens ein Merkmalsmodul (21) umfasst, das wie die übrigen Merkmalsmodule (21) mit einer Vielzahl von Zustandsvariablen-Speicherplätzen (23) und einer Recheneinheit (26) aufgebaut ist, wobei die latenten Zustandsvariablen des mindestens einen Merkmalsmoduls oder davon abgeleitete Größen als Ausgabedaten ausgegeben werden.

32. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 31, bei dem
- die Ausgabe-Einrichtung (30) einen Klassifikator, insbesondere basierend auf Perzeptrons oder einer Support-Vector-Maschine, umfasst, der auf der Basis der latenten Zustandsvariablen von Merkmalsmodulen (21) operiert, mit denen die Ausgabe-Einrichtung (30) gekoppelt ist.

33. Datenverarbeitungsverfahren gemäß einem der Ansprüche 18 bis 32, bei dem
- die Eingabedaten als Ergebnis von Messungen, Beobachtungen, Informationssammlungen oder numerischen Verfahren vorliegen und die Ausgabedaten aus den Eingabedaten ermittelt werden.

34. Computerprogrammprodukt, das auf einem Computer-lesbaren Speichermedium gespeichert und zur Ausführung eines Verfahrens gemäß einem der Ansprüche 18 bis 33 eingerichtet ist.

35. Computer-lesbares Speichermedium, auf dem ein Computerprogrammprodukt gespeichert ist, das zur Ausführung des Verfahrens gemäß einem der Ansprüche 18 bis 33 eingerichtet ist.

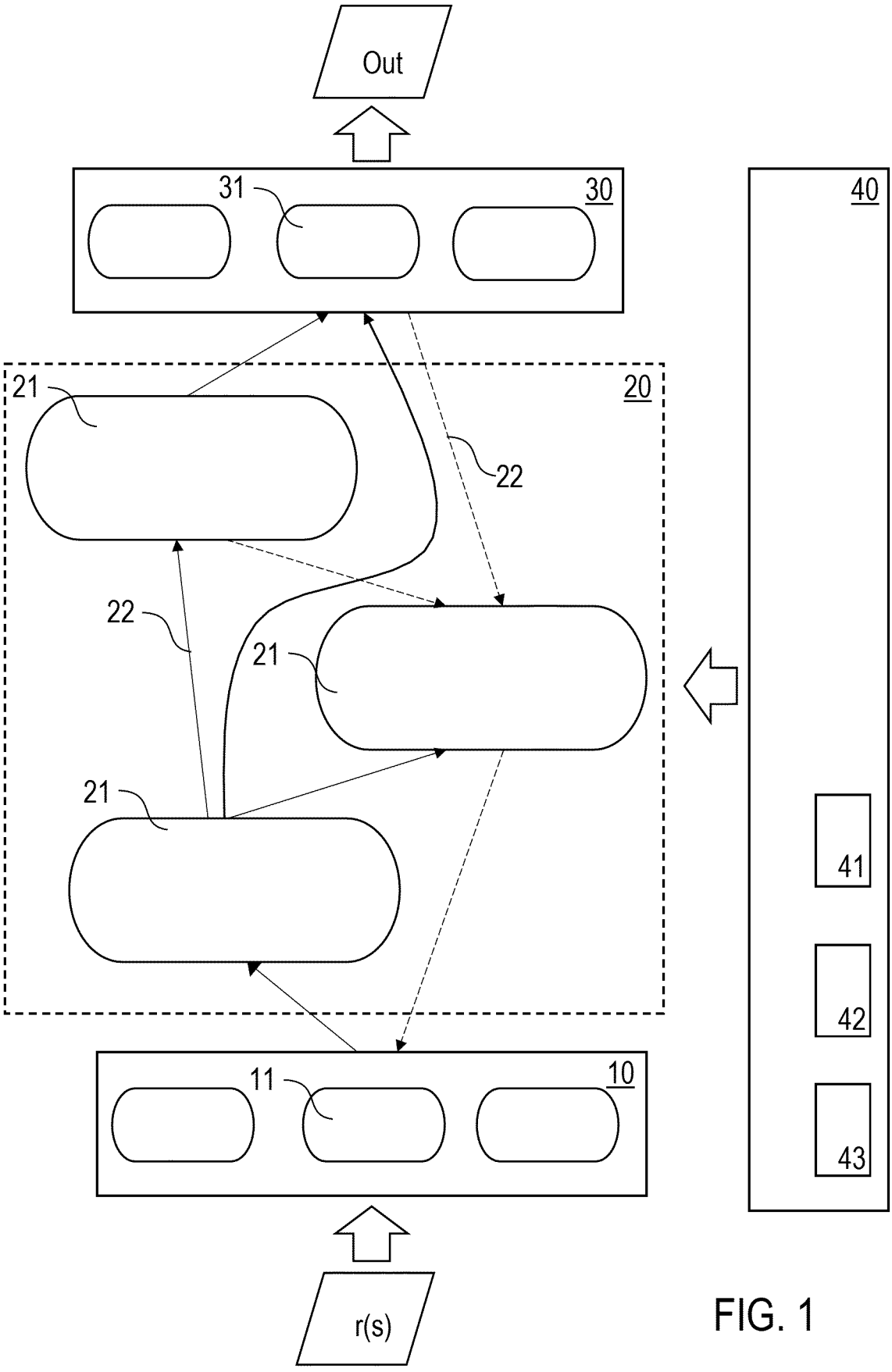


FIG. 1

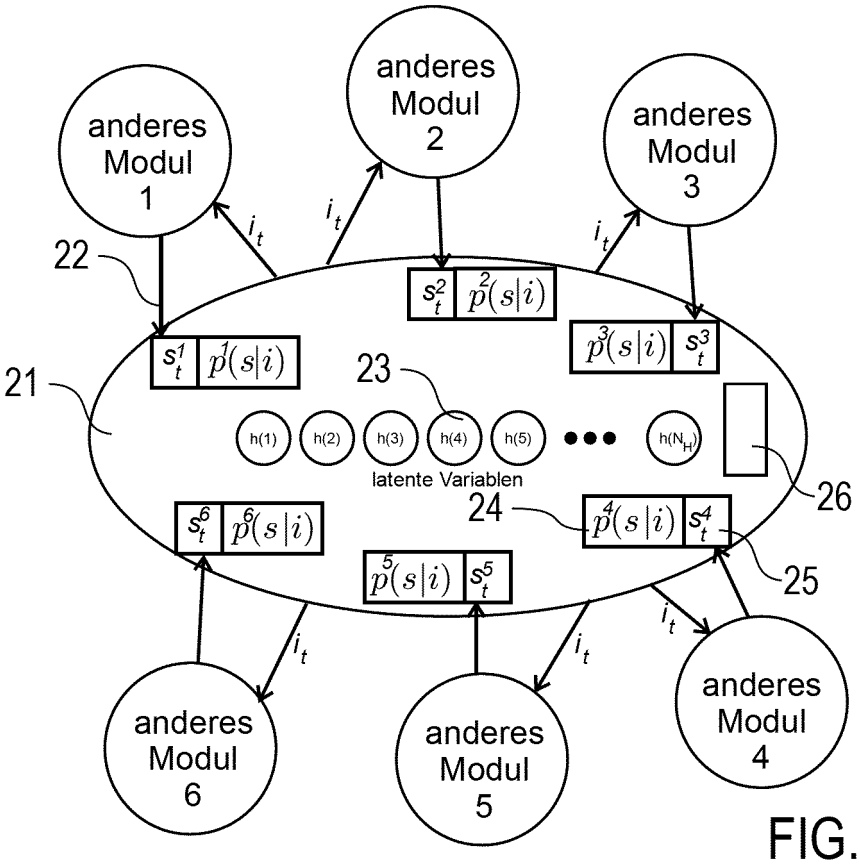

anderes Modul 1
anderes Modul 2
anderes Modul 3
anderes Modul 4
anderes Modul 5
anderes Modul 6
i_t
22
21
23
24
25
26
s_t^1 $p^1(s|i)$
s_t^2 $p^2(s|i)$
$p^3(s|i)$ s_t^3
$p^4(s|i)$ s_t^4
$p^5(s|i)$ s_t^5
s_t^6 $p^6(s|i)$
h(1) h(2) h(3) h(4) h(5) ••• h(N_H)
latente Variablen

FIG. 2

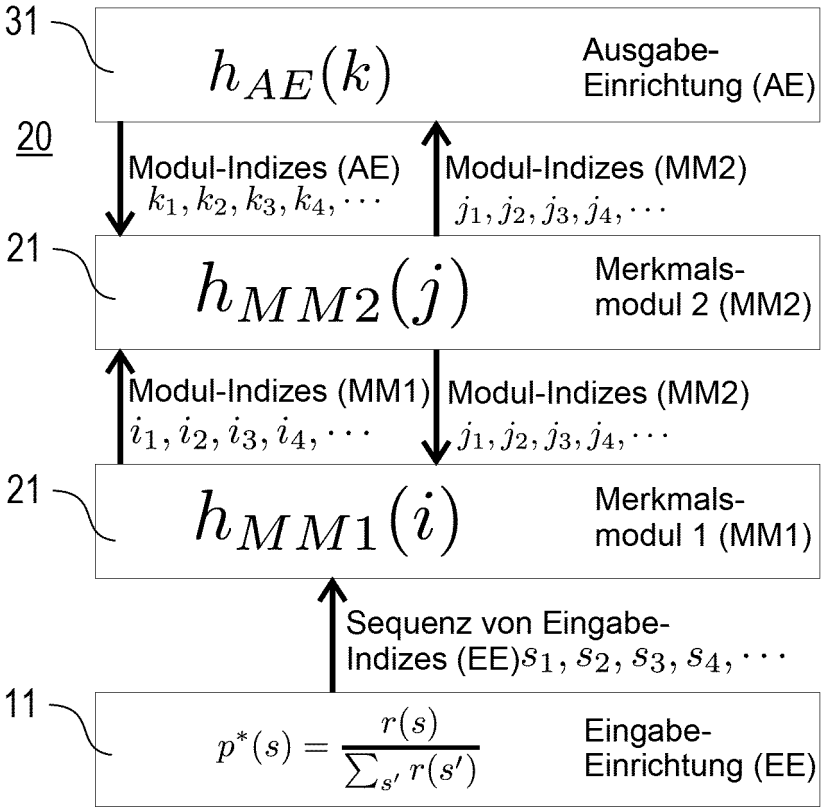

31
$h_{AE}(k)$
Ausgabe-Einrichtung (AE)
20
Modul-Indizes (AE)
$k_1, k_2, k_3, k_4, \cdots$
Modul-Indizes (MM2)
$j_1, j_2, j_3, j_4, \cdots$
21
$h_{MM2}(j)$
Merkmals-modul 2 (MM2)
Modul-Indizes (MM1)
$i_1, i_2, i_3, i_4, \cdots$
Modul-Indizes (MM2)
$j_1, j_2, j_3, j_4, \cdots$
21
$h_{MM1}(i)$
Merkmals-modul 1 (MM1)
Sequenz von Eingabe-Indizes (EE)$s_1, s_2, s_3, s_4, \cdots$
11
$p^*(s) = \frac{r(s)}{\sum_{s'} r(s')}$
Eingabe-Einrichtung (EE)

FIG. 3

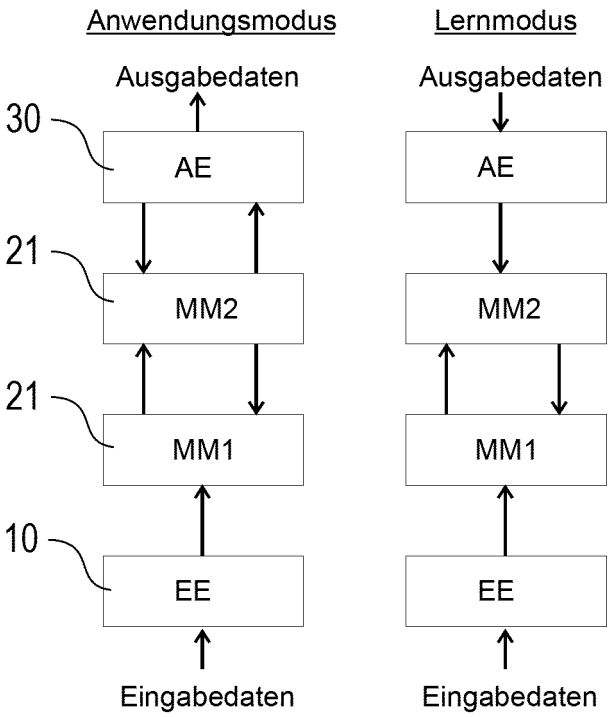

Anwendungsmodus
Lernmodus
Ausgabedaten
Ausgabedaten
30
AE
AE
21
MM2
MM2
21
MM1
MM1
10
EE
EE
Eingabedaten
Eingabedaten

FIG. 4

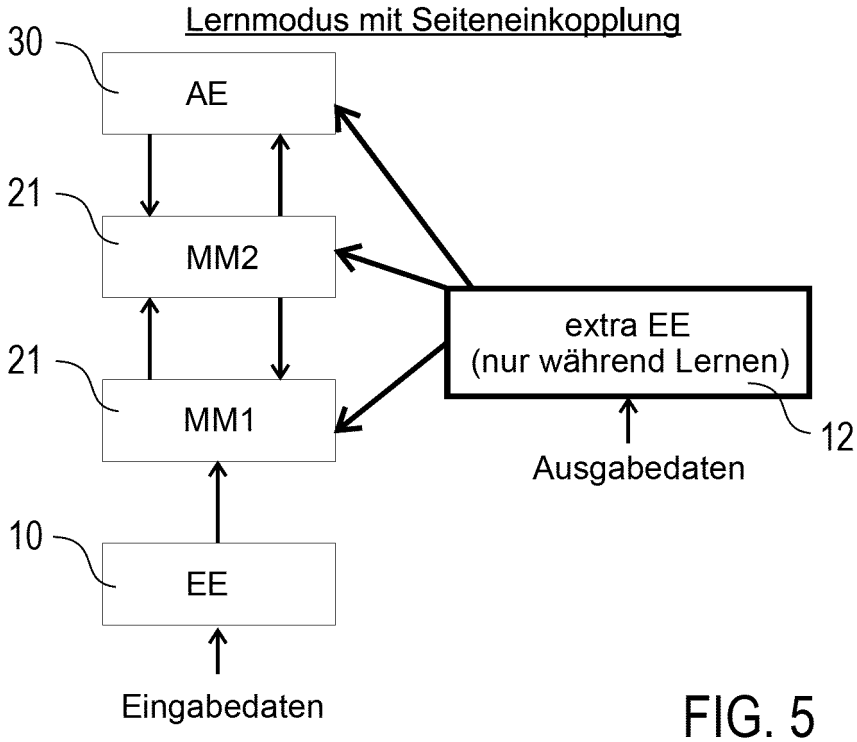

Lernmodus mit Seiteneinkopplung
30
AE
21
MM2
21
MM1
10
EE
Eingabedaten
extra EE
(nur während Lernen)
12
Ausgabedaten

FIG. 5

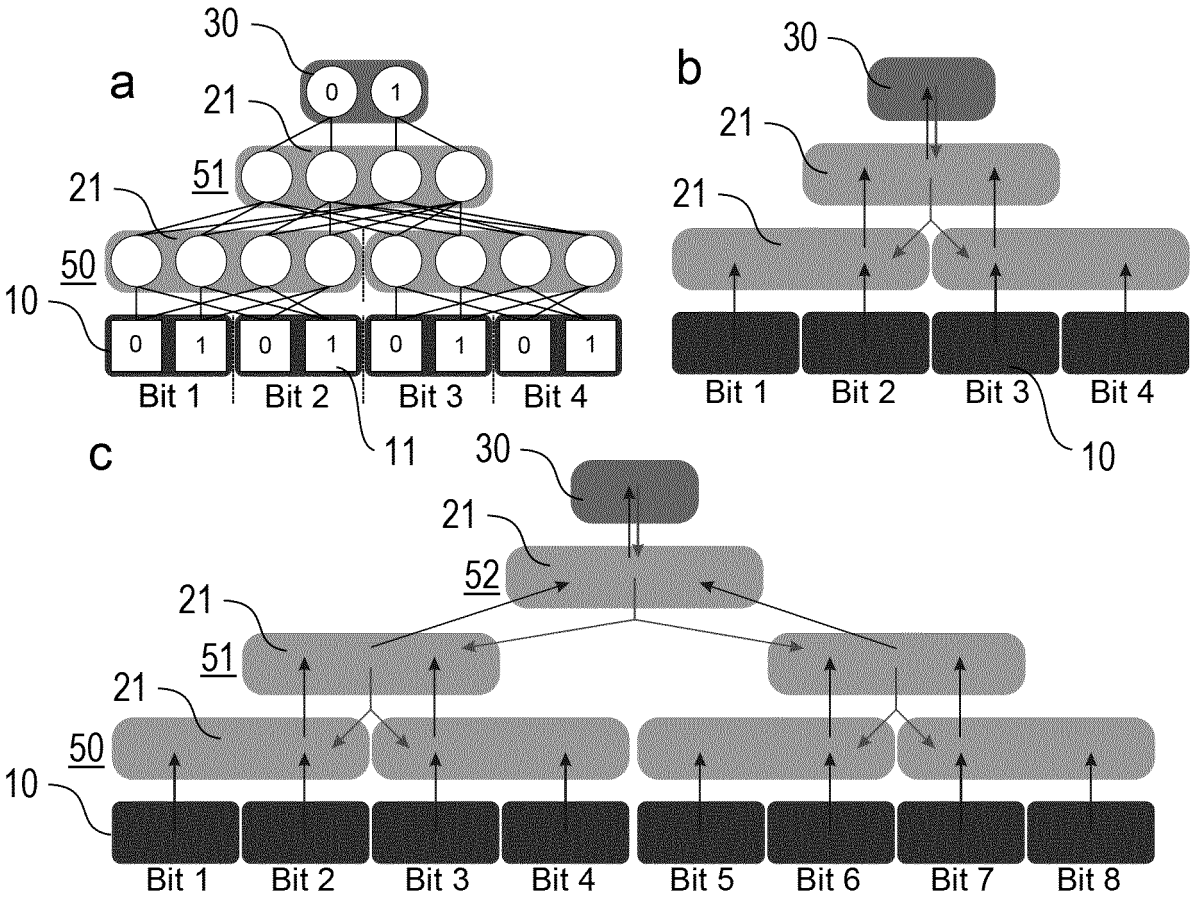

a
b
c
30
21
51
50
10
11
52
0
1
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Bit 7
Bit 8

FIG. 6

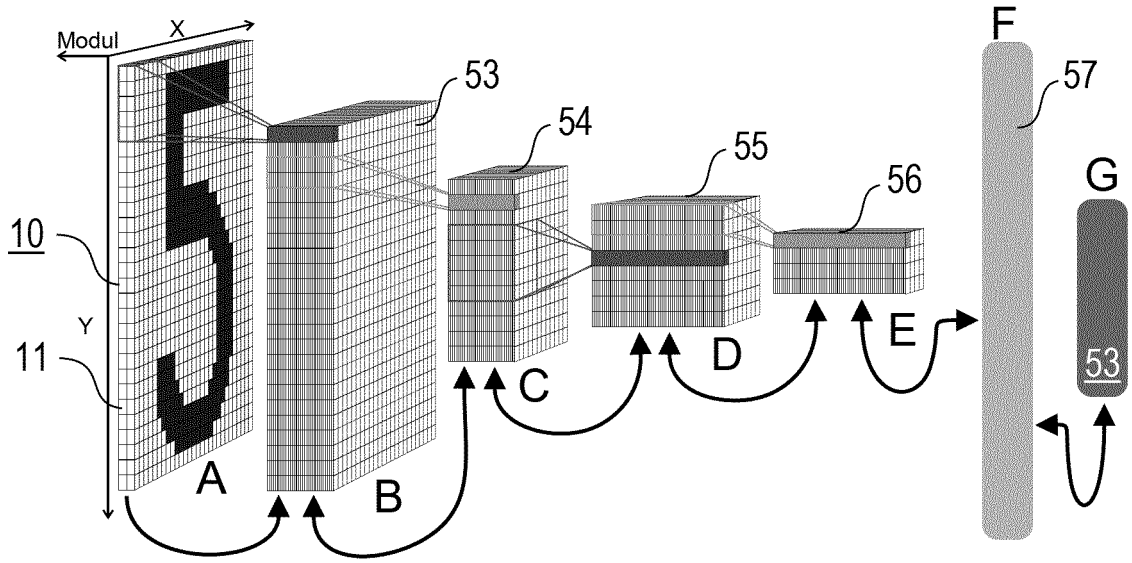

Modul
X
Y
10
11
53
54
55
56
57
A
B
C
D
E
F
G

FIG. 7

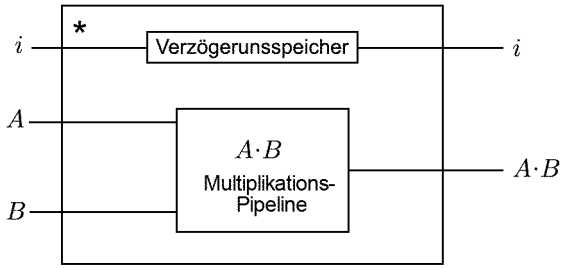


FIG. 8

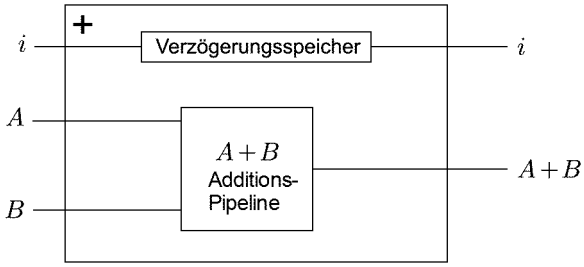


FIG. 9

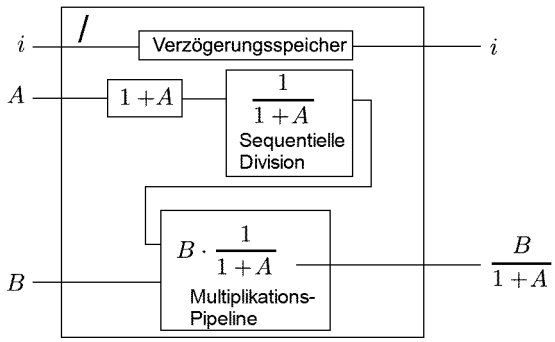


FIG. 10

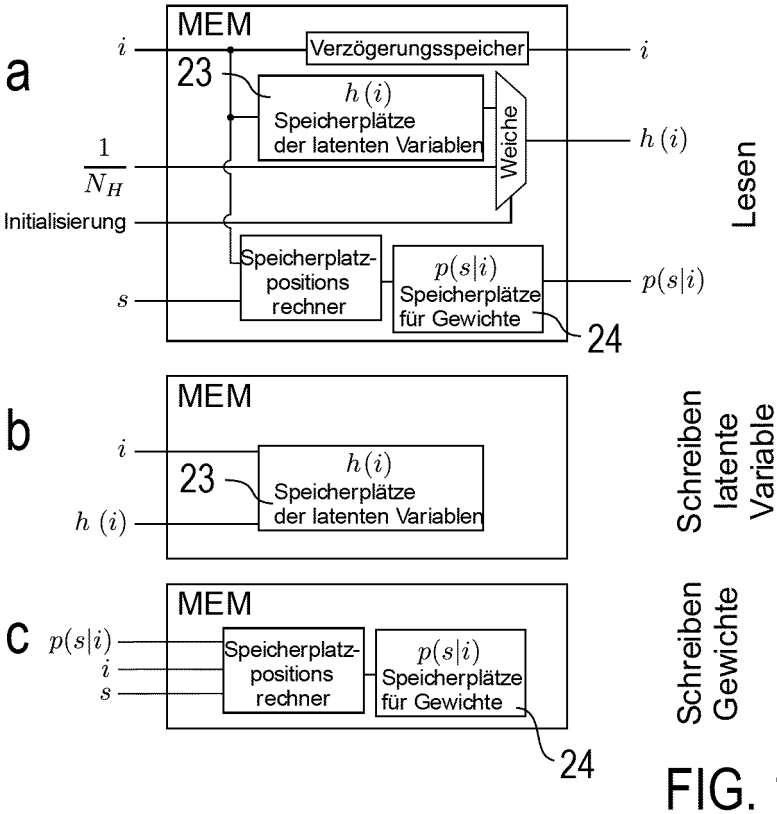


FIG. 11

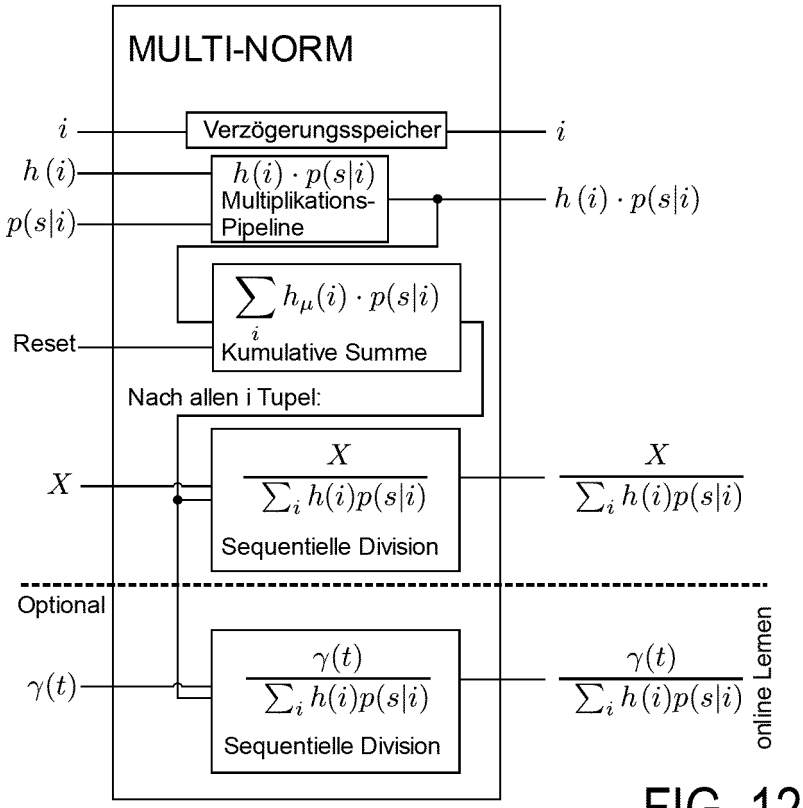

MULTI-NORM
i
Verzögerungsspeicher
i
$h(i)$
$h(i) \cdot p(s|i)$
Multiplikations-Pipeline
$h(i) \cdot p(s|i)$
$p(s|i)$
$\sum_i h_\mu(i) \cdot p(s|i)$
Reset
Kumulative Summe
Nach allen i Tupel:
X
$\frac{X}{\sum_i h(i)p(s|i)}$
Sequentielle Division
$\frac{X}{\sum_i h(i)p(s|i)}$
Optional
$\gamma(t)$
$\frac{\gamma(t)}{\sum_i h(i)p(s|i)}$
Sequentielle Division
$\frac{\gamma(t)}{\sum_i h(i)p(s|i)}$
online Lernen

FIG. 12

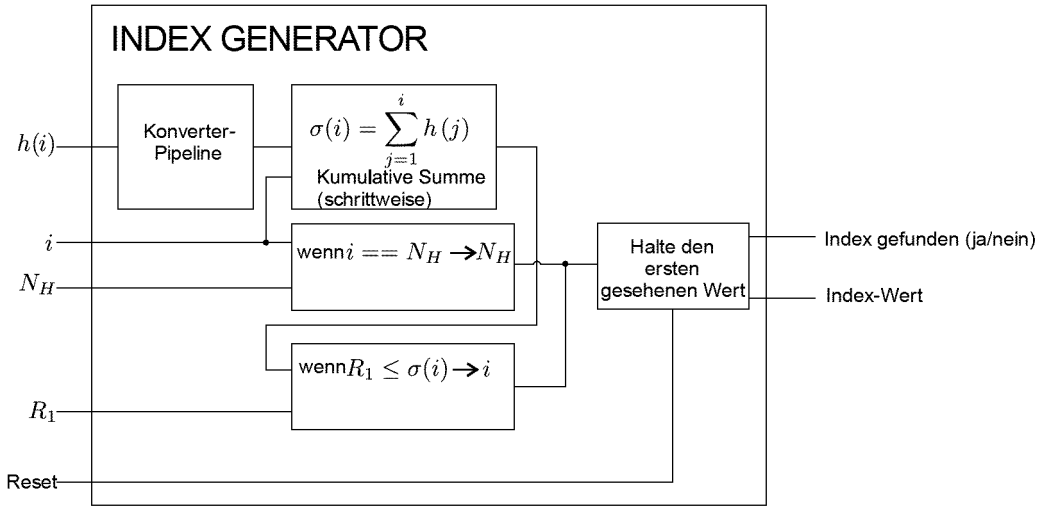

INDEX GENERATOR
$h(i)$
Konverter-Pipeline
$\sigma(i) = \sum_{j=1}^{i} h(j)$
Kumulative Summe (schrittweise)
i
wenn $i == N_H \rightarrow N_H$
N_H
Halte den ersten gesehenen Wert
Index gefunden (ja/nein)
Index-Wert
wenn $R_1 \leq \sigma(i) \rightarrow i$
R_1
Reset

FIG. 13

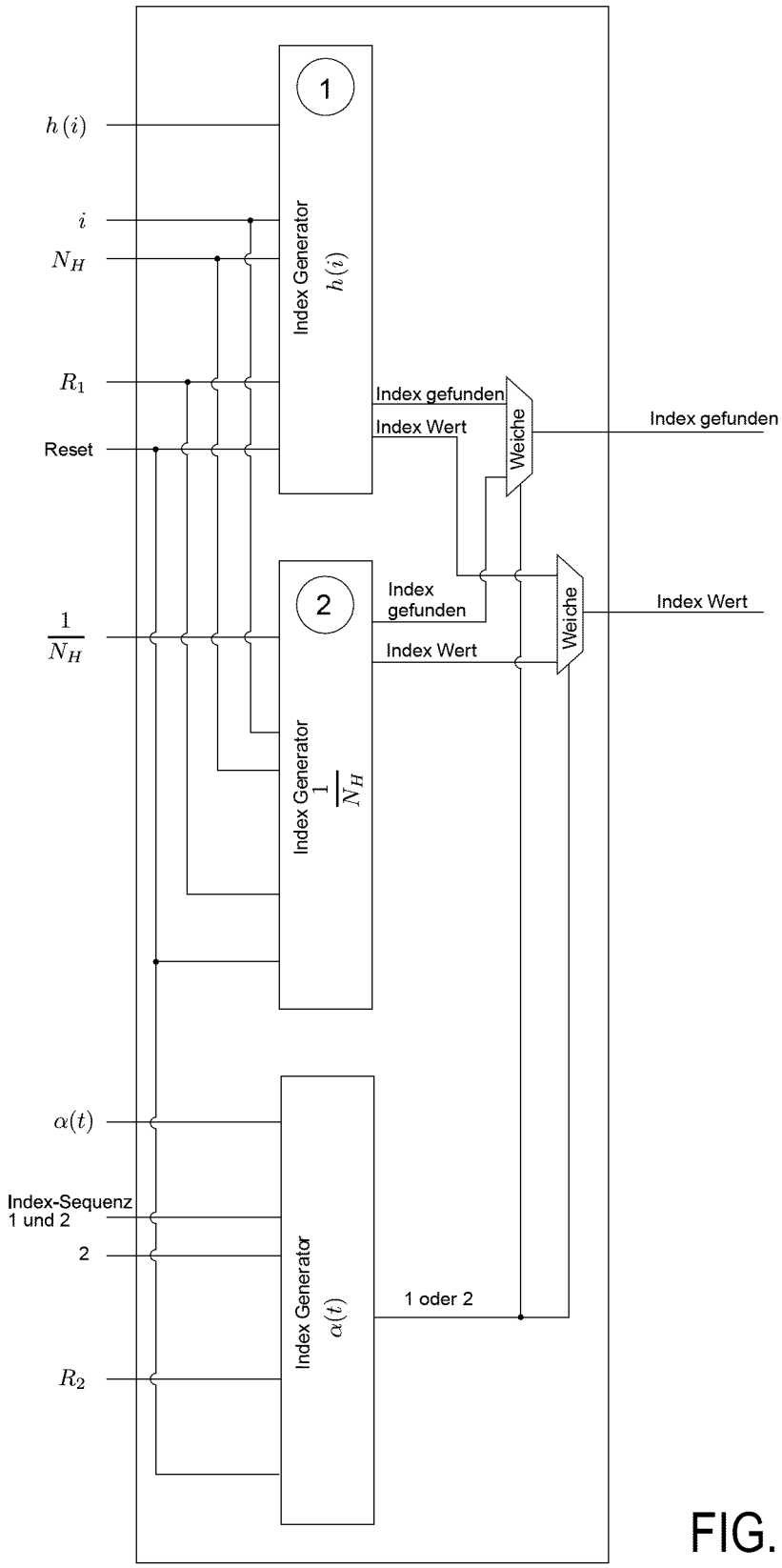
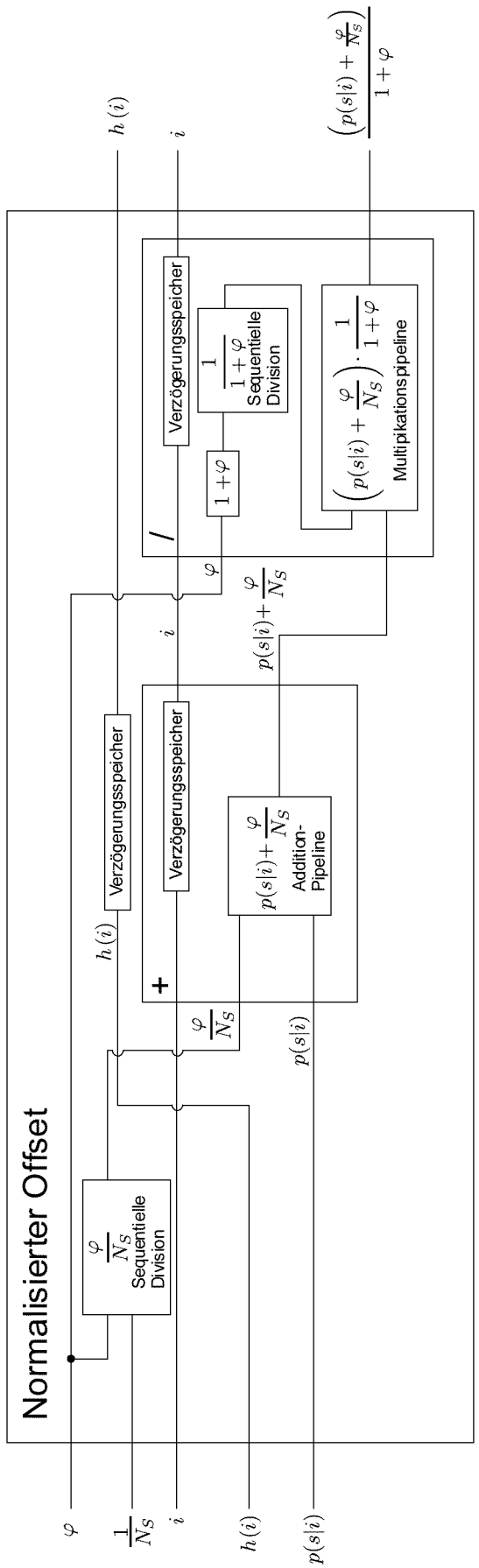

1
$h(i)$
i
N_H
Index Generator $h(i)$
R_1
Reset
Index gefunden
Index Wert
Weiche
Index gefunden
2
$\frac{1}{N_H}$
Index gefunden
Index Wert
Weiche
Index Wert
Index Generator $\frac{1}{N_H}$
$\alpha(t)$
Index-Sequenz 1 und 2
2
Index Generator $\alpha(t)$
1 oder 2
R_2

FIG. 14

FIG. 15


Normalisierter Offset
φ
$\frac{1}{N_S}$
i
$h(i)$
$p(s|i)$
$\frac{\varphi}{N_S}$ Sequentielle Division
Verzögerungsspeicher
$h(i)$
+
Verzögerungsspeicher
i
$\frac{\varphi}{N_S}$
$p(s|i)+\frac{\varphi}{N_S}$ Addition-Pipeline
$p(s|i)$
$p(s|i)+\frac{\varphi}{N_S}$
/
Verzögerungsspeicher
$1+\varphi$
$\frac{1}{1+\varphi}$ Sequentielle Division
φ
$\left(p(s|i)+\frac{\varphi}{N_S}\right)\cdot\frac{1}{1+\varphi}$ Multiplikationspipeline
$h(i)$
i
$\frac{\left(p(s|i)+\frac{\varphi}{N_S}\right)}{1+\varphi}$

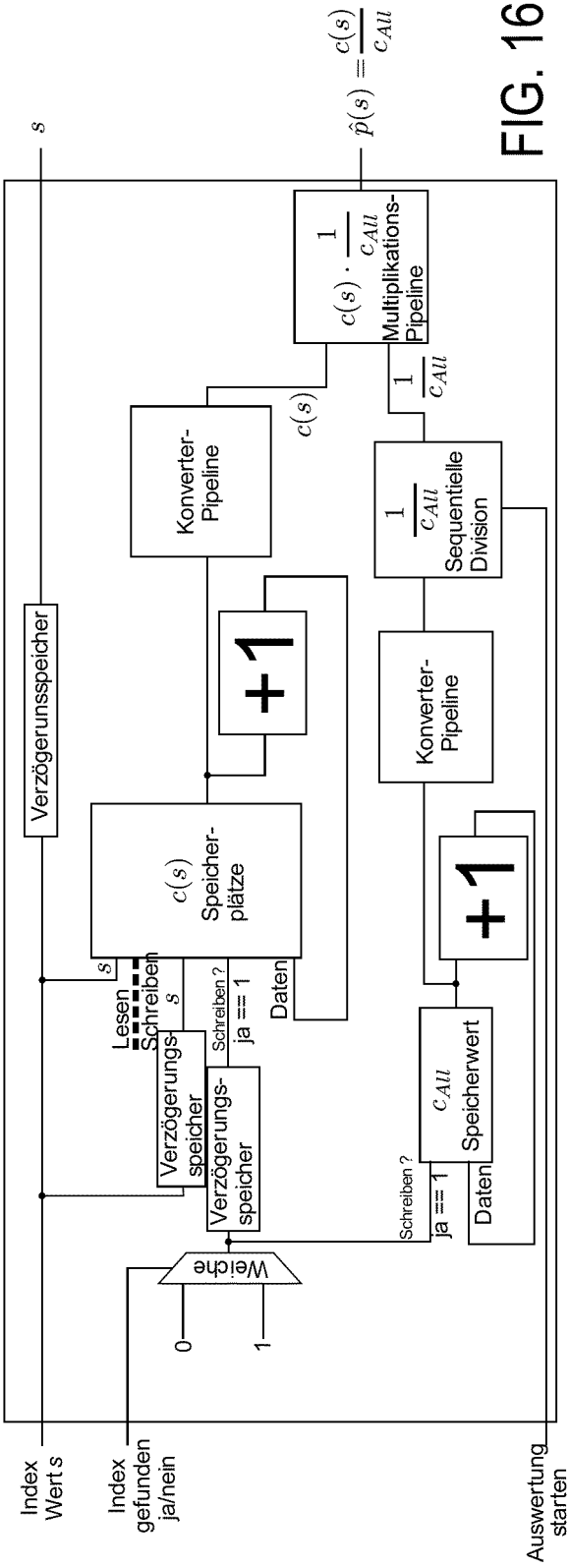

Index
Wert s
Index
gefunden
ja/nein
Auswertung
starten
Verzögerunsspeicher
Konverter-
Pipeline
Konverter-
Pipeline
Multiplikations-
Pipeline
Sequentielle
Division
Speicher-
plätze
Verzögerungs-
speicher
Verzögerungs-
speicher
Lesen
Schreiben
Schreiben ?
ja == 1
Daten
Schreiben ?
ja == 1
Speicherwert
Daten
Weiche
0
1
+1
+1

FIG. 16

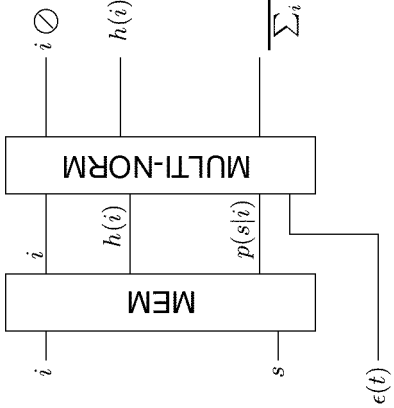

MEM
MULTI-NORM

FIG. 17

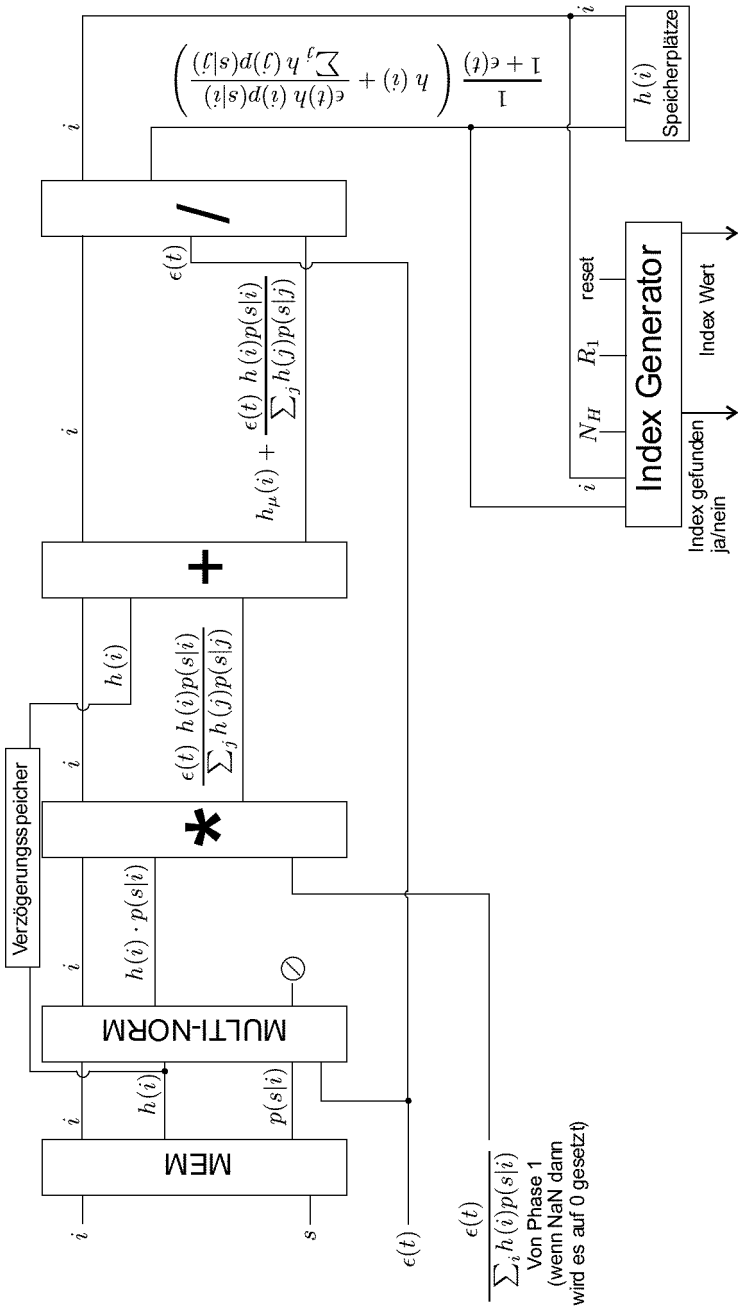

MEM
MULTI-NORM
Verzögerungsspeicher
Index Generator
reset
Index Wert
Index gefunden
ja/nein
Speicherplätze
Von Phase 1
(wenn NaN dann
wird es auf 0 gesetzt)

FIG. 18

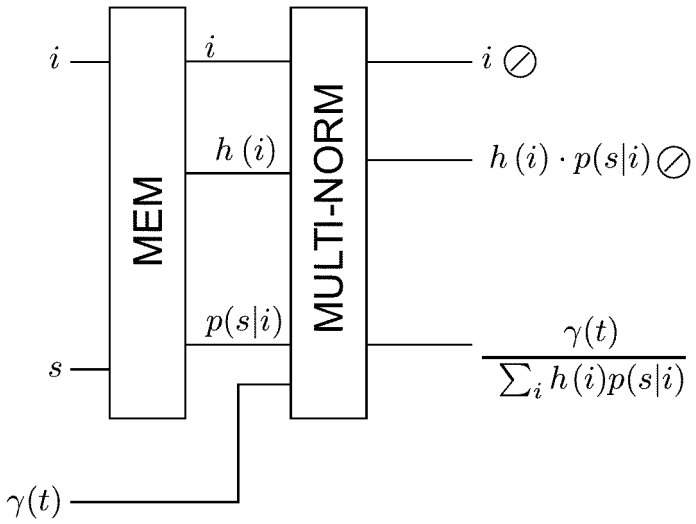

i
s
γ(t)
MEM
i
h (i)
p(s|i)
MULTI-NORM
i ⊘
h (i) · p(s|i) ⊘
γ(t) / ∑i h (i)p(s|i)

FIG. 19

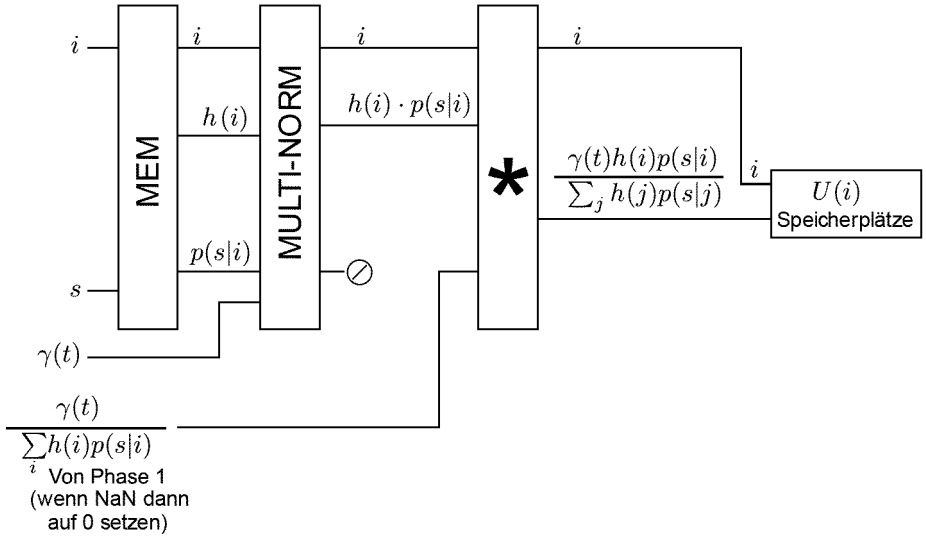

i
s
γ(t)
γ(t) / ∑i h(i)p(s|i)
Von Phase 1
(wenn NaN dann
auf 0 setzen)
MEM
h(i)
p(s|i)
MULTI-NORM
h(i) · p(s|i)
⊘
*
γ(t)h(i)p(s|i) / ∑j h(j)p(s|j)
U(i)
Speicherplätze

FIG. 20

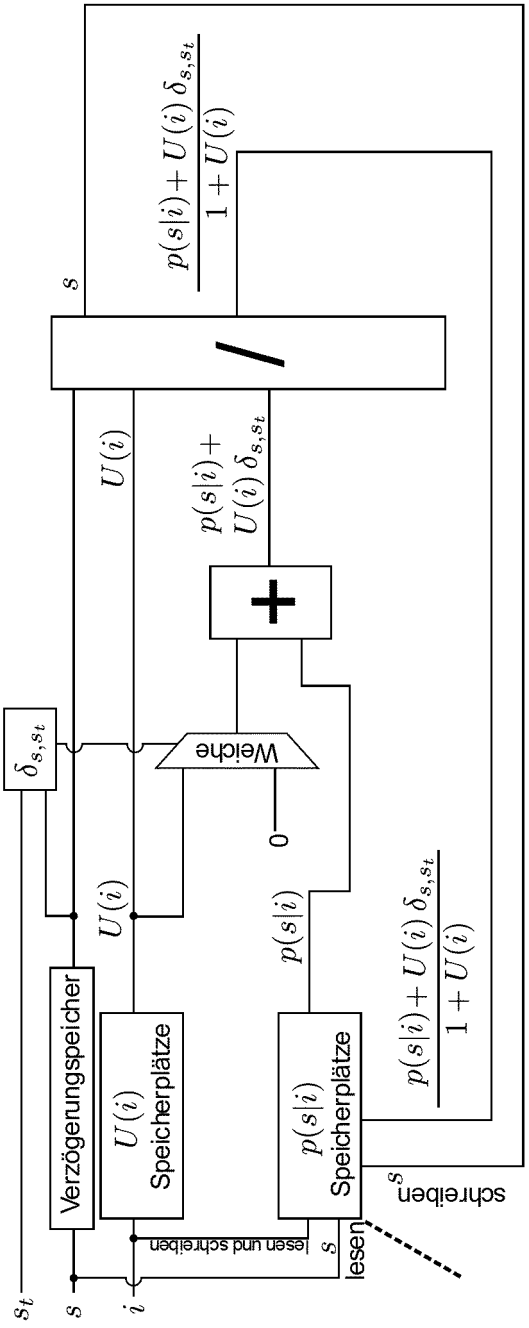

$\frac{p(s|i)+U(i)\,\delta_{s,s_t}}{1+U(i)}$
s
/
$U(i)$
$p(s|i)+$
$U(i)\,\delta_{s,s_t}$
+
δ_{s,s_t}
Weiche
0
$U(i)$
$p(s|i)$
Verzögerungspeicher
$U(i)$
Speicherplätze
$p(s|i)$
Speicherplätze
$\frac{p(s|i)+U(i)\,\delta_{s,s_t}}{1+U(i)}$
schreiben
s
lesen und schreiben
s
lesen
s_t
s
i

FIG. 21

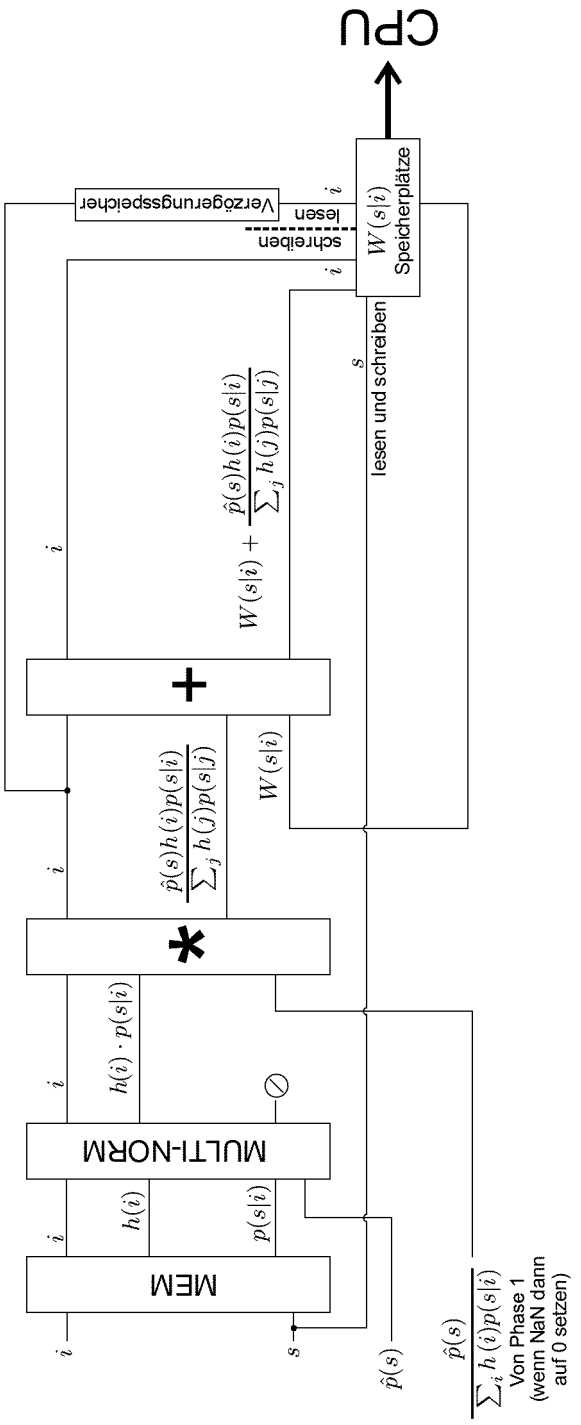

CPU
Speicherplätze
$W(s|i)$
schreiben
lesen
Verzögerungsspeicher
lesen und schreiben
$W(s|i) + \frac{\hat{p}(s)h(i)p(s|i)}{\sum_j h(j)p(s|j)}$
$\frac{\hat{p}(s)h(i)p(s|i)}{\sum_j h(j)p(s|j)}$
$W(s|i)$
$h(i) \cdot p(s|i)$
MULTI-NORM
MEM
$h(i)$
$p(s|i)$
$\frac{\hat{p}(s)}{\sum_i h(i)p(s|i)}$
Von Phase 1
(wenn NaN dann
auf 0 setzen)

FIG. 23

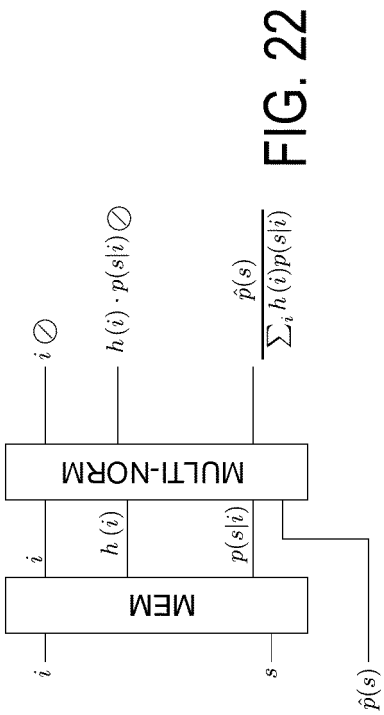

MULTI-NORM
MEM
$h(i) \cdot p(s|i) \oslash$
$\frac{\hat{p}(s)}{\sum_i h(i)p(s|i)}$
$h(i)$
$p(s|i)$
$\hat{p}(s)$

FIG. 22

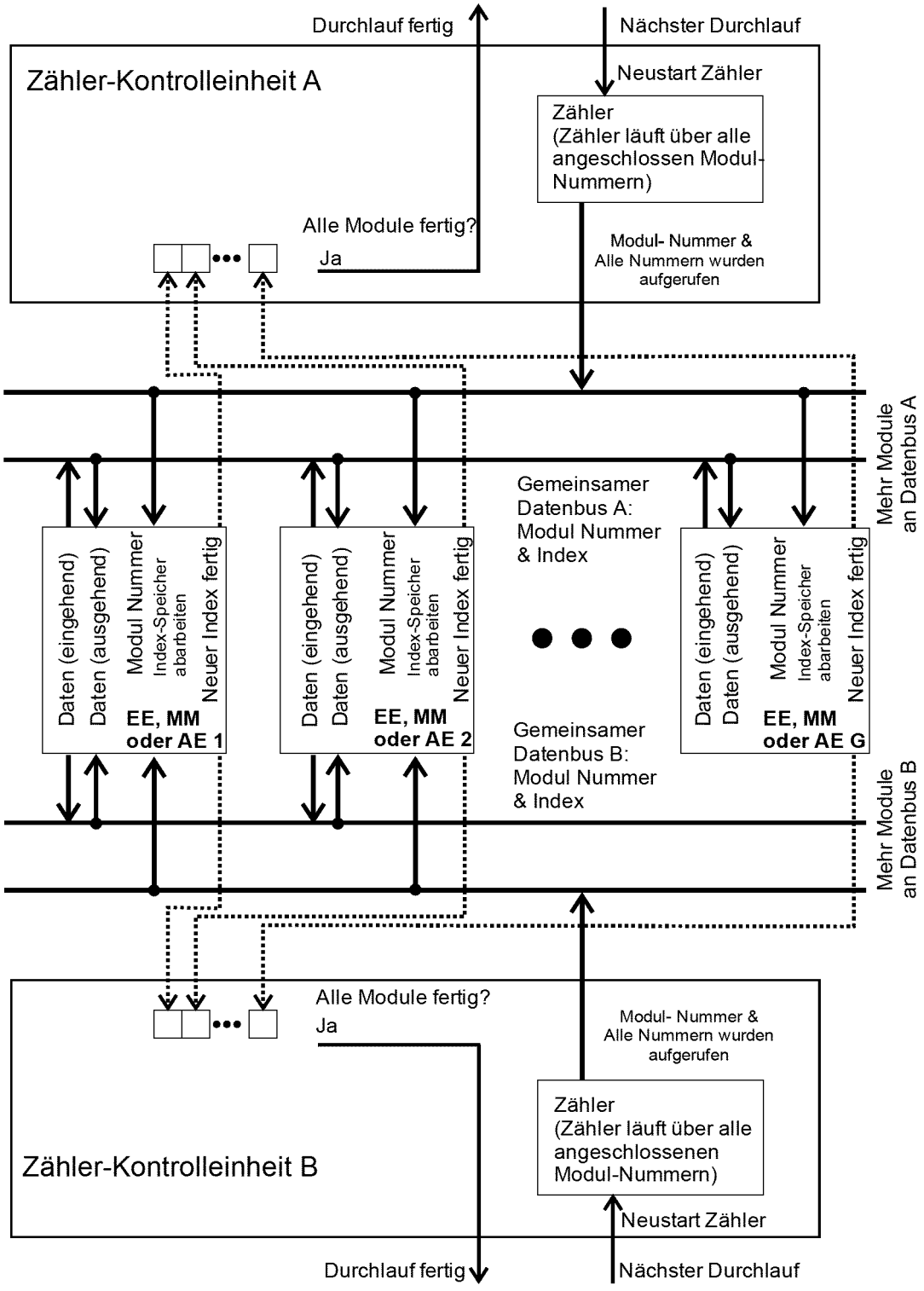

Durchlauf fertig
Nächster Durchlauf
Zähler-Kontrolleinheit A
Neustart Zähler
Zähler
(Zähler läuft über alle angeschlossen Modul-Nummern)
Alle Module fertig?
Ja
Modul- Nummer & Alle Nummern wurden aufgerufen
Mehr Module an Datenbus A
Gemeinsamer Datenbus A: Modul Nummer & Index
Daten (eingehend)
Daten (ausgehend)
Modul Nummer
Index-Speicher abarbeiten
Neuer Index fertig
EE, MM oder AE 1
EE, MM oder AE 2
EE, MM oder AE G
Gemeinsamer Datenbus B: Modul Nummer & Index
Mehr Module an Datenbus B
Alle Module fertig?
Ja
Modul- Nummer & Alle Nummern wurden aufgerufen
Zähler
(Zähler läuft über alle angeschlossenen Modul-Nummern)
Zähler-Kontrolleinheit B
Neustart Zähler
Durchlauf fertig
Nächster Durchlauf

FIG. 24

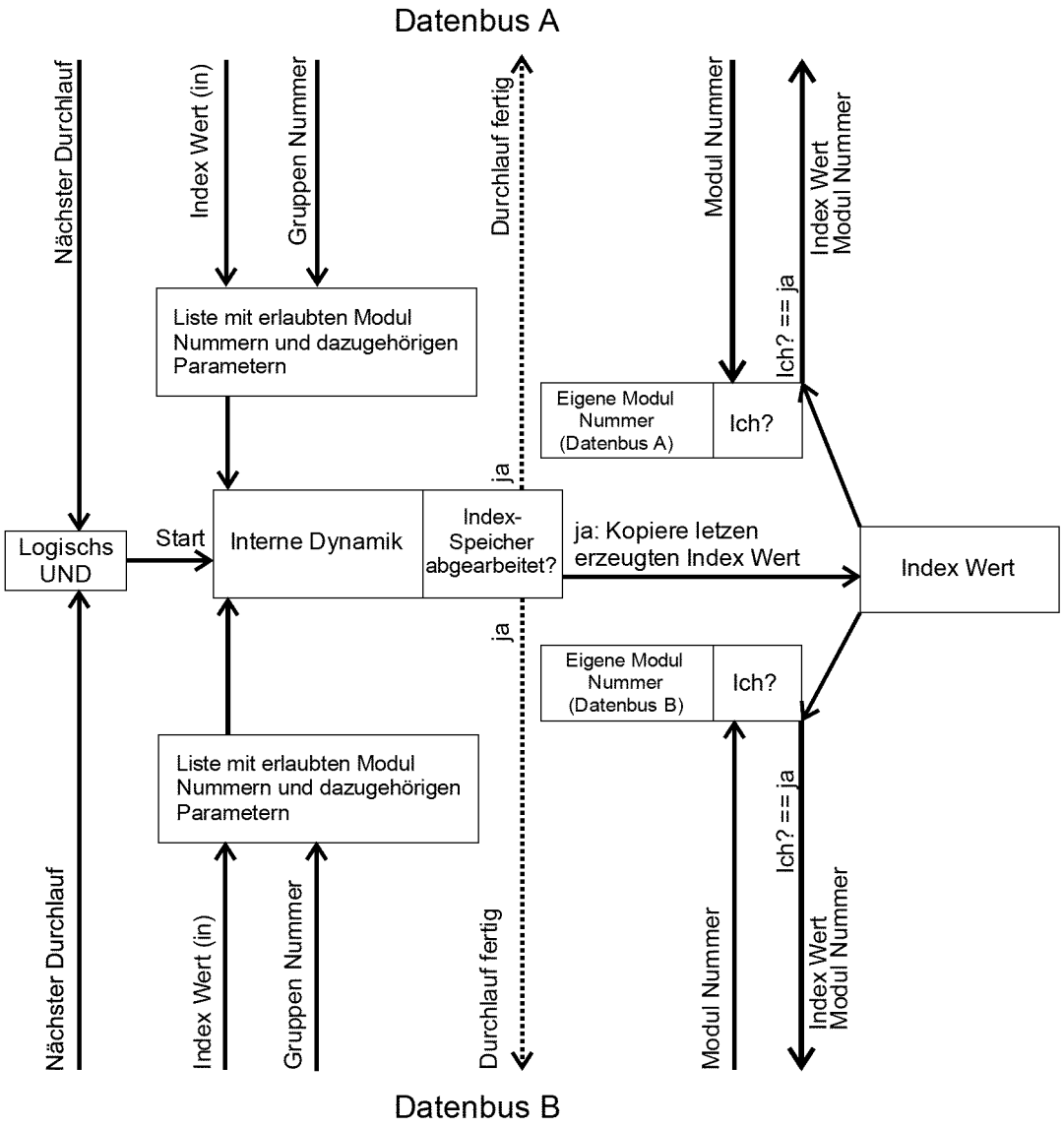

Datenbus A
Nächster Durchlauf
Index Wert (in)
Gruppen Nummer
Durchlauf fertig
Modul Nummer
Index Wert Modul Nummer
Liste mit erlaubten Modul Nummern und dazugehörigen Parametern
Eigene Modul Nummer (Datenbus A)
Ich?
Ich? == ja
ja
Logischs UND
Start
Interne Dynamik
Index-Speicher abgearbeitet?
ja: Kopiere letzen erzeugten Index Wert
Index Wert
ja
Eigene Modul Nummer (Datenbus B)
Ich?
Liste mit erlaubten Modul Nummern und dazugehörigen Parametern
Nächster Durchlauf
Index Wert (in)
Gruppen Nummer
Durchlauf fertig
Modul Nummer
Ich? == ja
Index Wert Modul Nummer
Datenbus B

FIG. 25

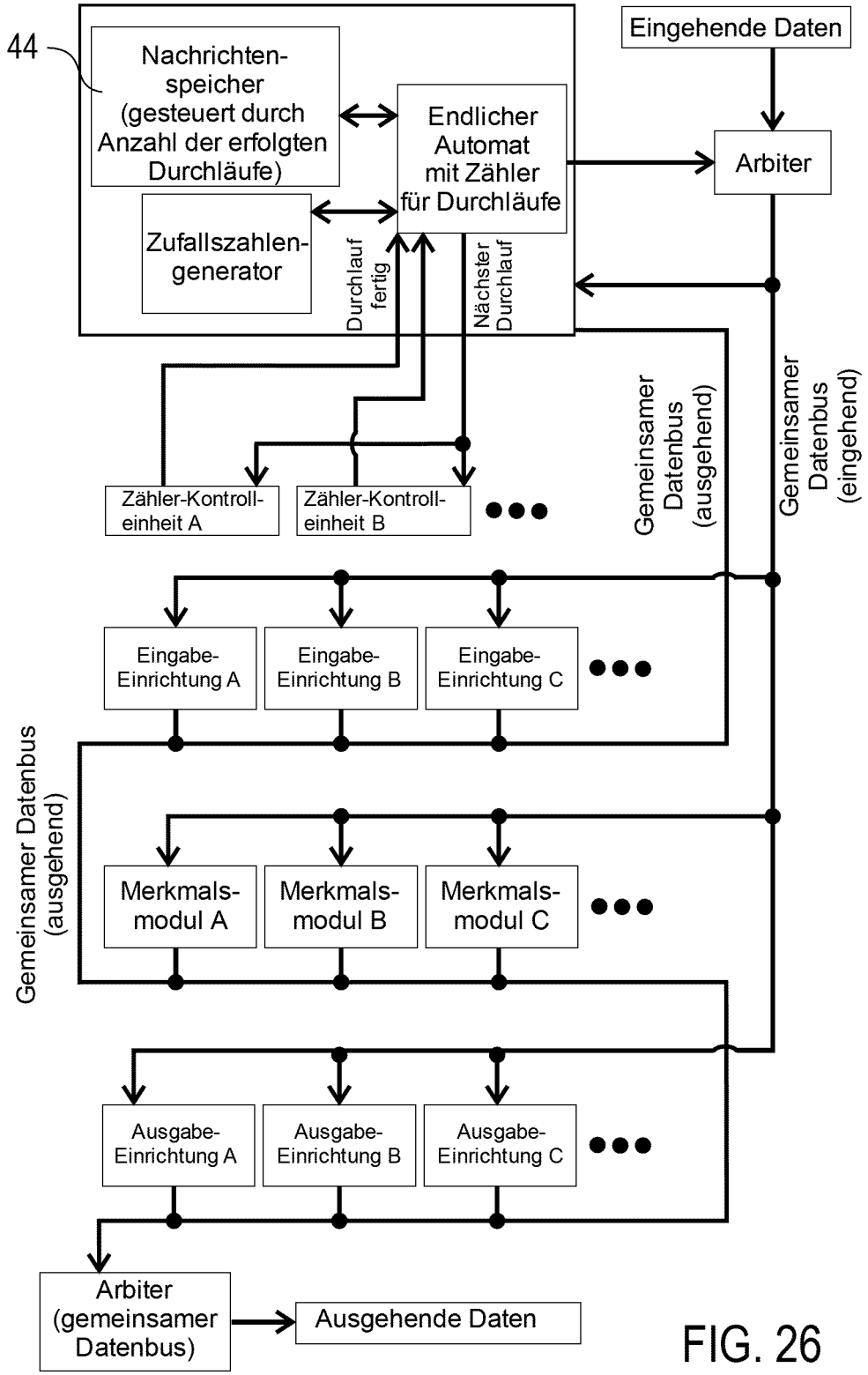

44
Nachrichten-speicher (gesteuert durch Anzahl der erfolgten Durchläufe)
Endlicher Automat mit Zähler für Durchläufe
Zufallszahlen-generator
Durchlauf fertig
Nächster Durchlauf
Eingehende Daten
Arbiter
Gemeinsamer Datenbus (eingehend)
Gemeinsamer Datenbus (ausgehend)
Zähler-Kontroll-einheit A
Zähler-Kontroll-einheit B
Eingabe-Einrichtung A
Eingabe-Einrichtung B
Eingabe-Einrichtung C
Gemeinsamer Datenbus (ausgehend)
Merkmals-modul A
Merkmals-modul B
Merkmals-modul C
Ausgabe-Einrichtung A
Ausgabe-Einrichtung B
Ausgabe-Einrichtung C
Arbiter (gemeinsamer Datenbus)
Ausgehende Daten

FIG. 26

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2019/079645

A. CLASSIFICATION OF SUBJECT MATTER

G06N 3/08(2006.01)i; ***G06N 3/04***(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	UDO ERNST ET AL. "Efficient Computation Based on Stochastic Spikes" *NEURAL COMPUTATION.*, US, Vol. 19, No. 5, 01 May 2007 (2007-05-01), pages 1313-1343 DOI: 10.1162/neco.2007.19.5.1313 ISSN: 0899-7667, XP055656095 cited in the application figures 1, 2-3, 11 paragraph [02.1] - paragraph [02.4] appendix B, F	1-35
A	AMIRHOSSEIN TAVANAEI ET AL. "Deep Learning in Spiking Neural Networks" *ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853,* 22 April 2018 (2018-04-22), XP081228628 abstract; figures 2, 6	1,9,10,18,26,27
A	US 2015242745 A1 (WANG XIN [US] ET AL) 27 August 2015 (2015-08-27) abstract paragraph [0007] paragraph [0038]	1,16,18

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 January 2020	**27 January 2020**

Name and mailing address of the ISA/EP	Authorized officer
European Patent Office **p.b. 5818, Patentlaan 2, 2280 HV Rijswijk** **Netherlands** Telephone No. **(+31-70)340-2040** Facsimile No. **(+31-70)340-3016**	**Bohn, Patrice** Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2019/079645

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MICHAEL PFEIFFER ET AL. "Deep Learning With Spiking Neurons: Opportunities and Challenges" *FRONTIERS IN NEUROSCIENCE,* CH, Vol. 12, 25 October 2018 (2018-10-25), DOI: 10.3389/fnins.2018.00774 ISSN: 1662-4548, XP055570520 page 4 - page 11	1-35

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/EP2019/079645

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2015242745 A1	27 August 2015	CA 2937949 A1	27 August 2015
		CN 106030620 A	12 October 2016
		EP 3108410 A2	28 December 2016
		JP 2017509978 A	06 April 2017
		KR 20160123309 A	25 October 2016
		TW 201541374 A	01 November 2015
		US 2015242745 A1	27 August 2015
		WO 2015127110 A2	27 August 2015

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/EP2019/079645

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
INV. G06N3/08 G06N3/04
ADD.

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
G06N

Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)
EPO-Internal, WPI Data

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	UDO ERNST ET AL: "Efficient Computation Based on Stochastic Spikes", NEURAL COMPUTATION., Bd. 19, Nr. 5, 1. Mai 2007 (2007-05-01), Seiten 1313-1343, XP055656095, US ISSN: 0899-7667, DOI: 10.1162/neco.2007.19.5.1313 in der Anmeldung erwähnt Abbildungen 1, 2-3, 11 Absatz [02.1] - Absatz [02.4] Appendix B, F ----- -/--	1-35

[X] Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen [X] Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :
"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
"E" frühere Anmeldung oder Patent, die bzw. das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist
"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absendedatum des internationalen Recherchenberichts
16. Januar 2020	27/01/2020

Name und Postanschrift der Internationalen Recherchenbehörde	Bevollmächtigter Bediensteter
Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Bohn, Patrice

2

C. (Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	AMIRHOSSEIN TAVANAEI ET AL: "Deep Learning in Spiking Neural Networks", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 22. April 2018 (2018-04-22), XP081228628, Zusammenfassung; Abbildungen 2, 6	1,9,10, 18,26,27
A	US 2015/242745 A1 (WANG XIN [US] ET AL) 27. August 2015 (2015-08-27) Zusammenfassung Absatz [0007] Absatz [0038]	1,16,18
A	MICHAEL PFEIFFER ET AL: "Deep Learning With Spiking Neurons: Opportunities and Challenges", FRONTIERS IN NEUROSCIENCE, Bd. 12, 25. Oktober 2018 (2018-10-25), XP055570520, CH ISSN: 1662-4548, DOI: 10.3389/fnins.2018.00774 Seite 4 - Seite 11	1-35

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen
PCT/EP2019/079645

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 2015242745 A1	27-08-2015	CA 2937949 A1	27-08-2015
		CN 106030620 A	12-10-2016
		EP 3108410 A2	28-12-2016
		JP 2017509978 A	06-04-2017
		KR 20160123309 A	25-10-2016
		TW 201541374 A	01-11-2015
		US 2015242745 A1	27-08-2015
		WO 2015127110 A2	27-08-2015