



(10) **DE 20 2024 100 807 U1** 2024.05.16

(12) **Gebrauchsmusterschrift**

(21) Aktenzeichen: **20 2024 100 807.3**

(51) Int Cl.: **G06N 3/092 (2023.01)**

(22) Anmeldetag: **20.02.2024**

(47) Eintragungstag: **04.04.2024**

(45) Bekanntmachungstag im Patentblatt: **16.05.2024**

(30) Unionspriorität:
18/441,920 14.02.2024 US

(74) Name und Wohnsitz des Vertreters:
Prock, Thomas, Dr., EC4A 1BW London, GB

(73) Name und Wohnsitz des Inhabers:
DeepMind Technologies Limited, London, GB

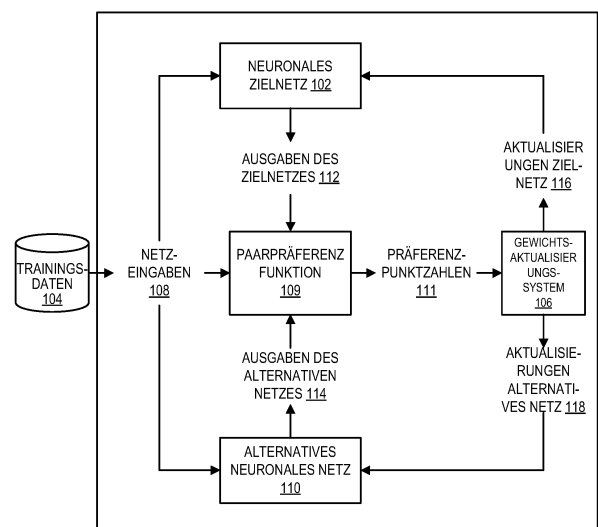
Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Verstärkungslernen durch Präferenz-Rückmeldung**

(57) Hauptanspruch: System, Folgendes umfassend:
einen oder mehrere Computer; und
eine oder mehrere Speichereinrichtungen, die kommunika-
tiv mit dem einen oder den mehreren Computern gekoppelt
sind, wobei die eine oder die mehreren Speichereinrichtun-
gen Anweisungen speichern, die, wenn sie durch den einen
oder die mehreren Computer ausgeführt werden, den einen
oder die mehreren Computer veranlassen, Vorgänge zum
Trainieren eines neuronalen Zielnetzes durchzuführen, das
eine Vielzahl von Gewichten des neuronalen Zielnetzes auf-
weist und dafür konfiguriert ist, eine Netzeingabe gemäß
den Gewichten des neuronalen Zielnetzes zu verarbeiten,
um eine Netzausgabe zu erzeugen, wobei die Vorgänge,
bei jedem von einer Vielzahl von Trainingsschritten, Folgen-
des umfassen:

Empfangen von einer oder mehreren Netzeingaben;
für jede der einen oder der mehreren Netzeingaben:
Verarbeiten der Netzeingabe unter Verwendung des neuro-
nalen Zielnetzes, um eine erste Netzausgabe zu erzeugen;
Verarbeiten der Netzeingabe unter Verwendung eines alter-
nativen neuronalen Netzes für den Trainingsschritt, um eine
zweite Netzausgabe zu erzeugen; und
Anwenden einer Präferenzfunktion auf die erste Netzaus-
gabe und die zweite Netzausgabe, um eine Präferenzpunkt-
zahl zu erzeugen, die eine Likelihood angibt, dass die erste
Netzausgabe eine Ausgabe von höherer Qualität als die
zweite Netzausgabe ist; und
Aktualisieren der Gewichte des neuronalen Zielnetzes unter
Verwendung einer Zielfunktion, die einen ersten Term ein-
schließt, der es begünstigt, gemäß den Präferenzpunktzah-
len die ersten Netzausgaben gegenüber den entsprechen-
den zweiten Netzausgaben zu bevorzugen.

TRAININGSSYSTEM 100



Beschreibung**HINTERGRUND**

[0001] Die vorliegende Patentschrift betrifft das Verarbeiten von Daten unter Verwendung von Maschinenlernmodellen.

[0002] Maschinenlernmodelle empfangen eine Eingabe und erzeugen auf der Grundlage der empfangenen Eingabe eine Ausgabe, z.B. eine vorhergesagte Ausgabe. Einige Maschinenlernmodelle sind parametrische Modelle und erzeugen die Ausgabe auf der Grundlage der empfangenen Eingabe und von Werten der Parameter des Modells.

[0003] Einige Maschinenlernmodelle sind tiefe Modelle, die mehrere Schichten von Modellen einsetzen, um eine Ausgabe für eine empfangene Eingabe zu erzeugen. Zum Beispiel ist ein tiefes neuronales Netz ein tiefes Maschinenlernmodell, das eine Ausgabeschicht und eine oder mehrere verborgene Schichten einschließt, die jeweils eine nichtlineare Transformation auf eine empfangene Eingabe anwenden, um eine Ausgabe zu erzeugen.

ZUSAMMENFASSUNG

[0004] Die vorliegende Patentschrift beschreibt Systeme und Verfahren, die als Computerprogramme auf einem oder mehreren Computern an einem oder mehreren Standorten implementiert sind, die neuronale Netze unter Verwendung einer Präferenzfunktion trainieren können, die ein Maß an Qualität zwischen Netzausgaben vergleicht.

[0005] Gemäß einem ersten Aspekt wird ein Verfahren bereitgestellt, das durch einen oder mehrere Computer durchgeführt wird, und zum Trainieren eines neuronalen Zielnetzes dient, das eine Vielzahl von Gewichten des neuronalen Zielnetzes aufweist und dafür konfiguriert ist, eine Netzeingabe gemäß den Gewichten des neuronalen Zielnetzes zu verarbeiten, um eine Netzausgabe zu erzeugen. Bei jedem von einer Vielzahl von Trainingsschritten schließt das Verfahren das Empfangen einer oder mehrerer Netzeingaben ein. Für jede der Netzeingaben schließt das Verfahren das Verarbeiten der Netzeingabe unter Verwendung des neuronalen Zielnetzes, um eine erste Netzausgabe zu erzeugen, das Verarbeiten der Netzeingabe unter Verwendung eines alternativen neuronalen Netzes für den Trainingsschritt, um eine zweite Netzausgabe zu erzeugen, und das Anwenden einer Präferenzfunktion auf die erste Netzausgabe und die zweite Netzausgabe ein, um eine Präferenzpunktzahl zu erzeugen, die eine Likelihood, angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist. Für jeden Trainingsschritt schließt das Verfahren das Aktualisieren der Gewichte des neuronalen Zielnetzes unter Verwendung einer Zielfunktion ein, die einen ersten Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen die ersten Netzausgaben gegenüber den entsprechenden zweiten Netzausgaben zu bevorzugen.

[0006] In einigen Implementierungen schließt das Verfahren ferner bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben (i) das Bestimmen einer ersten Likelihood-Punktzahl, die der ersten Netzausgabe durch das neuronale Zielnetz bei der gegebenen Netzeingabe zugewiesen wird, und (ii) das Bestimmen einer zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, ein, und die Zielfunktion schließt einen zweiten Term ein, der das neuronale Zielnetz für das Erzeugen erster Likelihood-Punktzahlen abstraf, die von entsprechenden zweiten Likelihood-Punktzahlen abweichen.

[0007] In einigen Implementierungen ist bei jedem der Vielzahl von Trainingsschritten das neuronale Referenznetz für den Trainingsschritt das alternative neuronale Netz für den Trainingsschritt.

[0008] In einigen Implementierungen schließt das Anwenden der Präferenzfunktion auf die erste Netzausgabe und die zweite Netzausgabe, um die Präferenzpunktzahl zu erzeugen, welche die Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist, Folgendes ein: (i) Verarbeiten der ersten Netzausgabe und der zweiten Netzausgabe unter Verwendung eines neuronalen Präferenzmodellnetzes, um eine Präferenznetzausgabe zu erzeugen, worin das neuronale Präferenzmodellnetz dafür trainiert ist, Paare von neuronalen Netzausgaben zu verarbeiten und für jedes verarbeitete Paar von neuronalen Netzausgaben eine entsprechende Präferenznetzausgabe zu erzeugen, die eine Likelihood angibt, dass ein Element des verarbeiteten Paares eine Ausgabe von höherer Qualität als das andere Element des verarbeiteten Paares ist, und (ii) auf der Grundlage der Präferenznetzausgabe erfolgreiches

Erzeugen der Präferenzpunktzahl, welche die Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist.

[0009] In einigen Implementierungen schließt das Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben Folgendes ein: Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe zugewiesen wird, auf der Grundlage einer geometrischen Mischung der ersten Likelihood-Punktzahl für die erste Netzausgabe und einer Likelihood-Punktzahl für die erste Netzausgabe unter einer festen Referenzverteilung und auf der Grundlage einer Normierungskonstante für den Trainingsschritt.

[0010] In einigen Implementierungen weist das alternative neuronale Netz für den Trainingsschritt bei jedem der Vielzahl von Trainingsschritten eine Vielzahl von Gewichten des alternativen neuronalen Netzes für den Trainingsschritt auf und weist die gleiche Architektur auf wie das neuronale Zielnetz, und die Gewichte des alternativen neuronalen Netzes für den Trainingsschritt werden durch einen exponentiellen gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes über die aktuellen und vorhergehenden Trainingsschritte bestimmt.

[0011] In einigen Implementierungen basiert die Zielfunktion für jede Netzeingabe und die entsprechenden ersten und zweiten Netzausgaben auf Folgendem: eine durch die Präferenzfunktion bestimmte Präferenzpunktzahl, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist; eine durch das neuronale Zielnetz bestimmte erste Likelihood-Punktzahl für die erste Netzausgabe; und eine durch eine feste Referenzverteilung bestimmte zweite Likelihood-Punktzahl für die erste Netzausgabe.

[0012] In einigen Implementierungen schließt das Aktualisieren der aktuellen Gewichte des neuronalen Zielnetzes unter Verwendung der Zielfunktion, die einen ersten Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen erste Netzeingaben gegenüber entsprechenden zweiten Netzeingaben zu bevorzugen, Folgendes ein : Bestimmen eines Gradienten der Zielfunktion in Bezug auf die Gewichte des neuronalen Zielnetzes und Aktualisieren der Gewichte des neuronalen Zielnetzes auf der Grundlage des Gradienten der Zielfunktion.

[0013] In einigen Implementierungen schließt das Bestimmen des Gradienten der Zielfunktion in Bezug auf die aktuellen Gewichte des neuronalen Zielnetzes für jede Netzeingabe und entsprechende erste und zweite Netzausgaben das Berechnen von Folgendem ein: einen Gradient einer ersten Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird, in Bezug auf die aktuellen Gewichte des neuronalen Zielnetzes; und eine Präferenzpunktzahl, die durch die Präferenzfunktion bestimmt wird, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist; die erste Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird; und eine Likelihood-Punktzahl für die erste Netzausgabe, die durch eine feste Referenzverteilung bestimmt wird.

[0014] In einigen Implementierungen ist das neuronale Zielnetz ein großes Sprachmodell.

[0015] In einigen Implementierungen ist das neuronale Präferenznetz ein großes Sprachmodell.

[0016] In einigen Implementierungen ist die erste Netzausgabe eine erste Netzausgabefolge; die zweite Netzausgabe ist eine zweite Netzausgabefolge; und das Verarbeiten der Netzeingabe unter Verwendung des alternativen neuronalen Netzes, um die zweite Netzausgabe zu erzeugen, schließt das autoregressive Erzeugen der zweiten Netzausgabefolge unter Verwendung des alternativen neuronalen Netzes ein.

[0017] In einigen Implementierungen schließt das Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben Folgendes ein: für jedes Ausgabeelement der ersten Netzausgabefolge erfolgreiches Bestimmen einer ersten Likelihood-Punktzahl für das Ausgabeelement bei der gegebenen Netzeingabe unter Verwendung des neuronalen Zielnetzes und Bestimmen einer zweiten Likelihood-Punktzahl für das Ausgabeelement auf der Grundlage einer geometrischen Mischung der ersten Likelihood-Punktzahl für das Ausgabeelement und einer Likelihood-Punktzahl für das Ausgabeelement unter einer festen Referenzverteilung und auf der Grundlage einer Normierungskonstante für den Trainingsschritt; und Bestimmen der zweiten

Likelihood-Punktzahl, die der ersten Netzausgabe zugewiesen wird, auf der Grundlage der zweiten Likelihood-Punktzahlen der Ausgabeelemente in der ersten Netzausgabefolge.

[0018] In einigen Implementierungen kann die Präferenzfunktion eine nicht-transitive Präferenzbeziehung modellieren.

[0019] Gemäß einem weiteren Aspekt wird ein System bereitgestellt, das einen oder mehrere Computer und eine oder mehrere Speichereinrichtungen einschließt, die kommunikativ mit dem einen oder den mehreren Computern gekoppelt sind und Anweisungen speichern, die, wenn sie durch den einen oder die mehreren Computer ausgeführt werden, den einen oder die mehreren Computer veranlassen, Vorgänge des zuvor beschriebenen Verfahrens durchzuführen.

[0020] Gemäß einem anderen Aspekt werden ein oder mehrere nichtflüchtige Computerspeichermedien bereitgestellt, auf denen Anweisungen gespeichert sind, die, wenn sie durch einen oder mehrere Computer ausgeführt werden, den einen oder die mehreren Computer veranlassen, Vorgänge des zuvor beschriebenen Verfahrens durchzuführen.

[0021] Bestimmte Ausführungsformen des in der vorliegenden Patentschrift beschriebenen Gegenstandes können so implementiert werden, dass dadurch einer oder mehrere der folgenden Vorteile verwirklicht werden.

[0022] Die beschriebenen Systeme können eine Paarpräferenz-Funktion verwenden, um das neuronale Ziel zu trainieren, das ein Paar von Netzausgaben verarbeitet, um eine Präferenzpunktzahl zu bestimmen. In einigen Implementierungen können die beschriebenen Systeme ein Präferenzmodell dafür trainieren, die Paarpräferenz-Funktion zu modellieren. Anders als herkömmliche Verfahren zum Trainieren neuronaler Netze auf der Grundlage von Präferenzen, die häufig auf dem Anpassen eines Belohnungsmodells beruhen, um einzelnen Netzausgaben numerische Werte zuzuweisen, die sich an den Präferenzen ausrichten, können die beschriebenen Systeme nicht-transitive Präferenzen modellieren. Durch Verwenden einer Paarpräferenz-Funktion können die beschriebenen Systeme auch transitive Präferenzen genauer modellieren als herkömmliche Verfahren. Die beschriebenen Systeme können daher neuronale Netze auf der Grundlage von Präferenzen für ein breiteres Spektrum von Anwendungen (z.B. Modellieren menschlicher Präferenzen, Vorhersagen von Ergebnissen von Wettbewerbsspielen usw.) als herkömmliche Verfahren genau trainieren.

[0023] Durch Verwendung der Paarpräferenz-Funktion, um Präferenzen für die bedingte Erzeugungsaufgabe genauer zu modellieren, können die beschriebenen Systeme das neuronale Zielnetz so trainieren, dass es einen bestimmten Schwellenwert des Leistungsvermögens bei der bedingten Erzeugungsaufgabe (der z.B. auf der Grundlage der Präferenzfunktion bestimmt wird) in weniger Trainingsiterationen als herkömmliche Verfahren erreicht. Die beschriebenen Systeme können daher neuronale Netze auf der Grundlage von Präferenzen effizienter (etwa in Bezug auf Trainingszeit, Rechenressourcen usw.) trainieren als herkömmliche Verfahren.

[0024] Herkömmliche Verfahren trainieren oft neuronale Netze dafür, Belohnungsmodelle zu optimieren, die auf der Grundlage von Präferenzdaten angepasst worden sind, und können anfällig für Belohnungs-Hacking sein, bei dem die Netzausgaben, welche die Belohnungen gemäß dem Belohnungsmodell maximieren, nicht gegenüber Netzausgaben bevorzugt werden, die schlechtere Belohnungen empfangen. Im Gegensatz dazu können die beschriebenen Systeme die neuronalen Zielnetze dafür trainieren, sich einem Nash-Gleichgewicht der Präferenzen in Bezug auf einen Satz von alternativen neuronalen Netzen anzunähern (z.B. um Netzausgaben zu erzeugen, die im Vergleich zu den durch den Satz von alternativen neuronalen Netzen erzeugten Netzausgaben mindestens genauso bevorzugt sind).

[0025] Insbesondere können die beschriebenen Systeme die neuronalen Zielnetze durch Vergleichen von Ausgaben aus den neuronalen Zielnetzen mit entsprechenden Ausgaben der alternativen neuronalen Netze trainieren. Die beschriebenen Systeme können eine Zielfunktion verwenden, die einen Term einschließt, der die Präferenzen der Ausgaben des neuronalen Zielnetzes im Vergleich zu entsprechenden alternativen neuronalen Netzausgaben angibt. Während des Trainings der neuronalen Zielnetze können die beschriebenen Systeme bei jedem Trainingsschritt die alternativen neuronalen Netze auf der Grundlage des Trainings der neuronalen Zielnetze aktualisieren. Durch Vergleichen von Präferenzen für die Ausgaben des neuronalen Zielnetzes mit entsprechenden Netzausgaben aus alternativen neuronalen Netzen, die bei jedem Trainingsschritt aktualisiert werden, kann die Zielfunktion der beschriebenen Systeme die neuronalen Zielnetze darin bestärken, sich einem Nash-Gleichgewicht der Präferenzen in Bezug auf einen Satz von alternativen neuro-

nen Netzen anzunähern. Indem die neuronalen Zielnetze dafür trainiert werden, sich einem Nash-Gleichgewicht der Präferenzen anzunähern, können die beschriebenen Systeme im Vergleich zu herkömmlichen Verfahren weniger anfällig für Belohnungs-Hacking sein.

[0026] Die beschriebenen Systeme können außerdem das Training des neuronalen Zielnetzes in Bezug auf ein neuronales Referenznetz reglementieren. Insbesondere können die beschriebenen Systeme eine Zielfunktion verwenden, die einen Term einschließt, der das neuronale Zielnetz dafür abstrafte, zu stark vom neuronalen Referenznetz abzuweichen. Zum Beispiel kann das neuronale Referenznetz bekannte sichere oder erwünschte Ausgaben erzeugen, und das beschriebene System kann das neuronale Zielnetz auf der Grundlage der Präferenzen trainieren und dabei Ausgaben erzeugen, die denen des neuronalen Referenznetzes ähnlich sind.

[0027] Das beschriebene System kann daher ein genaueres und effizienteres Training von neuronalen Netzen auf der Grundlage von Präferenzen ermöglichen und gleichzeitig sicherstellen, dass das neuronale Zielnetz bevorzugte Ausgaben erzeugen kann.

[0028] Die Einzelheiten einer oder mehrerer Ausführungsformen des Gegenstandes der vorliegenden Patentschrift sind in den beigefügten Zeichnungen und der nachstehenden Beschreibung dargelegt. Andere Merkmale, Aspekte und Vorteile des Gegenstandes werden aus der Beschreibung, den Zeichnungen und den Ansprüchen ersichtlich.

KURZBESCHREIBUNG DER ZEICHNUNGEN

Abb. 1 ist ein Blockschaltbild eines beispielhaften Trainingssystems.

Abb. 2 ist ein Blockschaltbild eines beispielhaften neuronalen Zielnetzes.

Abb. 3 ist ein Ablaufplan eines beispielhaften Prozesses zum Trainieren eines neuronalen Zielnetzes.

Abb. 4 ist ein Ablaufplan eines beispielhaften Prozesses zum Trainieren eines neuronalen Präferenzmodellnetzes.

Abb. 5 ist ein Ablaufplan eines beispielhaften Prozesses zum Bestimmen einer Zielfunktion für einen Trainingsschritt.

[0029] Gleiche Bezugszeichen und Bezeichnungen in den verschiedenen Zeichnungen geben gleiche Elemente an.

AUSFÜHRLICHE BESCHREIBUNG

[0030] **Abb. 1** zeigt ein beispielhaftes Trainingssystem 100. Das Trainingssystem 100 ist ein Beispiel für ein System, das als Computerprogramme auf einem oder mehreren Computern an einem oder mehreren Standorten implementiert ist, in dem die nachstehend beschriebenen Systeme, Komponenten und Methoden implementiert sind.

[0031] Das Trainingssystem 100 kann ein neuronales Zielnetz 102 unter Verwendung eines Satzes von Trainingsdaten 104 dafür trainieren, eine bedingte Erzeugungsaufgabe durchzuführen. Insbesondere kann das System 100 das neuronale Zielnetz 102 dafür trainieren, die bedingte Erzeugungsaufgabe im Einklang mit Präferenzen für die Aufgabe durchzuführen. Das neuronale Zielnetz 102 weist einen Satz von Gewichten des neuronalen Zielnetzes auf, die beispielsweise die Gewichte und optional die Verzerrungen der Schichten des neuronalen Zielnetzes einschließen. Das neuronale Zielnetz 102 kann ein neuronales Netz sein, das dafür trainiert ist (z.B. vortrainiert), die bedingte Erzeugungsaufgabe durchzuführen, und das System 102 kann das neuronale Zielnetz 102 weiter trainieren (z.B. feinabstimmen), um die bedingte Erzeugungsaufgabe im Einklang mit Präferenzen für die Aufgabe durchzuführen.

[0032] Das Trainingssystem 100 schließt ein Gewichtsaktualisierungssystem 106 ein, das über eine Reihe von Trainingsschritten die Gewichte des neuronalen Zielnetzes aktualisieren kann, um das neuronale Zielnetz 102 zu trainieren. Bei jedem Trainingsschritt kann das Trainingssystem 100 das neuronale Zielnetz 102 dafür trainieren, die bedingte Erzeugungsaufgabe an einem Satz 108 (z.B. einem Mini-Batch) von Netzeingaben durchzuführen.

[0033] Das neuronale Zielnetz 102 kann jedes beliebige generative Modell sein, das geeignet ist, Ausgabemuster auf der Grundlage der Netzeingaben 108 zu erzeugen. Das neuronale Zielnetz 102 wird nachstehend unter Bezugnahme auf **Abb. 2** ausführlicher beschrieben.

[0034] Die bedingte Erzeugungsaufgabe kann eine beliebige aus einer Vielfalt von Aufgaben sein. Zum Beispiel können die Netzeingaben 108 Textprompts sein, und die bedingte Erzeugungsaufgabe kann darin bestehen, Ausgabemuster zu erzeugen, die durch die Textprompts beschrieben werden. Als weiteres Beispiel kann das neuronale Zielnetz 102 dafür konfiguriert sein, z.B. Musterbilder, Mustervideos, Musteraudiosignale usw. auszugeben, die durch Textprompts aus den Netzeingaben 108 beschrieben werden.

[0035] Die bedingte Erzeugungsaufgabe kann eine Stilübertragungsaufgabe sein, um einen Stil von z.B. Bildern, Videos, Audio usw. anzupassen, der durch die Netzeingaben 108 spezifiziert wird. Zum Beispiel kann das neuronale Zielnetz 102 dafür konfiguriert sein, z.B. Musterbilder, Mustervideos, Musteraudiosignale usw. auszugeben, die zu Stilen der Netzeingaben 108 passen.

[0036] Die bedingte Erzeugungsaufgabe kann eine Transkriptionsaufgabe sein, bei der das neuronale Zielnetz 102 dafür konfiguriert ist, Texttranskriptionen z.B. für Video, Audio usw. zu erzeugen, die durch die Netzeingaben 108 spezifiziert wird.

[0037] Die bedingte Erzeugungsaufgabe kann eine Zusammenfassungsaufgabe sein, bei der das neuronale Zielnetz 102 dafür konfiguriert ist, Textzusammenfassungen von z.B. Text, Bildern, Video, Audio usw. zu erzeugen, die durch die Netzeingaben 108 spezifiziert werden.

[0038] Die bedingte Erzeugungsaufgabe kann eine Aufgabe der Verarbeitung natürlicher Sprache (NLP) sein, und das neuronale Zielnetz 102 kann ein großes Sprachmodell sein, das dafür konfiguriert ist, die Sprachverarbeitungsaufgabe durchzuführen. Zum Beispiel kann das neuronale Zielnetz 102 dafür konfiguriert sein, Textantworten auf Netzeingaben 108 zu erzeugen, die durch einen Benutzer bereitgestellt werden. Als weiteres Beispiel kann das neuronale Zielnetz 102 dafür konfiguriert sein, Code für Computerprogramme (z.B. in einer Programmiersprache wie Python, Java, C++ usw.) zu erzeugen, um Aufgaben durchzuführen, die durch die Netzeingaben 108 spezifiziert werden, die durch einen Benutzer bereitgestellt werden.

[0039] Die bedingte Erzeugungsaufgabe kann eine Aktionsauswahlaufgabe für einen Agenten sein, der mit einer Umgebung interagiert. Zum Beispiel können die Netzeingaben 108 etwa Sensordaten für den Agenten, beobachtete Trajektorien anderer Agenten innerhalb der Umgebung usw. sein, die Beobachtungen der Umgebung kennzeichnen, und das neuronale Zielnetz 102 kann dafür konfiguriert werden, Aktionen für den Agenten auf der Grundlage von Beobachtungen der Umgebung zu beproben. Als weiteres Beispiel können die Netzeingaben 108 etwa LIDAR-Daten, Kameradaten usw. sein, die für ein autonomes Fahrzeug gesammelt wurden, und das neuronale Zielnetz 102 kann dafür konfiguriert werden, Aktionen zur Steuerung des autonomen Fahrzeugs auszuwählen. Als weiteres Beispiel können die Netzeingaben 108 den Stand eines Spiels kennzeichnen, und das neuronale Zielnetz 102 kann dafür konfiguriert werden, Aktionen für einen Spieler des Spiels auszuwählen.

[0040] Das Trainingssystem 100 trainiert das neuronale Zielnetz 102 unter Verwendung einer Paarpräferenz-Funktion 109 dafür, die bedingte Erzeugungsaufgabe durchzuführen. Die Paarpräferenz-Funktion 109 kann ein gegebenes Ausgabemuster aus dem neuronalen Zielnetz 102 mit einem entsprechenden Ausgabemuster aus einem alternativen neuronalen Netz 110 vergleichen, um eine Präferenzpunktzahl 111 zu erzeugen, die eine Likelihood angibt, dass das Ausgabemuster aus dem neuronalen Zielnetz 102 eine Ausgabe von höherer Qualität als das Ausgabemuster aus dem alternativen neuronalen Netz 110 ist.

[0041] Das alternative neuronale Netz 110 kann ein beliebiges aus einer Vielfalt von Netzen sein, die alternative (relativ zum neuronalen Zielnetz 102) Ausgabemuster für die bedingte Erzeugungsaufgabe bereitstellen können. Beispiele für das alternative neuronale Netz 110 werden nachstehend unter Bezugnahme auf **Abb. 5** beschrieben.

[0042] Zum Beispiel kann die Präferenzpunktzahl 111 angeben, ob ein Ausgabemuster aus dem neuronalen Zielnetz 102 oder ein entsprechendes Ausgabemuster aus dem alternativen neuronalen Netz 110 mit größerer Wahrscheinlichkeit aus einer Grundwahrheitsverteilung abgeleitet wird.

[0043] Als ein anderes Beispiel kann die Präferenzpunktzahl 111 angeben, ob ein Ausgabemuster aus dem neuronalen Zielnetz 102 oder ein entsprechendes Ausgabemuster aus dem alternativen neuronalen Netz

110 mit größerer Wahrscheinlichkeit eine geringere Verzerrung, Wahrnehmungsverzerrung usw., für die bedingte Erzeugungsaufgabe aufweist.

[0044] Als ein anderes Beispiel kann die Präferenzpunktzahl 111 angeben, ob ein Ausgabemuster aus dem neuronalen Zielnetz 102 oder ein entsprechendes Ausgabemuster aus dem alternativen neuronalen Netz 110 mit größerer Wahrscheinlichkeit eine größere Belohnung für die bedingte Erzeugungsaufgabe empfängt.

[0045] Als noch ein anderes Beispiel kann die Präferenzpunktzahl 111 eine menschliche Präferenz zwischen einem Ausgabemuster aus dem neuronalen Zielnetz 102 und einem entsprechenden Ausgabemuster aus dem alternativen neuronalen Netz 110 quantifizieren.

[0046] Insbesondere kann die Präferenzfunktion 109 nicht-transitive Präferenzen zwischen Netzausgaben modellieren, und zwar für Netzausgaben A, B, und C, wenn die Präferenzfunktion bestimmt, dass A gegenüber B zu bevorzugen ist (d.h., dass es wahrscheinlicher ist, dass es eine Ausgabe von höherer Qualität als B ist) und dass B gegenüber C zu bevorzugen ist, und die Präferenzfunktion 109 kann bestimmen, dass C gegenüber A zu bevorzugen ist. Im obigen Beispiel würde ein Modell, das auf die Modellierung transitiver Präferenzen beschränkt ist, zwangsläufig bestimmen, dass A gegenüber C zu bevorzugen ist. Nicht-transitive Präferenzen können in einer Vielfalt von Anwendungen gefunden werden (wie etwa Modellierung menschlicher Präferenzen, Vorhersage der Leistung von Spielern in einem Wettbewerbs spiel usw.). Die Präferenzfunktion 109 kann daher im Vergleich zu Modellen, die auf die Modellierung transitiver Präferenzen beschränkt sind, Präferenzen für ein breiteres Spektrum von Anwendungen genau modellieren.

[0047] Bei jedem Trainingsschritt verwendet das System 102 die Paarpräferenz-Funktion 109, um Zielnetzausgaben 112, die Ausgabemuster aus dem neuronalen Zielnetz 102 einschließen, mit entsprechenden Ausgaben des alternativen Netzes 114, die Ausgabemuster aus dem alternativen neuronalen Netz 110 einschließen, zu vergleichen, um Präferenzpunktzahlen 111 für den Trainingsschritt zu erzeugen. Auf der Grundlage der Präferenzpunktzahlen 111 erzeugt das Gewichtaktualisierungssystem 106 Aktualisierungen 116 für die Gewichte des neuronalen Zielnetzes 102 und Aktualisierungen 118 für das alternative neuronale Netz 110. Das Gewichtaktualisierungssystem 106 kann die Gewichtaktualisierungen 116 und 118 erzeugen, um das neuronale Zielnetz 102 dafür zu trainieren, die bedingte Erzeugungsaufgabe durchzuführen. Der Prozess des Trainings des neuronalen Zielnetzes 102 wird nachstehend unter Bezugnahme auf **Abb. 3** ausführlicher beschrieben.

[0048] **Abb. 2** zeigt ein Beispiel für ein neuronales Zielnetz 102.

[0049] Das neuronale Zielnetz 102 schließt ein generatives Modell 202 ein, das die Netzeingaben 108 verarbeiten und Ausgabemuster 204 erzeugen kann, um eine bedingte Erzeugungsaufgabe durchzuführen.

[0050] Das generative Modell 202 kann jegliche zweckdienliche Architektur für das Verarbeiten der Netzeingaben 108 und das Erzeugen der Ausgabemuster 204 aufweisen, um die bedingte Erzeugungsaufgabe durchzuführen. Als Beispiel kann das generative Modell 202 ein autoregressives generatives Modell sein (wie etwa ein Transformer, ein rekurrentes neuronales Netz usw.), das auf der Grundlage der Netzeingaben 108 autoregressiv eine Folge als die Ausgabemuster 204 erzeugen kann. Das generative Modell kann zum Beispiel ein großes Sprachmodell (LLM) sein, das autoregressiv tokenisierte Darstellungen von Textdaten erzeugen kann, ein Sicht-Sprachmodell (VLM), das autoregressiv tokenisierte Darstellungen von Bild- oder Videodaten erzeugen kann, ein Audio-Sprachmodell, das autoregressiv tokenisierte Darstellungen von Textdaten erzeugen kann, und so weiter.

[0051] Als ein anderes Beispiel kann das generative Modell 202 ein Diffusionsmodell sein (wie etwa ein entauschendes Diffusionsmodell, ein punktzahlgestütztes Diffusionsmodell, ein latentes Diffusionsmodell usw.), das die Ausgabemuster 204 durch wiederholtes Transformieren von Mustern aus einer Rauschverteilung (wie etwa einer Gauß-Verteilung) auf der Grundlage der Netzeingaben 108 über eine Folge von Iterationen erzeugen kann. Zum Beispiel kann das generative Modell 202 ein Diffusionsmodell sein, das Muster aus der Rauschverteilung unter Verwendung eines entauschenden neuronalen Netzes mit jeglicher zweckdienlichen Architektur (wie etwa ein faltendes neuronales Netz, ein rekurrentes neuronales Netz usw.) transformiert.

[0052] Als ein anderes Beispiel kann das generative Modell 202 ein neuronales Netz sein, das die Ausgabemuster 204 durch Transformation von Mustern aus einer Rauschverteilung (wie etwa einer Gauß-Verteilung) erzeugen kann. Das generative Modell 202 kann z.B. ein Generatorknetz eines generativen gegnerischen Netzes, ein Decodierer eines Variationsautocodierers, ein normierter Fluss, usw. sein.

[0053] Das generative Modell 202 kann Netzausgaben 208 erzeugen, die jeweilige bedingte Wahrscheinlichkeiten oder Likelihoods der Ausgabemuster 204 bei den gegebenen entsprechenden Netzeingaben 108 definieren. Zum Beispiel können die Netzausgaben 208 bedingte Wahrscheinlichkeiten für einen Satz von Werten für jedes der Ausgabemuster 204 bei den gegebenen entsprechenden Netzeingaben einschließen. Als ein anderes Beispiel können die Netzausgaben 208 statistische Eigenschaften (wie etwa Mittelwerte, Varianzen usw.) in Bezug auf bedingte Verteilungen für jedes der Ausgabemuster 204 bei den gegebenen entsprechenden Netzeingaben 108 bestimmen. Als ein anderes Beispiel kann das generative Modell 202 eine Vielzahl von Verarbeitungsschichten und eine Vielzahl von Schichtaktivierungsfunktionen einschließen, und die Netzausgaben 208 können Ausgabewerte von den Verarbeitungsschichten und Schichtaktivierungsfunktionen einschließen, die sich aus der Verarbeitung der entsprechenden Netzeingaben 108 durch das generative Modell 202 ergeben.

[0054] Das neuronale Zielnetz 102 kann ein Muster-Likelihood-System 206 einschließen. Das Muster-Likelihood-System 206 kann die Daten verarbeiten, welche die bedingten Verteilungen 208 spezifizieren, um Likelihood-Punktzahlen 210 zu erzeugen, die Likelihoods von jedem der Ausgabemuster 204 unter den bedingten Verteilungen 208 bei den gegebenen entsprechenden Netzeingaben 108 spezifizieren.

[0055] Die Likelihood-Punktzahlen 210 können beliebige aus einer Vielfalt von numerischen Werten sein, welche bedingte Wahrscheinlichkeiten von Likelihoods der Ausgabemuster 204 bei den gegebenen entsprechenden Netzeingaben 108 angeben. Zum Beispiel können die Likelihood-Punktzahlen 210 die bedingten Wahrscheinlichkeiten oder Likelihoods der Ausgabemuster 204 bei den gegebenen entsprechenden Netzeingaben 108 sein. Als ein anderes Beispiel können die Likelihood-Punktzahlen 210 Transformationen der bedingten Wahrscheinlichkeiten oder Likelihoods (wie etwa logarithmische Wahrscheinlichkeiten, logarithmische Likelihoods, Likelihoods mit negativem Logarithmus, nicht normierte Werte usw.) der Ausgabemuster 204 bei den gegebenen Netzeingaben 108 sein. Als ein anderes Beispiel kann das Muster-Likelihood-System 206 Hilfsverteilungen bestimmen, welche bedingte Verteilungen der Ausgabemuster 204 bei den gegebenen entsprechenden Netzeingaben 108 approximieren, und die Likelihood-Punktzahlen 210 können Wahrscheinlichkeiten oder Likelihoods der Ausgabemuster 204 unter den Hilfsverteilungen sein.

[0056] Das Muster-Likelihood-System 206 kann die Likelihood-Punktzahlen 210 auf der Grundlage der Netzausgaben 208 durch ein beliebiges aus einer Vielfalt von Verfahren bestimmen. Zum Beispiel können die Netzausgaben 208 Sätze von bedingten Wahrscheinlichkeiten der Ausgabemuster 204 einschließen, und das Muster-Likelihood-System kann die bedingten Wahrscheinlichkeiten der Ausgabemuster 204 als die Likelihood-Punktzahlen 210 zurückgeben. Als ein anderes Beispiel können die Netzausgaben 208 Daten einschließen, die bedingte Verteilungen für die Ausgabemuster 204 spezifizieren, und das Muster-Likelihood-System 206 kann die Likelihoods der Ausgabemuster 204 unter den spezifizierten bedingten Verteilungen als die Likelihood-Punktzahlen 210 zurückgeben. Als ein anderes Beispiel können die Netzausgaben 208 Daten einschließen, die bedingte Verteilungen für die Ausgabemuster 204 spezifizieren, und das Muster-Likelihood-System 206 kann approximative Likelihoods der Ausgabemuster 204 unter den spezifizierten bedingten Verteilungen als die Likelihood-Punktzahlen 210 bestimmen und zurückgeben. Als spezielles Beispiel können die Netzausgaben 208 Daten einschließen, welche Schichtausgaben aus dem generativen Modell 202 spezifizieren, und das Muster-Likelihood-System kann approximative Likelihoods bestimmen (wie etwa durch Bestimmen von Näherungen der bedingten Verteilungen, Verwenden von Wichtigkeitsgewichten usw.), um sie als die Likelihood-Punktzahlen 210 zurückzugeben. Als ein anderes spezielles Beispiel kann das Muster-Likelihood-System 206 ein neuronales Netz einschließen, das dafür trainiert ist, bedingte Verteilungen der Ausgabemuster 204 bei den gegebenen Netzeingaben 108 zu approximieren, und kann die Netzeingaben 108 verarbeiten, um approximative Likelihoods der Ausgabemuster 204 zu bestimmen, um sie als die Likelihood-Punktzahlen 210 zurückzugeben.

[0057] **Abb. 3** ist ein Ablaufplan eines beispielhaften Prozesses zum Trainieren eines neuronalen Zielnetzes, um eine bedingte Erzeugungsaufgabe durchzuführen. Der Einfachheit halber wird der Prozess 300 so beschrieben, dass er durch ein System aus einem oder mehreren Computern durchgeführt wird, die sich an einem oder mehreren Standorten befinden. Zum Beispiel kann ein Trainingssystem wie etwa das Trainingssystem 100 von **Abb. 1**, das gemäß der vorliegenden Patentschrift zweckdienlich programmiert ist, den Prozess 300 durchführen.

[0058] Das Trainingssystem kann das neuronale Zielnetz über eine Vielzahl von Trainingsschritten trainieren.

[0059] Bei jedem Trainingsschritt kann das Trainingssystem einen Satz aus einer oder mehreren Netzeingaben für den Trainingsschritt empfangen (Schritt 302). Zum Beispiel kann das System einen Mini-Batch von Netzeingaben aus einem Satz von Trainingsdaten empfangen.

[0060] Das System kann die Netzeingaben für den Trainingsschritt unter Verwendung des neuronalen Zielnetzes verarbeiten, um entsprechende Netzausgaben zu erzeugen (Schritt 304). Insbesondere kann das neuronale Zielnetz jede der Netzeingaben verarbeiten, um ein entsprechendes Ausgabemuster zu erzeugen, um die bedingte Erzeugungsaufgabe durchzuführen.

[0061] Das System kann die Netzeingaben für den Trainingsschritt unter Verwendung eines alternativen neuronalen Netzes für den Trainingsschritt verarbeiten, um entsprechende Netzausgaben zu erzeugen (Schritt 306). Insbesondere kann das alternative neuronale Netz jede der Netzeingaben verarbeiten, um ein entsprechendes Ausgabemuster zu erzeugen, um die bedingte Erzeugungsaufgabe durchzuführen.

[0062] Für jede der Netzeingaben kann das System eine Präferenzfunktion auf die entsprechenden Netzausgaben aus dem Ziel- und dem alternativen neuronalen Netz anwenden, um eine Präferenzpunktzahl zu bestimmen, welche die Netzausgaben vergleicht (Schritt 308). Für jedes Paar von Netzausgaben aus dem Ziel- und dem alternativen neuronalen Netz gibt die Punktzahl eine Likelihood dafür an, dass die Netzausgabe aus dem neuronalen Zielnetz eine Ausgabe von höherer Qualität als die Netzausgabe aus dem alternativen neuronalen Netz ist.

[0063] In einigen Implementierungen ist die Präferenzfunktion ein neuronales Präferenzmodellnetz. Das neuronale Präferenzmodellnetz kann dafür trainiert werden, Paare von Ausgaben des neuronalen Netzes zu verarbeiten und für jedes verarbeitete Paar von Ausgaben des neuronalen Netzes eine entsprechende Präferenznetzausgabe zu erzeugen, die eine Likelihood angibt, dass ein Element des verarbeiteten Paares eine Ausgabe von höherer Qualität ist als das andere Element des verarbeiteten Paares. Ein Beispielprozess zum Trainieren des neuronalen Präferenzmodellnetzes wird nachstehend unter Bezugnahme auf **Abb. 4** ausführlicher beschrieben.

[0064] Das System kann bestimmen, ob der Trainingsschritt abgeschlossen ist (Schritt 310). Zum Beispiel kann das System bestimmen, dass der Trainingsschritt abgeschlossen ist, wenn das System alle Netzeingaben in dem Satz von Netzeingaben verarbeitet hat. Wenn der Trainingsschritt nicht abgeschlossen ist, kann das System eine nächste Netzeingabe für den Trainingsschritt verarbeiten.

[0065] Wenn der Trainingsschritt abgeschlossen ist, kann das System die Gewichte des neuronalen Zielnetzes auf der Grundlage der für den Trainingsschritt bestimmten Präferenzpunktzahlen aktualisieren (Schritt 312). Insbesondere kann das System die Gewichte des neuronalen Zielnetzes aktualisieren, indem es eine Zielfunktion für den Trainingsschritt verwendet, die einen Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen die Netzausgaben aus dem neuronalen Zielnetz gegenüber entsprechenden Netzausgaben aus dem alternativen neuronalen Netz zu bevorzugen. Eine Beispiel-Zielfunktion für den Trainingsschritt wird nachstehend unter Bezugnahme auf **Abb. 5** ausführlicher beschrieben.

[0066] **Abb. 4** ist ein Ablaufplan für einen Beispielprozess zum Trainieren eines neuronalen Präferenzmodellnetzes. Der Einfachheit halber wird der Prozess 400 so beschrieben, dass er durch ein System aus einem oder mehreren Computern durchgeführt wird, die sich an einem oder mehreren Standorten befinden. Zum Beispiel kann ein Trainingssystem wie etwa das Trainingssystem 100 von **Abb. 1**, das gemäß der vorliegenden Patentschrift zweckdienlich programmiert ist, den Prozess 400 durchführen.

[0067] Das System empfängt einen Satz von Trainingsdaten (Schritt 402). Die Trainingsdaten schließen beispielhafte Netzeingaben zum Trainieren des neuronalen Zielnetzes, um die bedingte Erzeugungsaufgabe durchzuführen, ein. Die Trainingsdaten können Daten einschließen, die eine Präferenzfunktion für Netzausgaben spezifizieren. Zum Beispiel können die Trainingsdaten für jede beispielhafte Netzeingabe ein Paar von beispielhaften Netzausgaben und eine entsprechende Angabe, dass eine Netzausgabe des Paares eine Ausgabe von höherer Qualität als die andere ist, einschließen. Als spezielles Beispiel können die Trainingsdaten für jedes Paar von beispielhaften Netzausgaben eine Ziel-Likelihood dafür einschließen, dass eine Netzausgabe des Paares eine Ausgabe von höherer Qualität als die andere Ausgabe des Paares ist. Als ein anderes spezielles Beispiel können die Trainingsdaten für jedes Paar von beispielhaften Netzausgaben eine Zielklassifizierung dafür einschließen, welche Netzausgabe des Paares eine Ausgabe von höherer Qualität ist.

[0068] Das System trainiert das neuronale Präferenzmodellnetz, um die durch die Trainingsdaten spezifizierte Präferenzfunktion zu modellieren (Schritt 404). Wie oben beschrieben, kann das System das neuronale Präferenzmodellnetz dafür trainieren, Paare von neuronalen Netzausgaben zu verarbeiten und eine entsprechende Präferenznetzausgabe für jedes verarbeitete Paar von neuronalen Netzausgaben zu erzeugen, die eine Likelihood dafür angibt, dass ein Element des verarbeiteten Paares eine Ausgabe von höherer Qualität ist als das andere Element des verarbeiteten Paares. Zum Beispiel kann das System das neuronale Präferenzmodellnetz dafür trainieren, die Ziel-Likelihoods aus dem Trainingsdatensatz durch Optimieren einer Regressionsverlustfunktion (wie etwa ein L2-Verlust) zwischen den Präferenznetzausgaben und entsprechenden Ziel-Likelihoods zu reproduzieren. Als ein anderes Beispiel kann das System das neuronale Präferenzmodellnetz dafür trainieren, Zielklassifizierungen für Paare von Netzausgaben durch Optimieren einer Klassifizierungsverlustfunktion (wie etwa ein Kreuzentropieverlust) zwischen den Präferenznetzausgaben und entsprechenden Zielklassifizierungen zu reproduzieren.

[0069] Das neuronale Präferenzmodellnetz kann eine beliebige aus einer Vielfalt von Netzarchitekturen aufweisen, die für das Verarbeiten von Paaren von Netzausgaben geeignet sind. Wenn zum Beispiel die Netzausgaben Bilder sind, kann das neuronale Präferenzmodellnetz neuronale Netze einschließen, die für das Verarbeiten von Bilddaten geeignet sind (wie etwa faltende neuronale Netze, Vision Transformer usw.). Als ein anderes Beispiel: Wenn die Netzausgaben Folgen von Werten sind (wie etwa Zeitreihendaten, Textsequenzen, Videosequenzen, Audiosequenzen usw.), kann das neuronale Präferenzmodellnetz neuronale Netze einschließen, die für das Verarbeiten von Folgen geeignet sind (wie etwa rekurrente neuronale Netze, Transformer-Netze usw.). Als spezielles Beispiel kann das neuronale Präferenzmodellnetz ein großes Sprachmodell sein.

[0070] Das System kann das neuronale Präferenzmodellnetz dafür trainieren, eine beliebige aus einer Vielfalt von Präferenzfunktionen für Netzausgaben zu modellieren. Zum Beispiel kann die Präferenzfunktion für jedes Paar von Netzausgaben eine Likelihood dafür angeben, dass eine Netzausgabe des Paares eine höhere Qualität als die andere aufweist, was durch ein Qualitätsmaß bestimmt wird. Als ein weiteres Beispiel kann die Präferenzfunktion eine Likelihood dafür angeben, dass eine Ausgabe des Paares eine geringere Verzerrung, wie etwa Wahrnehmungsverzerrung usw. aufweist. Noch ein anderes weiteres Beispiel: Wenn die Netzausgaben Aktionen für einen Agenten sind, der mit einer Umgebung interagiert, kann die Präferenzfunktion eine Likelihood dafür angeben, dass eine Ausgabe des Paares eine größere Belohnung empfängt.

[0071] Die Präferenzfunktion kann Präferenzen eines Menschen oder einer Gruppe von Menschen darstellen. Zum Beispiel kann die Präferenzfunktion für jedes Paar von Netzausgaben eine menschliche Präferenz für eine der Netzausgaben als die Ausgabe von höherer Qualität des Paares angeben. Als ein anderes Beispiel kann die Präferenzfunktion eine menschliche Überzeugung oder Gewissheit angeben, dass eine der Netzausgaben mit größerer Wahrscheinlichkeit die Ausgabe von höherer Qualität des Paares ist.

[0072] Insbesondere kann die Präferenzfunktion eine nicht-transitive Präferenzfunktion sein. Zum Beispiel, wenn y_1 , y_2 , und y_3 Netzausgaben sind, kann die Präferenzfunktion angeben, dass ein Mensch y_1 gegenüber y_2 bevorzugt, y_2 gegenüber y_3 bevorzugt, aber auch y_3 gegenüber y_1 bevorzugt. Als ein anderes Beispiel: Wenn die Netzausgaben Aktionen darstellen, die durch Agenten durchgeführt werden, die in einer Umgebung interagieren, kann die Präferenzfunktion eine nicht-transitive Likelihood für den Erfolg der Ausgabeaktionen angeben, wenn sie durch konkurrierende Agenten durchgeführt werden.

[0073] Wie oben beschrieben, kann das System dann das neuronale Zielnetz dafür trainieren, die bedingte Erzeugungsaufgabe unter Verwendung des Trainingsdatensatzes und des trainierten neuronalen Präferenzmodellnetzes durchzuführen (Schritt 406).

[0074] **Abb. 5** ist ein Ablaufplan eines beispielhaften Prozesses zum Bestimmen einer Zielfunktion zum Trainieren eines neuronalen Zielnetzes, um eine bedingte Erzeugungsaufgabe durchzuführen. Der Einfachheit halber wird der Prozess 500 so beschrieben, dass er durch ein System aus einem oder mehreren Computern durchgeführt wird, die sich an einem oder mehreren Standorten befinden. Zum Beispiel kann ein Trainingssystem wie etwa das Trainingssystem 100 von **Abb. 1**, das gemäß der vorliegenden Patentschrift zweckdienlich programmiert ist, den Prozess 500 durchführen.

[0075] Wie oben beschrieben, trainiert das System das neuronale Zielnetz über eine Folge von Trainingsschritten. Bei jedem Trainingsschritt empfängt das System einen Satz von Netzeingaben.

[0076] Das System verarbeitet den Satz von Netzeingaben, um einen Satz von Netzausgaben aus dem neuronalen Zielnetz und einen Satz von entsprechenden Netzausgaben aus einem alternativen neuronalen Netz für den Trainingsschritt zu erzeugen (Schritt 502).

[0077] Das alternative neuronale Netz für jeden Zeitschritt kann auf der Grundlage des neuronalen Zielnetzes bestimmt werden. Wie oben beschrieben, kann das alternative neuronale Netz ein beliebiges aus einer Vielfalt von Netzen sein, die alternative (im Vergleich zum neuronalen Zielnetz) Ausgabemuster für die bedingte Erzeugungsaufgabe bereitstellen können.

[0078] Zum Beispiel kann das alternative neuronale Netz ein Netz sein, das dafür trainiert ist, die bedingte Erzeugungsaufgabe mit einer verschiedenen Netzarchitektur im Vergleich zum neuronalen Zielnetz durchzuführen. Als ein anderes Beispiel kann das alternative neuronale Netz die gleiche Netzarchitektur aufweisen wie das neuronale Zielnetz. Als ein weiteres Beispiel kann das alternative neuronale Netz die gleiche Architektur aufweisen wie das neuronale Zielnetz, aber mit verschiedenen Hyperparametern, wie etwa Netzgrößen, Netztiefen, Anzahl von Eingaben, Lernraten usw. (also z.B. mit verschiedenen Hyperparametern trainiert worden sein, Inferenz unter Verwendung verschiedener Hyperparameter durchführen usw.). Als ein weiteres Beispiel kann das alternative neuronale Netz eine Neuparametrisierung des neuronalen Zielnetzes mit verschiedenen Netzgewichten sein, oder es kann eine Kopie des neuronalen Zielnetzes sein, die auf einen bestimmten Satz von Trainingsdaten für das alternative neuronale Netz feinabgestimmt wurde. Als ein weiteres Beispiel kann das alternative neuronale Netz im Vergleich zum neuronalen Zielnetz modifizierte Netzeingaben verarbeiten (wie etwa durch bedingtes Erzeugen von Netzausgaben auf der Grundlage modifizierter Eingaben in das neuronale Netz, die das Erzeugen der Ausgaben in einem verschiedenen Stil als das neuronale Zielnetz spezifizieren, durch Verarbeiten modifizierter Eingabeprompts im Vergleich zum neuronalen Zielnetz usw.).

[0079] Insbesondere kann das alternative neuronale Netz die gleiche Netzarchitektur wie das neuronale Zielnetz aufweisen und kann Gewichte des alternativen neuronalen Netzes aufweisen, die auf der Grundlage der Gewichte des neuronalen Zielnetzes in den aktuellen und vorhergehenden Trainingsschritten bestimmt werden. Als Beispiel können die Gewichte des alternativen neuronalen Netzes die Gewichte des neuronalen Zielnetzes aus einem vorhergehenden Zeitschritt sein (z.B. können Gewichte des alternativen neuronalen Netzes im T-ten Trainingsschritt, $\hat{\theta}_T$, die Gewichte des neuronalen Zielnetzes im T-K-ten Zeitschritt, θ_{T-K} , sein).

[0080] Als ein anderes Beispiel können die Gewichte des alternativen neuronalen Netzes durch einen gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes in den aktuellen und vorhergehenden Trainingsschritten bestimmt werden. Und zwar kann das alternative neuronale Netz eine bedingte Verteilung $p'_{\hat{\theta}}(y|x)$ modellieren, wobei die Gewichte des alternativen neuronalen Netzes im T-ten Trainingsschritt, $\hat{\theta}_T$, durch den gleitenden Durchschnitt folgendermaßen bestimmt werden:

$$\hat{\theta}_T = \frac{1}{K} \sum_{i=T-K+1}^T \theta_i$$

[0081] Dabei ist $K > 0$ eine Fensterlänge und θ_i sind die Gewichte des neuronalen Zielnetzes im i-ten Trainingsschritt.

[0082] Als ein anderes Beispiel können die Gewichte des alternativen neuronalen Netzes durch einen exponentiellen gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes in den aktuellen und vorhergehenden Trainingsschritten bestimmt werden. Und zwar kann das alternative neuronale Netz eine bedingte Verteilung $p'_{\hat{\theta}}(y|x)$ wobei die Gewichte des alternativen neuronalen Netzes im T-ten Trainingsschritt, $\hat{\theta}_T$, durch den gleitenden Durchschnitt folgendermaßen bestimmt werden:

$$\hat{\theta}_T = \sum_{i=0}^T \alpha^{T-i} \theta_i$$

[0083] Dabei ist $\alpha \in [0, 1]$ ein Gewichtungssparameter und θ_i sind die Gewichte des neuronalen Zielnetzes im i-ten Trainingsschritt.

[0084] Das System verarbeitet jede Netzausgabe aus dem neuronalen Zielnetz und entsprechende Netzausgabe aus dem alternativen neuronalen Netz für den Zeitschritt unter Verwendung einer Paarpräferenz-Funktion, um eine Präferenzpunktzahl für das Paar zu bestimmen (Schritt 504).

[0085] In einigen Implementierungen kann das System eine Likelihood-Punktzahl für jede Netzausgabe gemäß dem neuronalen Zielnetz bestimmen (Schritt 506). Die Likelihood-Punktzahl für eine bestimmte Netzausgabe gemäß dem neuronalen Zielnetz gibt eine Likelihood dafür an, dass die Netzausgabe unter der durch das neuronale Zielnetz modellierten bedingten Verteilung bei der gegebenen Netzeingabe der bestimmten Netzausgabe entspricht. Wenn das neuronale Zielnetz zum Beispiel eine auf der Netzeingabe x basierende Netzausgabe y unter Verwendung der bedingten Verteilung p_θ erzeugt, worin θ die Gewichte des neuronalen Zielnetzes sind (z.B. $y \sim p_\theta(\cdot|x)$) kann die Likelihood-Punktzahl für y gemäß dem neuronalen Zielnetz folgendermaßen lauten:

$$p_\theta(y|x)$$

[0086] In einigen Implementierungen können die Netzausgaben Ausgabefolgen sein. Zum Beispiel kann eine bestimmte Netzausgabe y eine Folge von NElementen sein, $y = (y_1, \dots, y_N)$. Das neuronale Zielnetz kann die Ausgabefolgen autoregressiv erzeugen, und die Likelihood-Punktzahl gemäß dem neuronalen Zielnetz für eine Ausgabe $y = (y_1, \dots, y_N)$ kann folgendermaßen lauten:

$$p_\theta(y|x) = \prod_{i=1}^N p_\theta(y_i|x, y_{j<i})$$

[0087] Dabei ist $p_\theta(y_i|x, y_{j<i})$ die Likelihood-Punktzahl des i -ten Elements y_i gemäß dem neuronalen Zielnetz bei der gegebenen Netzeingabe x und den vorherigen Elementen in der Ausgabefolge $y_{j < i}$.

[0088] In einigen Implementierungen kann das System eine Likelihood-Punktzahl für jede Netzausgabe gemäß einem neuronalen Referenznetz für den Trainingsschritt bestimmen (Schritt 508). Das neuronale Referenznetz kann ein beliebiges generatives Modell sein, das sich zur Mustererzeugung von durch die Netzeingaben bedingten Netzausgaben eignet.

[0089] Zum Beispiel kann das neuronale Referenznetz ein Netz sein, das dafür trainiert ist, die bedingte Erzeugungsaufgabe mit einer verschiedenen Netzarchitektur im Vergleich zum neuronalen Zielnetz durchzuführen. Als ein anderes Beispiel kann das neuronale Referenznetz die gleiche Netzarchitektur wie das neuronale Zielnetz aufweisen. Als ein weiteres Beispiel kann das alternative neuronale Netz die gleiche Architektur wie das neuronale Zielnetz aufweisen, unter Verwendung von Unterschieden wie etwa Netzgrößen, Netztiefen, Anzahl von Eingaben, Lernraten usw. (also z.B. mit verschiedenen Hyperparametern trainiert worden sein, Inferenz unter Verwendung verschiedener Hyperparameter durchführen usw.). Als ein weiteres Beispiel kann das neuronale Referenznetz eine Neuparametrisierung des neuronalen Zielnetzes mit verschiedenen Netzgewichten sein, oder es kann eine Kopie des neuronalen Zielnetzes sein, die auf einen bestimmten Satz von Trainingsdaten für das neuronale Referenznetz feinabgestimmt wurde. Als ein weiteres Beispiel kann das neuronale Referenznetz im Vergleich zum neuronalen Zielnetz modifizierte Netzeingaben verarbeiten (wie etwa durch bedingtes Erzeugen von Netzausgaben auf der Grundlage modifizierter Eingaben in das neuronale Netz, die das Erzeugen der Ausgaben in einem verschiedenen Stil als das neuronale Zielnetz spezifizieren, durch Verarbeiten modifizierter Eingabeprompts im Vergleich zum neuronalen Zielnetz usw.).

[0090] Die Likelihood-Punktzahl für eine bestimmte Netzausgabe gemäß dem neuronalen Referenznetz für den Zeitschritt gibt eine Likelihood dafür an, dass die Netzausgabe unter der durch das neuronale Referenznetz modellierten bedingten Verteilung bei der gegebenen Netzeingabe der bestimmten Netzausgabe entspricht. Wenn zum Beispiel q die durch das neuronale Referenznetz modellierte bedingte Verteilung bezeichnet, kann die Likelihood-Punktzahl für eine Netzausgabe y bei einer gegebenen Netzeingabe x gemäß dem neuronalen Referenznetz folgendermaßen lauten:

$$q(y|x)$$

[0091] Das neuronale Referenznetz für jeden Zeitschritt kann auf der Grundlage des neuronalen Zielnetzes bestimmt werden. Insbesondere kann das neuronale Referenznetz die gleiche Netzarchitektur wie das neuronale Zielnetz aufweisen und Gewichte des alternativen neuronalen Netzes aufweisen, die auf der Grundlage der Gewichte des neuronalen Zielnetzes in den aktuellen und vorhergehenden Trainingsschritten bestimmt werden. Zum Beispiel können die Gewichte des neuronalen Referenznetzes die Gewichte des neuronalen Zielnetzes aus einem vorhergehenden Zeitschritt sein (z.B. können Gewichte des neuronalen Referenznetzes im T -ten Trainingsschritt, $\hat{\theta}_T$, die Gewichte des neuronalen Zielnetzes im $T-K$ -ten Zeitschritt, θ_{T-K} , sein).

[0092] Als ein anderes Beispiel können die Gewichte des neuronalen Referenznetzes durch einen gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes in den aktuellen und vorhergehenden Trainingsschritten bestimmt werden. Und zwar kann das neuronale Referenznetz eine bedingte Verteilung $p_{\hat{\theta}}'(y|x)$ modellieren, wobei die Gewichte des neuronalen Referenznetzes im T-ten Trainingsschritt, $\hat{\theta}_T$, durch den gleitenden Durchschnitt folgendermaßen bestimmt werden:

$$\hat{\theta}_T = \frac{1}{K} \sum_{i=T-K+1}^T \theta_i$$

[0093] Dabei ist $K > 0$ eine Fensterlänge und θ_i sind die Gewichte des neuronalen Zielnetzes im i-ten Trainingsschritt.

[0094] Als ein anderes Beispiel können die Gewichte des neuronalen Referenznetzes durch einen exponentiell gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes in den aktuellen und vorhergehenden Trainingsschritten bestimmt werden. Und zwar kann das neuronale Referenznetz eine bedingte Verteilung $q_{\hat{\theta}}(y|x)$ modellieren, wobei die Gewichte des neuronalen Referenznetzes im T-ten Trainingsschritt, $\hat{\theta}_T$, durch den exponentiell gleitenden Durchschnitt folgendermaßen bestimmt werden:

$$\hat{\theta}_T = \sum_{i=0}^T \alpha^{T-i} \theta_i$$

[0095] Dabei ist $\alpha \in [0, 1]$ ein Gewichtungssparameter und θ_i sind die Gewichte des neuronalen Zielnetzes im i-ten Trainingsschritt.

[0096] Das neuronale Referenznetz kann eine Verteilung modellieren, die durch eine feste Referenzverteilung, $\mu(y|x)$, bestimmt ist. Wenn das neuronale Referenznetz durch die feste Referenzverteilung $\mu(y|x)$ bestimmt wird, kann das System das Training des neuronalen Zielnetzes unter Verwendung des neuronalen Referenznetzes reglementieren. Zum Beispiel kann die feste Referenzverteilung ein zuvor trainiertes neuronales Netz sein, das verifiziert worden ist, um sichere oder erwünschte Netzausgaben zu erzeugen.

[0097] Das neuronale Referenznetz kann eine Verteilung modellieren, die auf einer Mischung aus dem neuronalen Zielnetz und der festen Referenzverteilung $\mu(y|x)$ basiert.

[0098] Zum Beispiel kann das neuronale Referenznetz als eine additive Mischung zwischen dem neuronalen Zielnetz und einer festen Referenzverteilung $\mu(y|x)$ bestimmt werden. Zum Beispiel kann mit dem Gewichtungssparameter $A \in [0, 1]$ das neuronale Referenznetz folgendermaßen bestimmt werden:

$$q(y|x) = A p_{\theta}(y|x) + (1-A) \mu(y|x)$$

[0099] Als ein anderes Beispiel kann das neuronale Referenznetz als geometrische Mischung zwischen dem neuronalen Zielnetz und einer festen Referenzverteilung $\mu(y|x)$ bestimmt werden. Zum Beispiel kann das neuronale Referenznetz mit einem Gewichtungssparameter $\beta \in [0, 1]$ folgendermaßen bestimmt werden:

$$q(y|x) = \frac{p_{\theta}(y|x)^{1-\beta} \mu(y|x)^{\beta}}{c(x)}$$

[0100] Dabei ist $c(x)$ eine Normierungskonstante auf der Grundlage der Netzeingabe x . Insbesondere kann $c(x)$ folgendermaßen definiert werden:

$$c(x) = \sum_{y'} p_{\theta}(y'|x)^{1-\beta} \mu(y'|x)^{\beta}$$

[0101] Wenn die Netzausgaben Ausgabefolgen sind, kann das neuronale Referenznetz die Ausgabefolgen autoregressiv verarbeiten, und die Likelihood-Punktzahl gemäß dem neuronalen Zielnetz für eine Ausgabe $y = (y_1, \dots, y_N)$ kann folgendermaßen lauten:

$$q(y|x) = \prod_{i=1}^N q(y_i|x, y_{j<i})$$

[0102] Dabei ist $q(y_i|x, y_{j<i})$ die Likelihood-Punktzahl des i -ten Elements y_i gemäß dem neuronalen Referenznetz bei der gegebenen Netzeingabe x und den vorherigen Elementen in der Ausgabefolge $y_{j < i}$.

[0103] Wenn das neuronale Referenznetz zum Beispiel eine geometrische Verteilung zwischen dem neuronalen Zielnetz und einer festen Referenzverteilung $\mu(y|x)$ mit einem Gewichtungsparemeter $\beta \in [0, 1]$ ist, kann die Likelihood-Punktzahl des i -ten Elements y_i gemäß dem neuronalen Referenznetz folgendermaßen bestimmt werden:

$$q(y|x, y_{j<i}) = \frac{p_\theta(y|x, y_{j<i})^{1-\beta} \mu(y|x, y_{j<i})^\beta}{c(x, y_{j<i})}$$

[0104] Dabei ist $c(x, y_{j<i})$ eine Normierungskonstante auf der Grundlage der Netzeingabe x und der vorherigen Elemente $y_{j < i}$. Insbesondere kann $c(x)$ folgendermaßen definiert werden:

$$c(x, y_{j<i}) = \sum_{y'} p_\theta(y'|x, y_{j<i})^{1-\beta} \mu(y'|x, y_{j<i})^\beta$$

[0105] In einigen Implementierungen ist das neuronale Referenznetz das alternative neuronale Netz.

[0106] Das System aktualisiert dann das neuronale Zielnetz unter Verwendung einer Zielfunktion, um die Aufgabe der bedingten Mustererzeugung durchzuführen (Schritt 510).

[0107] Die Zielfunktion schließt einen ersten Term ein, der es begünstigt, gemäß den Präferenzpunktzahlen Netzausgaben aus dem neuronalen Zielnetz gegenüber den entsprechenden Netzausgaben aus dem alternativen neuronalen Netz zu bevorzugen.

[0108] In einigen Implementierungen kann die Zielfunktion einen zweiten Term einschließen, der das neuronale Zielnetz abstrafft, wenn es Netzausgaben mit Likelihood-Punktzahlen gemäß dem neuronalen Zielnetz erzeugt, die von entsprechenden Likelihood-Punktzahlen gemäß dem neuronalen Referenznetz abweichen.

[0109] Zum Beispiel kann die Zielfunktion mit dem Gewichtungsparemeter $\eta \in [0, 1]$ folgendermaßen lauten:

$$\mathbb{E}_x \left[\mathbb{E}_{y \sim p_\theta(\cdot|x), y' \sim p'(\cdot|x)} \left[\eta \mathcal{P}(y \succ y'|x) \right] - KL(p_\theta(\cdot|x) \parallel q(\cdot|x)) \right]$$

[0110] Dabei sind x Netzeingaben, p_θ ist die durch das neuronale Zielnetz modellierte bedingte Verteilung, p' ist die durch das alternative neuronale Netz modellierte bedingte Verteilung, q ist die durch das neuronale Referenznetz modellierte bedingte Verteilung, $P(y \succ y'|x)$ ist die Präferenzpunktzahl, die für die Zielnetzausgabe y gegenüber der Ausgabe des alternativen Netzes y' bestimmt wird, und KL ist die Kullback-Leibler-Divergenz.

[0111] Das System kann die Zielfunktion als eine Summierung von Verlusttermen für jede Netzeingabe x bestimmen. Zum Beispiel kann das System den Verlustterm für die Ausgabe des neuronalen Zielnetzes y und die Ausgabe des alternativen neuronalen Netzes y' bei gegebener Netzeingabe x folgendermaßen bestimmen:

$$\mathcal{L}(y, y', x) = \eta \mathcal{P}(y \succ y'|x) - \log \left(\frac{p_\theta(y|x)}{q(y|x)} \right)$$

[0112] Der Verlustterm $P(y \succ y'|x)$ bestärkt das neuronale Zielnetz darin, Ausgaben zu erzeugen, die gegenüber denen, die durch das alternative neuronale Netz erzeugt werden, bevorzugt werden. Der Verlustterm

$\log \left(\frac{p_\theta(y|x)}{q(y|x)} \right)$ strafft das neuronale Zielnetz für das Erzeugen von Netzausgaben mit Likelihood-Punktzahlen gemäß dem neuronalen Zielnetz, die von entsprechenden Likelihood-Punktzahlen gemäß dem neuronalen Referenznetz abweichen, ab.

[0113] In einigen Implementierungen kann das System das neuronale Zielnetz durch Berechnen eines Gradienten der Zielfunktion in Bezug auf die Gewichte des neuronalen Zielnetzes θ aktualisieren. Zum Beispiel

kann das System den Gradienten der Zielfunktion als eine Summe von Gradienten von Verlusttermen für jede Netzeingabe folgendermaßen bestimmen:

$$\nabla_{\theta} \mathcal{L}(y, y', x) = \left(\nabla_{\theta} \log(p_{\theta}(y|x)) \right) \left(\eta P(y \succ y'|x) - \log \left(\frac{p_{\theta}(y|x)}{q(y|x)} \right) \right)$$

[0114] Als spezielles Beispiel, wenn die Präferenzpunktzahl, $P(y \succ y'|x)$, ein Wert zwischen 0 und 1 ist, um eine Likelihood dafür anzugeben, dass y eine Ausgabe von höherer Qualität als y' ist, und wenn das neuronale Referenznetz eine geometrische Mischung zwischen dem neuronalen Zielnetz und einer festen Referenzverteilung $\mu(y|x)$ mit dem Gewichtungssparameter β ist, kann das System die Gradienten von Verlusttermen für jede Netzeingabe folgendermaßen bestimmen:

$$\nabla_{\theta} \mathcal{L}(y, y', x) = \left(\nabla_{\theta} \log(p_{\theta}(y|x)) \right) \left(\eta P(y \succ y'|x) - \frac{1}{2} - \beta \log \left(\frac{p_{\theta}(y|x)}{\mu(y|x)} \right) \right)$$

[0115] Das System kann die Gewichte des alternativen neuronalen Netzes und des neuronalen Referenznetzes aktualisieren (Schritt 512). Das System kann auf der Grundlage eines beliebigen einer Vielzahl von Kriterien bestimmen, ob die Gewichte des alternativen neuronalen Netzes und des neuronalen Referenznetzes zu aktualisieren sind. Zum Beispiel kann das System die Gewichte des alternativen neuronalen Netzes und des neuronalen Referenznetzes bei jedem Trainingsschritt aktualisieren.

[0116] Insbesondere kann das System die Gewichte des alternativen neuronalen Netzes und des neuronalen Referenznetzes aktualisieren, indem es die aktualisierten Gewichte des neuronalen Zielnetzes für den Trainingsschritt verwendet. Wenn zum Beispiel das alternative neuronale Netz auf der Grundlage eines Durchschnitts der Gewichte des neuronalen Zielnetzes über mehrere Zeitschritte bestimmt wird, kann das System die Gewichte des alternativen neuronalen Netzes so aktualisieren, dass sie die aktualisierten Gewichte des neuronalen Zielnetzes einschließen. Ein anderes Beispiel: Wenn das neuronale Referenznetz auf der Grundlage eines Durchschnitts der Gewichte des neuronalen Zielnetzes über mehrere Zeitschritte bestimmt wird, kann das System die Gewichte des neuronalen Referenznetzes so aktualisieren, dass sie die aktualisierten Gewichte des neuronalen Zielnetzes einschließen. Als ein anderes Beispiel kann das System, wenn das neuronale Referenznetz auf der Grundlage einer Mischverteilung zwischen dem neuronalen Zielnetz und einer festen Referenzverteilung bestimmt wird, die Mischverteilung für das neuronale Referenznetz so aktualisieren, dass sie das aktualisierte neuronale Zielnetz einschließt.

[0117] Die vorliegende Patentschrift verwendet den Begriff „konfiguriert“ in Verbindung mit Systemen und Computerprogrammkomponenten. Dass ein System aus einem oder mehreren Computern dafür konfiguriert ist, bestimmte Vorgänge oder Aktionen durchzuführen, bedeutet dabei, dass auf dem System Software, Firmware, Hardware oder eine Kombination daraus installiert ist, die das System im Betrieb veranlassen, die Vorgänge oder Aktionen durchzuführen. Dass ein oder mehrere Computerprogramme dafür konfiguriert sind, bestimmte Vorgänge oder Aktionen durchzuführen, bedeutet, dass das eine oder die mehreren Programme Anweisungen einschließen, die, wenn sie durch eine Datenverarbeitungsvorrichtung ausgeführt werden, die Vorrichtung veranlassen, die Vorgänge oder Aktionen durchzuführen.

[0118] Ausführungsformen des Gegenstandes und der funktionalen Vorgänge, die in der vorliegenden Patentschrift beschrieben sind, können in digitalen elektronischen Schaltungen, in greifbar verkörperter Computersoftware oder Firmware, in Computerhardware, welche die in der vorliegenden Patentschrift offengelegten Strukturen und ihrer strukturellen Äquivalente einschließt, oder in Kombinationen aus einem oder mehreren davon implementiert werden. Ausführungsformen des in der vorliegenden Patentschrift beschriebenen Gegenstandes können als ein oder mehrere Computerprogramme implementiert werden, d.h. als ein oder mehrere Module von Computerprogrammanweisungen, die auf einem greifbaren nichtflüchtigen Speichermedium codiert sind, um durch eine Datenverarbeitungsvorrichtung ausgeführt zu werden oder um deren Vorgang zu steuern. Das Computerspeichermedium kann eine maschinenlesbare Speichereinrichtung, ein maschinenlesbares Speichersubstrat, eine Speichereinrichtung mit wahlfreiem oder serielltem Zugriff oder eine Kombination aus einem oder mehreren davon sein. Alternativ oder zusätzlich können die Programmanweisungen auf ein künstlich erzeugtes übertragenes Signal codiert sein, wie etwa ein maschinell erzeugtes elektrisches, optisches oder elektromagnetisches Signal, das erzeugt wird, um Informationen zur Übertragung zu einer geeigneten Empfangsvorrichtung zur Ausführung durch eine Datenverarbeitungsvorrichtung zu codieren.

[0119] Der Begriff „Datenverarbeitungsvorrichtung“ bezieht sich auf Datenverarbeitungshardware und umfasst alle Arten von Vorrichtungen, Einrichtungen und Maschinen zur Verarbeitung von Daten, die zum Beispiel einen programmierbaren Prozessor, einen Computer oder mehrere Prozessoren oder Computer einschließen. Die Vorrichtung kann auch eine spezielle Logikschaltung sein oder diese ferner einschließen, wie etwa ein FPGA (anwenderprogrammierbares Gate-Array) oder ein ASIC (anwendungsspezifischer integrierter Schaltkreis). Die Vorrichtung kann optional zusätzlich zur Hardware Code einschließen, der eine Ausführungsumgebung für Computerprogramme erzeugt, wie etwa einen Code, der eine Prozessor-Firmware, einen Protokollstack, ein Datenbankverwaltungssystem, ein Betriebssystem oder eine Kombination aus einem oder mehreren davon darstellt.

[0120] Ein Computerprogramm, das auch als Programm, Software, Softwareanwendung, App, Modul, Softwaremodul, Skript oder Code bezeichnet oder beschrieben werden kann, kann in jeder beliebigen Form von Programmiersprache geschrieben werden, einschließlich kompilierter oder interpretierter Sprachen oder deklarativer oder prozeduraler Sprachen, und es kann in jeder beliebigen Form eingesetzt werden, einschließlich als eigenständiges Programm oder als Modul, Komponente, Unterprogramm oder andere Einheit, die zur Verwendung in einer Computerumgebung geeignet ist. Ein Programm kann, muss aber nicht, einer Datei in einem Dateisystem entsprechen. Ein Programm kann in einem Abschnitt einer Datei gespeichert werden, die andere Programme oder Daten enthält, wie etwa ein oder mehrere Skripte, die in einem Dokument in einer Auszeichnungssprache gespeichert sind, in einer einzelnen Datei, die für das betreffende Programm vorgesehen ist, oder in mehreren koordinierten Dateien, wie etwa Dateien, die ein oder mehrere Module, Unterprogramme oder Codeabschnitte speichern. Ein Computerprogramm kann eingesetzt werden, um auf einem Computer oder auf mehreren Computern ausgeführt zu werden, die sich an einem Standort befinden oder über mehrere Standorte verteilt und durch ein Datenkommunikationsnetzwerk miteinander verbunden sind.

[0121] In der vorliegenden Patentschrift wird der Begriff „Engine“ im weitesten Sinne verwendet und bezieht sich auf ein softwaregestütztes System, Teilsystem oder Prozess, das so programmiert ist, dass es eine oder mehrere spezifische Funktionen durchführt. Grundsätzlich wird eine Engine als ein oder mehrere Softwaremodule oder -komponenten implementiert, die auf einem oder mehreren Computern an einem oder mehreren Standorten installiert sind. In einigen Fällen sind ein oder mehrere Computer für eine bestimmte Engine vorgesehen; in anderen Fällen können mehrere Engines auf demselben Computer oder denselben Computern installiert sein und laufen.

[0122] Die in der vorliegenden Patentschrift beschriebenen Prozesse und logischen Abläufe können durch einen oder mehrere programmierbare Computer durchgeführt werden, die ein oder mehrere Computerprogramme ausführen, um Funktionen durchzuführen, indem sie auf Eingabedaten operieren und Ausgaben erzeugen. Die Prozesse und logischen Abläufe können auch durch spezielle Logikschaltungen, wie etwa ein FPGA oder einen ASIC, oder durch eine Kombination aus speziellen Logikschaltungen und einem oder mehreren programmierten Computern durchgeführt werden.

[0123] Computer, die für die Ausführung eines Computerprogramms geeignet sind, können auf allgemeinen oder speziellen Mikroprozessoren oder auf beiden oder auf jeder beliebigen anderen Art von Zentralverarbeitungseinheit basieren. Grundsätzlich empfängt eine Zentralverarbeitungseinheit Anweisungen und Daten aus einem Festwertspeicher oder einem Direktzugriffsspeicher oder aus beiden. Die wesentlichen Elemente eines Computers sind eine Zentralverarbeitungseinheit zur Durchführung oder Ausführung von Anweisungen und eine oder mehrere Speichereinrichtungen zum Speichern von Anweisungen und Daten. Die Zentralverarbeitungseinheit und der Speicher können durch eine spezielle Logikschaltung ergänzt werden oder in diese integriert sein. Grundsätzlich schließt ein Computer auch eine oder mehrere Massenspeichereinrichtungen zum Speichern von Daten ein, wie etwa magnetische, magnetooptische oder optische Speicherplatten, und/oder ist betrieblich damit gekoppelt, um Daten von dort zu empfangen oder dorthin zu übertragen, oder beides. Ein Computer muss jedoch keine derartigen Einrichtungen aufweisen. Darüber hinaus kann ein Computer in eine andere Einrichtung eingebettet sein, wie etwa ein Mobiltelefon, ein persönlicher digitaler Assistent (PDA), ein mobiler Audio- oder Videoplayer, eine Spielkonsole, ein Empfänger für GPS (Globales Positionsbestimmungssystem) oder eine tragbare Speichereinrichtung, wie etwa ein Flash-Laufwerk mit USB (Universeller Serieller Bus), um nur einige zu nennen.

[0124] Computerlesbare Medien, die zum Speichern von Computerprogrammanweisungen und -daten geeignet sind, schließen alle Formen von nichtflüchtigen Speichern, Medien und Speichereinrichtungen ein, beispielsweise Halbleiterspeichereinrichtungen, wie etwa EPROM-, EEPROM- und Flash-Speichereinrichtungen.

gen; Magnetspeicherplatten, wie etwa interne Festplatten oder Wechselfestplatten; magnetooptische Speicherplatten sowie CD-ROM- und DVD-ROM-Speicherplatten.

[0125] Um für Interaktion mit einem Benutzer zu sorgen, können Ausführungsformen des in der vorliegenden Patentschrift beschriebenen Gegenstandes auf einem Computer implementiert werden, der eine Anzeigeeinrichtung, wie etwa einen CRT-(Kathodenstrahlröhren-) oder LCD-(Flüssigkristallanzeige-)Monitor, zum Anzeigen von Informationen für den Benutzer sowie eine Tastatur und eine Zeigevorrichtung, wie etwa eine Maus oder einen Trackball, mit denen der Benutzer dem Computer Eingaben bereitstellen kann, aufweist. Ebenso können andere Arten von Einrichtungen verwendet werden, um für Interaktion mit dem Benutzer zu sorgen; zum Beispiel kann die Rückmeldung an den Benutzer jede Form von sensorischer Rückmeldung sein, wie etwa sichtbare Rückmeldung, hörbare Rückmeldung oder taktile Rückmeldung; und Eingaben vom Benutzer können in jeder beliebigen Form empfangen werden, einschließlich akustischer, sprachlicher oder taktiler Eingaben. Darüber hinaus kann ein Computer mit einem Benutzer interagieren, indem er Dokumente an eine durch den Benutzer verwendete Einrichtung sendet und von dort empfängt; zum Beispiel durch das Senden von Webseiten an einen Webbrowser auf einer Einrichtung des Benutzers als Antwort auf vom Webbrowser empfangene Anfragen. Auch kann ein Computer mit einem Benutzer interagieren, indem er Textnachrichten oder andere Formen von Nachrichten an eine persönliche Einrichtung, wie etwa ein Smartphone, auf dem eine Nachrichtenapplication läuft, sendet und im Gegenzug Antwortnachrichten vom Benutzer empfängt.

[0126] Datenverarbeitungsvorrichtungen für das Implementieren von Maschinenlernmodellen können zum Beispiel auch spezielle Hardware-Beschleunigungseinheiten für das Verarbeiten allgemeiner und rechenintensiver Teile von Maschinenlerntrainings oder Produktion, d.h. Inferenz, Arbeitslast, einschließen.

[0127] Maschinenlernmodelle können unter Verwendung eines Frameworks für maschinelles Lernen, wie etwa eines TensorFlow-Frameworks oder eines Jax-Frameworks, implementiert und eingesetzt werden.

[0128] Ausführungsformen des in der vorliegenden Patentschrift beschriebenen Gegenstandes können in einem Computersystem implementiert werden, das eine Backend-Komponente, wie etwa als Datenserver, einschließt, oder das eine Middleware-Komponente, wie etwa einen Anwendungsserver einschließt, oder das eine Frontend-Komponente, wie etwa einen Client-Computer mit einer grafischen Benutzeroberfläche, einem Webbrowser oder einer App, über die ein Benutzer mit einer Implementierung des in der vorliegenden Patentschrift beschriebenen Gegenstandes interagieren kann, oder jegliche Kombination aus einer oder mehreren solcher Backend-, Middleware- oder Frontend-Komponenten einschließt. Die Komponenten des Systems können durch jegliche Form oder jegliches Medium von digitaler Datenkommunikation miteinander verbunden werden, wie etwa durch ein Kommunikationsnetzwerk. Beispiele für Kommunikationsnetzwerke schließen ein lokales Netz (LAN) und ein Weitverkehrsnetz (WAN), wie etwa das Internet, ein.

[0129] Das Computersystem kann Clients und Server einschließen. Ein Client und ein Server sind grundsätzlich voneinander abgesetzt und interagieren in der Regel über ein Kommunikationsnetzwerk. Die Beziehung zwischen Client und Server ergibt sich daraus, dass Computerprogramme auf den jeweiligen Computern laufen und eine Client-Server-Beziehung zueinander aufweisen. In einigen Ausführungsformen überträgt ein Server Daten, wie etwa eine HTML-Seite, zu einer Benutzereinrichtung, etwa um Daten für einen Benutzer anzuzeigen, der mit der Vorrichtung interagiert, die als Client fungiert, und Benutzereingaben von ihm zu empfangen. In der Benutzereinrichtung erzeugte Daten, wie etwa ein Ergebnis der Benutzerinteraktion, können am Server von der Einrichtung empfangen werden.

[0130] Wenngleich die vorliegende Patentschrift viele spezifische Einzelheiten der Implementierung enthält, sollten diese nicht als Beschränkungen des Schutzbereichs einer beliebigen Erfindung oder des Umfangs der möglichen Ansprüche verstanden werden, sondern als Beschreibungen von Merkmalen, die für bestimmte Ausführungsformen bestimmter Erfindungen spezifisch sein können. Bestimmte Merkmale, die in der vorliegenden Patentschrift im Kontext separater Ausführungsformen beschrieben sind, können auch in Kombination in einer einzigen Ausführungsform implementiert werden. Umgekehrt können verschiedene Merkmale, die im Kontext einer einzelnen Ausführungsform beschrieben werden, auch in mehreren Ausführungsformen separat oder in jeder beliebigen geeigneten Unterkombination implementiert werden. Darüber hinaus können, obwohl Merkmale oben als in bestimmten Kombinationen wirkend beschrieben werden und sogar ursprünglich als solche beansprucht werden können, ein oder mehrere Merkmale aus einer beanspruchten Kombination in einigen Fällen aus der Kombination herausgenommen werden, und die beanspruchte Kombination kann auf eine Unterkombination oder eine Variante einer Unterkombination gerichtet sein.

[0131] Und auch wenn Vorgänge in einer bestimmten Reihenfolge in den Zeichnungen durchgeführt und in den Ansprüchen genannt sind, ist dies nicht so zu verstehen, dass diese Vorgänge in der konkreten gezeigten Reihenfolge oder in sequenzieller Reihenfolge durchgeführt werden müssen oder dass alle dargestellten Vorgänge durchgeführt werden müssen, um erwünschte Ergebnisse zu erzielen. Unter gewissen Umständen können Multitasking und Parallelverarbeitung vorteilhaft sein. Darüber hinaus ist die Trennung verschiedener Systemmodule und -komponenten in den oben beschriebenen Ausführungsformen nicht so zu verstehen, dass eine solche Trennung in allen Ausführungsformen erforderlich ist, und es versteht sich, dass die beschriebenen Programmkomponenten und Systeme grundsätzlich zusammen in ein einziges Softwareprodukt integriert oder in mehrere Softwareprodukte gepackt werden können.

[0132] Bestimmte Ausführungsformen des Gegenstandes sind beschrieben worden. Andere Ausführungsformen fallen in den Schutzbereich der folgenden Ansprüche. Zum Beispiel können die in den Ansprüchen genannten Aktionen in einer verschiedenen Reihenfolge durchgeführt werden und dennoch zu erwünschten Ergebnissen führen. Zum Beispiel erfordern die in den begleitenden Figuren dargestellten Prozesse nicht unbedingt die konkrete gezeigte Reihenfolge oder sequenzielle Reihenfolge, um erwünschte Ergebnisse zu erzielen. In einigen Fällen können Multitasking und Parallelverarbeitung vorteilhaft sein.

Schutzansprüche

1. System, Folgendes umfassend:
einen oder mehrere Computer; und
eine oder mehrere Speichereinrichtungen, die kommunikativ mit dem einen oder den mehreren Computern gekoppelt sind, wobei die eine oder die mehreren Speichereinrichtungen Anweisungen speichern, die, wenn sie durch den einen oder die mehreren Computer ausgeführt werden, den einen oder die mehreren Computer veranlassen, Vorgänge zum Trainieren eines neuronalen Zielnetzes durchzuführen, das eine Vielzahl von Gewichten des neuronalen Zielnetzes aufweist und dafür konfiguriert ist, eine Netzeingabe gemäß den Gewichten des neuronalen Zielnetzes zu verarbeiten, um eine Netzausgabe zu erzeugen, wobei die Vorgänge, bei jedem von einer Vielzahl von Trainingsschritten, Folgendes umfassen:
Empfangen von einer oder mehreren Netzeingaben;
für jede der einen oder der mehreren Netzeingaben:
Verarbeiten der Netzeingabe unter Verwendung des neuronalen Zielnetzes, um eine erste Netzausgabe zu erzeugen;
Verarbeiten der Netzeingabe unter Verwendung eines alternativen neuronalen Netzes für den Trainingsschritt, um eine zweite Netzausgabe zu erzeugen; und
Anwenden einer Präferenzfunktion auf die erste Netzausgabe und die zweite Netzausgabe, um eine Präferenzpunktzahl zu erzeugen, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist; und
Aktualisieren der Gewichte des neuronalen Zielnetzes unter Verwendung einer Zielfunktion, die einen ersten Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen die ersten Netzausgaben gegenüber den entsprechenden zweiten Netzausgaben zu bevorzugen.
2. System nach Anspruch 1, wobei die Vorgänge ferner bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben, Folgendes umfassen:
Bestimmen einer ersten Likelihood-Punktzahl, die der ersten Netzausgabe durch das neuronale Zielnetz bei der gegebenen Netzeingabe zugewiesen wird; und
Bestimmen einer zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, wobei die Zielfunktion einen zweiten Term einschließt, der das neuronale Zielnetz für das Erzeugen erster Likelihood-Punktzahlen abstrafft, die von entsprechenden zweiten Likelihood-Punktzahlen abweichen.
3. System nach Anspruch 2, wobei bei jedem der Vielzahl von Trainingsschritten das neuronale Referenznetz für den Trainingsschritt das alternative neuronale Netz für den Trainingsschritt ist.
4. System nach Anspruch 1, wobei das Anwenden der Präferenzfunktion auf die erste Netzausgabe und die zweite Netzausgabe, um die Präferenzpunktzahl zu erzeugen, welche die Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist, Folgendes umfasst:
Verarbeiten der ersten Netzausgabe und der zweiten Netzausgabe unter Verwendung eines neuronalen Präferenzmodellnetzes, um eine Präferenznetzausgabe zu erzeugen, wobei das neuronale Präferenzmodellnetz dafür trainiert wurde, Paare von neuronalen Netzausgaben zu verarbeiten und eine entsprechende Präferenznetzausgabe für jedes verarbeitete Paar von neuronalen Netzausgaben zu erzeugen, die eine

Likelihood angibt, dass ein Element des verarbeiteten Paares eine Ausgabe von höherer Qualität als das andere Element des verarbeiteten Paares ist; und
 auf der Präferenznetzausgabe basierendes Erzeugen der Präferenzpunktzahl, welche die Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist.

5. System nach Anspruch 2, wobei bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben das Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, Folgendes umfasst:

Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe zugewiesen wird, auf der Grundlage einer geometrischen Mischung der ersten Likelihood-Punktzahl für die erste Netzausgabe und einer Likelihood-Punktzahl für die erste Netzausgabe unter einer festen Referenzverteilung und auf der Grundlage einer Normierungskonstante für den Trainingsschritt.

6. System nach Anspruch 1, wobei bei jedem der Vielzahl von Trainingsschritten:
 das alternative neuronale Netz für den Trainingsschritt eine Vielzahl von Gewichten des alternativen neuronalen Netzes für den Trainingsschritt aufweist und die gleiche Architektur wie das neuronale Zielnetz aufweist; und
 die Gewichte des alternativen neuronalen Netzes für den Trainingsschritt durch einen exponentiellen gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes über die aktuellen und vorhergehenden Trainingsschritte bestimmt werden.

7. System nach Anspruch 1, wobei die Zielfunktion für jede Netzeingabe und entsprechende erste und zweite Netzausgaben auf Folgendem basiert:

einer durch die Präferenzfunktion bestimmten Präferenzpunktzahl, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist;

einer ersten Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird; und

einer zweiten Likelihood-Punktzahl für die erste Netzausgabe, die durch eine feste Referenzverteilung bestimmt wird.

8. System nach Anspruch 1, wobei das Aktualisieren der aktuellen Gewichte des neuronalen Zielnetzes unter Verwendung der Zielfunktion, die einen ersten Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen erste Netzeingaben gegenüber entsprechenden zweiten Netzeingaben zu bevorzugen, Folgendes umfasst:

Bestimmen eines Gradienten der Zielfunktion in Bezug auf die Gewichte des neuronalen Zielnetzes; und
 Aktualisieren der Gewichte des neuronalen Zielnetzes auf der Grundlage des Gradienten der Zielfunktion.

9. System nach Anspruch 8, wobei das Bestimmen des Gradienten der Zielfunktion in Bezug auf die aktuellen Gewichte des neuronalen Zielnetzes das Berechnen von Folgendem für jede Netzeingabe und entsprechende ersten und zweiten Netzausgaben umfasst:

einem Gradient einer ersten Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird, in Bezug auf die aktuellen Gewichte des neuronalen Zielnetzes; und

eine durch die Präferenzfunktion bestimmte Präferenzpunktzahl, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist;

die erste Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird; und

eine Likelihood-Punktzahl für die erste Netzausgabe, die durch eine feste Referenzverteilung bestimmt wird.

10. System nach Anspruch 1, wobei das neuronale Zielnetz ein großes Sprachmodell ist.

11. System nach Anspruch 1, wobei das neuronale Präferenznetz ein großes Sprachmodell ist.

12. System nach Anspruch 2, wobei:
 die erste Netzausgabe eine erste Netzausgabefolge umfasst;
 die zweite Netzausgabe eine zweite Netzausgabefolge umfasst; und
 das Verarbeiten der Netzeingabe unter Verwendung des alternativen neuronalen Netzes, um die zweite Netzausgabe zu erzeugen, das autoregressive Erzeugen der zweiten Netzausgabefolge unter Verwendung des alternativen neuronalen Netzes umfasst.

13. System nach Anspruch 12, wobei bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben das Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, Folgendes umfasst:

für jedes Ausgabeelement der ersten Netzausgabefolge:

Bestimmen einer ersten Likelihood-Punktzahl für das Ausgabeelement bei der gegebenen Netzeingabe unter Verwendung des neuronalen Zielnetzes; und

Bestimmen einer zweiten Likelihood-Punktzahl für das Ausgabeelement auf der Grundlage einer geometrischen Mischung aus der ersten Likelihood-Punktzahl für das Ausgabeelement und einer Likelihood-Punktzahl für das Ausgabeelement unter einer festen Referenzverteilung und auf der Grundlage einer Normierungskonstante für den Trainingsschritt; und

Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe zugewiesen wird, auf der Grundlage der zweiten Likelihood-Punktzahlen der Ausgabeelemente in der ersten Netzausgabefolge.

14. System nach Anspruch 1, wobei die Präferenzfunktion eine nicht-transitive Präferenzbeziehung modellieren kann.

15. Ein oder mehrere nichtflüchtige Computerspeichermedien, die Anweisungen speichern, die, wenn sie durch einen oder mehrere Computer ausgeführt werden, den einen oder die mehreren Computer veranlassen, Vorgänge zum Trainieren eines neuronalen Zielnetzes durchzuführen, das eine Vielzahl von Gewichten des neuronalen Zielnetzes aufweist und dafür konfiguriert ist, eine Netzeingabe gemäß den Gewichten des neuronalen Zielnetzes zu verarbeiten, um eine Netzausgabe zu erzeugen, wobei die Vorgänge, bei jedem von einer Vielzahl von Trainingsschritten, Folgendes umfassen:

Empfangen von einer oder mehreren Netzeingaben;

für jede der einen oder der mehreren Netzeingaben:

Verarbeiten der Netzeingabe unter Verwendung des neuronalen Zielnetzes, um eine erste Netzausgabe zu erzeugen;

Verarbeiten der Netzeingabe unter Verwendung eines alternativen neuronalen Netzes für den Trainingsschritt, um eine zweite Netzausgabe zu erzeugen; und

Anwenden einer Präferenzfunktion auf die erste Netzausgabe und die zweite Netzausgabe, um eine Präferenzpunktzahl zu erzeugen, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist; und

Aktualisieren der Gewichte des neuronalen Zielnetzes unter Verwendung einer Zielfunktion, die einen ersten Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen die ersten Netzausgaben gegenüber den entsprechenden zweiten Netzausgaben zu bevorzugen.

16. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei die Vorgänge ferner, bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben, Folgendes umfassen:

Bestimmen einer ersten Likelihood-Punktzahl, die der ersten Netzausgabe durch das neuronale Zielnetz bei der gegebenen Netzeingabe zugewiesen wird; und

Bestimmen einer zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, wobei

die Zielfunktion einen zweiten Term einschließt, der das neuronale Zielnetz für das Erzeugen erster Likelihood-Punktzahlen abstrafte, die von entsprechenden zweiten Likelihood-Punktzahlen abweichen.

17. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 16, wobei bei jedem der Vielzahl von Trainingsschritten das neuronale Referenznetz für den Trainingsschritt das alternative neuronale Netz für den Trainingsschritt ist.

18. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei das Anwenden der Präferenzfunktion auf die erste Netzausgabe und die zweite Netzausgabe, um die Präferenzpunktzahl zu erzeugen, welche die Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist, Folgendes umfasst:

Verarbeiten der ersten Netzausgabe und der zweiten Netzausgabe unter Verwendung eines neuronalen Präferenzmodellnetzes, um eine Präferenznetzausgabe zu erzeugen, wobei das neuronale Präferenzmodellnetz dafür trainiert wurde, Paare von neuronalen Netzausgaben zu verarbeiten und eine entsprechende Präferenznetzausgabe für jedes verarbeitete Paar von neuronalen Netzausgaben zu erzeugen, die eine Likelihood angibt, dass ein Element des verarbeiteten Pairs eine Ausgabe von höherer Qualität als das andere Element des verarbeiteten Pairs ist; und

auf der Präferenznetzausgabe basierendes Erzeugen der Präferenzpunktzahl, welche die Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist.

19. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 16, wobei bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben das Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, Folgendes umfasst:

Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe zugewiesen wird, auf der Grundlage einer geometrischen Mischung der ersten Likelihood-Punktzahl für die erste Netzausgabe und einer Likelihood-Punktzahl für die erste Netzausgabe unter einer festen Referenzverteilung und auf der Grundlage einer Normierungskonstante für den Trainingsschritt.

20. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei bei jedem der Vielzahl von Trainingsschritten:

das alternative neuronale Netz für den Trainingsschritt eine Vielzahl von Gewichten des alternativen neuronalen Netzes für den Trainingsschritt aufweist und die gleiche Architektur wie das neuronale Zielnetz aufweist; und

die Gewichte des alternativen neuronalen Netzes für den Trainingsschritt durch einen exponentiellen gleitenden Durchschnitt der Gewichte des neuronalen Zielnetzes über die aktuellen und vorhergehenden Trainingsschritte bestimmt werden.

21. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei die Zielfunktion für jede Netzeingabe und entsprechende erste und zweite Netzausgaben auf Folgendem basiert:

einer durch die Präferenzfunktion bestimmten Likelihood-Punktzahl, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist;

einer ersten Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird; und

einer zweiten Likelihood-Punktzahl für die erste Netzausgabe, die durch eine feste Referenzverteilung bestimmt wird.

22. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei das Aktualisieren der aktuellen Gewichte des neuronalen Zielnetzes unter Verwendung der Zielfunktion, die einen ersten Term einschließt, der es begünstigt, gemäß den Präferenzpunktzahlen erste Netzausgaben gegenüber entsprechenden zweiten Netzausgaben zu bevorzugen, Folgendes umfasst:

Bestimmen eines Gradienten der Zielfunktion in Bezug auf die Gewichte des neuronalen Zielnetzes; und Aktualisieren der Gewichte des neuronalen Zielnetzes auf der Grundlage des Gradienten der Zielfunktion.

23. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 22, wobei das Bestimmen des Gradienten der Zielfunktion in Bezug auf die aktuellen Gewichte des neuronalen Zielnetzes das Berechnen, für jede Netzeingabe und entsprechende erste und zweiten Netzausgaben, von Folgendem umfasst:

einem Gradienten einer ersten Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird, in Bezug auf die aktuellen Gewichte des neuronalen Zielnetzes; und

einer durch die Präferenzfunktion bestimmte Präferenzpunktzahl, die eine Likelihood angibt, dass die erste Netzausgabe eine Ausgabe von höherer Qualität als die zweite Netzausgabe ist;

der ersten Likelihood-Punktzahl für die erste Netzausgabe, die durch das neuronale Zielnetz bestimmt wird; und

einer Likelihood-Punktzahl für die erste Netzausgabe, die durch eine feste Referenzverteilung bestimmt wird.

24. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei das neuronale Zielnetz ein großes Sprachmodell ist.

25. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei das neuronale Präferenznetz ein großes Sprachmodell ist.

26. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 16, wobei:

die erste Netzausgabe eine erste Netzausgabefolge umfasst;

die zweite Netzausgabe eine zweite Netzausgabefolge umfasst; und

das Verarbeiten der Netzeingabe unter Verwendung des alternativen neuronalen Netzes, um die zweite

Netzausgabe zu erzeugen, das autoregressive Erzeugen der zweiten Netzausgabefolge unter Verwendung des alternativen neuronalen Netzes umfasst.

27. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 26, wobei bei jedem der Vielzahl von Trainingsschritten und für jede der einen oder der mehreren Netzeingaben das Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe durch ein neuronales Referenznetz für den Trainingsschritt bei der gegebenen Netzeingabe zugewiesen wird, Folgendes umfasst:

für jedes Ausgabeelement der ersten Netzausgabefolge:

Bestimmen einer ersten Likelihood-Punktzahl für das Ausgabeelement bei der gegebenen Netzeingabe unter Verwendung des neuronalen Zielnetzes; und

Bestimmen einer zweiten Likelihood-Punktzahl für das Ausgabeelement auf der Grundlage einer geometrischen Mischung aus der ersten Likelihood-Punktzahl für das Ausgabeelement und einer Likelihood-Punktzahl für das Ausgabeelement unter einer festen Referenzverteilung und auf der Grundlage einer Normierungskonstante für den Trainingsschritt; und

Bestimmen der zweiten Likelihood-Punktzahl, die der ersten Netzausgabe zugewiesen wird, auf der Grundlage der zweiten Likelihood-Punktzahlen der Ausgabeelemente in der ersten Netzausgabefolge.

28. Ein oder mehrere nichtflüchtige Computerspeichermedien nach Anspruch 15, wobei die Präferenzfunktion eine nichttransitive Präferenzbeziehung modellieren kann.

Es folgen 5 Seiten Zeichnungen

TRAININGSSYSTEM 100

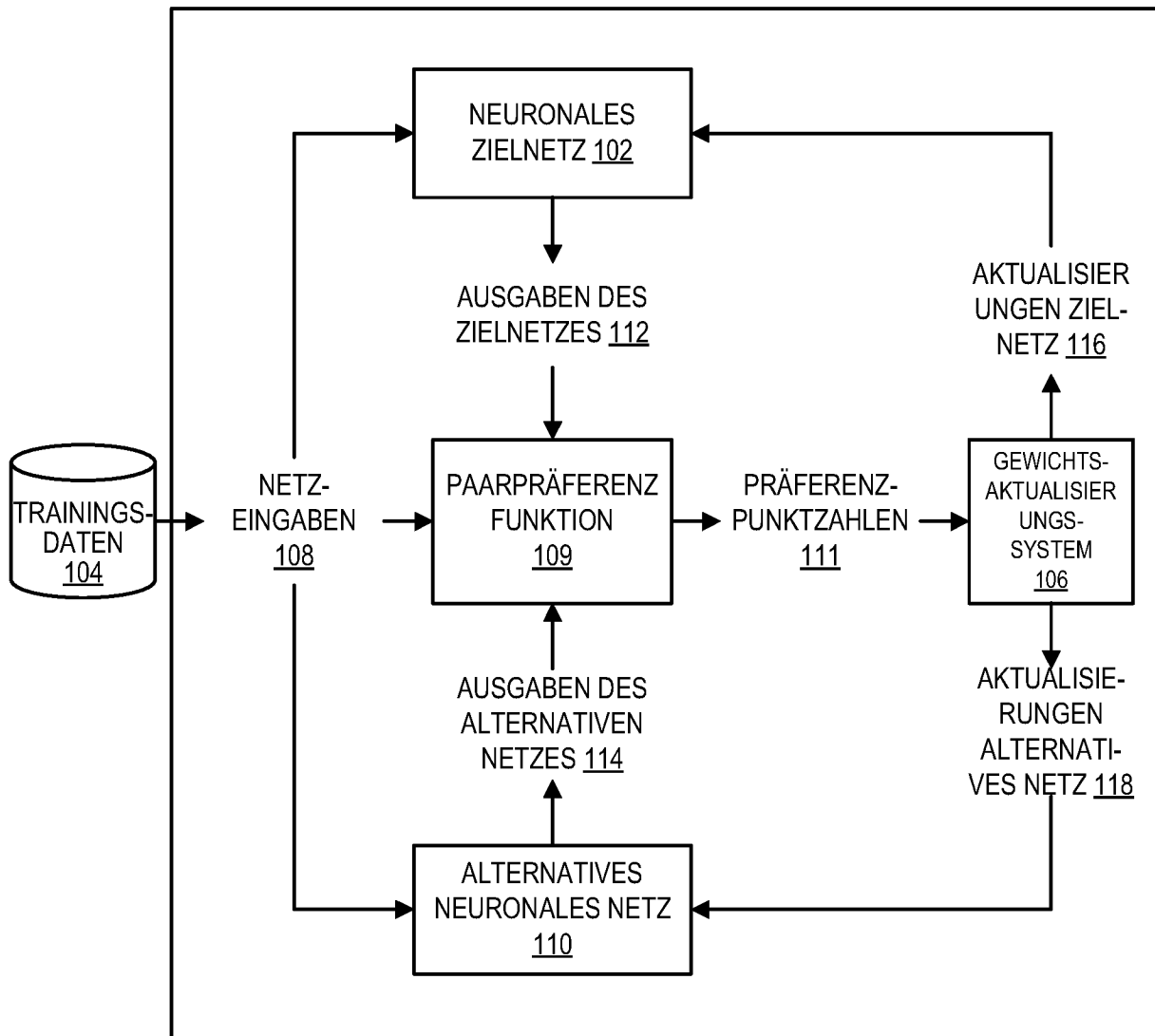


ABB. 1

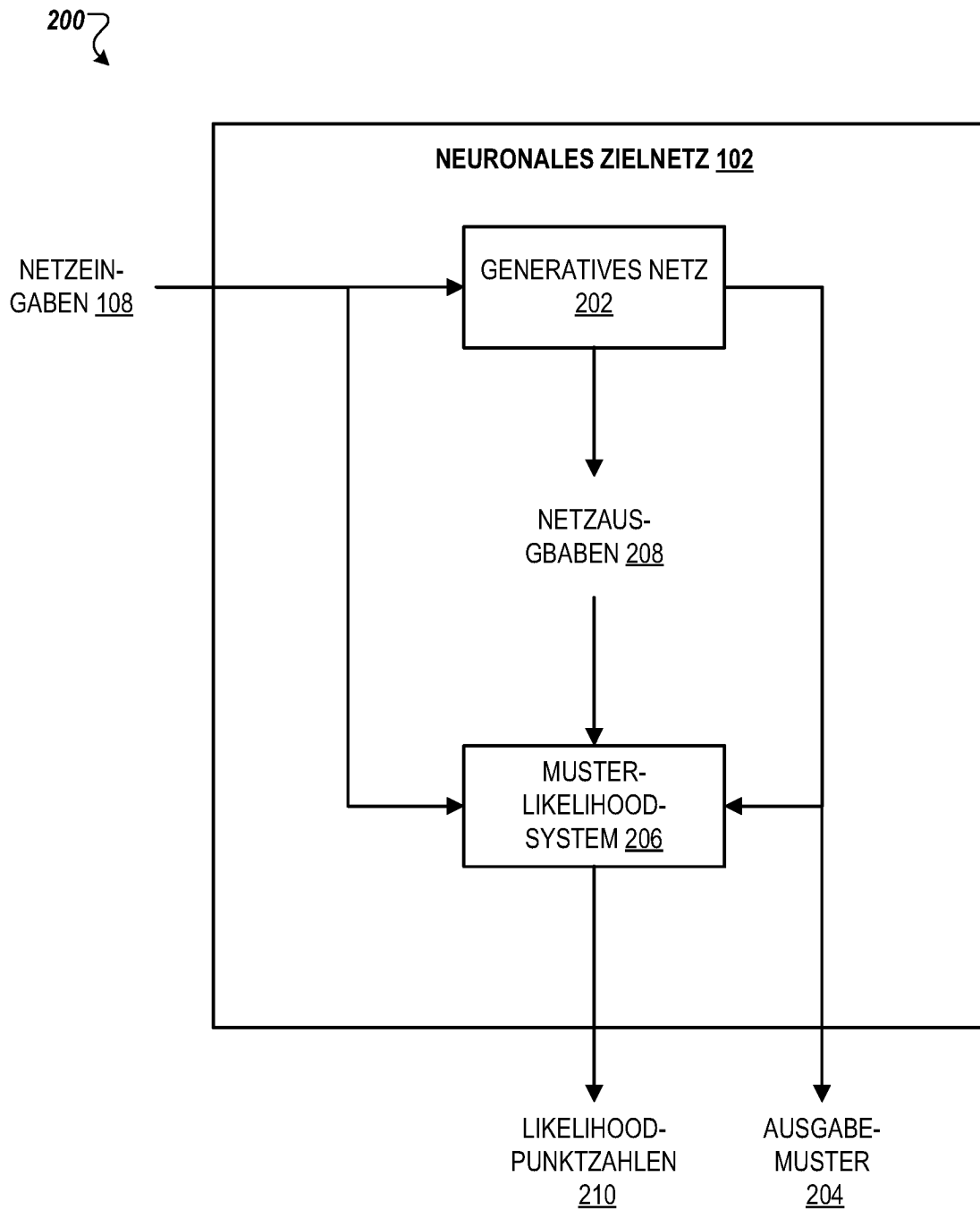


ABB. 2

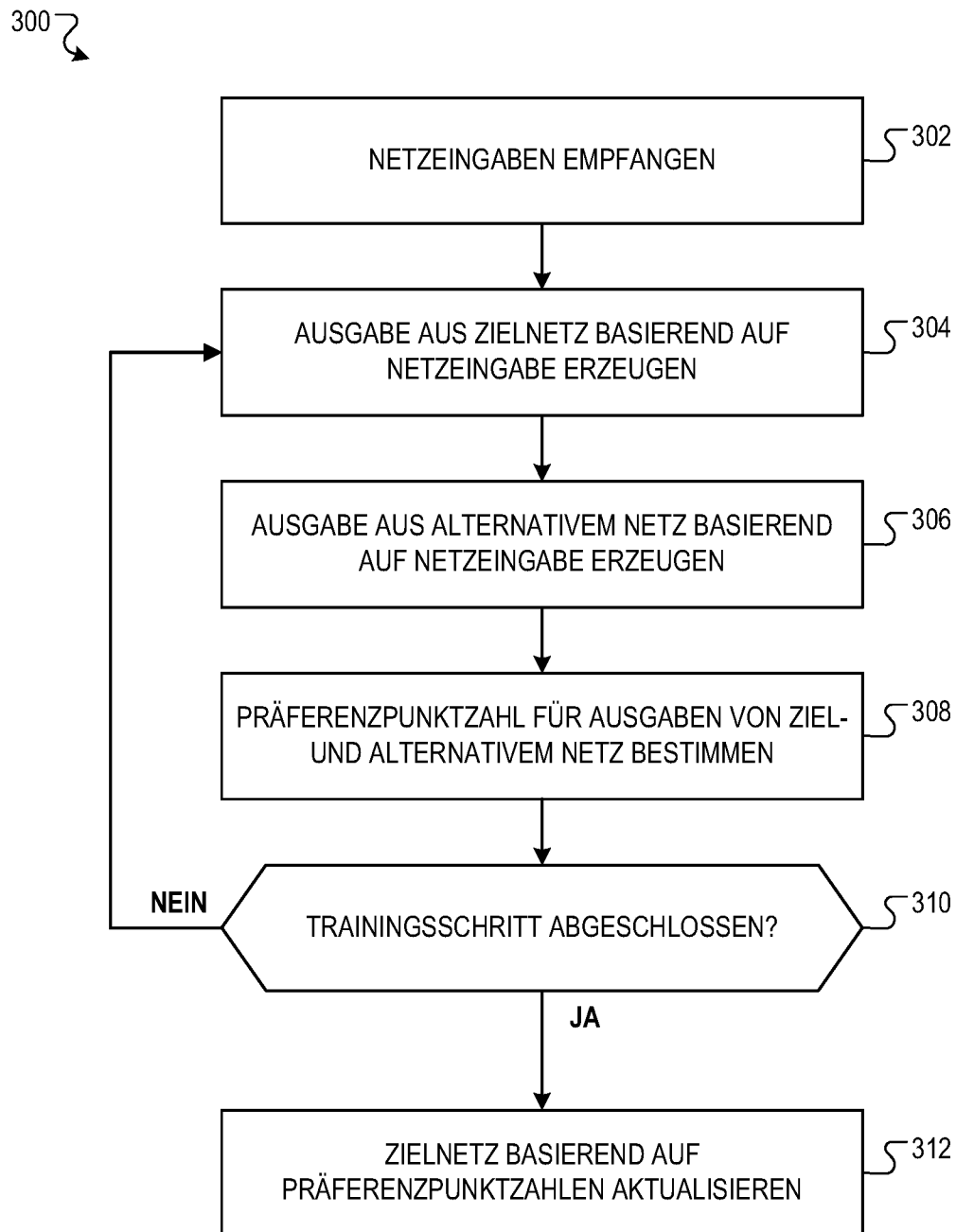


ABB. 3

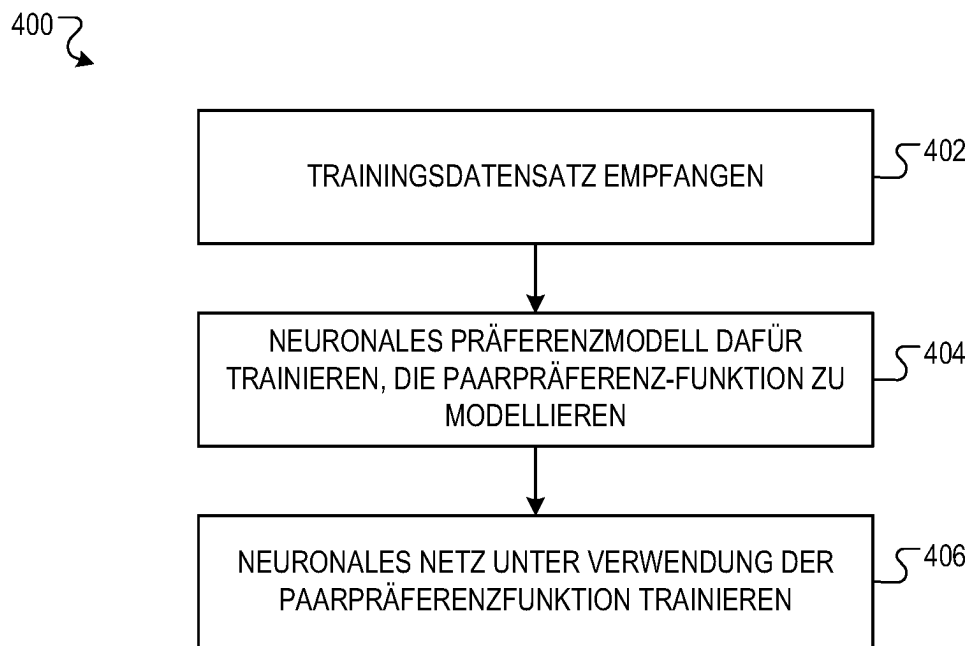


ABB. 4

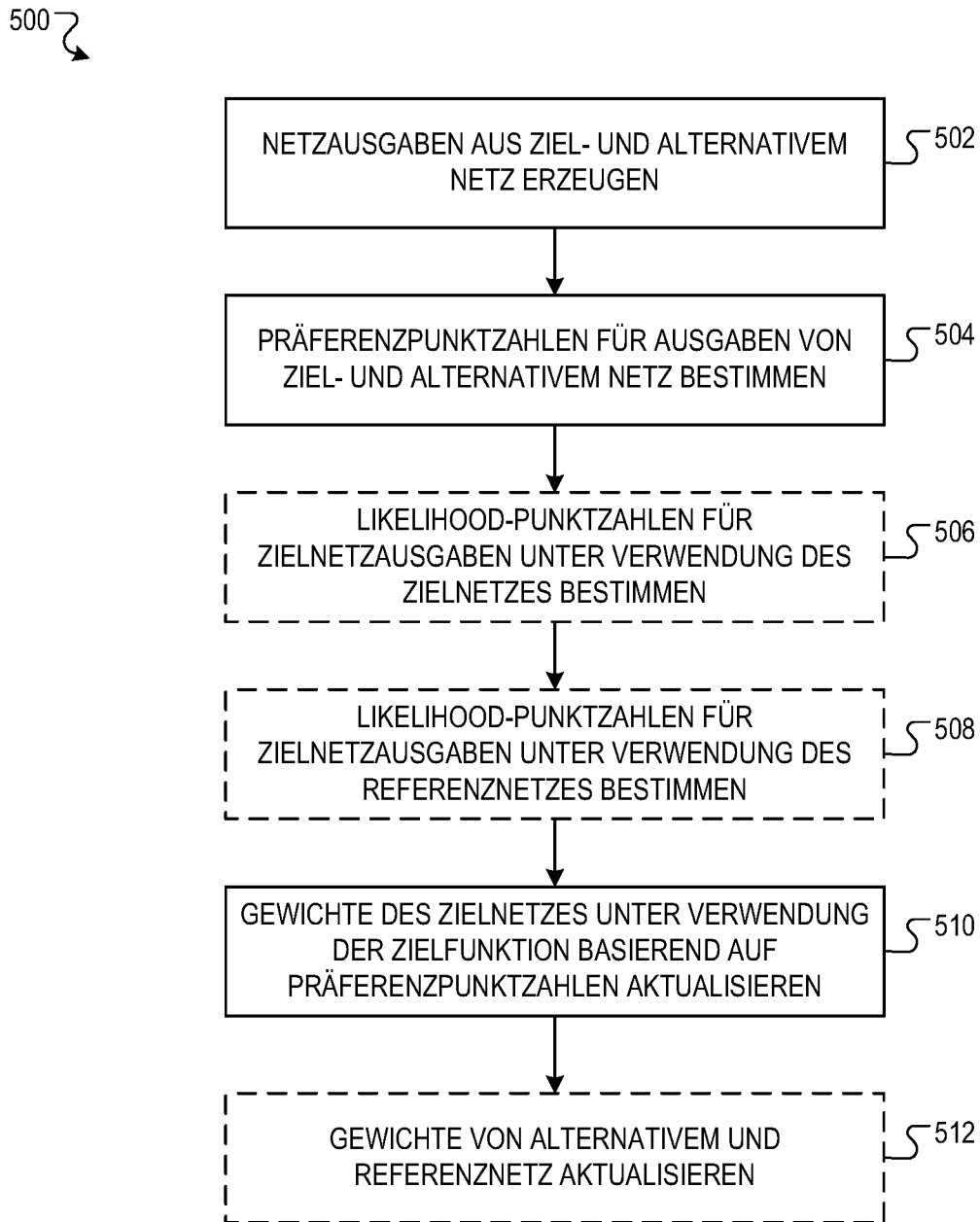


ABB. 5