



(10) **DE 10 2023 203 277 A1** 2024.10.17

(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2023 203 277.5**

(22) Anmeldetag: **11.04.2023**

(43) Offenlegungstag: **17.10.2024**

(51) Int Cl.: **G06N 20/00 (2019.01)**

(71) Anmelder:
**Robert Bosch Gesellschaft mit beschränkter
Haftung, 70469 Stuttgart, DE**

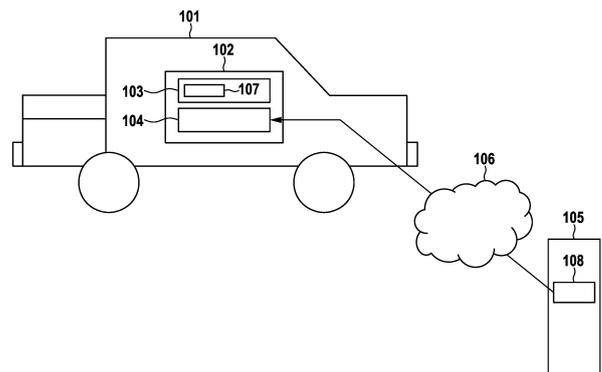
(72) Erfinder:
**Goth, Johannes, 72218 Wildberg, DE; Coors,
Benjamin, 70178 Stuttgart, DE; Cimurs, Reinis,
70176 Stuttgart, DE**

Rechercheantrag gemäß § 43 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Verfahren zum Trainieren eines maschinellen Lernmodells**

(57) Zusammenfassung: Gemäß verschiedenen Ausführungsformen wird ein Verfahren zum Trainieren eines maschinellen Lernmodells (108) bereitgestellt, aufweisend Einteilen, für Trainingsdaten, die eine Vielzahl von Trainingsdatenelementen enthalten, die jeweils ein Paar aus Trainingseingabe und Zielverarbeitungsergebnis für die Trainingseingabe aus einem Ergebnisraum angeben, des Ergebnisraums in Bereiche (403, 404, 405), Ermitteln einer Akzeptanzrate abhängig von der Anzahl der Bereiche (403, 404, 405), für die jeweils gilt, dass die Trainingsdaten (402) mindestens ein Trainingsdatenelement aufweisen, das ein Zielverarbeitungsergebnis angibt, das in dem jeweiligen Bereich (403, 404, 405) liegt; und, für jede Trainingsiteration Trainingsdatenelementen (406) aus den Trainingsdaten (402) abhängig von der Akzeptanzrate ausgewählt werden und das maschinelle Lernmodell (108) mittels der für die Trainings-Iteration ausgewählten Trainingsdatenelemente trainiert wird.



Beschreibung

[0001] Die vorliegende Offenbarung bezieht sich auf Verfahren zum Trainieren eines maschinellen Lernmodells.

[0002] Maschinelle Lernmodelle werden unter Verwendung von Trainingsdaten trainiert. Das maschinelle Lernmodell passt sich dabei an die Trainingsdaten an, die es beim Training „sieht“, d.h. an die Trainingsdatenelemente, die ihm im Training zugeführt werden und anhand derer es bewertet und angepasst wird. Sieht das maschinelle Lernmodell im Training Trainingsdaten für einen bestimmten Bereich von Szenarien wenig (oder auch im Verhältnis zu anderen Trainingsdaten selten), ist es für diesen Bereich von (z.B. Steuerungs-)Szenarien im Vergleich zu anderen Bereichen verhältnismäßig schlecht trainiert und liefert möglicherweise schlechte Ergebnisse für diese Bereiche, d.h. generalisiert schlecht hinsichtlich dieser Bereiche. Es sind deshalb Herangehensweisen wünschenswert, die ein Training eines maschinellen Lernmodells ermöglichen, sodass es eine hohe Generalisierungsfähigkeit hat und damit in einem breiten Spektrum von Szenarien, z.B. Verkehrssituationen, gute Ergebnisse liefert.

[0003] Gemäß verschiedenen Ausführungsformen wird ein Verfahren zum Trainieren eines maschinellen Lernmodells, insbesondere zum Implementieren einer Steuerungsstrategie für eine Robotervorrichtung (d.h. sodass das maschinelle Lernmodell eine Steuerungsstrategie für eine Robotervorrichtung wie z.B. ein autonomes Fahrzeug repräsentiert), bereitgestellt, aufweisend Einteilen, für Trainingsdaten, die eine Vielzahl von Trainingsdatenelementen enthalten, die jeweils ein Paar aus Trainingseingabe, insbesondere Trainings-Sensordaten, die einen Zustand einer Robotervorrichtung und/oder einer Umgebung der Robotervorrichtung beschreiben, und Zielverarbeitungsergebnis für die Trainingseingabe aus einem Ergebnisraum, insbesondere einen Robotervorrichtungs-Steuerungsaktionsraum, angeben, des Ergebnisraums in Bereiche, Ermitteln einer Akzeptanzrate abhängig von der Anzahl der Bereiche, für die jeweils gilt, dass die Trainingsdaten mindestens ein Trainingsdatenelement aufweisen, das ein Zielverarbeitungsergebnis angibt, das in dem jeweiligen Bereich liegt; und, für jede Trainingsiteration,

- Auswählen von Trainingsdatenelementen aus den Trainingsdaten, durch wiederholtes
- Sampeln eines Trainingsdatenelements aus den Trainingsdaten;
- Ermitteln des Verhältnisses der Anzahl von für die Trainings-Iteration ausgewählten Trainingsdatenelementen, die Zielverarbeitungsergebnisse angeben, die in dem Bereich liegen, in

dem das Zielverarbeitungsergebnis, das das gesampelte Trainingsdatenelement angibt, liegt, zu der Anzahl von für die Trainings-Iteration ausgewählten Trainingsdatenelementen und

- Auswählen des gesampelten Trainingsdatenelements für die Trainings-Iteration, falls das ermittelte Verhältnis kleiner ist als die Akzeptanzrate; und
- Trainieren des maschinellen Lernmodells mittels der für die Trainings-Iteration ausgewählten Trainingsdatenelemente.

[0004] Das oben beschriebene Verfahren verringert (gegenüber einer Verwendung von Trainingsdaten ohne eine Filterung von Trainingsdatenelementen, wie sie das beschriebene Verfahren vorsieht, indem es gesampelte Trainingsdatenelemente ggf. nicht akzeptiert) das Risiko einer Überanpassung beim Training des maschinellen Lernmodells an in den Trainingsdaten überrepräsentierten Zielverarbeitungsergebnisse (d.h. z.B. Bereiche von Szenarien, z.B. Arten von Steuerungsaktionen). So wird die Generalisierungsfähigkeit des maschinellen Lernmodells erhöht. Trainingsdatenelemente, die unwichtig sind in dem Sinne, dass sie keine neuen Informationen (hinsichtlich der Zielverarbeitungsergebnisse) enthalten, werden herausgefiltert, wobei die Filterung von den Zielverarbeitungsergebnissen (z.B. Aktionen) abhängt und nicht von der Trainingseingabe. Damit wird die Bedeutung der Eingabe (z.B. eines Zustands) durch sein Zielverarbeitungsergebnis (z.B. Aktion) bestimmt, nicht durch die Eingabe selbst (da es z.B. mehr als eine mögliche Aktion pro Zustand gibt). Damit wird beispielsweise der Reaktion auf einen Zustand Bedeutung beigemessen, nicht dem Zustand selbst.

[0005] Durch das oben beschriebene Verfahren wird das Trainingsergebnis verbessert und es kann auch die Trainingszeit reduziert werden, da durch das Herausfiltern unwichtiger Trainingsdatenelemente die Gesamtzahl der Trainingsdatenelemente reduziert werden kann und das Training für eine Epoche (mit mehreren Trainingsiterationen) über die (so gefilterten) Trainingsdaten somit kürzer gewählt werden kann. Es kann also durch effiziente Datennutzung eine Reduzierung der Trainingszeit erreicht werden, indem nicht auf vielen „ähnlichen“ Trainingsdatenelementen trainiert wird (wobei „ähnliche“ Trainingsdatenelemente beispielsweise solche sind, die dasselbe (oder sehr ähnliche) Zielverarbeitungsergebnis, z.B. Aktion, enthalten).

[0006] Beispielsweise ist das Training ein Verhaltensklonen, z.B. zum Lernen eines Fahrmodells oder einer Fahrstrategie. Bei diesem kann durch die beschriebene Filterung von Trainingsdatenelementen durch Online-Filterung (z.B. während des Trai-

nings, d.h. Online, gegenüber einer Filterung bei einer Vorverarbeitung, d.h. Offline) der Trainingsdaten im Aktionsraum eine Verbesserung der Realitätsnähe des Fahrermodells bzw. der Fahrstrategie durch Verbesserung der Auswahl von Trainingsdatenelementen erreicht werden. Das Herausfiltern von überrepräsentierten bzw. unwichtigen Trainingsdatenelementen bedeutet in so einem Anwendungsfall, dass in den Trainingsdaten unterrepräsentiertes aufgezeichnetes Fahrverhalten während des Trainings stärker berücksichtigt wird (gegenüber einem Sampeln aus den Trainingsdaten ohne Filterung).

[0007] Verhaltensklonen ist ein Beispiel für überwachtes Lernen. Das Verfahren kann auch auf andere Verfahren für das überwachte Lernen angewendet werden. Die Zielverarbeitungsergebnisse können Aktionen sein, das Verfahren ist aber nicht darauf beschränkt, die Zielverarbeitungsergebnisse können auch beispielsweise Labels für überwachtes Lernen sein, z.B. Klassen für ein Training eines maschinellen Lernmodells (z.B. neuronalen Netzwerks) für eine Bildklassifikation.

[0008] Im Folgenden werden verschiedene Ausführungsbeispiele angegeben.

[0009] Ausführungsbeispiel 1 ist ein Verfahren zum Trainieren eines maschinellen Lernmodells wie oben beschrieben.

[0010] Ausführungsbeispiel 2 ist ein Verfahren nach Ausführungsbeispiel 1, wobei die Akzeptanzrate so ermittelt wird, dass sie umgekehrt proportional zu der Anzahl der Bereiche, für die jeweils gilt, dass die Trainingsdaten mindestens ein Trainingsdatenelement aufweisen, das ein Zielverarbeitungsergebnis angibt, das in dem jeweiligen Bereich liegt, ist.

[0011] Dadurch kann gewährleistet werden, dass die Trainingsdatenelemente sich gleichmäßig über die Bereiche verteilen, für die die Trainingsdaten Trainingsdatenelemente enthalten.

[0012] Ausführungsbeispiel 3 ist ein Verfahren nach Ausführungsbeispiel 1 oder 2, wobei der Ergebnisraum mehrere Dimensionen hat und in die Bereiche eingeteilt wird, indem der Wertebereich jeder Dimension in Intervalle eingeteilt wird und jeder Bereich des Ergebnisraums als eine Kombination von Intervallen mit jeweils einem Intervall pro Dimension definiert wird.

[0013] Die Bereiche (mehrdimensionale Bins) ergeben sich also als Kombinationen von (eindimensionalen) Bins der einzelnen Dimensionen. Dadurch wird erreicht, dass sich die betrachteten Trainingsdatenelemente (und z.B. die betrachteten Kombinationen) über Kombinationen von pro Dimension auftretenden Fällen verteilen.

[0014] Ausführungsbeispiel 4 ist ein Verfahren nach einem der Ausführungsbeispiele 1 bis 3, wobei für jedes Trainingsdatenelement die Trainingseingabe Zustandsinformationen enthält und das Zielverarbeitungsergebnis eine Steuerungsaktion (eines Agenten) angibt.

[0015] Insbesondere im Anwendungsfall des Lernens einer Steuerungsstrategie wird durch das oben beschriebene Verfahren erreicht, dass relevante, aber in der Realität (und damit möglicherweise in den Trainingsdaten) im Vergleich selten auftretende Situationen (wie z.B. Unfallvermeidungsmanöver) beim Training so häufig berücksichtigt werden, dass die gelernte Steuerungsstrategie mit solchen Situationen korrekt umgehen kann.

[0016] Ausführungsbeispiel 5 ist ein Verfahren nach einem der Ausführungsbeispiele 1 bis 4, aufweisend Trainieren des maschinellen Lernmodells mittels der für die Trainings-Iteration ausgewählten Trainingsdatenelemente mittels verstärkenden Lernens oder überwachten Lernens.

[0017] Beispielsweise wird pro Trainingsiteration ein Loss (d.h. Verlust oder Abweichung) berechnet und das maschinelle Lernmodell (z.B. ein neuronales Netz) zur Reduzierung des Losses angepasst. Das Filtern von Trainingsdatenelementen wie oben beschrieben hat insbesondere für diese Trainingsverfahren die oben genannten Vorteile.

[0018] Ausführungsbeispiel 6 ist ein Verfahren zum Steuern eines technischen Systems, aufweisend Trainieren eines maschinellen Lernmodells zum Steuern des technischen Systems nach einem der Ausführungsbeispiele 1 bis 5 und Steuern des technischen Systems unter Verwendung des trainierten maschinellen Lernmodells.

[0019] Ausführungsbeispiel 7 ist eine Datenverarbeitungseinrichtung, eingerichtet zum Durchführen des Verfahrens nach einem der Ausführungsbeispiele 1 bis 6.

[0020] Ausführungsbeispiel 8 ist ein Computerprogramm mit Befehlen, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren nach einem der Ausführungsbeispiele 1 bis 6 durchführt.

[0021] Ausführungsbeispiel 9 ist ein computerlesbares Medium, das Befehle speichert, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren nach einem der Ausführungsbeispiele 1 bis 6 durchführt.

[0022] In den Zeichnungen beziehen sich ähnliche Bezugszeichen im Allgemeinen auf dieselben Teile in den ganzen verschiedenen Ansichten. Die Zeich-

nungen sind nicht notwendigerweise maßstäblich, wobei die Betonung stattdessen im Allgemeinen auf die Darstellung der Prinzipien der Erfindung gelegt wird. In der folgenden Beschreibung werden verschiedene Aspekte mit Bezug auf die folgenden Zeichnungen beschrieben.

Fig. 1 zeigt ein Fahrzeug.

Fig. 2 zeigt eine Visualisierung der Verteilung von Szenarien ursprünglichen Trainingsdaten.

Fig. 3 zeigt eine Visualisierung der Verteilung von Szenarien in gemäß einer Ausführungsform für ein Training ausgewählten Trainingsdaten.

Fig. 4 veranschaulicht ein Trainingsverfahren gemäß einem Ausführungsbeispiel.

Fig. 5 zeigt ein Ablaufdiagramm, das ein Verfahren zum Trainieren eines maschinellen Lernmodells gemäß einer Ausführungsform darstellt.

[0023] Die folgende ausführliche Beschreibung bezieht sich auf die begleitenden Zeichnungen, die zur Erläuterung spezielle Details und Aspekte dieser Offenbarung zeigen, in denen die Erfindung ausgeführt werden kann. Andere Aspekte können verwendet werden und strukturelle, logische und elektrische Änderungen können durchgeführt werden, ohne vom Schutzbereich der Erfindung abzuweichen. Die verschiedenen Aspekte dieser Offenbarung schließen sich nicht notwendigerweise gegenseitig aus, da einige Aspekte dieser Offenbarung mit einem oder mehreren anderen Aspekten dieser Offenbarung kombiniert werden können, um neue Aspekte zu bilden.

[0024] Im Folgenden werden verschiedene Beispiele genauer beschrieben.

[0025] **Fig. 1** zeigt ein Fahrzeug 101.

[0026] Im Beispiel von **Fig. 1** ist ein Fahrzeug 101, beispielsweise ein PKW oder LKW, mit einer Fahrzeugsteuereinrichtung (z.B. einer Electronic Control Unit (ECU)) 102 versehen.

[0027] Die Fahrzeugsteuereinrichtung 102 weist Datenverarbeitungsponenten auf, z.B. einen Prozessor (z.B. eine CPU (Zentraleinheit)) 103 und einen Speicher 104 zum Speichern von Steuerungssoftware 107, gemäß der die Fahrzeugsteuereinrichtung 102 arbeitet, und Daten, die von dem Prozessor 103 verarbeitet werden. Der Prozessor 103 führt die Steuerungssoftware 107 aus.

[0028] Beispielsweise weist die gespeicherte Steuerungssoftware (Computerprogramm) Anweisungen auf, die, wenn der Prozessor sie ausführt, bewirken, dass der Prozessor 103 Fahrerassistenz-Funktionen ausführt (oder auch Fahrdaten sammelt) oder sogar das Fahrzeug autonom steuert.

[0029] Die Steuerungssoftware 107 wird beispielsweise von einem Computersystem 105, z.B. über ein Netzwerk 106 (oder auch mit Hilfe eines Speichermediums wie einer Speicherkarte), an das Fahrzeug 101 übertragen. Dies kann auch im Betrieb geschehen (oder zumindest, wenn das Fahrzeug 101 beim Benutzer ist), da die Steuerungssoftware 107 beispielsweise im Laufe der Zeit auf neue Versionen aktualisiert wird.

[0030] Die Steuerungssoftware 107 kann beispielsweise mittels maschinellen Lernens (ML) trainiert werden, d.h. die Steuerungssoftware 107 implementiert ein ML-Modell 108 (oder „maschinelles Lernmodell“), das auf der Grundlage von Trainingsdaten 109, in diesem Beispiel vom Computersystem 105, trainiert wird. Das Computersystem 105 implementiert also einen ML-Trainingsalgorithmus zum Training des ML-Modells 108.

[0031] Im Anwendungsbeispiel von **Fig. 1** ist das Ziel somit das (Daten-basierte) Erlernen einer Steuerungsstrategie zur Steuerung eines automatisierten Fahrzeugs (Anwendungsfall Ego-Steuerungsstrategie) oder auch von Agenten für eine Simulation (Anwendungsfall Fahrmodell, z.B. zum Simulieren von Verkehrsteilnehmern für das Testen bzw. Bewerten einer weiteren Steuerungsstrategie). Die Trainingsdaten bestehen in diesem Fall z.B. aus aufgezeichneten Sequenzen aus der realen Welt. Jedes Trainingsdatenelement enthält (zumindest) ein Paar aus einer Trainingseingabe, in diesem Fall eine Beobachtung, und eines Zielverarbeitungsergebnisses (für die Trainingseingabe).

[0032] Jede Beobachtung enthält Informationen über den Zustand des Agenten (also in diesem Beispiel des Fahrzeugs) und seiner Umgebung zu einem jeweiligen Zeitpunkt t . Eine Beobachtung kann z.B. einen Vektor (mit Geschwindigkeiten etc.) oder ein oder mehrere gerasterte Bilder enthalten. Das Zielverarbeitungsergebnis (auch als Label bezeichnet) für eine Beobachtung enthält in diesem Anwendungsbeispiel die entsprechende Fahraktion zum Zeitpunkt t , die als Low-Level-Aktion(en) (z. B. Lenkwinkel und Beschleunigung) oder Wegpunkte (x-y-Delta in Bezug auf die globale Position und Lage des Agenten) repräsentiert sein kann.

[0033] In dem Anwendungsbeispiel von **Fig. 1** wird beispielsweise anhand solcher Trainingsdaten mit Hilfe von Behavioural Cloning das maschinelle Lernmodell derart trainiert, dass es eine Steuerungsstrategie implementiert, die ein realistisches Fahrverhalten, wie es durch die Trainingsdaten dargestellt wird, zu imitieren, indem für jeden mehrerer Batches, die jeweils eine Vielzahl von Trainingsdatenelementen enthalten, der Unterschied zwischen den von dem maschinellen Lernmodell 108 vorhergesagten Fahrweisen und den jeweiligen Fahrweisen, wie sie durch

die jeweiligen Zielverarbeitungsergebnisse angegebenen werden, durch Anpassung des maschinellen Lernmodells 108 (also z.B. der Gewichte eines neuronalen Netzwerks) minimiert (oder zumindest reduziert) wird.

[0034] Dieser Unterschied wird durch einen Trainings-Loss erfasst, der die mittlere Differenz über alle Trainingsdatenelemente des jeweiligen Batches berechnet und das maschinelle Lernmodell wird in Richtung des abnehmenden Losses optimiert.

[0035] Fahraktionen, die in den Trainingsdaten häufiger vertreten sind, werden daher den Loss stärker beeinflussen als unterrepräsentierte Fahraktionen. Dies erschwert das Training, da reale Datensätze viele ähnliche Szenarien enthalten. Beispielsweise fahren die meisten Fahrzeuge auf der Straße in derselben Spur und führen nur selten Spurwechsel oder Notfallmanöver wie Notbremsungen oder Notausweichmanöver durch. Von einer auf der Grundlage solcher Trainingsdaten trainierten Steuerungsstrategie kann nicht erwartet werden, dass sie bei unterrepräsentierten Fahrmanövern oder Szenarien (wie Notbremsungen oder Notausweichmanövern) gut funktioniert.

[0036] Das Ziel des Trainings einer datenbasierten Fahrstrategie besteht jedoch darin, eine Steuerungsstrategie zu trainieren, die sich gut auf viele verschiedene Szenarien verallgemeinern lässt, einschließlich Szenarien, die nur in geringer Häufigkeit im Trainingsdatensatz auftreten.

[0037] Gemäß verschiedenen Ausführungsformen wird eine Herangehensweise zum Training eines maschinellen Lernmodells bereitgestellt, die die Verallgemeinerung des maschinellen Lernmodells auf Szenarien, die in den Trainingsdaten vorkommen, aber im Vergleich zu anderen, häufiger auftretenden Szenarien unterrepräsentiert sind, verbessert.

[0038] Gemäß verschiedenen Ausführungsformen wird die Generalisierung des maschinellen Lernmodells verbessert, indem für sein Training eine Teilmenge der Trainingsdaten ausgewählt wird, die ausgewogener ist und in der verschiedene Szenarien (im Anwendungsfall von **Fig. 1** z.B. definiert durch ihre jeweiligen Fahraktionen) mit ähnlicher Häufigkeit auftreten.

[0039] **Fig. 2** zeigt eine Visualisierung der Verteilung von Szenarien (Fahren in einer Fahrspur 201, Notfallbremsung 202, Fahrspurwechsel 203) in den ursprünglichen, unausgewogenen Trainingsdaten.

[0040] **Fig. 3** zeigt eine Visualisierung der Verteilung von Szenarien (Fahren in einer Fahrspur 301, Notfallbremsung 302, Fahrspurwechsel 303) in den ausgewählten Trainingsdaten. In diesen sind die ver-

schiedenen Szenarien deutlich ausgeglichener Vorhanden.

[0041] Gemäß verschiedenen Ausführungsformen wird dieser Ausgleich durch eine Filterung von Trainingsdatenelementen, d.h. die Zurückweisung überrepräsentierter Szenarien (d.h. entsprechender Trainingselement-Samples) zugunsten unterrepräsentierter Trainingselement-Samples zur Trainingszeit realisiert. Dabei wird davon ausgegangen, dass für jede Trainingsiteration (d.h. z.B. jeden Batch) Trainingselemente aus den Trainingsdaten gesampelt werden (wobei dies rein zufällig erfolgen kann aber auch eine gewisse Explorationsstrategie berücksichtigt werden kann). Infolgedessen lernt das maschinelle Lernmodell ein realistischeres Verhalten für (in den ursprünglichen Trainingsdaten) unterrepräsentierte Szenarien (z.B. Aktionen) wie Spurwechsel oder Notbremsungen.

[0042] Um den Ausgleich zu erreichen, wird gemäß verschiedenen Ausführungsformen die Anzahl der akzeptierten, d.h. nicht abgelehnten gesampelten Trainingselemente aufgeschlüsselt nach Szenarien (d.h. Bereichen von Zielverarbeitungsergebnissen n , z.B. Fahraktionen) bei der Auswahl von Trainingsdatenelementen verfolgt.

[0043] Beispielsweise werden die Trainingsdatenelemente aller Trainingsdaten in Bereiche (Bins) diskreter Größe auf der Grundlage der Dimensionen der Zielverarbeitungsergebnisse (z.B. der (Fahr-) Aktionsdimensionen eingeteilt. Mit den Dimensionen sind hier die verschiedenen Komponenten gemeint: Ist ein Zielverarbeitungsergebnis (d.h. im aktuellen Anwendungsfall eine Fahraktion) z.B. ein Vektor von zwei Fließkommazahlen, so hat das Zielverarbeitungsergebnis zwei Dimensionen (z.B. Beschleunigung und Lenkwinkel).

[0044] Die Zielverarbeitungsergebnisdimensionen (z.B. Handlungsdimensionen) brauchen nicht den Dimensionen der Daten in den Trainingsdatenelementen entsprechen, die die Zielverarbeitungsergebnisse angeben.

[0045] Beispielsweise enthält jedes Trainingsdatenelement eine Angabe des Zielverarbeitungsergebnisses in Form einer Folge von Wegpunkten über z.B. fünf Zeitschritte, welche allerdings nicht den kinematischen Zustand ausdrücken (d.h. ob aktuell gebremst oder beschleunigt oder wie gelenkt wird). Um den kinematischen Zustand als Grundlage für das Filtern der Trainingsdatenelemente zu verwenden (d.h. also ein Action-Balancing durchzuführen), können die Wegpunkte (durch eine entsprechende Transformation) auf Beschleunigungs- und/oder Lenkaktionen abgebildet werden. Diese sind im Vergleich zu den Wegpunkten niedrigdimensional und können somit die Filterung verbessern. Nichtsdesto-

trotz kann jedes Zielverarbeitungsergebnis, das in einem Trainingsdatenelement explizit in Form von Wegpunkten angegeben ist, auch als Angabe eines Zielverarbeitungsergebnisses in Form von Aktionen angesehen werden (da die Aktionen durch eine entsprechende Abbildung bzw. Transformation durch die Wegpunkte angegeben sind). Wird hierin also darauf Bezug genommen, dass Trainingsdatenelemente Zielverarbeitungsergebnisse in einem Ergebnisraum angeben, so können sie das explizit (durch Elemente aus dem Ergebnisraum) tun oder durch Elemente aus einem anderen Raum, die Elemente aus dem Ergebnisraum angeben (z.B. Folgen von Wegpunkten, die wiederum Lenkaktionen angeben) tun.

[0046] Die Diskretisierung jeder Dimension (d.h. die Größe jedes einzelnen Bins pro Komponente der Fahraktion) wird beispielsweise wie folgt berechnet:

$$\delta = \frac{x_{\max} - x_{\min}}{\sqrt{n}} \quad (1)$$

wobei δ die Bin-Größe in der jeweiligen Dimension, x_{\max} der maximale Wert für die Komponente der Fahraktion der jeweiligen Dimension, der in den Trainingsdaten auftritt, x_{\min} der minimale Wert für die Komponente der Fahraktion der jeweiligen Dimension, der in den Trainingsdaten auftritt, und n die Gesamtzahl der Trainingsdatenelemente in den Trainingsdaten ist.

[0047] Bei Zielverarbeitungsergebnissen n mit zwei Dimensionen ergeben sich somit zweidimensionale Bins, deren Rasterung in jeder Dimension durch das jeweilige für diese Dimension bestimmte δ gegeben ist. Das Raster von Bins (zweidimensional oder auch höher) deckt den Bereich des Ergebnisraums (z.B. Aktionsraums), in dem die Trainingsdaten Zielverarbeitungsergebnisse enthalten, ab. Bei mehr als einer Dimension entspricht jedes Bin einer Kombination von Bereichen der einzelnen Komponente (z.B. Bin 1: Lenkwinkel zwischen 10 Grad und 11 Grad, Beschleunigung zwischen 2m/s² und 3m/s²; Bin 2: Lenkwinkel zwischen 11 Grad und 12 Grad, Beschleunigung zwischen 2m/s² und 3m/s²; ...; Bin X: Lenkwinkel zwischen 10 Grad und 11 Grad, Beschleunigung zwischen 3m/s² und 4m/s² ...).

[0048] Auf der Grundlage dieser Einteilung werden nun beim Training Trainingsdaten-Elemente ausgewählt (d.h. es wird eine „Online-Filterung“ der Trainingsdatenelemente aus den Trainingsdaten durchgeführt). Dabei wird ein aus den Trainingsdaten gesampeltes Trainingsdatenelement nur dann akzeptiert, wenn die Anzahl der Trainingsdatenelemente, die sich bereits in dem Bin befinden, zu dem das Trainingsdatenelement (gemäß seines Zielverarbeitungsergebnisses) gehört, geringer ist als eine Akzeptanzrate.

[0049] Die Akzeptanzrate ρ wird beispielsweise berechnet als:

$$\rho = \frac{10}{N} \quad (2)$$

wobei N die Gesamtzahl der Bins ist, zu denen mindestens ein Trainingsdatenelement aus den Trainingsdaten gehört.

[0050] Die Akzeptanzrate wird wie folgt angewendet. Ein Trainingsdatenelement, das zu dem i -ten Bin gehört, wird akzeptiert, falls

$$\frac{b_i}{\sum_{j=0}^{N'} b_j} < \rho \quad (3)$$

wobei b_i und b_j die Anzahl der für den aktuellen Batch bisher ausgewählter Trainingsdatenelemente, die zu dem i -ten bzw. j -ten Bin gehören, ist und N' die Gesamtzahl der Bins ist, die (gemäß der bisherigen Auswahl für den aktuellen Batch) mindestens ein Trainingsdatenelement enthalten.

[0051] Hierbei ist zur Einfachheit angenommen, dass die Bins mittels natürlicher Zahlen linear durchnummeriert sind. Bei einem höherdimensionalen Raster von Bins können die Indizes auch als Kombination von natürlichen Zahlen indiziert werden.

[0052] Auf diese Weise werden Batches der Größe k aus den Trainingsdaten gebildet und dann zum Trainieren eines ML-Modells (z.B. durch überwachtes Lernen oder verstärkendes Lernen) verwendet. Durch die Filterung (d.h. Akzeptanz von gesampelten Trainingsdatenelementen für das Training abhängig von der Bedingung von Gleichung (3)) wird eine Überanpassung an überrepräsentierte Szenarien (d.h. Zielverarbeitungsergebnisse) verhindert (oder zumindest eingeschränkt), da das ML-Modell auf einer ausgewogenen Menge von Trainingsdatenelementen trainiert wird. Dies führt gegenüber einem Training auf der Grundlage der ungefilterten Trainingsdaten zu einer verbesserten Modellgeneralisierung.

[0053] Im Anwendungsfall von **Fig. 1** wird wie oben erwähnt eine Fahraktionen beispielsweise durch einen Vektor von Fließkommazahlen (z.B. Lenkwinkel, Beschleunigung oder Wegpunkte) repräsentiert. Die Datenverarbeitungseinrichtung, die das Training durchführt, z.B. das Computersystem 105, ordnet somit jede Fahraktion einem diskreten Bereich zu und prüft, ob ein aktuell gesampeltes Trainingsdatenelement (d.h. im Anwendungsfall von **Fig. 1** ein Beobachtungs-Aktions-Paar) auf der Grundlage der Anzahl der bereits ausgewählten (d.h. akzeptierten) Trainingsdatenelemente in diesem Bereich akzeptiert werden sollte (siehe Gleichung (3)). Die Batches, die zum Training des maschinellen Lernmo-

dells 108 verwendet werden, werden dann nur mit den akzeptierten Trainingsdatenelementen gefüllt. Jedes akzeptierte Trainingsdatenelement wird einmal in den Batch für die aktuelle Trainingsiteration aufgenommen. Es kann auch vorgesehen sein, dass jedes akzeptierte Trainingsdatenelement pro Epoche (mit mehreren Batches) nur (d.h. genau) einmal in einen Batch aufgenommen wird.

[0054] Gemäß einer Ausführungsform führt die Datenverarbeitungseinrichtung, die das Training durchführt, z.B. das Computersystem 105, die folgenden Schritte durch:

1. Bestimmen der Parameter δ und ρ :
 - a. Laden des Satzes von Trainingsdaten
 - b. Iterieren über alle Trainingsdatenelemente der Trainingsdaten
 - c. Aufzeichnen der Minimal- und Maximalwerte für jede Dimension der Zielverarbeitungsergebnisse (d.h. x_{\min} und x_{\max} für jede Dimension)
 - d. Berechnen von δ (Gleichung (1))
 - e. Iterieren über alle Trainingsdatenelemente und Einordnen der Trainingsdatenelemente in die entsprechenden Bins (gemäß ihrer Zielverarbeitungsergebnisse).
 - f. Ermitteln von N und Berechnung von ρ (Gleichung (2)).
2. Online-Trainingsdatenfilterung beim Training für jeden Batch
 - a. Samplen eines Trainingsdatenelements aus den Trainingsdaten
 - b. Ermitteln des Bins i , zu dem das gesampelte Trainingsdatenelement gehört und der Anzahl von schon für den Batch ausgewählten Trainingsdatenelementen, die zu diesem Bin gehören, b_i
 - c. Prüfen, ob das Trainingsdatenelement gemäß der Akzeptanzquote ρ akzeptiert wird (Gleichung (3))
 - d. Wenn das Trainingsdatenelement akzeptiert wird, Hinzufügen des Trainingsdatenelements zu dem Batch (wenn nicht, Verwerfen des Trainingsdatenelements und Rückkehr zu Punkt 2a) und Erhöhen der Anzahl von b_i um eins (in anderen Worten Anpassen des Histogramms der ausgewählten Trainingsdatenelemente)
 - e. Wiederholen von 2.a. bis 2.d., bis der Batch bis zu einer vorgegebenen Größe k mit akzeptierten Trainingsdatenelementen gefüllt ist.
 - f. Trainieren des maschinellen Lernmodells mittels des Batches

[0055] Fig. 4 veranschaulicht das oben angegebene Trainingsverfahren.

[0056] Ein Trainingsdatenelement 401 wird aus den Trainingsdaten 402 gesampelt und auf seinem zugehörigen Bin 405 von (zur Einfachheit) drei Bins 403, 404, 405 zugeordnet. Es ist angenommen, dass der aktuelle Batch bisher zwei Trainingsdatenelemente enthält, die zum ersten Bin 403 gehören (d.h. $b_1 = 2$), vier Trainingsdatenelemente enthält (d.h. $b_2 = 4$), die zum zweiten Bin 404 gehören und ein Trainingsdatenelement enthält, das zum dritten Bin 405 gehört (d.h. $b_3 = 1$). Es wird nun Gleichung (3) überprüft, konkret ob $b_3/(b_1+b_2+b_3) = 1/(2+4+1)$ kleiner als die Akzeptanzrate ρ ist. Ist dies nicht der Fall, wird das gesampelte Trainingsdatenelement 401 verworfen und neu gesampelt. Ist $1/(2+4+1)$ kleiner als die Akzeptanzrate ρ , so wird das gesampelte Trainingsdatenelement 401 dem aktuellen Batch 406 hinzugefügt, b_3 von 1 auf 2 erhöht und neu gesampelt.

[0057] Dieses Trainingsverfahren kann auch offline für die Auswahl von Trainingsdaten während einer Vorverarbeitung angewendet werden. Bei einer Online-Filterung ist es nicht erforderlich, dass die Auswahl in einer solchen Vorverarbeitung stattfindet. Je nachdem kann das beschriebene Filterungsverfahren beispielsweise von einem Datenlademodul für das Training, das eine Online-Auswahl ermöglicht, oder von einem Datenvorverarbeitungsmodul implementiert werden.

[0058] Wie oben beschrieben kann das Trainingsverfahren auf Verhaltensklonen zum Lernen eines Fahrermodells oder einer Fahrstrategie für automatisierte Fahrfunktionen angewendet werden. Um die Generalisierung des Modells zu verbessern, werden die Trainingsmuster bei diesem Trainingsverfahren gefiltert, um ein realistischeres Fahrverhalten zu erhalten.

[0059] Zusammengefasst wird gemäß verschiedenen Ausführungsformen ein Verfahren bereitgestellt, wie in Fig. 5 dargestellt.

[0060] Fig. 5 zeigt ein Ablaufdiagramm 500, das ein Verfahren zum Trainieren eines maschinellen Lernmodells gemäß einer Ausführungsform darstellt.

[0061] In 501 wird für (gegebene) Trainingsdaten, die eine Vielzahl von Trainingsdatenelementen enthalten, die jeweils ein Paar aus Trainingseingabe und Zielverarbeitungsergebnis für die Trainingseingabe aus einem Ergebnisraum angeben, der Ergebnisraums in Bereiche (auch als Bins bezeichnet) eingeteilt (z.B. gleichmäßig).

[0062] In 502 wird eine Akzeptanzrate abhängig von der Anzahl (N im obigen Beispiel) der Bereiche, für

die jeweils gilt, dass die Trainingsdaten mindestens ein Trainingsdatenelement aufweisen, das ein Zielverarbeitungsergebnis angibt, das in dem jeweiligen Bereich liegt, ermittelt.

[0063] In 503 werden in 504 für jede Trainingsiteration Trainingsdatenelementen aus den Trainingsdaten ausgewählt (z.B. ein (Trainings-)Batch ausgewählt), durch wiederholtes (d.h. mehrmaliges, z.B. bis eine vorgegebene Anzahl von ausgewählten Trainingsdatenelementen (d.h. z.B. eine bestimmte Batch-Größe) erreicht ist):

- Sampeln (d.h. (z.B. zufälliges) Ziehen) eines Trainingsdatenelements aus den Trainingsdaten;
- Ermitteln des Verhältnisses der Anzahl (b_i im obigen Beispiel) von für die Trainings-Iteration ausgewählten Trainingsdatenelementen, die Zielverarbeitungsergebnisse angeben, die in dem Bereich liegen, in dem das Zielverarbeitungsergebnis, das das gesampelte Trainingsdatenelements angibt, liegt, zu der Anzahl von für die Trainings-Iteration (bisher) ausgewählten Trainingsdatenelementen und
- Auswählen des gesampelten Trainingsdatenelements für die Trainings-Iteration, falls das ermittelte Verhältnis kleiner ist als die Akzeptanzrate (und ansonsten z.B. Verwerfen des Trainingsdatenelements, d.h. das Trainingsdatenelement wird nicht für das Training der Trainings-Iteration verwendet); und

in 505 das maschinelle Lernmodell mittels der für die Trainings-Iteration ausgewählten Trainingsdatenelemente trainiert.

[0064] Gemäß verschiedenen Ausführungsformen wird mit anderen Worten eine Filterung von Trainingsdaten auf der Grundlage eines Sub-Samplings und einer Einteilung des Ergebnisraums in Bereiche (Bins) durchgeführt. Die Einteilung in Bereiche kann wie oben beschrieben auf einer Einteilung der verschiedenen Dimensionen der Zielverarbeitungsergebnisse basieren. Die Zielverarbeitungsergebnisse sind beispielsweise (niedrigdimensionale) wie z.B. eines oder eine Kombination von Geschwindigkeit, Lenkwinkel, Wegpunkt(en), Koordinaten, Beschleunigung, usw.

[0065] Das Verfahren von **Fig. 5** kann durch einen oder mehrere Computer mit einer oder mehreren Datenverarbeitungseinheiten durchgeführt werden. Der Begriff „Datenverarbeitungseinheit“ kann als irgendein Typ von Entität verstanden werden, die die Verarbeitung von Daten oder Signalen ermöglicht. Die Daten oder Signale können beispielsweise gemäß mindestens einer (d.h. einer oder mehr als einer) speziellen Funktion behandelt werden, die durch die Datenverarbeitungseinheit durchgeführt

wird. Eine Datenverarbeitungseinheit kann eine analoge Schaltung, eine digitale Schaltung, eine Logikschaltung, einen Mikroprozessor, einen Mikrocontroller, eine Zentraleinheit (CPU), eine Graphikverarbeitungseinheit (GPU), einen Digitalsignalprozessor (DSP), eine integrierte Schaltung einer programmierbaren Gatteranordnung (FPGA) oder auch Quantensensoren und Quantencomputer oder irgendeine Kombination davon umfassen oder aus dieser ausgebildet sein. Irgendeine andere Weise zum Implementieren der jeweiligen Funktionen, die hierin genauer beschrieben werden, kann auch als Datenverarbeitungseinheit oder Logikschaltungsanordnung verstanden werden. Es können ein oder mehrere der im Einzelnen hier beschriebenen Verfahrensschritte durch eine Datenverarbeitungseinheit durch eine oder mehrere spezielle Funktionen ausgeführt (z. B. implementiert) werden, die durch die Datenverarbeitungseinheit durchgeführt werden.

[0066] Das Verfahren ist also gemäß verschiedenen Ausführungen insbesondere computerimplementiert.

[0067] Je nach Art der Eingabe und Ausgabe und entsprechend der Art des maschinellen Lernmodells erfolgt das Training mittels der ausgewählten Trainingsdatenelemente beispielsweise durch Überwachtes Lernen bzw. Klonen von Verhaltensweisen (Behavioural Cloning) oder verstärkendem Lernen (Reinforcement Learning), z.B. Erlernen einer Steuerungsstrategie durch Off-Policy Verstärkungslernen, indem bei jeder Trainingsiteration der ausgewählte (also gefilterte) Satz von gesammelten Erfahrungen für die Steuerung eines technischen Systems zum Training verwendet wird.

[0068] Das maschinelle Lernmodell kann nach dem Training auf Sensordaten angewendet werden, die von mindestens einem Sensor ermittelt werden. Die Ausgabe des maschinellen Lernmodells liefert somit ein Ergebnis über einen physikalischen Zustand einer Umgebung des mindestens einen Sensors und/oder des mindestens einen Sensors selbst bzw. das Verfahren kann aufweisen, die Ausgabe des trainierten maschinellen Lernmodells, die es in Reaktion auf eine Eingabe von Sensordaten liefert, als ein solches Ergebnis zu verwenden.

[0069] Beispielsweise wird ein maschinelles Lernmodell, das eine Steuerungsstrategie repräsentiert, trainiert und nach dem Training zum Erzeugen eines Steuersignals für eine Robotervorrichtung eingesetzt, indem man ihm Sensordaten bzgl. der Robotervorrichtung und/oder seiner Umgebung zuführt. Der Begriff „Robotervorrichtung“ kann als sich auf irgendein technisches System (mit einem mechanischen Teil, dessen Bewegung gesteuert wird) beziehend verstanden werden, wie z. B. eine computergesteuerte Maschine, ein Fahrzeug, ein Haushaltsgerät, ein Elektrowerkzeug, eine Ferti-

gungsmaschine, einen persönlichen Assistenten oder ein Zugangssteuersystem. Es wird also beispielsweise eine Steuerungsvorschrift oder Steuerungsstrategie für ein solches technisches System gelernt und das technische System dann entsprechend gesteuert.

[0070] Das Training der Steuerungsstrategie bzw. eines maschinellen Lernmodells, das sie repräsentiert, kann als Training eines Agenten angesehen werden (der gemäß der Steuerungsstrategie handelt). Der allgemeine Begriff „Agent“ wird hierin auch insbesondere für alle Arten von technischen Systemen verwendet, die mit den hierin beschriebenen Ansätzen gesteuert werden können. Die hierin beschriebenen Ansätze können jedoch auf jede Art von Agenten angewendet werden (z.B. auch auf einen Agenten, der nur simuliert wird und nicht physisch existiert).

[0071] Verschiedene Ausführungsformen können als Trainingseingabe (bzw. nach dem Training als Eingabedaten) Sensorsignale von verschiedenen Sensoren wie z. B. Video, Radar, LiDAR, Ultraschall, Bewegung, Wärmeabbildung usw. empfangen und verwenden, beispielsweise um Sensordaten hinsichtlich Demonstrationen oder Zuständen des Systems (Roboter und Objekt oder Objekte) und Konfigurationen und Szenarios zu erhalten. Die Sensordaten können durch das maschinelle Lernmodell verarbeitet werden. Dies kann die Klassifikation der Sensordaten oder das Durchführen einer semantischen Segmentierung an den Sensordaten umfassen, beispielsweise um die Anwesenheit von Objekten (in der Umgebung, in der die Sensordaten erhalten wurden) zu detektieren. Ausführungsformen können zum Trainieren eines Maschinenlernsystems und Steuern einer Robotervorrichtung, z. B. autonom von Roboter manipulieren, um verschiedene Manipulationsaufgaben unter verschiedenen Szenarios zu erreichen, verwendet werden. Insbesondere sind Ausführungsformen auf die Steuerung und Überwachung der Ausführung von Manipulationsaufgaben anwendbar, z. B. in Montagelinien.

[0072] Das maschinelle Lernmodell kann beispielsweise zum Messen und Steuern trainiert werden, d.h. zum Analysieren von (Sensor-)Daten (z.B. skalare Zeitreihen), insbesondere Aktionsdaten, und anschließendes Betreiben eines jeweiligen technischen Systems. Ein Anwendungsbeispiel ist eine Fail-Safe-Operation, für die das maschinelle Lernmodell dazu trainiert (und nach dem Training verwendet) wird, um signifikante Eckfälle zu erkennen, indem es auf der Grundlage von Trainingsdaten, die solche signifikante Eckfälle enthalten, trainiert wird. Es kann dazu trainiert werden, sicherheitskritische Eckfälle zu erkennen und z.B. als Aktionen geeignete Gegenmaßnahmen auszugeben.

[0073] Beispielsweise ist es typischerweise wichtig, sicherzustellen, dass ein automatisiertes Fahrzeug nicht mit Fußgängern zusammenstößt und dass es anderen Fahrzeugen ausweicht. Gemäß der Herangehensweise von **Fig. 5** kann die Bedeutung von in den Trainingsdaten unterrepräsentierten Aktionen, wie z.B. Kollisionsvermeidung von Kollisionen, erhöht werden, sodass ein auf solchen Trainingsdaten maschinelles Lernmodell (d.h. in diesem Fall ein Fahrermodell bzw. eine Fahrstrategien) in der Lage ist, auf realistischere Fahrentscheidungen zu verallgemeinern.

[0074] Die Herangehensweise von **Fig. 5** kann auch für eine aktive Auswahl von Daten verwendet werden, die ein Datenverarbeitungssystem (z.B. entsprechend dem Computersystem 105), das einen maschinellen Lernalgorithmus implementiert, neben dem Training zum Testen, Verifizieren und/oder Validieren verwendet. Insbesondere wird ein Testen, Verifizieren und/oder Validieren auch als Teil des Trainings angesehen, d.h. eine Trainingsiteration kann auch eine Test-, Verifizierungs- oder Validierungsiteration sein (da sie beispielsweise darüber entscheidet, ob weitertrainiert wird oder nicht).

[0075] Beispielsweise wurde ein Trainingsdatensatz aufgezeichnet, der ein Fahrverhalten auf öffentlichen Straßen repräsentiert. Der Datensatz zeigt größtenteils ähnliches Verhalten (Fahren in der Spur, Einhalten der Geschwindigkeit, ähnliche Aktionen), enthält aber auch einige Eckfälle (Fahrspurwechsel, plötzliches Bremsen, Kollisionsvermeidung, Auffahren auf oder Abfahren von einer Straße etc.). Die Herangehensweise von **Fig. 5** filtert den Trainingsdatensatz auf der Grundlage der festgelegten Filterkriterien und wählt eine größere Varianz von Fahrmanövern aus, um den maschinellen Lernalgorithmus bei der Verallgemeinerung auf Eckfälle (neben dem regulärem Fahrverhalten) zu unterstützen.

Patentansprüche

1. Verfahren zum Trainieren eines maschinellen Lernmodells (108), insbesondere zum Implementieren einer Steuerungsstrategie für eine Robotervorrichtung, aufweisend:
Einteilen, für Trainingsdaten, die eine Vielzahl von Trainingsdatenelementen enthalten, die jeweils ein Paar aus Trainingseingabe, insbesondere Trainings-Sensordaten, die einen Zustand einer Robotervorrichtung und/oder einer Umgebung der Robotervorrichtung beschreiben, und Zielverarbeitungsergebnis für die Trainingseingabe aus einem Ergebnisraum, insbesondere einen Robotervorrichtungs-Steuerungsaktionsraum, angeben, des Ergebnisraums in Bereiche (403, 404, 405);
Ermitteln einer Akzeptanzrate abhängig von der Anzahl der Bereiche (403, 404, 405), für die jeweils

gilt, dass die Trainingsdaten (402) mindestens ein Trainingsdatenelement aufweisen, das ein Zielverarbeitungsergebnis angibt, das in dem jeweiligen Bereich (403, 404, 405) liegt; und für jede Trainingsiteration, Auswählen von Trainingsdatenelementen (406) aus den Trainingsdaten, durch wiederholtes Sampeln eines Trainingsdatenelements (401) aus den Trainingsdaten; Ermitteln des Verhältnisses der Anzahl von für die Trainings-Iteration ausgewählten Trainingsdatenelementen, die Zielverarbeitungsergebnisse angeben, die in dem Bereich (403, 404, 405) liegen, in dem das Zielverarbeitungsergebnis, das das gesampelte Trainingsdatenelements (401) angibt, liegt, zu der Anzahl von für die Trainings-Iteration ausgewählten Trainingsdatenelementen und Auswählen des gesampelten Trainingsdatenelements (401) für die Trainings-Iteration, falls das ermittelte Verhältnis kleiner ist als die Akzeptanzrate; und Trainieren des maschinellen Lernmodells (108) mittels der für die Trainings-Iteration ausgewählten Trainingsdatenelemente.

2. Verfahren nach Anspruch 1, wobei die Akzeptanzrate so ermittelt wird, dass sie umgekehrt proportional zu der Anzahl der Bereiche (403, 404, 405), für die jeweils gilt, dass die Trainingsdaten (402) mindestens ein Trainingsdatenelement aufweisen, das ein Zielverarbeitungsergebnis angibt, das in dem jeweiligen Bereich (403, 404, 405) liegt, ist.

3. Verfahren nach Anspruch 1 oder 2, wobei der Ergebnisraum mehrere Dimensionen hat und in die Bereiche (403, 404, 405) eingeteilt wird, indem der Wertebereich jeder Dimension in Intervalle eingeteilt wird und jeder Bereich (403, 404, 405) des Ergebnisraums als eine Kombination von Intervallen mit jeweils einem Intervall pro Dimension definiert wird.

4. Verfahren nach einem der Ansprüche 1 bis 3, wobei für jedes Trainingsdatenelement die Trainingseingabe Zustandsinformationen enthält und das Zielverarbeitungsergebnis eine Steuerungsaktion angibt.

5. Verfahren nach einem der Ansprüche 1 bis 4, aufweisend Trainieren des maschinellen Lernmodells (108) mittels der für die Trainings-Iteration ausgewählten Trainingsdatenelemente mittels verstärkenden Lernens oder überwachten Lernens.

6. Verfahren zum Steuern eines technischen Systems, aufweisend Trainieren eines maschinellen Lernmodells (108) zum Steuern des technischen Systems nach einem der Ansprüche 1 bis 5 und Steuern des technischen Systems unter Verwendung des trainierten maschinellen Lernmodells (108).

7. Datenverarbeitungseinrichtung (105), eingerichtet zum Durchführen des Verfahrens nach einem der Ansprüche 1 bis 6.

8. Computerprogramm mit Befehlen, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren nach einem der Ansprüche 1 bis 6 durchführt.

9. Computerlesbares Medium, das Befehle speichert, die, wenn sie durch einen Prozessor ausgeführt werden, bewirken, dass der Prozessor ein Verfahren nach einem der Ansprüche 1 bis 6 durchführt.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

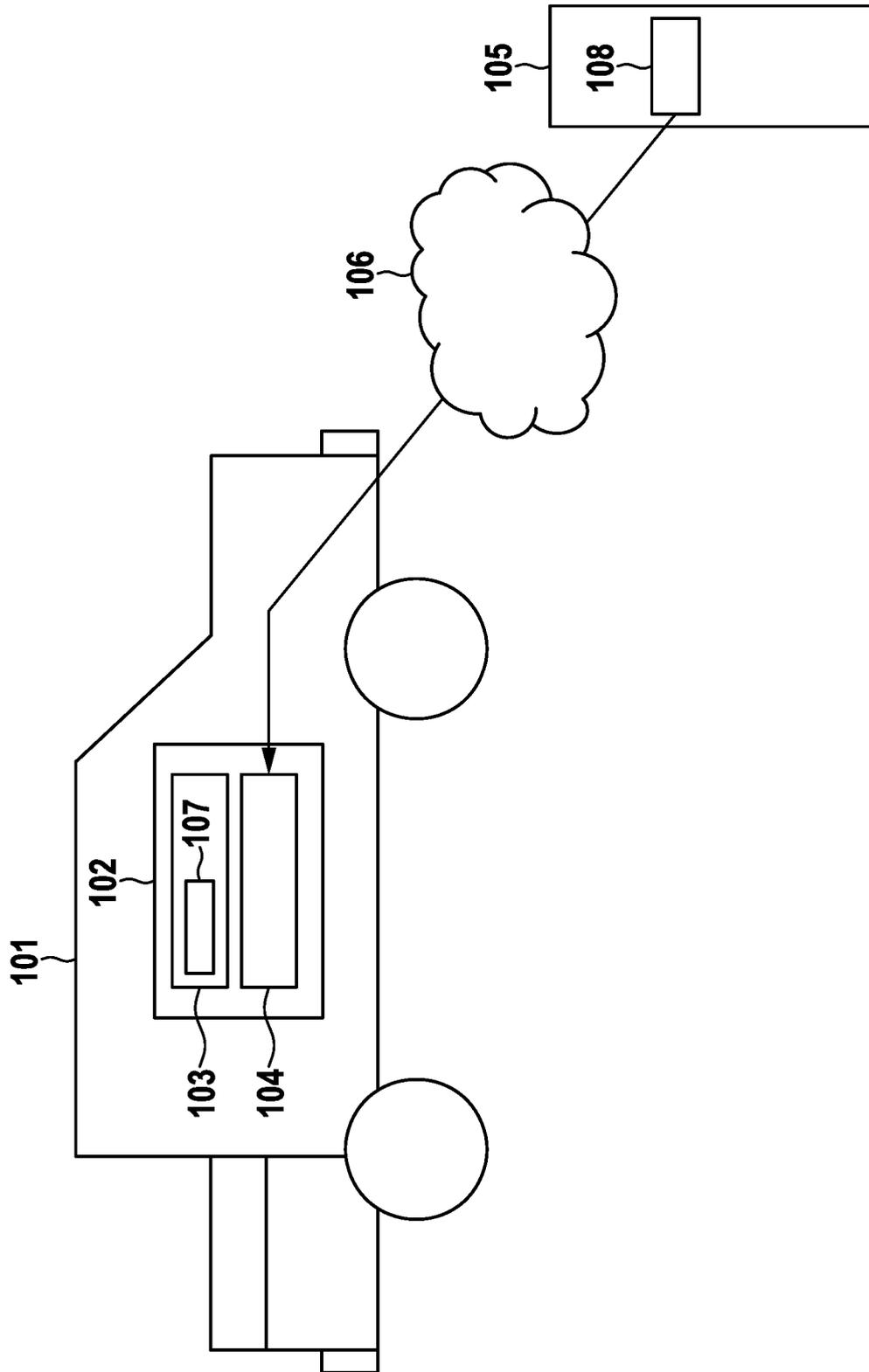
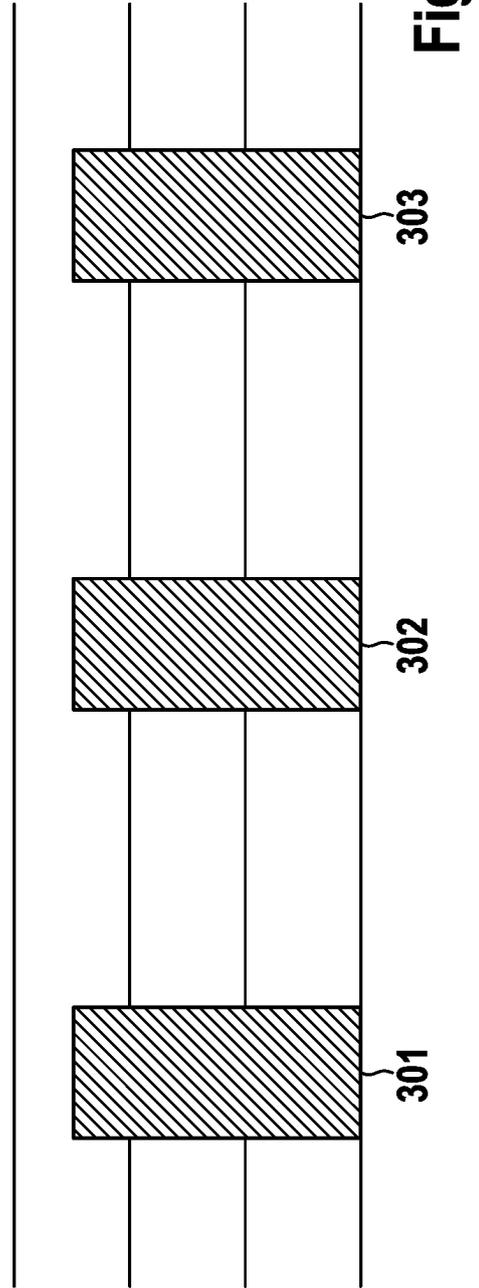
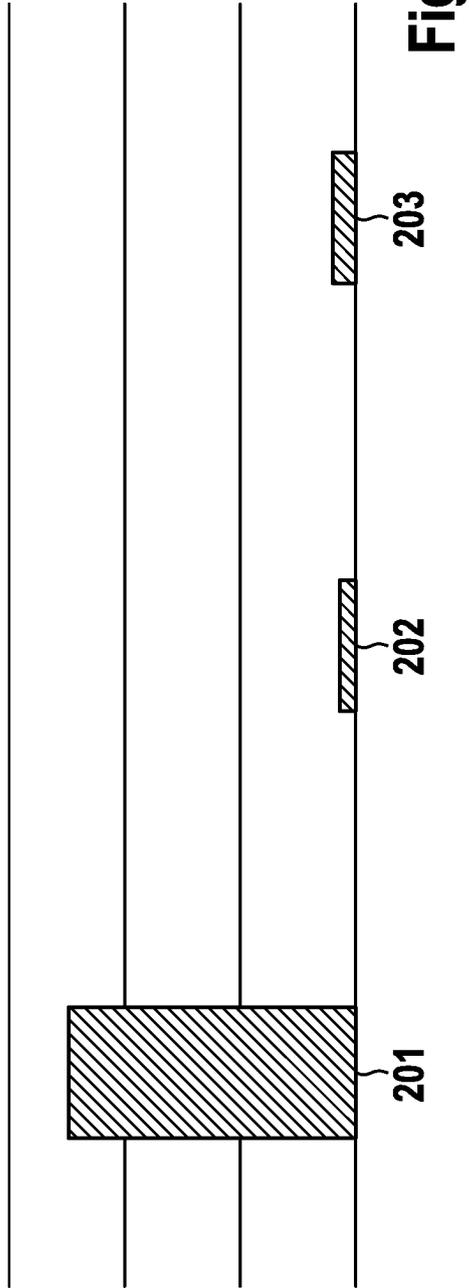


Fig. 1



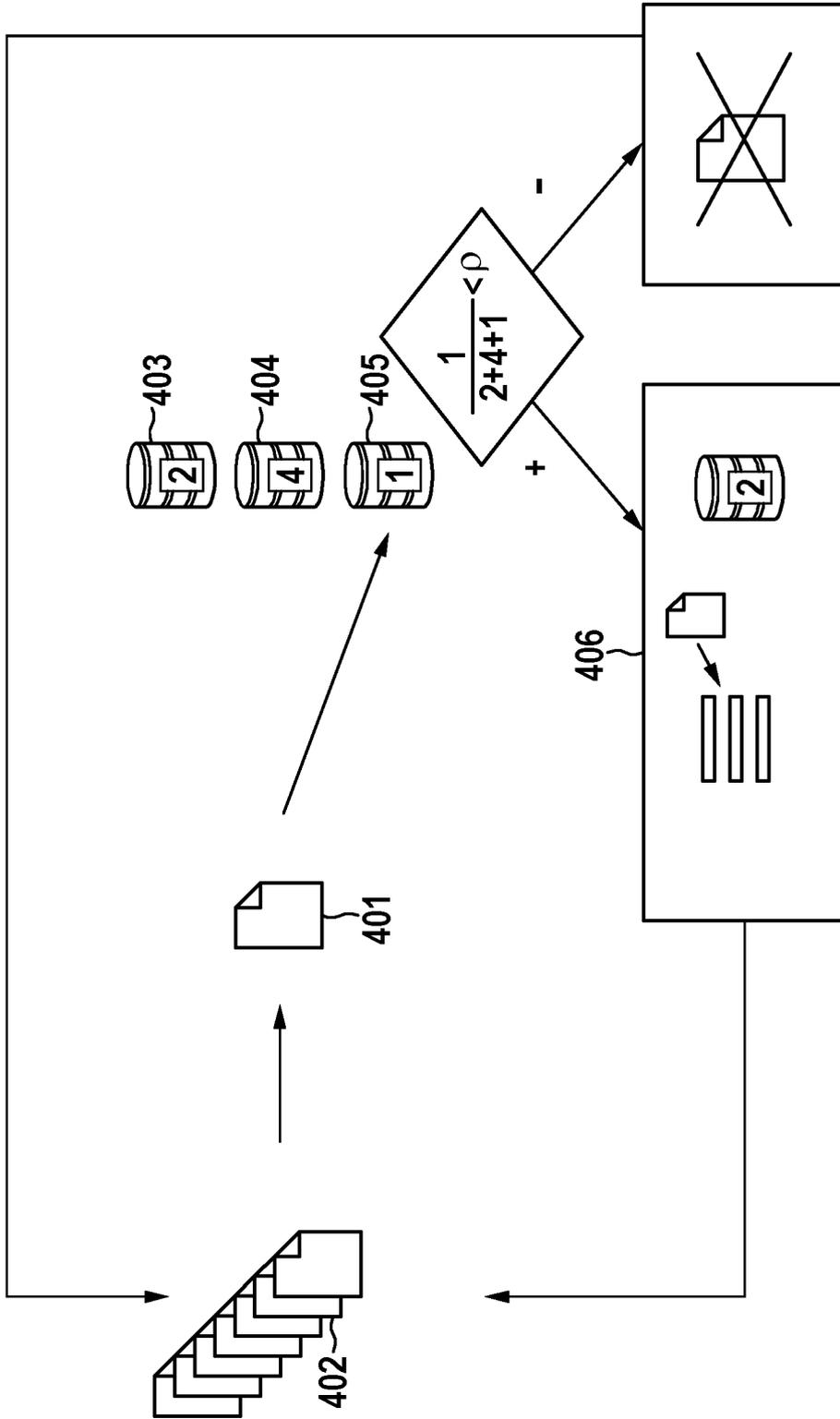


Fig. 4

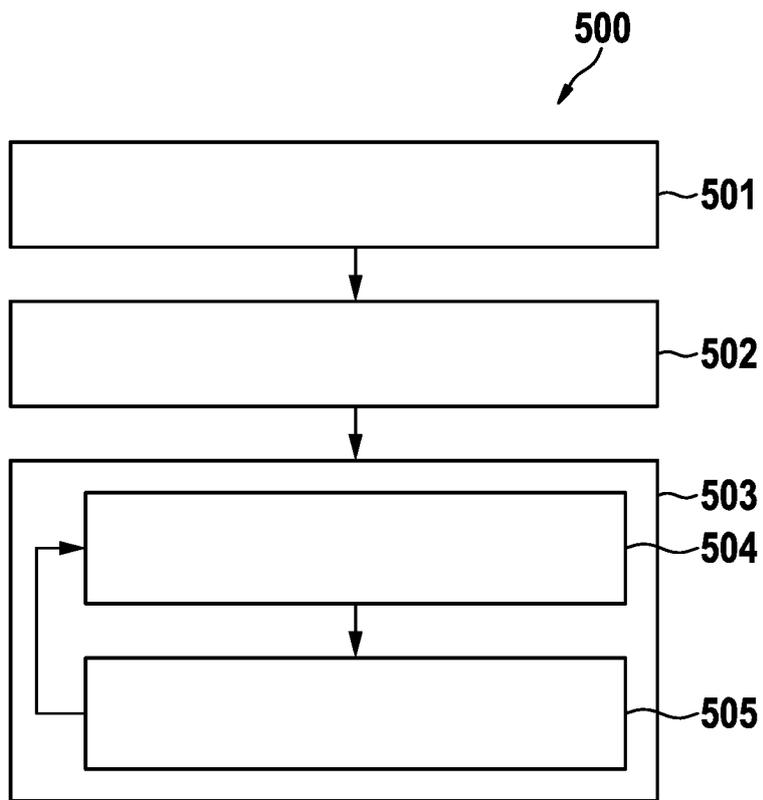


Fig. 5