



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2022 210 195.2**  
(22) Anmeldetag: **27.09.2022**  
(43) Offenlegungstag: **02.11.2023**

(51) Int Cl.: **G06N 3/04 (2023.01)**  
**G01M 17/00 (2006.01)**

(30) Unionspriorität:  
**10-2022-0052223 27.04.2022 KR**

(74) Vertreter:  
**HOFFMANN - EITLE Patent- und Rechtsanwälte  
PartmbB, 81925 München, DE**

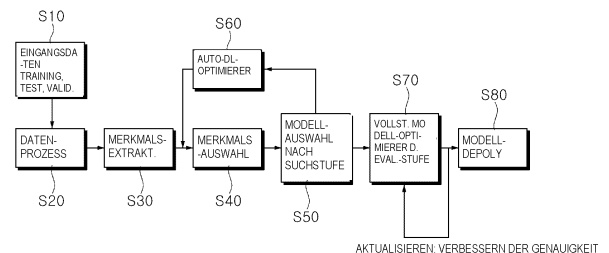
(71) Anmelder:  
**Hyundai Motor Company, Seoul, KR; IUCF-HYU  
(Industry-University Cooperation Foundation  
Hanyang University), Seoul, KR; Kia Corporation,  
Seoul, KR**

(72) Erfinder:  
**Lee, Dong-Chul, Anyang-si, Gyeonggi-do, KR;  
Jung, In-Soo, Goyang-si, Gyeonggi-do, KR; Lee,  
Joo-Hyun, Seoul, KR; Chang, Joon-Hyuk, Seoul,  
KR; Noh, Kyoung-Jin, Seoul, KR**

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Verfahren zur automatischen Erstellung eines KI-Diagnosemodells zur Diagnose eines anormalen Zustands auf der Grundlage von Geräusch- und Vibrationsdaten, auf die ENAS angewendet wird**

(57) Zusammenfassung: Ein Verfahren zum automatischen Erzeugen eines Diagnosemodells mit künstlicher Intelligenz (KI) zum Diagnostizieren eines anormalen Zustands eines Fahrzeugs beinhaltet: Erfassen von Geräusch- und Vibrationsdaten, die von einem Sensor des Fahrzeugs gemessen werden, als Eingangsdaten, Verarbeiten der Eingangsdaten, Suchen und Auswählen einer Architektur des KI-Diagnosemodells auf der Grundlage der verarbeiteten Eingangsdaten und Bereitstellen des KI-Diagnosemodells, um den anormalen Zustand des Fahrzeugs zu diagnostizieren, wobei eine effiziente neuronale Architektursuche (ENAS) angewendet wird, um das KI-Diagnosemodell und einen Parameter, der das KI-Diagnosemodell konfiguriert, zu aktualisieren, wobei die ENAS den Parameter mit dem aktualisierten KI-Diagnosemodell teilt.



## Beschreibung

### Technisches Gebiet

**[0001]** Die vorliegende Offenbarung betrifft ein Verfahren zur automatischen Erzeugung eines KI-Diagnosemodells zur Diagnose eines anormalen Zustands eines Teils, auf das eine ENAS angewendet wird, auf der Grundlage von Geräusch- und Vibrationsdaten unter Verwendung der ENAS, um das KI-Diagnosemodell automatisch zu entwerfen.

### Hintergrund

**[0002]** Bisher wurde versucht, bessere Ergebnisse zu erzielen, indem die auf der Grundlage des Sensors erfassten Daten, wie z. B. ein bestimmter Sensorwert oder ein Schwellenwert eines Signals, analysiert wurden, um einen anormalen Zustand eines zu beobachtenden Systems oder mechanischen Geräts zu analysieren, und KI angewendet wurde. Wenn jedoch ein auf KI basierendes Diagnosemodell verwendet wird, ist der Grund, warum hauptsächlich ein Diagnosealgorithmus, der sich auf das Know-how des Entwicklers stützt, angewandt wird, der, dass die Diagnoseleistung nur für ein bestimmtes Problem, auf das sich das Know-how des Entwicklers konzentriert, und nicht für das gesamte Problem ausgeübt und angewendet wurde. Dementsprechend kann es bei der wiederholten Erfassung und Analyse von Daten, die den Entwicklern vertraut sind, zu einem Problem der Überanpassung (over-fitting) der Daten kommen.

**[0003]** Wenn eine neue Problemsituation auftritt, muss entsprechend überprüft werden, ob es sich bei dem Modell um ein optimiertes Modell handelt, nachdem ein neues Modell konfiguriert wird, und daher kann ein enormer Rechenaufwand erforderlich sein. Entsprechend, wenn eine Reihe von Prozessen automatisiert wird, besteht ein Bedarf an einer Technik zur Erstellung eines optimierten Modells, ohne auf das Know-how des Entwicklers angewiesen zu sein.

## ZUSAMMENFASSUNG

**[0004]** Ein Ziel der vorliegenden Offenbarung ist es, ein Verfahren zur automatischen Erstellung eines KI-Diagnosemodells zur Diagnose eines anormalen Zustands auf der Grundlage von Geräusch- und Vibrationsdaten bereitzustellen, wobei eine KI-Diagnosemodell- (Deep Learning Model) Automatik-Optimierungs-Technologie, auf die eine effiziente neuronale Architektursuche (ENAS) angewendet wird, und ein „Framework“-Werkzeug verwendet werden.

**[0005]** Um das Ziel zu erreichen, wird ein Verfahren zur automatischen Erstellung eines KI-Diagnosemo-

dells zur Diagnose eines abnormalen Zustands auf der Grundlage von Geräusch- und Vibrationsdaten bereitgestellt, auf die eine ENAS gemäß einer Ausführungsform der vorliegenden Offenbarung angewendet wird, wobei das Verfahren Folgendes umfasst: Erfassen der Geräusch- und Vibrationsdaten als Eingangsdaten von einem Sensor eines Fahrzeugs (S10), Verarbeiten der Eingangsdaten (S20), Extrahieren eines Merkmals (S30), Auswählen einer Kombination von Merkmalen, die für das KI-Diagnosemodell geeignet sind, aus dem extrahierten Merkmal (S40), Suchen und Auswählen einer Architektur des KI-Diagnosemodells (S50), und Optimieren der Architektur des KI-Diagnosemodells (S60), wobei beim Suchen und Auswählen (S50) und Optimieren (S60) der Architektur des KI-Diagnosemodells, wenn das zuerst berechnete KI-Diagnosemodell und ein Parameter, der das KI-Diagnosemodell konfiguriert, auf einer effizienten neuronalen Architektursuche (ENAS) basieren und das KI-Diagnosemodell aktualisiert wird, der Parameter gemeinsam genutzt wird.

**[0006]** Es wird gemäß einer anderen Ausführungsform der vorliegenden Offenbarung ein Verfahren zur automatischen Erstellung eines KI-Diagnosemodells zur Diagnose eines abnormalen Zustands auf der Grundlage von Geräusch- und Vibrationsdaten bereitgestellt, auf die eine ENAS angewendet wird, wobei das Verfahren Folgendes umfasst: Erfassen der Geräusch- und Vibrationsdaten eines Fahrzeugs als Eingangsdaten durch einen Sensor (S10), Verarbeiten der Eingangsdaten (S20), Extrahieren eines Merkmals aus den verarbeiteten Eingangsdaten (S30), Auswählen einer Kombination von Merkmalen, die für das KI-Diagnosemodell geeignet sind, aus dem extrahierten Merkmal (S40), Suchen und Auswählen einer Architektur des KI-Diagnosemodells (S50), Optimieren der Architektur des KI-Diagnosemodells (S60), Validieren des KI-Diagnosemodells, das einen Trainingsprozess beendet, wenn eine Änderungsrate der Genauigkeit auf ein bestimmtes Niveau oder darunter konvergiert, selbst wenn die Genauigkeit höher als ein bestimmtes Niveau ist, und eine größere Anzahl von Schichten entsprechend einer Änderung in der Tiefe des KI-Diagnosemodells hinzugefügt werden (S70), und Bereitstellen des KI-Diagnosemodells, um den abnormalen Zustand des Fahrzeugs zu diagnostizieren, wobei, wenn das KI-Diagnosemodell zuerst berechnet wird und ein Parameter, der das KI-Diagnosemodell konfiguriert, auf einer effizienten neuronalen Architektursuche (ENAS) basiert und das KI-Diagnosemodell aktualisiert wird, der Parameter gemeinsam genutzt wird.

**[0007]** Das Verfahren zur automatischen Erzeugung des KI-Diagnosemodells zur Diagnose eines abnormalen Zustands auf Basis von Geräusch- und Vibrationsdaten, auf die eine ENAS gemäß der vorliegenden Offenbarung angewendet wird, kann das Deep-

Learning-Modell mit hoher Robustheit automatisch erzeugen, indem die Leistung des objektiven Generierungsmodells optimiert wird, ohne sich auf das Know-how des KI-Entwicklers zu verlassen.

**[0008]** Darüber hinaus ist es möglich, die Technologie der effizienten neuronalen Architektursuche (ENAS), die in der Lage ist, die GPU mit der kleinen Kapazität zu haben und das Modell im Vergleich zur Technik der neuronalen Architektursuche (NAS) schnell zu erzeugen, auf das Gebiet der Diagnose des anormalen Zustands von Fahrzeugen auszuweiten.

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

**Fig. 1** ist eine vollständige Blockansicht der vorliegenden Offenbarung.

**Fig. 2** ist eine Blockansicht einer automatischen Merkmalslernstufe und einer Stufe, auf die eine ENAS angewendet wird.

**Fig. 3** ist eine Blockansicht, die eine Modellauswahlstufe (S50) konfiguriert.

**Fig. 4** ist eine konzeptionelle Ansicht einer Bühne, auf welche die ENAS angewendet wird.

#### AUSFÜHRLICHE BESCHREIBUNG

**[0009]** Ein spezifischer Inhalt zur Durchführung der vorliegenden Offenbarung wird unter Bezugnahme auf die Zeichnungen beschrieben.

**[0010]** Die Suche nach einer neuronalen Architektur als eine Art des automatischen maschinellen Lernens (AutoML), das sich auf eine automatisierte Modellierung des maschinellen Lernens bezieht, ist ein automatisiertes Werkzeug sowie eine automatische Generierungsmethodik, die eine neuronale Architektur optimiert, die ein KI-Modell für maschinelles Lernen konfiguriert, und ist eine Technik der Suche nach einer optimalen künstlichen neuronalen Architektur durch Training eines neuronalen Netzwerks, das durch ein rekurrentes neuronales Netzwerk (RNN) abgeleitet wird.

**[0011]** Die neuronale Architektursuche kann nach der optimalen künstlichen neuronalen Architektur suchen, indem sie das neuronale Netzwerk trainiert, das durch das rekurrente neuronale Netzwerk abgeleitet wurde. Der RNN-Controller kann dazu dienen, Kandidaten für die neuronale Architektur zu erstellen, die neuronale Architektur zu trainieren und die Leistung zu messen. Die Messergebnisse können helfen, eine bessere neuronale Architektur zu finden. Der RNN-Controller kann die neuronale Architektur in die Lage versetzen, durch Training zu einem bestimmten Modell unter den Kandidaten für die neuronale Architektur zu konvergieren, und in diesem Prozess wird die Genauigkeit der neuronalen Archi-

tektur einer Gruppe von Kandidaten für die neuronale Architektur als Belohnungssignal verwendet.

**[0012]** In einigen Beispielen entfernt die vom RNN-Controller erzeugte Kandidatengruppe der neuronalen Architektur - dies wird als Kindmodell bezeichnet - alle trainierten Gewichte, wodurch sich der Rechenaufwand erheblich erhöht, da jedes Mal ein neues Training durchgeführt wird, wenn das Modell neu erzeugt wird.

**[0013]** Die Suche nach der neuronalen Architektur kann eine unbegrenzte Modellarchitektur erstellen und abschließen, um sie zu trainieren und dann alle Parameter zu initialisieren. Dementsprechend kann die für das Training des Modells benötigte Zeit exponentiell ansteigen, wodurch sich die Wahrscheinlichkeit der Bestimmung der Genauigkeit des Modells verringert, bis die endgültige Leistung des Modells überprüft wird.

**[0014]** Die ENAS kann sich auf eine effiziente neuronale Architektursuche beziehen. Die ENAS kann durch die Suche nach einer Architekturkombination für eine bestimmte Modelltiefe und die gemeinsame Nutzung von Parametern jeder Modellarchitektur von einem Ausgangsmodell bis zu einem anschließend berechneten Modell gekennzeichnet sein. So kann beispielsweise ein Klassifizierungsmodell für Geräusch- und Vibrationsdaten, das auf der ENAS basiert, innerhalb einer bestimmten Modelltiefe optimiert werden, und die Entwicklungszeit kann verkürzt und die Leistung leicht überprüft werden, durch eine Technik, die nur den besten Algorithmus aufweist, und Optimieren jeden Parameters.

**[0015]** Die ENAS und das verstärkte Training können bei der Automatisierung des KI-Diagnosemodells zur Diagnose des anormalen Zustands auf der Grundlage von Geräusch- und Vibrationsdaten eines Fahrzeugs, die von einem Sensor erfasst wurden, angewendet werden.

**[0016]** **Fig. 1** ist eine vollständige Blockansicht der vorliegenden Offenbarung, auf die die ENAS angewendet wird.

**[0017]** Eine Stufe (S10) kann als Eingangsdaten, die vom Sensor erfasst werden, Daten enthalten, die in Trainingsdaten, Testdaten und Validierungsdaten klassifiziert sind. Geräusch- und Vibrationsdaten können mit einem Sensor außerhalb des Fahrzeugs oder durch Installation eines Sensors innerhalb des Fahrzeugs gemessen oder gesammelt werden, und die gesammelten Daten können in einer separaten Speichervorrichtung oder auf einem externen Server gespeichert und dann auch abgerufen werden, wenn das Diagnosemodell trainiert wird.

**[0018]** Die Trainingsdaten können für das Training des Modells verwendet werden. Die Validierungsdaten können zur Überprüfung der Leistung in der Mitte des Trainings des Modells und zur Aktualisierung des Modells zusammen mit den Trainingsdaten verwendet werden. Die Testdaten können für die Validierung des konstruierten KI-Diagnosemodells verwendet werden.

**[0019]** S20 repräsentiert einen Datenverarbeitungsprozess, und ein Datensatz wird als Datenvorformatierungsstufe bestimmt. Der in der vorliegenden Offenbarung verwendete Datensatz ist ein Geräusch- oder Vibrationsdatensatz und umfasst beispielsweise in einem Fahrzeug gemessene Geräusch- oder Vibrationsdaten (dB) über die Zeit (t).

**[0020]** In der Stufe (S20) wird die Eignung der Daten dahingehend bestimmt, ob es sich um qualitativ hochwertige Daten mit geringer Störung unter den nach Typ gesammelten Lärm- oder Vibrationsdaten handelt. Eine Abtastrate wird angepasst, und zu diesem Zweck kann eine erneute Abtastung (Resampling) durchgeführt werden. Ein Hoch-/Tief-/Bandpassfilter kann auch selektiv als Frequenzfilter angewendet werden.

**[0021]** Die für die Datenverarbeitung verwendeten Algorithmen können selektiv eingesetzt werden, aus einem Crop, der Rauschen durch visuelle Inspektion entfernt und die Länge zwischen den Daten bei der Dateneingabe vereinheitlicht, einem Resampling, das die gesamte Datenabtastrate vereinheitlicht, einer harmonischen/perkussiven Schallseparation (HPSS), die Hoch-/Tief-/Bandpassfilter trennt und extrahiert, harmonische, perkussive Wellenformkomponenten, die spezifische Frequenzbänder entfernt oder extrahiert, einer Normalisierung, die automatisch eine Skalierung der Datenwerte vornimmt, einer Ausreißererkennung, die hauptsächlich in einem CAN verwendet wird und Ausreißer erkennt und entfernt, und einer PCA, die die Dimensionen reduziert.

**[0022]** Durch die Stufen (S10 und S20) kann eine Vorbereitung für die Durchführung des ENAS-Frameworks abgeschlossen werden.

**[0023]** S30 ist eine Stufe zur Merkmalsextraktion, und es kann eine oder eine Kombination verschiedener Filtertechniken und Signalverarbeitungsmethoden zur Extraktion des Merkmals ausgewählt werden, die zur Extraktion der Merkmale zu verwenden sind. Zum Beispiel können Techniken wie FFT, Mel-Spektrogramm und HPSS verwendet werden. Ein Größenwert eines wichtigen Frequenzbandes eines Zielgeräusches kann in eine dB-Skala transformiert und als Merkmalsvektor unter Verwendung der schnellen Fourier-Transformation (FFT) verwendet

werden. Das Mel-Spektrogramm kann als Merkmalsvektor das Spektrogramm verwenden, bei dem das FFT-transformierte Spektrogramm durch Anwendung einer Mel-Filterbank auf eine Frequenzachse in einer Mel-Einheit transformiert wird. Die harmonisch-perkussive Quellenseparation (HPSS) trennt die harmonischen und perkussiven Komponenten auf der Frequenzachse in Bezug auf das Spektrogramm nach der FFT, trennt dann eine H-Komponente durch Anwendung eines horizontalen Medianfilters entlang der Frequenzachse und trennt eine P-Komponente durch Anwendung eines vertikalen Medianfilters entlang einer Zeitachse. Eine binäre Maske kann durch Anwendung eines Schwellenwertes auf eine H/P- oder P/H-Rate erstellt werden, und ein STFT-Koeffizient eines Eingangssignals und die binäre Maske können einer elementweisen Multiplikation unterzogen werden, um schließlich die H- und P-Komponenten zu trennen. Wie oben beschrieben, werden drei Merkmalsextraktionstechniken angewandt, und von diesen können ein, zwei oder mehr Merkmale auf das Modell angewendet werden.

**[0024]** S40 ist eine Merkmalsauswahlstufe und wählt eine Kombination der für die Modellierung geeigneten Merkmale aus, um das ausgewählte Merkmal in die ENAS-Modellierung einfließen zu lassen. Es wird eine Kombination mit dem besten Belohnungssignal (Genauigkeit) gesucht und bei Durchführung jeder Epoche des Trainingsdatensatzes reflektiert.

**[0025]** S50 ist eine Stufe, die dazu dient, Strukturen einer normalen Zelle und einer Reduktionszelle festzulegen, die ein Einheitsmodell im RNN-Controller darstellen, der als Agent der ENAS dient. Das Einheitsmodell kann sich auf ein Modell beziehen, das aus einem Paar aus normaler Zelle und Reduktionszelle besteht, und dies ist die Grundlage für ein vollständiges Modell von S70, das auf mehrere Schichten erweitert ist.

**[0026]** In S50 wird das Deep Learning-Modell auf der Grundlage der effizienten neuronalen Architektursuche (ENAS) optimiert. Bei der Suche und Auswahl der Struktur für die Konstruktion des Modells innerhalb des Bereichs, in dem die Rechenleistung eines Servers, auf dem die Berechnung durchgeführt wird, möglich ist, wird die effiziente neuronale Architektursuche (ENAS) angewandt, bei der es sich um eine effiziente Modellsuchtechnik durch gemeinsame Nutzung der Modellparameter handelt.

**[0027]** S60 ist ein Prozess zur Erhöhung der Genauigkeit durch die Optimierung des Einheitsmodells, das aus der normalen Zelle und der Reduktionszelle besteht, indem die Einstellung der Parameter des vom RNN-Controller erzeugten Modells und eines Hyperparameters aktualisiert wird.

**[0028]** Durch wiederholte Durchführung der Stufen (S40, S50 und S60) kann das optimale Einheitsmodell automatisch erzeugt werden, und wenn das Modell, bei dem die Genauigkeit konvergiert, gefunden ist, kann das Einheitsmodell durch frühzeitiges Anhalten beendet werden.

**[0029]** S70 erzeugt das vollständige Modell mit einer tiefen Modellarchitekturschicht unter Verwendung des oben beschriebenen Einheitsmodells in S50, und zu diesem Zeitpunkt wird die Genauigkeit mit einem automatisierten Bereich verbessert, in dem die Reihenfolge und Anzahl der normalen Zellen und der Reduktionszellen mit einer Gittersuche optimiert werden. In Prozess von S70 werden alle im Einheitsmodell aktualisierten Parameter initialisiert und nur die Architektur erzeugt, und der Parameter wird durch erneutes Training des vollständigen Modells aktualisiert und optimiert, um die Genauigkeit zu verbessern. Beim Prozess von S70 wird das vollständige Modell kombiniert, indem alle im Einheitsmodell aktualisierten Parameter initialisiert werden und nur die Architektur erzeugt wird, wird der initialisierte Parameter aktualisiert, um das vollständige Modell erneut zu optimieren, um die Genauigkeit zu verbessern, und wenn eine bestimmte Genauigkeit erreicht ist, wird mit Stufe S80 fortgefahren.

**[0030]** S80 ist eine Stufe, in der das endgültige Diagnosemodell der ENAS in Form einer API bereitgestellt wird, in der Codes so implementiert sind, dass die Berechnung und Ausführung auf dem Server erfolgt, oder in Form einer Datei für jedes Gerät in Form einer Ausführungsdatei gespeichert wird, z. B. in der Form (Android, C++, C-Sprache usw.), die für Geräteumgebungen geeignet ist, die in einem Gerät des Benutzers verwendet werden sollen.

**[0031]** Fig. 2 zeigt die Merkmalsauswahlstufe (S40), die verfügbare Merkmale aus der Merkmalsextraktionsstufe (S30) kombiniert, auf die die ENAS im automatisierten Deep-Learning-Modellierungsprozess angewendet wird. Die Merkmalsextraktionsstufe (S30) und die Merkmalsauswahlstufe (S40) sind eine automatisierte Merkmalslernstufe. Ein KI-Diagnosemodell wird durch die Verwendung von Trainingsdaten aus den Eingangsdaten erstellt, und die obige Stufe ist ein Prozess der Suche nach einer Kombination von Merkmalen, in der die Klassifikationskategorie-Unterscheidung zwischen Merkmalen durch die Trainingsdaten gut ausgedrückt wird, insbesondere unter Verwendung einiger aller Trainingsdaten.

**[0032]** Fig. 2 zeigt eine Stufe der Suche nach dem und Auswahl des Einheitsmodells (S50) durch die ausgewählten Merkmale und eine Stufe der Suche nach einer neutralen Architektur aus dem Einheitsmodell von S50 und der Bewertung des erzeugten

Modells durch die neuronale Architektursuche (S70), um das beste Modell zu finden.

**[0033]** Die in Fig. 2 gezeigte Stufe der Suche und Auswahl des Einheitsmodells (S50) ist eine Stufe der Suche und Auswahl der Architektur des Einheitsmodells (normale Zelle/Reduktionszelle) mit hervorragender Leistung durch einen ENAS-Algorithmus, und die Stufe der Auswahl des vollständigen Modells (S70) ist als Auswahl des vollständigen Modells, das das beste Modell ist, aus dem Einheitsmodell (S70) dargestellt.

**[0034]** Die Stufe der Suche und Auswahl des Einheitsmodells in S50 ist ein Prozess der Erhöhung der Genauigkeit durch Aktualisierung der Parameter des Modells, das vom RNN-Controller unter Verwendung der Trainingsdaten festgelegt wurde, und die Genauigkeit wird mit den Validierungsdaten aus den Eingangsdaten bestätigt, und dieser Wert wird als das Belohnungssignal ausgewählt, um das Training in einer Richtung durchzuführen, in der das Belohnungssignal besser (verbessert) wird, und nach der Architektur des Einheitsmodells zu suchen. Eine tiefe Modellarchitekturschicht wird unter Verwendung des gesuchten Einheitsmodells erzeugt, und zu diesem Zeitpunkt werden als eine automatisierte Region, in der Reihenfolge und Anzahl normaler Zellen und von Reduktionszellen unter Verwendung einer Gittersuche optimiert sind, in dieser Stufe alle aktualisierten Parameter des Einheitsmodells initialisiert und nur wird die Architektur erzeugt, um die Parameter durch erneutes Training des vollständigen Modells zu aktualisieren und das Modell zu optimieren.

**[0035]** Fig. 3 ist eine konzeptionelle Ansicht, die zeigt, dass die Stufe der Suche und Auswahl des Modells (S50), auf das die ENAS angewendet wird, weiter unterteilt ist. Die Stufe der Auswahl des Modells in Fig. 3 ist ein Prozess der Suche und des Trainings der Modellarchitektur und wird in eine Parameterabstimmungs-Stufe (S50-A) und eine Controller-Trainings-Stufe (S50-B) unterteilt. Der Prozess der Suche und des Trainings der Modellarchitektur zielt auf 1 Epoche, und 1 Epoche bedeutet die gesamten Trainingsdaten.

**[0036]** Die Parameterabstimmungsstufe (S50-A) besteht aus einer Stufe der Erstellung eines Proxy-Modells in einer Umgebung durch Erzeugen eines Architektur-Strings durch den RNN-Controller, der ein Agent ist (S51), einer Stufe der Übertragung der Trainingsdaten von den Eingangsdaten an das Proxy-Modell mit einem Mini-Batch (S52) und einer Stufe der Aktualisierung der Parameter in dem Proxy-Modell (S53).

**[0037]** Mit anderen Worten ist das Proxy-Modell der Umgebung ein Trainingsmodus, und der RNN-Controller ist als Agent auf einen Validierungsmodus ein-

gestellt. Der RNN-Controller dient dazu, den Architektur-String durch eine Kombination von Operationen (arithmetische Operationen oder Berechnungen) und Datenfluss für jeden Mini-Batch für die Trainingsdaten unter den Eingangsdaten zu erfassen. Beim Deep Learning bezeichnet ein Batch ein Bündel von Stichproben, die zur einmaligen Aktualisierung des Gewichts des Modells verwendet werden. Im Vergleich zu einem Batch, bei dem die gesamten Daten trainiert werden, führt der Mini-Batch das Training mit einer Methode durch, welche die gesamten Daten  $N$  fach aufteilt, um alle Trainingsdaten zu disponieren, und der Mini-Batch kann die Zeit im Vergleich zum Batch reduzieren.

**[0038]** Die Modellarchitektur wird geändert, indem der abgetastete Architektur-String an das Proxy-Modell der Umgebung übertragen wird, und Daten der Mini-Batch-Größe werden in das geänderte Proxy-Modell eingegeben.

**[0039]** Da der RNN-Controller die Zufallsmodellarchitektur zu Beginn der Parameterabstimmungstufe (S50-A) abtastet, sollten die meisten Parameterwerte des Proxy-Modells abgestimmt werden. Im weiteren Verlauf der Suche konvergiert die Ausgabe des RNN-Controllers allmählich zu einer Form, und nur die häufig verwendeten Parameterwerte des Proxy-Modells werden aktualisiert.

**[0040]** Die Controller-Trainingsstufe (S50-B) besteht aus einer Stufe der Übertragung des abgetasteten Architekturstrings an das Proxy-Modell der Umgebung durch den RNN-Controller, der ein Agent ist (S55), einer Stufe der Übertragung der Validierungsdaten unter den Eingangsdaten an das Proxy-Modell mit dem Mini-Batch (S56), einer Stufe der Messung der Genauigkeit in dem Proxy-Modell mit geänderter Architektur (S57) und einer Stufe der Aktualisierung der Parameter unter Verwendung der gemessenen Genauigkeit als Belohnung für das verstärkte Training durch den RNN-Controller (S58). Mit anderen Worten, in der Controller-Trainingsstufe (S50-B) ist das Proxy-Modell der Validierungsmodus, und der RNN-Controller ist auf den Trainingsmodus eingestellt. Der RNN-Controller ändert die Architektur des Proxy-Modells, indem er den abgetasteten Architektur-String an das Proxy-Modell überträgt, dessen Parameter bis zu einem gewissen Grad optimiert werden. Die Genauigkeit innerhalb des Mini-Batch wird ausgegeben, indem die Validierungsdaten unter den Eingangsdaten in das geänderte Proxy-Modell mit dem Mini-Batch eingegeben werden. Die Genauigkeit kann für jede geänderte Architektur des Proxy-Modells gemessen werden, die Parameterwerte werden durch das verstärkte Training aktualisiert, das Belohnungen zur Erhöhung der gemessenen Genauigkeit durchführt, und der RNN-Controller wird trainiert.

**[0041]** Fig. 4 zeigt die Stufe der Suche nach der Architektur des Einheitsmodells (normale Zelle/Reduktionszelle) mit hervorragender Leistung durch den ENAS-Algorithmus in der Stufe der Suche und Auswahl des in Fig. 2 gezeigten Einheitsmodells (S50), und die Konfiguration des vollständigen Modells, das das beste Modell ist, aus dem Einheitsmodell (S70).

**[0042]** Die Architektur des vollständigen Modells in der Stufe der Suche und Auswahl des Modells (S50) besteht aus einer geringeren Anzahl von Schichten als die Modellarchitektur in der Stufe der Validierung des Modells (S70). Ein Parameter in jeder Operation (Edge computing) verbraucht Speicher. Daher wird das Modell in der Stufe des Suchens und Auswählens des Modells (S50) so konfiguriert, dass die Anzahl der Schichten reduziert wird, und nachdem die Stufe des Suchens nach dem Modell abgeschlossen ist, wird das Modell in der Stufe der Validierung des Modells (S70) durch eine Anzahl von gefundenen Zellarchitekturen trainiert, die größer ist als die Anzahl der Schichten in der Stufe des Suchens nach dem Modell.

**[0043]** Eine Technik, die in der Stufe der Validierung des Modells (S70) angewendet wird, kann eine Änderung der Tiefe des Einheitsmodells auswählen, indem eine Änderung der Lernrate und eine Gittersuchtechnik angewendet werden. Die Genauigkeit ist höher als ein bestimmtes Niveau, und wenn eine Änderungsrate der Genauigkeit auf ein bestimmtes Niveau oder weniger konvergiert, selbst wenn mehr Schichten in Abhängigkeit von einer Änderung in der Tiefe des Modells konfiguriert werden, wird der Trainingsprozess beendet.

**[0044]** In der Stufe der Validierung des Modells (S70) wird das vollständige Modell konfiguriert, indem der Parameter abgestimmt wird, während die  $N$  normalen Zellen und eine Reduktionszelle  $M$ -mal ( $M$ ,  $N$  sind eine natürliche Zahl) aus dem Einheitsmodell wiederholt werden, d.h. die normale Zelle und die Reduktionszelle, die in der Stufe der Suche und Auswahl des Modells (S50) gefunden wurden.

**[0045]** Deep Learning-Generalisierungstechniken wie eine Datenerweiterung, ein Cosinus-Annealing-Schema und ein Hilfskopf werden verwendet, um die Leistung des Modells zu maximieren. Der Trend der Änderung bei der Genauigkeit kann gesucht werden, und die Genauigkeit ist höher als ein bestimmtes Niveau, und wenn die Änderungsrate der Genauigkeit auf ein bestimmtes Niveau oder weniger konvergiert, selbst wenn mehr Schichten in Abhängigkeit von einer Änderung der Tiefe des Modells konfiguriert werden, kann der Trainingsprozess beendet werden, wodurch die Betriebszeit verkürzt wird.

**[0046]** Gemäß der vorliegenden Offenlegung ist die ENAS so rekonfiguriert, dass es für Geräusch- und Diagnoseaufgaben auf der Grundlage des „Inception“-Moduls von „Google Net“ als architektonisches Merkmal der Schicht des Modells optimiert wird. Die vorliegende Offenbarung zielt darauf ab, die Betriebskombination des Inception-Moduls unter Berücksichtigung der Eigenschaften der Fahrzeugdomäne und der Eigenschaften von Geräusch- und Vibrationssignalen zu finden.

**[0047]** Um eine Schicht zu konfigurieren, wird eine optimale Kombination von N verschiedenen Arten von Operationen gebildet, um eine Schicht zu bilden. Es gibt viele Arten von Operationen wie Identität, 3X3-Faltung, 5X5-Faltung, Durchschnitts-Pooling und Max-Pooling.

**[0048]** Gemäß der vorliegenden Offenbarung wird das Modell in der Verlustfunktion durch den Kreuzentropieverlust unter Verwendung der stochastischen Gradientenabweichung trainiert. Bei komplexen Aufgaben wird eine Kombination aus dem mittleren quadratischen Fehler (MSE), dem mittleren Wurzelquadrat-Fehler (RMSE), der binären Kreuzentropie, der kategorialen Kreuzentropie und den spärlichen kategorialen Kreuzentropie-Verlustfunktionen verwendet. Darüber hinaus ist es auch möglich, die Verlustfunktion zu verwenden, auf die das Gewicht angewendet wird, indem die Ensemble-Methode angewendet wird.

### Patentansprüche

1. Verfahren zur automatischen Erzeugung eines Diagnosemodells mit künstlicher Intelligenz (KI) zur Diagnose eines anormalen Zustands eines Fahrzeugs, wobei das Verfahren umfasst:  
Erfassung von Geräusch- und Vibrationsdaten, die von einem Sensor des Fahrzeugs gemessen werden, als Eingangsdaten;  
Verarbeitung der Eingangsdaten;  
Suchen und Auswählen einer Architektur des KI-Diagnosemodells auf Basis der verarbeiteten Eingangsdaten; und  
Bereitstellung des KI-Diagnosemodells zur Diagnose des anormalen Zustands des Fahrzeugs, wobei eine effiziente neuronale Architektursuche (ENAS) angewendet wird, um das KI-Diagnosemodell und einen Parameter, der das KI-Diagnosemodell konfiguriert, zu aktualisieren, wobei die ENAS den Parameter mit dem aktualisierten KI-Diagnosemodell teilt.

2. Verfahren nach Anspruch 1, wobei das Suchen und Auswählen der Architektur des KI-Diagnosemodells Parameterabstimmung und Controllertraining beinhaltet.

3. Verfahren nach Anspruch 2, wobei die Parameterabstimmung beinhaltet:  
Erzeugung eines abgetasteten Architekturstrings, um den erzeugten Architekturstring durch einen rekurrenten neuronalen Netzwerk-Controller (RNN) an ein Proxy-Modell zu übertragen.

4. Verfahren nach Anspruch 3, wobei die Parameterabstimmung beinhaltet:  
Übermitteln von Trainingsdaten unter den verarbeiteten Eingangsdaten an das Proxy-Modell, wobei die Trainingsdaten in eine Vielzahl von Daten unterteilt sind.

5. Verfahren nach Anspruch 4, wobei die Parameterabstimmung beinhaltet:  
die Aktualisierung des Parameters im Proxy-Modell.

6. Verfahren nach Anspruch 2, wobei das Controllertraining beinhaltet:  
Erzeugen eines abgetasteten Architekturstrings, um den erstellten Architekturstring durch einen RNN-Controller an ein Proxy-Modell zu übertragen.

7. Verfahren nach Anspruch 6, wobei das Training des Controllers weiterhin beinhaltet:  
Übermitteln von Validierungsdaten unter den Eingangsdaten an das Proxy-Modell, wobei die Validierungsdaten in eine Vielzahl von Daten unterteilt sind.

8. Verfahren nach Anspruch 7, wobei das Controllertraining weiterhin beinhaltet:  
Messen der Genauigkeit im Proxy-Modell mit einer anderen Architektur des KI-Diagnosemodells.

9. Verfahren nach Anspruch 8, wobei das Controllertraining weiterhin beinhaltet:  
Aktualisieren eines Wertes des Parameters unter Verwendung eines verstärkten Trainings, das die gemessene Genauigkeit erhöht, indem ein Verstärkungslernen für eine Belohnung durchgeführt wird, und  
Trainieren des RNN-Controllers mit dem aktualisierten Wert des Parameters.

10. Verfahren nach Anspruch 1, wobei das Suchen und Auswählen der Architektur des KI-Diagnosemodells beinhaltet:  
Suchen nach dem KI-Diagnosemodell, das ein Einheitsmodell mit einer Normalzelle und einer Reduktionzelle sucht.

11. Verfahren nach Anspruch 10, weiter umfassend:  
Validieren des KI-Diagnosemodells, wobei, basierend auf (i) einem Genauigkeitsniveau des KI-Diagnosemodells, das größer als ein vordefiniertes Niveau ist, und (ii) einer Anzahl von Schichten, die größer oder gleich einer vordefinierten

Anzahl ist, die zu dem KI-Diagnosemodell gemäß einer Änderung in einer Tiefe des KI-Diagnosemodells hinzugefügt werden, ein Trainingsprozess beendet wird, wenn eine Änderungsrate der Genauigkeit auf ein Niveau konvergiert, das gleich oder kleiner als ein vorbestimmtes Niveau ist.

12. Verfahren nach Anspruch 11, wobei das KI-Diagnosemodell (i) als API in einem Server bereitgestellt oder (ii) in einer Datei als Benutzergeräteumgebung gespeichert wird.

13. Verfahren zur automatischen Erzeugung eines Diagnosemodells mit künstlicher Intelligenz (KI) zur Diagnose eines anormalen Zustands eines Fahrzeugs, wobei das Verfahren umfasst:  
Erfassen von Geräusch- und Vibrationsdaten, die von einem Sensor des Fahrzeugs gemessen werden, als Eingangsdaten;  
Verarbeiten der Eingangsdaten;  
Extrahieren eines oder mehrerer Merkmale aus den verarbeiteten Eingangsdaten;  
Auswählen einer für das KI-Diagnosemodell geeigneten Merkmalskombination aus dem extrahierten einen oder mehreren Merkmalen;  
Suchen und Auswählen einer Architektur des KI-Diagnosemodells auf Basis der verarbeiteten Eingangsdaten;  
Optimieren der Architektur des KI-Diagnosemodells auf Basis eines Parameters;  
Validieren des KI-Diagnosemodells, das so konfiguriert ist, dass es auf Basis (i) einer Genauigkeit des KI-Diagnosemodells, die größer als ein vordefiniertes Niveau ist, und (ii) einer Anzahl von Schichten, die größer oder gleich einer vordefinierten Anzahl ist, die dem KI-Diagnosemodell entsprechend einer Änderung in einer Tiefe des KI-Diagnosemodells hinzugefügt wird, einen Trainingsprozess beendet, wenn eine Änderungsrate der Genauigkeit auf ein Niveau konvergiert, das gleich oder kleiner als ein vorbestimmtes Niveau ist; und Bereitstellen des KI-Diagnosemodells zur Diagnose des anormalen Zustands des Fahrzeugs, wobei eine effiziente neuronale Architektursuche (ENAS) angewendet wird, um das KI-Diagnosemodell und den Parameter, der das KI-Diagnosemodell konfiguriert, zu aktualisieren, wobei die ENAS den Parameter mit dem aktualisierten KI-Diagnosemodell teilt.

14. Verfahren nach Anspruch 13, wobei das Suchen und Auswählen der Architektur des KI-Diagnosemodells die Parameterabstimmung und das Controllertraining beinhaltet.

15. Verfahren nach Anspruch 14, wobei die Parameterabstimmung beinhaltet:  
Erzeugen eines abgetasteten Architekturstrings, um den erzeugten Architekturstring durch einen rekur-

renten neuronalen Netzwerk-Controller (RNN) an ein Proxy-Modell zu senden.

16. Verfahren nach Anspruch 15, wobei die Parameterabstimmung beinhaltet:  
Senden von Trainingsdaten unter den verarbeiteten Eingangsdaten an das Proxy-Modell, wobei die Trainingsdaten in eine Vielzahl von Daten unterteilt sind.

17. Verfahren nach Anspruch 16, wobei die Parameterabstimmung beinhaltet:  
Aktualisieren des Parameters im Proxy-Modell.

18. Verfahren nach Anspruch 14, wobei das Controllertraining beinhaltet:  
Erzeugen eines abgetasteten Architekturstrings, um den erzeugten Architekturstring durch einen RNN-Controller an ein Proxy-Modell zu übertragen.

19. Verfahren nach Anspruch 18, bei dem das Controllertraining außerdem beinhaltet:  
Senden von Validierungsdaten unter den Eingangsdaten an das Proxy-Modell, wobei die Validierungsdaten in eine Vielzahl von Daten unterteilt sind.

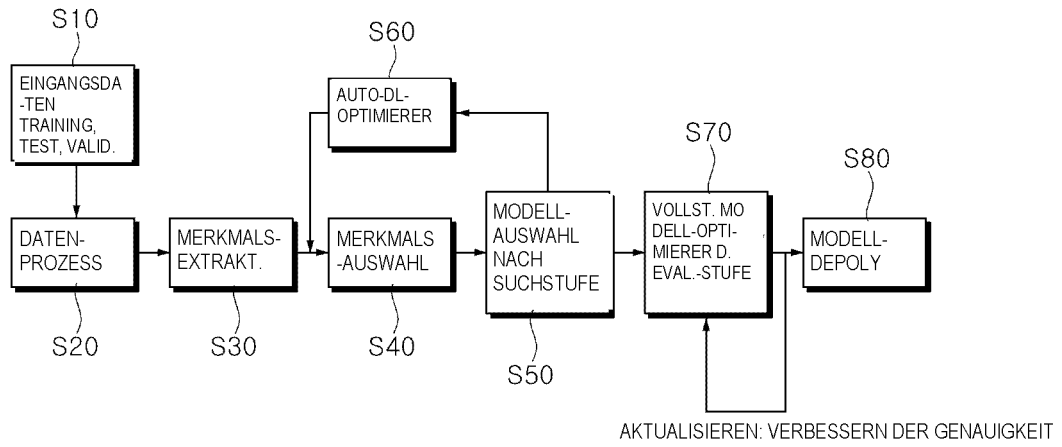
20. Verfahren nach Anspruch 13, wobei das KI-Diagnosemodell (i) als API in einem Server bereitgestellt oder (ii) in einer Datei als Benutzergeräteumgebung gespeichert wird.

Es folgen 3 Seiten Zeichnungen

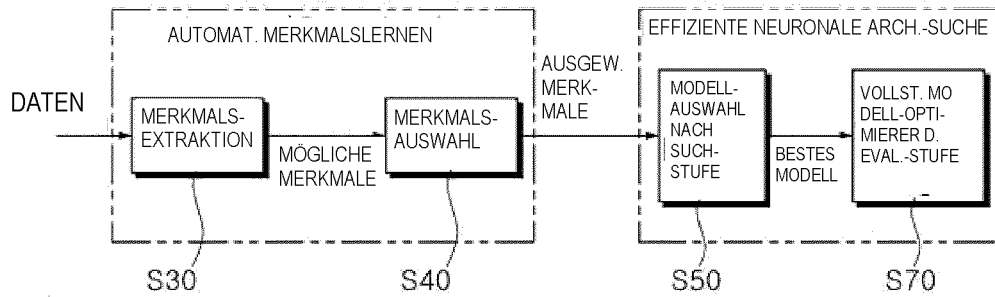


Anhängende Zeichnungen

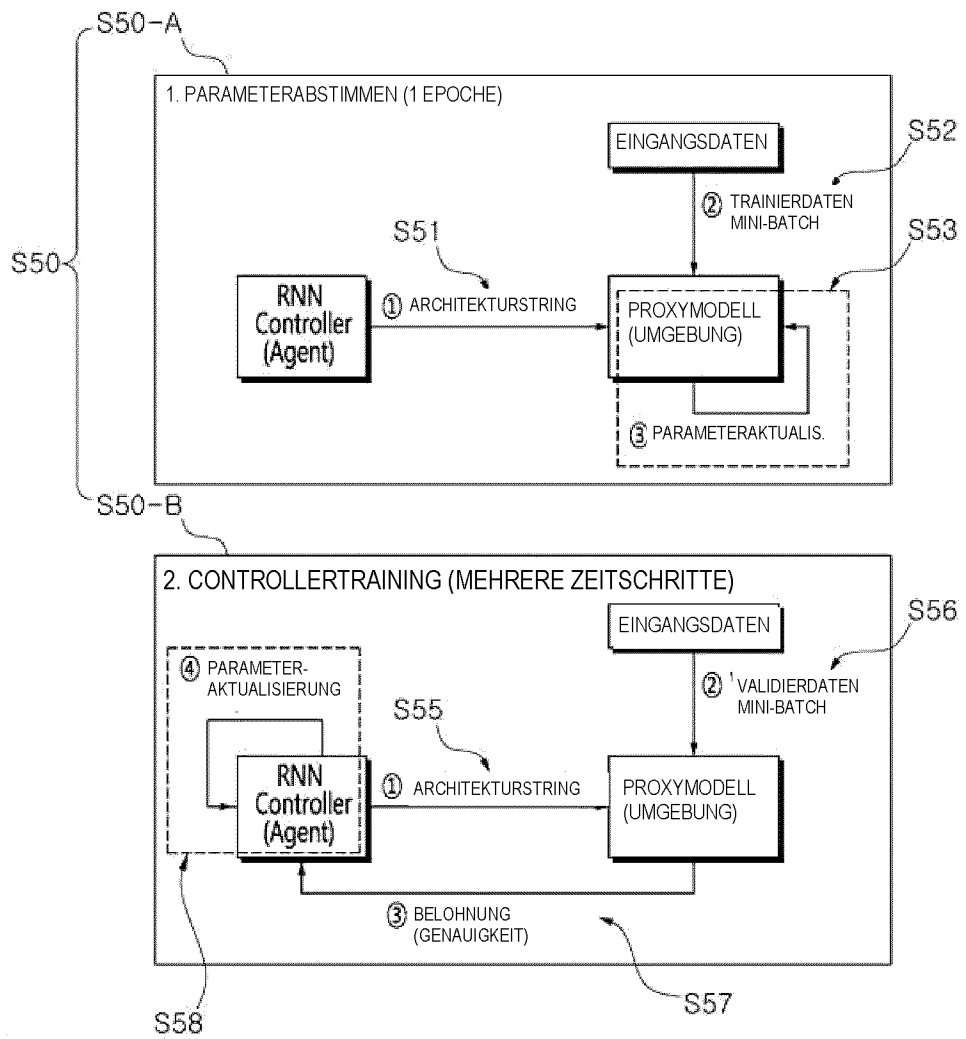
[FIG. 1]



[FIG. 2]



[FIG. 3]



[FIG. 4]

