

Neuronale Netze in der Automobilität

Die Anwendung neuronaler Netze für
autonome Fahrzeuge

Maximilian Nasert

31.09.2019

Was ist künstliche Intelligenz und wie hilft sie uns beim Autofahren? Sind neuronale Netze geeignet Autos zu steuern und welche mathematischen Prinzipien liegen ihnen zugrunde?

Inhalt

Einführung.....	2
Künstliche Intelligenz	3
<i>Was ist künstliche Intelligenz?</i>	3
<i>Welche Abstufungen sind bei künstlicher Intelligenz zu ziehen?</i>	4
Die starke künstliche Intelligenz.....	4
Die schwache künstliche Intelligenz.....	4
„Deep Learning“	7
Das „Perzeptron“	7
Das „Multilayer-Perzeptron“	9
Neuronale Netze in der Automobilität.....	12
Auswertung des Projektes.....	14
Ausblick	15
Literaturverzeichnis.....	17
Anhang	18
Grafiken	18
Interviews:.....	21

Einführung

In folgender Arbeit möchte ich mich damit beschäftigen, ob es möglich ist, Autos von neuronalen Netzen, also stochastischen Verarbeitungsmethoden im Bereich der Informatik, steuern zu lassen.

Zunächst wird dafür die Frage geklärt, was künstliche Intelligenz ist und welche Gebiete es im Bereich dieser gibt. So muss unterschieden werden, zwischen maschinellern Lernen und „Deep Learning“.

Danach wird betrachtet, warum neuronale Netze „künstliche Intelligenzen“ sind und was sie so viel performanter macht als herkömmliche Datenstrukturen und das „einfache“ maschinelle Lernen. Dafür untersuche ich, welche mathematischen Algorithmen dahinter stecken und wie man passende Methoden aussucht.

Als Praxisteil ziehe ich eine Analogie zwischen Automobilität und Robotik, als Argumentationsgrundlage dient mir hierbei ein Programm, dass ich im Rahmen der „Robotino AG“ am OSZ-Oder-Spree Palmnicken geschrieben habe. Dabei habe ich ein Perzeptron in der Entwicklungsumgebung „Robotinoview 3“ darauf trainiert, Objekten dynamisch auszuweichen, indem Sensordaten gewichtet werden und indirekt Einfluss auf die verschiedenen Motoren des Robotinos- also des Roboters bzw. analog des Autos- nimmt.

Abschließend werde ich anhand zweier Interviews mit Forschern des „Deutschen Forschungsinstituts für künstliche Intelligenz“, Dr. Leonhard Hennig und Robert Schwarzenberg, beleuchten, ob autonomes Fahren tatsächlich möglich ist, oder ob es doch ein Wunschtraum bleiben wird.

Künstliche Intelligenz

Was ist künstliche Intelligenz?

Künstliche Intelligenz beschreibt zunächst nur ein Teilgebiet der Informatik. Man versucht dabei das menschliche „Lernen“, also das selbstständige aneignen von Fähigkeiten und Wissen zu simulieren.

Der Begriff an sich wurde von John McCarty geprägt, einem amerikanischen Informatiker, der 1956 für die „Dartmouth Konferenz“ am Dartmouth College in den USA den Begriff „künstliche Intelligenz“ als Titel wählte. Auf ihr wurden Programme vorgestellt, die Dame oder Schach spielten.[Vergleich 7]

Ihm selbst wird zugeschrieben, Erfinder des „Alpha-Beta-Algorithmus“ zu sein, welcher anhand zweier Werte -Alpha und Beta- Entscheidungsbäume relevant oder irrelevant für künftige Kalkulationen macht, also demnach die Menge an möglichen Zügen eines Schachcomputers einschränkt, indem die Züge bewertet werden.

Des weiteren hat er die Programmiersprache LISP entwickelt, die zu einer der ersten objektorientierten Sprachen zählt.[Vergleich 7]

Welche Abstufungen sind bei künstlicher Intelligenz zu ziehen?

Künstliche Intelligenz spaltete sich in zwei Themengebiete auf:[Vergleich 6]

Die starke künstliche Intelligenz

Als starke künstliche Intelligenz versteht man einen Apparat, der nicht nur in der Lage ist, Lernen zu simulieren, sondern der Zusammenhänge erkennt und selbst Modelle unabhängig von seinem Einsatzgebiet aufbaut. Das wird auch eine „Singularität“ genannt. Wissenschaftler sind sich recht uneinig darüber, ob es jemals eine starke künstliche Intelligenz geben wird. So behauptet Dr. Leonhard Hennig, ein Experte auf dem Gebiet der „Data Science“ und der Computerlinguistik, dass auch mit den Quantencomputern, welche den unglaublichen Vorteil der nahezu unbegrenzten Rechenleistung besitzen, keine starke künstliche Intelligenz kommen werde. Robert Schwarzenberg ist da nicht so sicher. Er meint, dass die Quantencomputer, die ja an sich schon prädestiniert sind für „Big Data“, durchaus in der Lage sein könnten, aufgrund ihrer charakteristischen Unsicherheit in Berechnungen, mehr als herkömmliche Systeme zu leisten. Diese Unsicherheit findet sich auch beim Menschen, denn anders als binäre Systeme kann ein Mensch mehr als nur ja oder nein sagen und so kann ein Quantencomputer auch mehrere Zustände gleichzeitig annehmen.

Zusammengefasst hat eine starke KI eine logische Planungsfähigkeit, etwas das wir unter dem Begriff „Kreativität“ fassen würden, außerdem kann sie für mehrere Aufgaben zugeeteilt werden, ohne sie zu verändern.[Vergleich 6]

Die schwache künstliche Intelligenz

Zur schwachen künstlichen Intelligenz zählen alle zielorientierten Algorithmen, die sich nicht von allein weiterentwickeln, somit alles was wir bisher an intelligenten Systemen besitzen. Vom Sprachassistenten im Handy, bis zur Handschrifterkennung ist alles eine schwache künstliche Intelligenz.

Maschinelles Lernen

Zum maschinellen Lernen zählen vergleichsweise einfache stochastische Methoden, wie zum Beispiel die lineare Regression. Hierbei werden n Datentupel aufgespalten in x Eingabewerte und y erwartete Ausgabewerte, die dann der

Form $(x; y)$ sind. Nun wird eine lineare Funktion über alle Datentupel n gemittelt, sodass eine Funktion der Form $ax + b$ entsteht. Zusätzlich wird mit sogenannten „Hyperparametern“ verhindert, dass die entstandene Funktion auf einige wenige Merkmale perfekt passt, aber nicht generalisiert. [1, Seite 108 ff.]

Dann gibt es die polynomielle Regression, bei der mit Funktionsrekonstruktion, ein Polynom über die Datentupel gelegt wird, sodass eine Funktion der Form $ax^i + bx^{i-1} + \dots + c$ entsteht. [1, Seite 122 ff.]

Als drittes Beispiel möchte ich den „RandomForest“ vorstellen, bei dem anhand der erwarteten Ein- und Ausgabewerte ein Entscheidungsbaum erstellt wird, bei dem jedoch eine gewisse Randomisierung vorliegt.

Es wird aus einer Menge $|n|$ Datentupel gezogen und statistisch verteilt, dann wird ein Entscheidungsbaum aufgebaut, bei dem aus den Eingabewerten $x = [x_n; x_{n-1}; \dots; x_0]$ eine Menge an Kriterien $k \leq |x|$ ausgewählt wird.

Analog dazu lässt sich dieses Verfahren folgend versinnbildlichen:

Ein Würfel ist ein Quader (Kriterium 1), ist größtenteils weiß (Kriterium 2) und hat schwarze Punkte (Kriterium 3), egal in welcher Reihenfolge man diese Eigenschaften untersucht, man findet heraus, dass das Objekt ein Würfel ist. Natürlich müssen keine booleschen Werte genutzt werden im „RandomForest“, sondern es können auch Dezimalwerte, Hexadezimalwerte, Binärwerte usw. benutzt werden, die Ausgabewerte sind also unabhängig vom Datenmodell, sondern von den Eingabewerten. [1, Seite 72 ff.]

Es gibt natürlich noch viele weitere Algorithmen, wie zum Beispiel k-Nachbarn, den „Bayes“ Klassifikator und viele mehr, die ich hier aber nicht vorstellen möchte. [Vergleich 1, Seite 1-50]

Neuronale Netze

Obwohl neuronale Netze gewissermaßen ebenfalls stochastische Methoden sind, besitzen sie die Fähigkeit, sich den Daten anzupassen, indem ein Abbild der Daten in ihnen entsteht. So lernt ein „Convolutional Neural Net“, zu Deutsch ein „faltendes neuronales Netz“, Ecken, Kanten und charakteristische Merkmale von Bildern zu erkennen.

Zum Beispiel kann bei der Handschrifterkennung ein solches Netz zwischen den

Rundungen der 3 und den Ecken der 7 unterscheiden.[1, Seite 361 ff.]

Ein „Recurrent Neural Net“, also ein rekurrentes, bzw. rückwirkendes neuronales Netz merkt sich die vorangegangenen Daten und versucht die nächsten vorherzusagen und wird auch mittlerweile an Börsen zur Spekulation benutzt.[1, Seite 385 ff.]

Hauptsächlich werde ich mich, aufgrund der enormen mathematischen und thematischen Komplexität mit den einfachen „Multilayer Perceptrons“, den „mehrlagigen Perzeptronen“, beschäftigen, die im Vergleich mit den anderen Netzarchitekturen recht primitiv sind.[1, Seite 253 ff.]

Außerdem ist verständnishaft zu sagen, dass der Name „neuronales Netz“ sehr irreführend ist. Als Leser entsteht sofort eine Verknüpfung zum Gehirn und vielleicht zum Aktionspotential und Ruhepotential aus dem Biologieunterricht, aber das neuronale Netz ist nur bedingt eine Nachbildung des Gehirns. Gewissermaßen sollte ein tiefes neuronales Netz eher „Stochastische Funktion n-ten Grades“ heißen, weil auch hier nur eine stochastische Analyse der eingegebenen Daten stattfindet und die entstandene Funktion den Grad der Tiefe des Netzes hat und auf das Datenset generalisiert, weil aus den Eingabewerten eine Funktion entsteht.

Der Name existiert aber natürlich nicht ohne Grund, das Gehirn bekommt auch eine Menge an Eingabewerten und gibt bestimmte Ausgabewerte, aber das menschliche Gehirn kann mit mehreren Daten gleichzeitig umgehen und verschiedene Aufgaben lösen. Ein neuronales Netz ist auf eine Aufgabe spezialisiert.[Vergleich 8]

„Deep Learning“

Das „Perzeptron“

Das Perzeptron ist das einfachste neuronale Netz, es hat nur zwei Eingänge und einen Ausgang. Hierfür sei der Inputvektor $x = [(0,1), (1,1), (1,0), (0,0)]$.

Jeweils ein Tupel in der Matrix ist für die Eingänge [0], [1], das heißt, wenn dazu der Ausgangsvektor $y = [0,1,0,0]$ betrachtet wird, dass wir die logische UND-Funktion abbilden. Natürlich muss mit einem Fehler gerechnet werden und die Daten, die das Perzeptron ausgibt müssen gedeutet werden, aber es wird bei guter Performance diese logistisch-logische Funktion abbilden.

Zunächst ist wichtig, dass mathematisch geklärt ist, wie dieses einschichtige Netz funktioniert:

$$y = wx.$$

In diesem Fall wird aber die Sigmoidfunktion als Ausgabefunktion verwendet, damit das Multilayer-Perzeptron verständlicher ist.[4, Seite 2-4]

$$y = \sigma(wx)$$
$$\sigma(wx) = \frac{1}{1 + e^{-(wx)}}$$

Als Lernregel für das Netz muss dann w den Daten x angepasst werden, dafür gilt folgende Formel, die im als „Deltaregel“ bezeichnet wird:

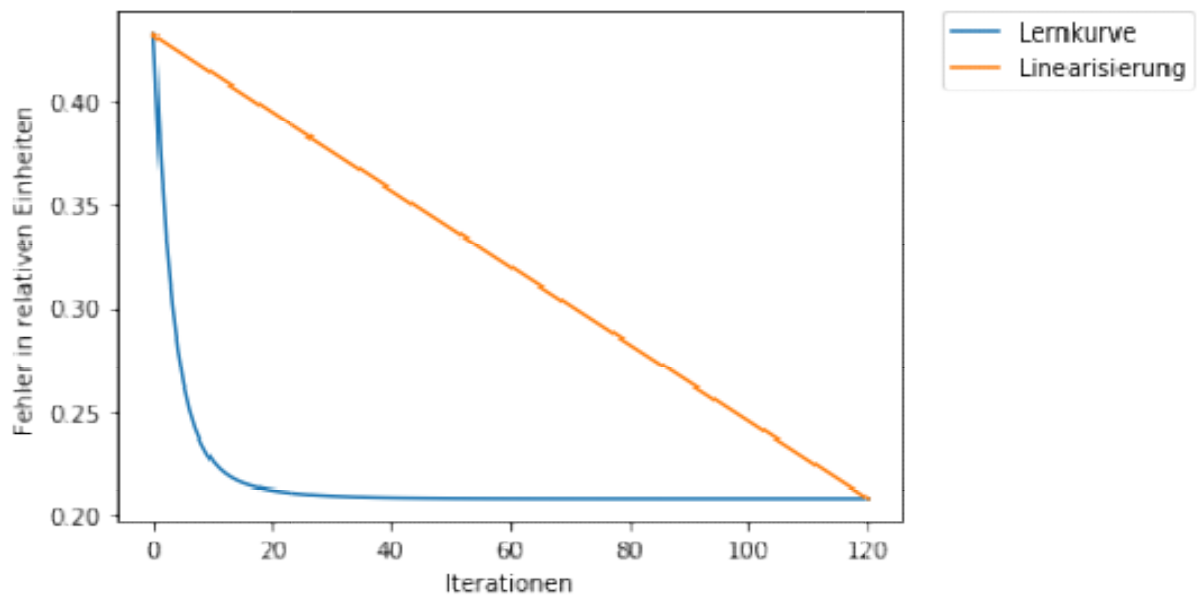
$$w_{x+1} = w(x) - (Y - y) * x.$$

Damit haben wir die Vorform eines Gradienten, denn wie aus dem Newton Verfahren bekannt ist, gilt bei der Nullstellensuche nicht umstellbarer Funktionen wie z.B. $f(x) = x^3 + x^2 - 1$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Als Veranschaulichung für den Lernprozess des Perzeptrons ist in folgender Grafik der Fehler in Abhängigkeit zur Iteration, also zum Gesamtdurchlauf aller

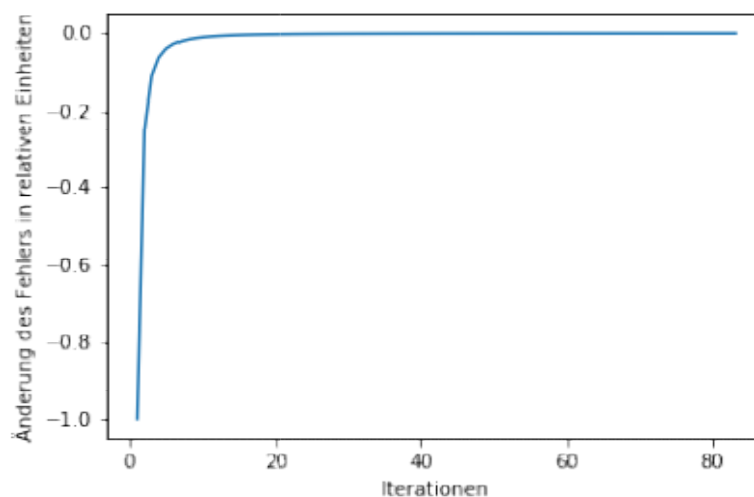
Daten mit Anpassung des Gewichtes, dargestellt:



1 <https://github.com/MNasert/Seminararbeit/blob/master/Perzeptron.ipynb>

Besonders wichtig ist hier, dass der Fehler permanent fällt, wie an der Linearisierung erkennbar ist. Interessant ist, dass der dargestellte Graph sich als $f(x) = \frac{1}{x}$ interpretieren lässt, und damit lässt sich auch diese Lernkurve ableiten als $f'(x) = -\frac{1}{x^2}$.

2 <https://github.com/MNasert/Seminararbeit/blob/master/Perzeptron.ipynb>



Die Schlussfolgerung ist, dass methodisch ein „Breakpoint“, also ein Punkt an dem das Netz nicht mehr auf die Werte trainiert wird, eingebaut wird.

Das „Multilayer-Perzeptron“

Das Multilayer Perzeptron ist eine der gängigsten Methoden der neuronalen Datenverarbeitung, weil es ein solides Modell ist, das je nach Tiefe sehr komplexe Funktionen abbilden kann.

Mathematisch zugrunde liegt hier ein Netz 2. Grades, mit der Sigmoid Funktion als Aktivierungsfunktion[1, Seite 136, 31.08.2019]:

$$y = \sigma(\sigma(xw_1 + b_1)w_2 + b_2).$$

y ist dabei der Output des Netzes. Es ist ein Input Channel vorhanden und ein „hidden Layer“. Um den Fehler des Netzes zu berechnen nutzt man im einfachsten Fall den mittleren quadratischen Fehler, also die stochastische Varianz auf j Objekte:[1, Seite 37, 31.08.2019]

$$L_i = \sum_{i=0}^j \frac{1}{2} (Y_i - y_i)^2$$

Durch Verwendung des quadratischen Fehlers werden große Fehler stärker bestraft als kleine, da Zahlen kleiner eins quadriert immer kleiner sind als ihre Wurzel:

$$0 < x < 1 \quad ; \quad x^2 < x.$$

Zur Berechnung des Gradienten nach w_n muss erstmal die Grundlage dazu geschaffen werden:

$$\sigma(wx + b) = \frac{1}{1 + e^{-(wx+b)}}$$

$$\sigma'(wx + b) = \sigma(wx + b) * (1 - \sigma(wx + b)).$$

Um jetzt zur Ableitung des ersten Gewichtes zu gelangen muss die Kettenregel angewandt werden:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial z_2} \frac{\partial z_2}{\partial \sigma_1} \frac{\partial \sigma_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}.$$

Durch Substitution lässt sich diese Schachtelung der Funktionen vereinfachen:

$$z_1 = w_1 x + b_1$$

$$\frac{\partial z_1}{\partial w_1} = x$$

$$\sigma_1 = \sigma(z_1)$$

$$\frac{\partial \sigma_1}{\partial z_1} = \sigma(z_1) * (1 - \sigma(z_1)).$$

Ab diesem Punkt wiederholt sich die Ableitung bis $\frac{\partial L}{\partial y}$:

$$z_2 = \sigma_1 w_2 + b_2$$

$$\frac{\partial z_2}{\partial \sigma_1} = w_2$$

$$y = \sigma(z_2) * (1 - \sigma(z_2)).$$

Nun ist Vorsicht geboten, da nicht nach Y sondern nach y abgeleitet wird.

$$\frac{\partial L}{\partial y} = y - Y$$

Besonders interessant ist, dass sich auch hier das Perzeptron wiederfindet $w(x + 1) = (y - Y)(x)$ das entspricht der erläuterten Deltaregel des Perzeptrons. Der Gradient sieht allerdings anders aus:

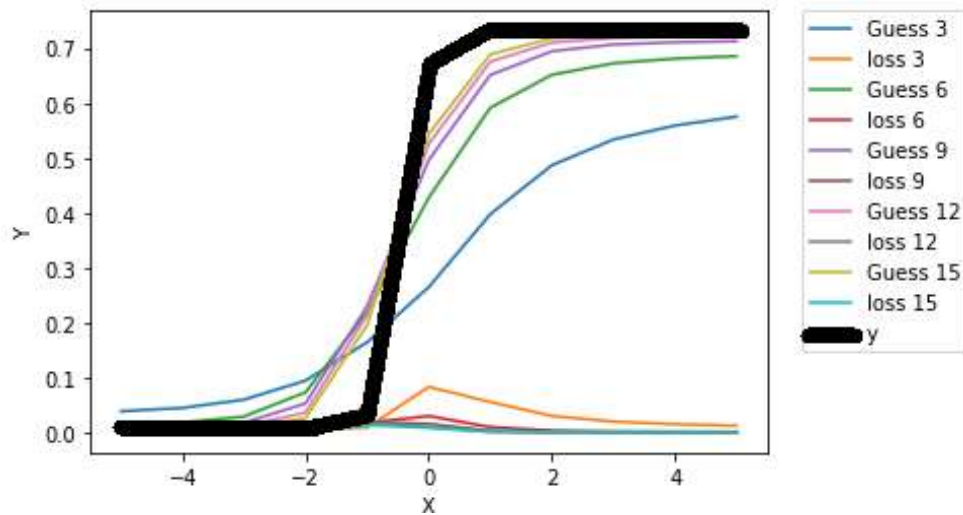
$$\frac{\partial L}{\partial w_1} = (y - Y)(w_2)(\sigma(z_1) * (1 - \sigma(z_1)))(x)$$

Da alle Werte zu jedem Zeitpunkt einen numerischen Wert im Programm haben lässt sich alles berechnen, ohne auf Probleme zu stoßen.

Der Gradient deutet auf das Maximum, weshalb er vom Gewicht abgezogen werden muss, um zum Minimum des Fehlers zu gelangen:

$$w_1 = w_1 - \frac{\partial L}{\partial w_1}(w_{ij}, b_{ij})$$

Nachdem das Netz dann trainiert wurde, ist es in der Lage eine Funktion nachzubilden. In einem selbst entwickelten Programm lies ich ein solches neuronales Netz eine Funktion mit der Form $y = \sigma(\sigma(xw_1 + b_1)w_2 + b_2)$ nachahmen, durch randomisiertes Initialisieren aller Variablen. Das Plotten der Funktionen führte zu folgender Grafik:



3 <https://github.com/MNasert/ML-projects/blob/master/Backprop.ipynb>

Die „loss“ Linien beschreiben den x abhängigen Fehler zur Funktion. Die „Guess“ Linien beschreiben nach durchgerechneter Epoche sortierter Reihenfolge, die neue Funktion und in schwarz ist die erwartete Funktion sichtbar.

Erkennbar ist, dass das Gradientenverfahren funktioniert, weil sich die „Guess“ Funktionen der „y“ Funktion immer weiter nähern. Wenn die Funktion mehrere tausendmal rechnen würde, dann wäre auch nicht garantiert, dass man die Ausgangsfunktion erhält, aber da neuronale Netze eher für Generalisierungen verwendet werden ist das nicht schlimm, weil man so bessere Aussagen über Gruppen treffen kann.

Dieses Netz könnte auch mit zwei Eingabewerten umgehen, die dann aus zwei dieser Funktionen bestehen. Dann würde das Netz versuchen zwischen diesen beiden Funktionen die Mitte zu finden, weil dort der Fehler am geringsten ist. Als Resultat entstünde dann eine Generalisierung beider Funktionen, die immer den Mittelwert von beiden y Werten ergäbe.

Neuronale Netze in der Automobilität

Als Analogie zu einem selbstfahrendem Auto gelte hier folgendes Robotinoviewprojekt, das ich im Rahmen der „Robotino AG“ am OSZ-Oder-Spree angefertigt habe.

Zugrunde liegt die Deltaregel des Perzeptrons, also die vereinfachte Version des bereits ausführlich erläuterten mittleren quadratischen Fehlers. Als Bedingung gilt, dass der Roboter neun Sensoren hat, die wie folgt angeordnet sind.

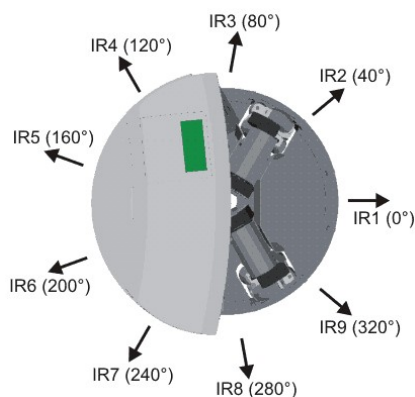


Abbildung 4 [5, 24.09.2019]

Das heißt, dass mehrere Sensoren vor-, oder wie im Programm realisiert, nach Gewichtung zusammengefasst werden müssen.

Zunächst ist aber wichtig, dass alle Werte eingelesen werden und das entsprechende Δy errechnet wird (siehe Anhang, Abbildung 1), dazu muss noch beachtet werden, dass der Roboter nur begrenzte Rechenleistung hat, weshalb es Sinn macht, die Gewichtsbeziehungsweise Deltaberechnungen nur nach einer gewissen Zeit zu wiederholen, hier alle 50 Millisekunden.

Nach Sichtung der verfügbaren Daten müssen diese verarbeitet werden, bevor eine tatsächliche Berechnung stattfindet, deshalb ist eine Funktion eingebaut, die nur ein x weitergibt, wenn es größer als ein beliebiger Wert z ist (siehe Anhang, Abbildung 2).

Nach Verarbeitung der Daten greift in halb-unüberwachter Form das Perzeptron ein. Halb-unüberwacht heißt für dieses Perzeptron, dass nur vorgegeben ist, was es erreichen soll, nicht was es machen soll, bei Eingabewert x . Sobald es einen Wert erhält wird nach der Form $w_i(x_i + 1) = w_i + w_i * \Delta y * x_i * \alpha$ (α ist ein zusätzlicher Lernfaktor, der das Lernen beschleunigt, aber ungenauer macht.) ein Gewicht

berechnet für den Sensor x_i (siehe Anhang, Abbildung 3). Dabei ist wichtig, dass das Perzeptron nicht übersteuert, also dass nur weil ein imminenter Wert (sehr wichtig, also sehr nah in diesem Kontext.) am Sensor liegt, das Programm nicht plötzlich gigantische Werte ausgibt und der Roboter gefährlich schnell wegfährt.

Das wäre nicht nur schlecht im Sinne der Programmierung, sondern kann auch Menschen verletzen, da der Roboter recht schwer ist.

Um dieser Fehlerquelle vorzubeugen wurde im Programm willkürlich der Wert 100 gewählt, was dann multipliziert mit dem maximalen Sensorwert (rund 2,5) 250 Einheiten ergibt, was laut Robotinoview einer Sollgeschwindigkeit von 250 Millimetern pro Sekunde entspricht.

Ganzheitlich sieht das Programm so aus (siehe Anhang, Abbildung 5). Es ist knapp die jeweilige Bedeutung des Programmblocks angegeben und online ist das gesamte Programm veröffentlicht.

Auswertung des Projektes

Das Programm läuft zur aktuellen Version komplett fehlerfrei. Der Roboter erkennt Objekte und weicht ihnen dynamisch aus. Die Zeit bis der Roboter einem Objekt ausweicht ist nicht mit dem menschlichem Auge sichtbar, was bedeutet, dass der Roboter mindestens so schnell wie ein echter Mensch reagiert. Die Bedeutung für ein autonomes Fahrzeug ist daraus ersichtlich, was dieser Roboter kann ist in etwa das, was eine Fußgängererkennung kann. Er könnte fahren und bei einem Kontakt entscheiden, ob zu bremsen ist und wenn ja wie stark, dass kein anderes Objekt gerammt wird oder rammt.

Die Idee hinter dem Projekt ist, den Roboter als Kellner in einer Gaststätte fahren zu lassen, der eine Programmteil ist ein Koordinatensystem, in dem der Roboter von Koordinate A zu B fährt und der zweite Teil ist das Perzeptron.

Das Perzeptron dient der Erkennung von Hindernissen und Objekten und gibt vor, wie der Roboter auszuweichen hat. Da ist es wichtig, dass der Roboter immer richtig agiert und ausweicht, würde er das nicht tun, also hätte man zum Beispiel ein festes Programm, das immer in die selbe Richtung fährt, bei Sensorbeanspruchung, dann würde der Roboter definitiv irgendwann ein Objekt rammen. Das Problem bei solcher Programmierung wäre, dass keine Zusammenhänge von so einer Methodik erkannt werden würden, wohingegen das Perzeptron wenigstens auf einem sehr primitiven Level den Zusammenhang zwischen Sensor A und seinem Gegenpart B erkennt und somit nicht ohne Rücksicht rückwärts fährt.

Es lässt sich zusammenfassen, dass das Projekt gelungen ist und so wie es ist fehlerfrei eingesetzt werden könnte, mit minimalem Wartungsaufwand, weil die Aufgabe immer die selbe bleibt und sich die Umwelt auch nicht verändern kann, dass die Sensoren plötzlich keine oder falsche Werte ausgeben.

Ausblick

Viele Wissenschaftler meinen, dass autonomes Fahren sehr bald kommerzialisiert wird und für die Bevölkerung zugänglich wird. Robert Schwarzenberg, ein Wissenschaftler am „Deutschen Forschungsinstitut für künstliche Intelligenz DFKI“ sagt dazu: „Ja, also ich denke das wird doch schon gezeigt wenn man sich die Tesla Autopiloten angeguckt, (...)im Großen und Ganzen von den Ressourcen her ist das machbar.“[Siehe Anhang, Interviews, Protokoll mit Robert Schwarzenberg]. Dr. Leonhard Hennig, ebenfalls Forscher am DFKI, schließt sich Robert Schwarzenberg an:

„Ich denke ja, das was da an Bildverarbeitungs- und Sensorverarbeitungsalgorithmen eingesetzt werden muss, um autonomes Fahren zu ermöglichen ist mittlerweile so ausgereift, dass es eine sinnvolle Ergänzung, also eine sinnvolle Softwareergänzung eines physischen Autos ist und das autonomes Fahren sozusagen in der Richtung tatsächlich nützlich sein wird und eine nützliche Komponente eines Autos ein wird.“[siehe Anhang, Protokoll mit Dr. Leonhard Hennig]

Nach Betrachtung meines Perzeptrons kann auch ich mich nur anschließen. Auch wenn der Mensch wahrscheinlich noch einige Jahrzehnte das Auto wenigstens als Kontrollfunktion steuern muss, wird er irgendwann abgelöst. Wie von Herrn Schwarzenberg bereits beschrieben können die Autopiloten von Tesla bereits auf Highways fahren ohne von einem Menschen kontrolliert werden zu müssen. Der Stadtverkehr ist natürlich noch einige Schwierigkeitsgrade höher angesiedelt, mit Kreuzungen, Ampeln und vielem mehr. Vielleicht erübrigen sich diese Probleme aber mit der Digitalisierung oder durch „kluge“ Straßen die z.B. Autos anleiten könnten oder zumindest die Positionen aller auf ihnen fahrenden Autos mitteilen.

Das Robotinoprogramm ist da ein sehr einfacher Anfang, Wissenschaftler von IEEE ROBOTICS AND AUTOMATION LETTERS[2] haben in ihrem Paper aus dem Juni die tatsächlichen Algorithmen zusammengefasst: „Autonomous vehicles are controlled today either based on sequences of decoupled perception-planning-action operations, either based on End2End or Deep Reinforcement Learning (DRL) systems. Current deep learning solutions for autonomous driving are subject to several limitations (e.g. they estimate driving actions through a direct mapping of sensors to actuators, or require complex reward shaping methods).“[2, Seite 1]. Mit dem in meiner Arbeit präsentierten „Deep Learning“ lässt sich ansatzweise vorstellen, wie komplex so ein autonomes Fahrzeug ist, das dann mit „reinforcement learning“, also

Belohnungsorientiertem Lernen lernt und wie „einfach“ in diesem Zusammenhang ein Perzeptron ist, das aber auch schon gute Erfolge erzielt.

Literaturverzeichnis

1. **Geron, Aurelien.** *Praxiseinstieg Machine Learning mit Scikit-Learn und Tensorflow*. 69123 Heidelberg : O'Reilly, 2018. 978-3-96009-061-8.
2. **Sorin M. Grigorescu, Bogdan Trasnea, Liviu Marina, Andrei Vasilcoi, Tiberiu Cocias.** *NeuroTrajectory: A Neuroevolutionary Approach to Local State Trajectory Learning for Autonomous Vehicles*. [Dokument] <https://arxiv.org/pdf/1906.10971.pdf> : IEEE ROBOTICS AND AUTOMATION LETTERS, 2019.
3. **Hennig, Jon.** *Neuronale Steuerungsparadigmen für autonome Roboter realisiert durch ein flexibles Software-Tool*. [Dokument] <http://ul.qucosa.de/api/qucosa%3A16620/attachment/ATT-0/> : Universität Leipzig, Universität Leipzig, 2003.
4. **Otte, Sebastian.** *Künstliche neuronale Netze Das Perzeptron*. [Dokument] <https://www.cs.hs-rm.de/~panitz/prog3WS08/perceptron.pdf> : Hochschule RheinMain, 2009.
5. **Didactic, Festo.** Robotino View. *Robotino View 3.3.1*. 2019.
6. **Vowinkel, Bernd.** Der Humanistische Pressedienst. *hpd.de*. [Online] Humanistischer Pressedienst (hpd) e.V., 2. September 2016. [Zitat vom: 24. September 2019.] <https://hpd.de/artikel/kommt-technologische-singularitaet-13480>.
7. **Wichert, Andreas.** *spektrum.de. spektrumsverlag.de*. [Online] Spektrum der Wissenschaft Verlagsgesellschaft mbH, 2000. [Zitat vom: 24. September 2019.] <https://www.spektrum.de/lexikon/neurowissenschaft/kuenstliche-intelligenz/6810>.
8. **Michaels, Johanna.** Frankfurter Allgemeine. *faz.net*. [Online] Frankfurter Allgemeine Zeitung GmbH, 31. Juli 2019. [Zitat vom: 30. September 2019.] <https://www.faz.net/aktuell/wissen/computer-mathematik/kuenstliche-intelligenz-per-velo-zur-superintelligenz-16312008.html>.

Anhang

Grafiken

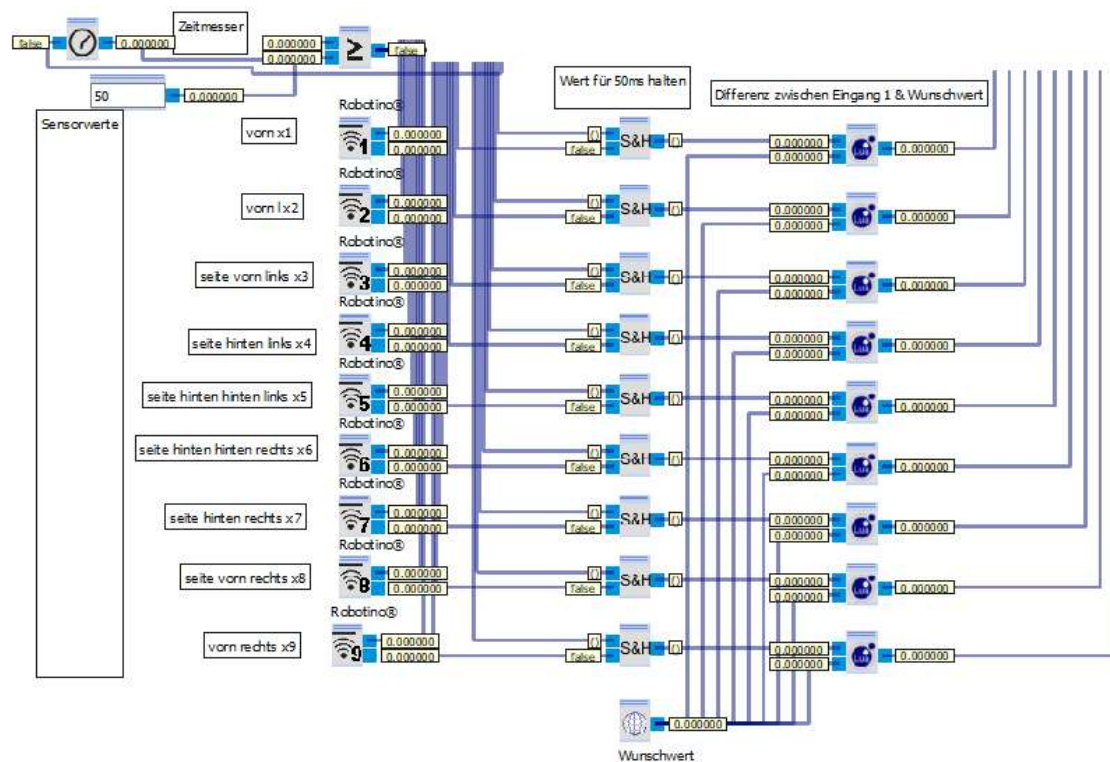


Abbildung 1

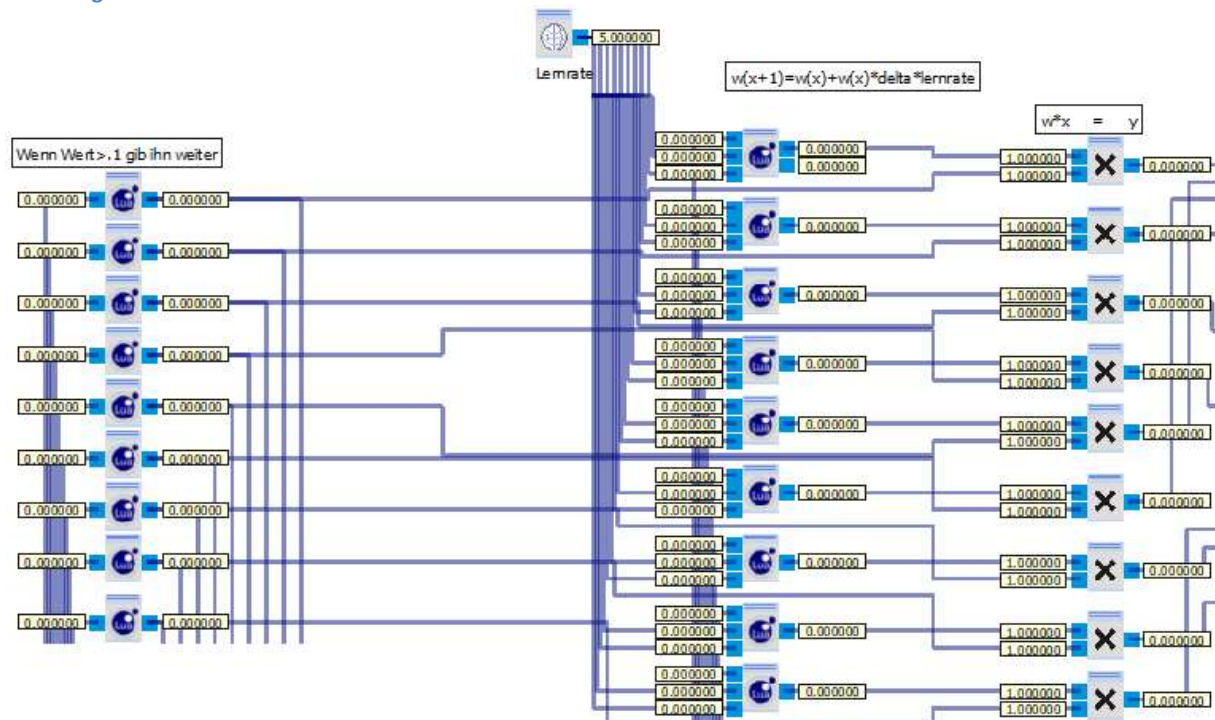


Abbildung 2

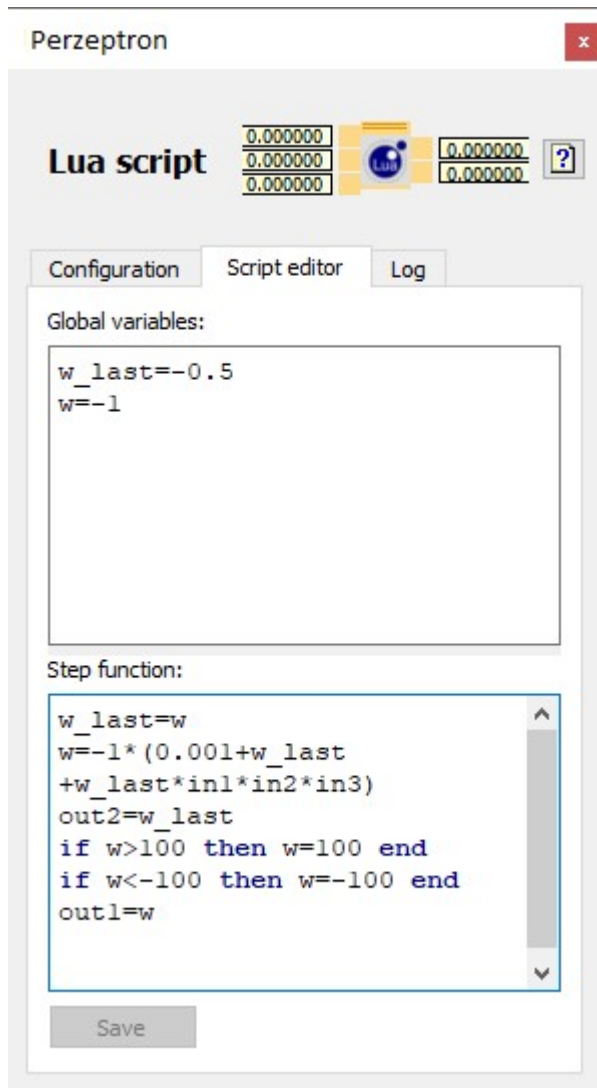


Abbildung 3

Interviews:

Protokoll mit Dr. Leonhard Hennig vom 13.06.2019

Maximilian Nasert: So, ich wollte sie mal bitten sich vorstellen.

Mein Name ist Leonard Hennig, ich bin Senior Researcher am deutschen Forschungszentrum für künstliche Intelligenz ähm arbeite seit knapp 5 Jahren hier. Ich habe im Prinzip ne Ausbildung äh in Osnabrück und an der TU Berlin als Computerlinguist und Informatiker.

Maximilian Nasert: Okay, Dann würde ich erstmal ein paar allgemeine Fragen stellen.

Okay.

Maximilian Nasert: Woran arbeiten sie denn beziehungsweise, was ist ihr Arbeitsgebiet?

Also mein Arbeitsgebiet beschäftigt sich mit der Extraktion von Informationen aus Texten, das heißt wir arbeiten mit geschriebener Sprache, Nachrichtendokumenten, Blogposts, Twitter und ähnlichen Dokumentenarten und versuchen aus diesem, sozusagen für eine Maschine unstrukturierten Texten, strukturierte Informationen zu extrahieren also sowas wie Eigennamen erkennen also Orte, Person, Firmenname.

Wir versuchen Informationen über diese erkannten Eigennamen also Personen zu finden oder Firmen so zum Beispiel Standorte von Firmen, Firmenübernahmen, Lieferkettenschwierigkeiten und Ähnliches, mit dem Ziel ähm diese Informationen sozusagen in einer Art Datenbank zur Verfügung stellen zu können damit sie in weitergehender Anwendung genutzt werden kann.

Dass so ein Teilgebiet andere Teilgebiete, an den ich aktuell nicht so intensiv arbeite während sowas wie automatische Textzusammenfassung äh sprich ein oder mehrere längere Texte in eine zusammengefasste Version zu überführen. Chatbots und Dialogsysteme und was fällt mir noch ein als Drittes – ja, maschinelle Übersetzung wäre zum Beispiel auch noch ein Gebiet was hier in der Gruppe bearbeitet wird.

Maximilian Nasert: Okay, danke.

Maximilian Nasert: Was denken sie denn was kann in 15 oder 20 Jahren alles übernehmen, was heute noch nicht so möglich ist?

Hmm... also das was man in sozusagen der Öffentlichkeit äh berichtet oder diskutiert wird ähm ist ja schon einiges also sowas wie autonomes Fahren, ähm Bilderkennung also zum Beispiel die Diagnose von Hautkrebs ähm äh gezielte Sachen wie Textgenerierung für bestimmte Arten von Nachrichtenartikeln und so weiter also sehr spezialisierte Aufgaben vielleicht auch sowas von wahrscheinlich viele mittlerweile gehört haben äh bestimmte Arten von Spielen spielen. „Go“ hat wahrscheinlich jeder mittlerweile gehört, dass es da ein Programm gibt was der menschlichen Groß- oder Weltmeister geschlagen hat, analog andere Arten von Spielen, Computerspiele und Ähnliches. Das heißt äh, meiner Meinung nach, werden wir bestimmte Arten von Algorithmen die heutzutage künstliche Intelligenz genannt werden oder unter dieses Gebiet fallen in der Praxis sehen für bestimmte hoch spezialisierte Aufgaben.

Was wir nicht sehen werden ist eine generelle künstliche Intelligenz die vielleicht wie ein Mensch ähm handeln kann, sich in der Welt zum Beispiel bewegen kann. Also ein anderes Teilgebiet was was jetzt sozusagen ja noch gar nicht erwähnt wurde, dass ich noch gar nicht erwähnt hätte wäre sowas wie

Robotik, dass wir Roboter haben werden, die äh auch nur annäherungsweise menschliche Emotionen oder emotionale Intelligenz nachbilden können wird auch in 20 Jahren noch nicht so weit sein. Äh vielleicht wieder etwas näher bei meinem Gebiet, Sprachtechnologie. Wir haben Algorithmen die Texte übersetzen können ähm aus indoeuropäischen Sprachen also wir können gut oder einigermaßen gut Deutsch-Englisch, Englisch-Deutsch. Wir können vielleicht auch für Englisch-Russisch, Russisch-Englisch, wir können es nicht für äh kleinere Sprachen tschechisch, polnisch moldawisch ähm ganz zu schweigen von vielen afrikanischen Sprachen weil es einfach nicht genug Ressourcen gibt und selbst wenn das dann hoffentlich soweit sein wird, können diese Algorithmen halt eben nur das eine nämlich übersetzen aber, nicht gleichzeitig Fragen beantworten und nicht gleichzeitig ein Gespräch führen und nicht gleichzeitig sich vielleicht sogar in der Welt bewegen uns sehen und Sachen wahrnehmen die nicht über Sprache, gesprochene Sprache kommuniziert werden.

Maximilian Nasert: Ähm...damit würde ich diesen relativ allgemeinen Themenbereich abschließen und mich den neuronalen Netzen widmen.

Was denken Sie dann warum neuronale Netze so erfolgreich sind?

Ähm, ich glaube der Kollege hat schon einen Grund genannt, nämlich die Verfügbarkeit von entsprechende Hardware ähm im vorherigem Interview.

Der zweite Grund den ich auch nicht unwichtig finde ist, dass das äh viele Forscher in den Jahren davor mit den existierenden Methoden an Grenzen gestoßen sind ähm das heißt in der in der klassischen statistischen, im klassischen statistischen, maschinellen Lernen ging halt viel darum Daten tatsächlich zu verstehen äh sich Eigenschaften der Daten zu überlegen die für ein maschinelles Lernverfahren relevant sind und diese halt zu programmieren und als Input für den Klassifikator oder für ein Klassifikationsalgorithmus ähm zu nutzen. Das war ein sehr hoher ähm Aufwand, man braucht sozusagen Domainexpertise und und aha sozusagen Wissen wie eine Data Science ist zum Beispiel arbeitet äh das alles also dieses Versprechen dass ich kein Featureengineering machen muss, so nennt man das, äh hat sehr viele Leute angelockt mit neuronalen Netzen zu arbeiten weil es natürlich einfacher ist wenn äh neuronale Netze es, oder neuronale Algorithmen es hinbekommen solche, solches Wissen über äh die Inputdaten selbst äh lernen zu können sozusagen. Das heißt es erspart mir Engineeringarbeit, dass ist der Grund warum das so erfolgreich eingesetzt wird, in Unternehmen. Weil man niemand mehr braucht der Features entwickeln kann. Das ist eine wesentlicher Grund, meiner Meinung nach.

Maximilian Nasert: Okay, gibt es eine Herangehensweise mit der man neuronale Netze modelliert oder muss man zufällig ein Aufbau auswählen, also jetzt in Bezug auf die Layer und wie groß diese Layer sind?

Es gibt, also da kann ich nicht so viel zu sagen, weil ich tatsächlich äh aktiv nicht mehr so viel äh selber neuronale Netze entwickle. Das machen die Doktoranten, ähm meines Wissens äh denke ich dass da momentan der Ingenieursaufwand drin steckt. Es gibt gewisse Erfahrungswissen welche Arten von Netzen welche Layer für bestimmte Aufgaben funktioniert, wie viele Layer mit welcher Tiefe, mit welchem Parameter, Hyperparametern äh ich einsetze um am Ende gegen gewisse Aufgabe ist tatsächlich äh wieder genau der Aufwand, der Engineeringaufwand den ich vorher bei der Definition von Features hatte. Es gab eine Zeit lang, das ist jetzt zum Glück in letzten 2 Jahren nicht mehr so viel aber die 2 - 3 Jahre davor war der Großteil der Paperware auf wissenschaftlichen Konferenzen, in der Art, ich nehme ein existierendes neuronales Netz und schmeiße noch ein, und füge ein weiterer Layer hinzu, ähm...keine Ahnung- .nen Pooling Layer oder ein äh aha ja keine Ahnung oder Residual Layer Connection,

also ich ich versuche Verbindung in meinem neuen neuronale Netz hinzuzufügen oder wegzunehmen um zu schauen wie der Effekt auf die auf die komplette Aufgabe ist.

Maximilian Nasert:Dankeschön.

Äm..nun schreibe ich ja demnächst meiner Seminararbeit über neuronale Netze in Bezug auf Automobiltechnik und ich möchte Sie dahingehendfragen ob Sie glauben dass neuronale Netze in der Lage sind, ein Auto zu Steuern ohne dabei unangemessen viele Ressourcen zu verbrauchen

ähm Ich denke ja.

Das äh, dass die sagen, dass was da an Bildverarbeitungs- und Sensorverarbeitungsalgorithmen eingesetzt werden muss um autonomes Fahren zu ermöglichen äh mittlerweile so ausgereift ist, dass ähm es eine sinnvolle Ergänzung, also eine sinnvolle Softwareergänzung eines physischen Autos ist und das autonomes Fahren sozusagen in der Richtung tatsächlich nützlich sein wird und eine nützliche Komponente eines Autos ein wird.

Maximilian Nasert:Okay äh noch eine letzte formale Frage.

Darf ich dieses Interview für meine Facharbeit benutzen?

Ja

Maximilian Nasert:Okay, dann bedanke ich mich und auf Wiedersehen

Protokoll mit Robert Schwarzenberg vom 13.06.2019

Maximilian Nasert:Dann stellen Sie sich doch bitte mal vor.

Hallo Ich bin Robert Schwarzenberg, ich bin Informatiker im deutschen Forschungszentrum für künstliche Intelligenz wo ich als Softwareengineer arbeite aber auch an meiner Doktorarbeit für die Forschung betreiben hier.

Maximilian Nasert:Okay, dann werde ich jetzt mit dem Interview direkt beginnen.

Gerne!

Maximilian Nasert:Woran arbeiten Sie denn, was ist Ihr Arbeitsgebiet?

Also das DFKI bemüht sich aktuellsten Forschungsergebnisse zu übertragen und zu übersetzen in Industrieapplikation, so dass sozusagen der normale Anwender gleich davon profitiert das heißt dasDFKI ist ein Katalysator, das ist eine meiner Aufgaben, Industriepartner dabei zu begleiten wie wir statistische Lernverfahren ihre Produkte und Lösung einbauen können.

Als zweite äh zweite Sache mit der ich beschäftigt bin, wie schon gesagt, dass meine Doktorarbeit und hier forsche ich in die Richtung erklärbare neuronale Netze in Bereich der Sprachtechnologie. Das bedeutet wir würden gerne erfahren auf welche Grundlage die Entscheidungen basieren die neuronale Netze treffen ähm und das ist gar keine so triviale Aufgabe.

Maximilian Nasert:Okay, was denken Sie denn was KI in 15 oder 20 Jahren alles übernehmen kann was, heute so noch nicht möglich ist?

Ich denke wir werden in 5, 10 oder 20 Jahren mit Ausnahme eines Durchbruchs bei den Quantencomputern, dass äh ist für mich schwer abzuschätzen, wenn die sich durchsetzen sollten was das für die KI bedeutet.Aber wenn das bei konventionellen Rechenmaschinen bleibt werden wir hoch spezialisierte Anwendungen haben aber keine starke künstliche Intelligenz, dass bedeutet wir werden nicht den Punkt der Singularität erreicht haben in dem die künstliche Intelligenz den Menschen jedweder Domäne ablöst, aber hoch spezialisierte Anwendungen wie sie jetzt schon beobachten im täglich im Alltag. Also ähm ich könntemir vorstellen dass es ähm sehr effiziente äh auch in der Sprachtechnologie schon sehr effizienten, ähm wie das deutsche Wort für „summerizer“ - „Zusammenfassen“ gibt, ja so dass man dein groß Buch rein gibt und schnell -ne gute Zusammenfassung kriegt oder man hat ein selbstfahrendes Auto aber selbst fahren Auto kann halt nur nicht die Texte zusammenfassen und der „Zusammenfasser“, wenn man ihn so nenne darf, kann das Auto nicht steuern, also sind spezialisierte Systeme geben.

Maximilian Nasert:Vielen Dank.

Ähm...nun würde ich gerne etwas spezielle werden und direkt auf neuronale Netzte eingehen.

Gerne!

Maximilian Nasert:Warum sind denn neuronale Netze so erfolgreich?

Warum die genau äh in spezielle Aufgabe sehr erfolgreich sind, dass wird noch erforscht. Das ist ja ein Teil dessen ähm, womit ich mich beschäftige dass wir zurückführen was die Grundlage für eine normale Entscheidung war und so hoffentlich dann auch verstehen wie die neuronale Netze zu Entscheidungen

kommen sind insbesondere ist das schwierig ist die die zwischenzeitlichen Repräsentation eines Problems zu interpretieren und erscheint das neuronale Netz sehr gute Möglichkeiten gefunden zu haben Probleme zu abstrahiert.

Aber da ist die Forschung noch am Anfang würde ich vermuten, obwohl ich schon seit einigen Jahrzehnten geht. Warum gerade sehr erfolgreich sind im Sinne von warum sie überall Anwendung finden, ist natürlich weil sie objektiv die beste Leistung bringen und das hängt damit zusammen dass die Hardware der letzten Zeit äh große Sprünge gemacht hat, was übrigens der Gamingindustrie zu verdanken ist. Die nämlich dazu geführt hat dass die Nachfrage nach leistungsfähigen Grafikkarten da war und diese leistungsfähigen Grafikkarten erlauben es nun in der Tiefe normale Netze zu trainieren also diese tiefen neuronalen Netze haben die Möglichkeit viele viele sehr effiziente Repräsentation zwischenzeitlich zu finden abstrahiert.

Genau...und da wir jetzt äh die Hardware haben um Tiefenneuronale Netze zutrainieren, wurde etwas überwunden was man in „neuronalen Winter“ nennt das war eine Zeit in der viele Menschen nicht daran geglaubt haben, dass neuronale Netze noch bei effektiv werden, eben weil sie so aufwendig zu trainieren sind.

Maximilian Nasert: Danke schön.

Ähm...nun gibt es ja die Hardware und auch die Software um damit umzugehen, aber gibt es da noch eine Herangehensweisen mit der man neuronale Netze modelliert oder, muss man zufällig ein Aufbau auswählen, also wie viele Layer ein neuronales Netz hat und wieviel Eingang, Ausgang man braucht?

Ja, das ist schon richtig also ähm das neuronale Netz findet durch eine ein Verfahren, dass wir Backpropagation nennen, Internparameter die das Problem möglichst gut abbilden aber, man muss noch externe Entscheidung treffen zum Beispiel zur Architektur und da ist es oft so dass die die Forscherinnen und Forscher eher eine Intuition haben warum bestimmte neuronale Baustein äh hilfreich sein könnte. Ja also man schaut sich das Problem an, man fragt sich intuitiv wie müsste er müsste die Informationsfluss sein um das Problem zu lösen und entscheidet darauf hin welche Komponenten neuronale Komponenten die alle spezialisiert sind auf bestimmte Informationsflüsse man nutzt um das Problem anzugehen und da ist viel Trial und Error dabei, da ist viel ausprobieren dabei, was aber auch mittlerweile automatisiert wird. Da testet der Computer viele Architekturen und entscheidet sich dann für die äh äh Effizienteste.

Maximilian Nasert: Danke schön.

Ähm...nun schreibe ich ja eine Arbeit bei der ich neuronale Netzeversuche auf Automobiltechnik anzuwenden.

Maximilian Nasert: Glauben Sie dann als Spezialist, dass neuronale Netze in der Lage sind ein Auto zu steuern, ohne dass man dabei unangemessen viele Ressourcen verbraucht?

Ja, also ich denke das wird doch schon gezeigt wenn man sich die äh Tesla Autopiloten angeguckt, die sind in vielen Situationen, also ich hab die bisher nur auf Highways gesehen, was natürliches stark vereinfachte Situation für ein Auto ist, da ist wahrscheinlich die Linechange das größte Problem aber im Großen und Ganzen von den Ressourcen her ist das machbar, Das ist auch ein Onboard-Computer also das werden die Berechnung, die Bildverarbeitung äh im Auto ähm durchgeführt ähm. Es gibt natürlich auch Anzeichen dafür, dass es zum Beispiel Stadtverkehr eventuell viel zu aufwendig ist ähm. Ich denke da insbesondere an Googles Versprechen ein selbstfahrendes Auto auszuliefern, das bis heute nicht passiert ist und ich denke sie sind auf Probleme gestoßen, die sich nicht so einfach lösen das vielleicht eben weils zu rechenaufwendig ist für ein Onboard-Computer.

Maximilian Nasert: So dann noch eine letzte formelle Frage. Ähm darf ich dieses Interview benutzen für meine Facharbeit?

Für deine Facharbeit kannst du dieses Interview benutzen.

Maximilian Nasert: Dann vielen Dank und auf Wiedersehen.

Ja, ich bedanke mich auch für deine gute Arbeit.

Anmerkung: Diese Interviews entstanden aus Tonaufnahmen und wurden von Microsoft Office transkribiert

Selbstständigkeitserklärung

Ich erkläre, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

Datum/ Unterschrift