

Anwendung neuronaler Netzer zur Verkehrsflussoptimierung

Till Hildebrandt, inf102835*

University of Applied Sciences, Wedel

E-mail: till.hildebrandt@gmail.com

INHALTSVERZEICHNIS

1	Einleitung	3
1.1	Motivation	3
1.2	Gegenwärtige Situation	6
2	Neuronale Netze	9
2.1	Grundprinzipien	9
2.1.1	Perzeptron	10
2.1.2	Sigmoid-Neuronen	12
2.2	Arten und Anwendungsgebiete // Architektur	13
2.3	Lernen - Das Anpassen der Gewichte	14
2.4	Überwachtes und Unüberwachtes Lernen	16
3	Verkehrsmodelle	17

1 EINLEITUNG

Die folgende Ausarbeitung befasst sich mit der Optimierung von Ampelschaltungen in Verkehrsnetzen mithilfe neuronaler Netze, die sich den erweiterten Verkehrsmanagementsystemen (*engl.* Advanced Traffic Management System ¹) zuordnen lässt. Zunächst wird das Problem beschrieben, die Motivation hergeleitet und die gegenwärtige Situation aufgezeigt. Anschließend werden neuronale Netze² in ihren Arten und Prinzipien sowie die Darstellung von Verkehrsnetzen in datenverarbeitenden Systemen behandelt. Diese Abschnitte bilden die Grundlage, um den Optimierungsansatz mit dem Hopfield-Modell³ genau vorzustellen und dessen Resultate mit denen anderer Lösungen zu vergleichen.

Ziel ist es den Verkehrsfluss in Städten so zu optimieren, dass der Verkehr möglichst störungsfrei fließt. Ein besserer Verkehrsfluss bedeutet weniger Staus, heißt weniger Umweltbelastung und weniger Unfälle.

1.1 Motivation

Die aktuelle politische Situation im Kleinen, sowie der Klimawandel im Großen, treiben uns immer weiter dazu nach Möglichkeiten zu streben verantwortungsvoller mit unserer Umwelt umzugehen. Ein verbesserter Verkehrsfluss ermöglicht es nicht nur dem Reisenden (im Durchschnitt) schneller sein Ziel zu erreichen, er geht optimaler Weise auch einher mit weniger Standzeiten, weniger Motoren im Leer-

¹ Wikipedia, [Advanced Traffic Management System](#)

² Wikipedia, [Künstliches neuronales Netz](#)

³ Wikipedia, [Hopfield-Netz](#)

lauf und in energieaufwändigen Prozessen wie dem Anfahren, bei dem viel Energie aufgebracht wird und damit entsprechend viel CO₂ wie andere Schadstoffe produziert werden. Dies bedeutet auch weniger Verschleiß, was zu einem längeren Verwenden des Fahrzeugs und somit zu einer Reduktion der CO₂-Bilanz⁴ der Herstellung zugute kommt.

Zudem steigen die Anzahl zugelassener Fahrzeuge und das damit verbundene Verkehrsaufkommen seit Aufkommen des Automobils an fast stetig an^{5,6}. Einzig die von der Bundesregierung ausgesprochene Abwrackprämie⁷ im Jahr 2008 hat, zumindest in Deutschland, Wirkung gezeigt. Dennoch sind die Zahlen weiterhin am Steigen und somit ist auch künftig mit einer zunehmenden Belastung des Straßennetzes zu rechnen. Ein besserer Verkehrsfluss kann von einer höheren Effizienz der Auslastung der Kapazitäten der Straßen profitieren und dabei helfen Investitionen in infrastrukturelle Projekte im besten Fall zu vermeiden.

Abschließend besteht die sowohl emotionale als auch statistisch begründbare Vermutung, dass ein verbesserter Verkehrsfluss mit weniger und kürzeren Standzeiten zu einer angenehmeren Reise mit mehr Umsicht, Ruhe und weniger Unfällen führt. Statistisch ist eine Korrelation zwischen Fahrtzeit und der Wahrscheinlichkeit, dass ein Unfall eintritt zu erwarten.

$$P_{\text{strecke-kurz}}(\text{Unfall}) < P_{\text{strecke-lang}}(\text{Unfall})$$

⁴ Wikipedia, **CO₂-Bilanz**

⁵ Statista, **engl. Number of vehicles in use worldwide**,
<https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide/>

⁶ Statista, **PKW-Bestand in Deutschland**,
<https://de.statista.com/statistik/daten/studie/12131/umfrage/pkw-bestand-in-deutschland/>

⁷ Wikipedia, **Umweltprämie**

Es hat sich gezeigt, dass neuronale Netze ein gutes Werkzeug im Umgang mit Problemen sein können, die mit klassischen Verfahren nur schwer adressierbar sind. Dabei handelt es sich in der Regel um Probleme, die mit Daten umgehen, in denen es irgendeine Form an statistischen Zusammenhängen/Mustern gibt. So wurde mithilfe neuronaler Netze Software produziert, die eine künstliche Intelligenz für das Spiel Go⁸, bei dem der Mensch lange als ungeschlagen galt, realisiert oder eine, die eine Synchronisation von geschriebenem Text auf Mundbewegungen eines Videos bewerkstelligt⁹. Weitere Beispiele sind:

- Schrifterkennung¹⁰
- Kaufempfehlungen
- Spracherkennung
- Aktienkursanalysen
- Gesichtserkennung
- Textübersetzungen

Bei dem hier behandelten Problem der Verkehrsflussoptimierung kann, ähnlich wie bei den oben genannten Probleme, ein starker statistischer Zusammenhang angenommen werden. In bestimmten Rhythmen fahren mehr oder weniger Menschen bestimmte Straßen in bestimmte Richtungen. Der Verkehr ist dabei unterschiedlich dicht und schnell. Manche Strecken sind zu bestimmten Zeiten mehr ausgelastet, andere weniger. Klingt nach einem großartigen Anwendungsfall für neuronale Netze.

⁸ Wikipedia, [Go \(Spiel\)](#)

⁹ Joon Son Chung and Andrew Zisserman, Oxford University, [Out of time: automated lip sync in the wild](#)

¹⁰ Pythonprogramming, [Image Recorgnition with Python](#), <https://pythonprogramming.net/image-recognition-python/>

1.2 Gegenwärtige Situation

Derweil gibt es keine einheitlich angewandtes System zur Ampelsteuerung in Deutschland. Ampelphasen können fest definiert sein oder verkehrsabhängig gesetzt werden. Bei festen Definitionen werden die Zeiten in einem Signalzeitenplan¹¹ festgehalten.

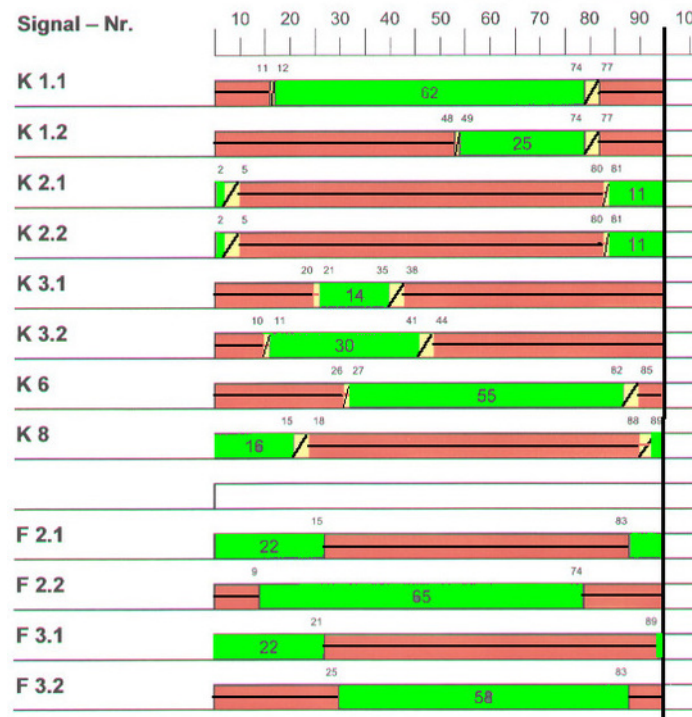


Abbildung 1: Signalzeitenplan

Auf der X-Achse befindet sich die Zeit in Sekunden, auf der Y-Achse die diskreten Werte der beteiligten Signalgeber. Rot- und Grünphasen sind in der jeweiligen Farbe dargestellt. Die übrigen Elemente entsprechen den beiden Gelbphasen Gelb und Rot/Gelb. Zu verstehen ist das Diagramm im Zeitfluss von links nach rechts. Rechts angekommen, geht es links wieder los.

Bei der verkehrsabhängigen Steuerung werden die einzelnen Verkehrsströme je nach Bedarf bedient. Sie funktioniert mithilfe vielerlei Technologien, so gibt es Induktionsschleifen und Bewegungsmelder,

¹¹ Wikipedia, [Signalzeitenplan](#)

aber auch Videokameras unterstützen die Optimierung des Verkehrsflusses bei manchen Systemen. Vorwiegend handelt es sich dabei um Systeme, die an einzelnen Kreuzungen wirken und keinen größeren Bereich in Betracht ziehen. Sie sind zum Beispiel so konstruiert, dass alle ankommenden Fahrzeuge einer ankommenden Fahrzeugwelle die Kreuzung passieren können, die Ampelphasen werden dementsprechend angepasst. Dabei ist darauf zu achten, dass die Wartezeiten der anderen Verkehrsteilnehmer nicht unzumutbar werden oder ein Rückstau der Abbiegespuren entsteht, die die anderen Spuren einengt.

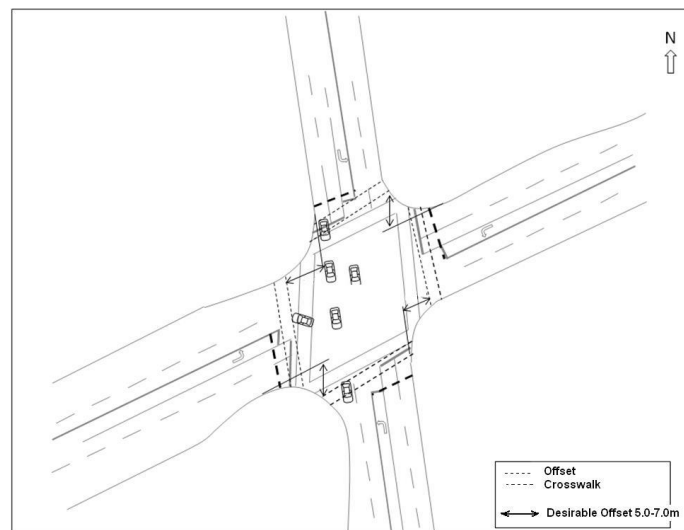


Abbildung 2: Rückstau an einer Kreuzung

Daraus ergibt sich eine variable Umlaufzeit der Ampelphasen. Fest definierte Abläufe können zudem per Automatik zwischen verschiedenen Programmen wechseln, so kann auf verschiedene Verkehrsbelastungen (Berufs-, Tages- und Nachtverkehr usw.) reagiert werden.

Weitere Elemente des Straßenverkehrs, die den Verkehrsfluss direkt beeinflussen können, sind zum Beispiel Fußgängerampeln mit Anforderung oder einem Zebrastreifen. Es ist zwar wichtig, möchte man ein ganzheitliches System implementieren, diese Elemente zu betrach-

ten, in dieser Ausarbeitung werden diese Elemente jedoch noch außen vorgelassen.

Der im Folgenden vorgestellte Optimierungsansatz mit dem Hopfield-Netzwerk betrachtet hingegen ein komplexes System an Straßen. Sozusagen einen Teilgraph des Verkehrsnetzes. Angewandt wurde das System im Zuge der Olympischen Spiele in Atlanta 1996¹².

NOTES: umlaufszeit 45 - 120 sek, Situation in HH smart city atms
atlanta olympic anforderung/bedarfsampeln

¹² John F. G. and Khalid J. E., *Traffic Management Applications of Neural Networks*,
<https://www.aaai.org/Papers/Workshops/1993/WS-93-04/WS93-04-012.pdf>

2 NEURONALE NETZE

Neuronale Netze beziehen sich eigentlich auf biologische Nervenzentren. In der Informatik hat man nur das grundlegende Prinzip adaptiert und versucht Strukturen zu erzeugen, die anhand von Daten lernen können. Es wird nicht versucht ein *echtes* Gehirn nachzuempfinden, in dem Metastrukturen auftreten, in denen bestimmte Bereiche bestimmte Aufgaben haben. An dem Bereich wird natürlich auch geforscht, ist aber Gegenstand der Computational Neuroscience TO-DO.

2.1 Grundprinzipien

Die nachfolgenden Ausführungen und Grafiken zum Schaffen eines grundlegenden Verständnisses für neuronale Netze, die hier verwendet werden, basieren im wesentlichen auf dem E-Book “Neural Networks and Deep Learning”¹³.

Im Rahmen des maschinellen Lernens stellen die neuronalen Netze einen elementaren Ansatz dar, der in vielen weiteren Modellen Verwendung findet. Neuronale Netze wie das maschinelle Lernen an sich stellen eine andere Herangehensweise dar, als die klassischer, deterministischer Algorithmen. Anstatt dem System eine eindeutige Abfolge von Anweisungen mitzuteilen, um eine konkrete Problemstellung zu lösen, wird ein Modell definiert und dieses mit verschiedenen Beispielen konfrontiert - die Beispiele sind dabei Tupel aus Eingangsgröße und erwarteter Ausgangsgröße. Die Dimensionen von Eingangs- und Ausgangsgröße können sich dabei gleich sein, müssen es aber nicht.

¹³ Michael Nielsen, Neural Networks and Deep Learning.
(<http://neuralnetworksanddeeplearning.com/index.html>)

So können als Eingabe Bilder dienen und als Ausgaben konkrete Klassen, um beispielsweise Hunde von Katzen unterscheiden zu können. TODO Anstatt nun algorithmisch zu definieren, was einen Hund von einer Katze unterscheidet, wird es dem zuvor erstellten Modell überlassen, anhand der gegebenen Eingaben und erwarteten Ausgaben, eigenständig Regeln abzuleiten, um mit dessen Hilfe auch unbekannte Eingaben klassifizieren zu können.

Dieser Ansatz wird auch als *Soft Computing* TODO bezeichnet.

2.1.1 Perzeptron

Als elementaren Bestandteil eines neuronalen Netzes dient das *Perzeptron* - dieses stellt die kleinste Einheit eines neuronalen Netzes dar und wird auch als "künstliches Neuron" bezeichnet.

Grundsätzlich akzeptiert ein Neuron einen beliebig großen Input bestehend aus Features x_1, x_2, \dots, x_n und berechnet daraus ein Ergebnis.

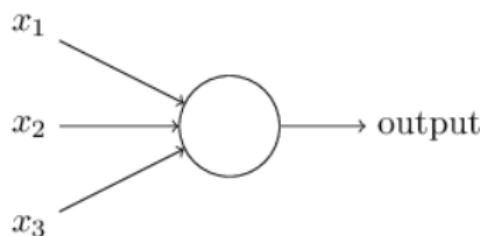


Abbildung 3: Perzeptron

Im gezeigten Bild ist beispielsweise ein Neuron dargestellt, das drei Inputgrößen akzeptiert und daraus einen Output produziert. Um den Output zu berechnen werden Gewichte (*engl. weights*) eingeführt. Ob das Neuron 0 oder 1 als Output liefert, hängt dann davon ab, ob die gewichtete Summe der Eingangsgrößen einen zu definierenden Schwellwert überschreitet.

Dies kann anhand der nachfolgenden Formel verdeutlicht werden:

$$\text{output} = \begin{cases} 0 & \text{if } \sum \omega_i x_i \leq \text{threshold (Schwellwert)} \\ 1 & \text{if } \sum \omega_i x_i > \text{threshold (Schwellwert)} \end{cases}$$

Abbildung 4: Berechnung des Outputs.

Dies ist das grundlegende Modell. Grundsätzlich kann sich das Perzeptron auch als ein “Entscheidungs-Unterstützer” vorstellen, der eine Entscheidung trifft, in dem er konkrete Fakten mit einem bestimmten Gewicht versieht.

Das gezeigte Modell ist augenscheinlich sehr simpel und noch sehr weit von dem entfernt, was als ein neuronales Netz bezeichnet werden würde. Es ist allerdings ohne Weiteres denkbar, das gezeigte Modell komplexer zu gestalten, indem mehrere Perzeptrons miteinander verknüpft werden, so dass beispielsweise das nachfolgende Netzwerk entstehen könnte:

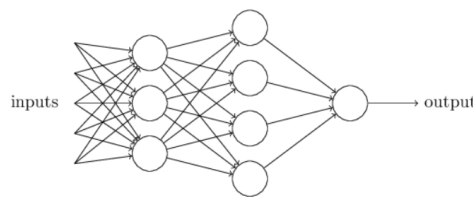


Abbildung 5: Mehrschichtiges neuronales Netz.

In Grafik⁴ wurde ein Schwellwert eingeführt, der überschritten werden muss, damit ein Perzeptron aktiviert wird. Um das Modell zu vereinheitlichen, kann der *Bias* definiert werden, der den negativen Schwellwert darstellt. Durch diese Maßnahme kann die Aktivierungsfunktion des Perzeptrons dann geschrieben werden als:

$$\text{output} = \begin{cases} 0 & \text{if } \sum \omega \cdot x + b \leq 0 \\ 1 & \text{if } \sum \omega \cdot x + b > 0 \end{cases}$$

Abbildung 6: Berechnung des Outputs bei Verwendung eines Bias.

Inhaltlich kann der Bias als ein Maß verstanden werden, aus dem hervorgeht, wie *leicht* ein Perzeptron aktiviert werden kann. Nimmt der Bias einen großen Wert an, so kann das Perzeptron einen Wert von 1 annehmen, auch wenn das Produkt aus den Gewichten und den Eingangsgrößen einen negativen Wert annimmt. Gleiches gilt selbstverständlich auch für einen kleinen Bias, der zur Folge hat, dass ein Perzeptron träger reagiert.

NOTE: wiederholung gehirn menschlich lernen

2.1.2 Sigmoid-Neuronen

Eine Weiterentwicklung des zuvor vorgestellten Modells stellen Sigmoid-Neuronen dar. Diese Weiterentwicklung wird dann erforderlich, wenn das Anpassen der Gewichte - also letztlich das Lernen - betrachtet wird. Dabei ist das Ziel, dass eine kleine Anpassung eines Gewichts auch nur eine kleine Änderung des Outputs zur Folge hat. Das zuvor betrachtete Perzeptron ist lediglich in der Lage 0 oder 1 als Output zu liefern, so dass Änderungen an den Gewichten keine stetige Änderung des Outputs zur Folge haben, sondern folgenlos bleiben können bis irgendwann ein Sprung von 0 auf 1 oder umgekehrt stattfindet, was wiederum eine große Änderung darstellt.

Die Weiterentwicklung besteht nun in einer Verfeinerung der Aktivierungsfunktion. Anstatt eine Sprungfunktion^{8b} zu verwenden, die lediglich 0 und 1 als Funktionswert annehmen kann, wird die Sigmoid Funktion^{8a} eingeführt, die die folgende Form hat:

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}}$$

Abbildung 7: Sigmoid-Funktion.

Der entscheidende Unterschied kann an den beiden nachfolgenden Grafiken verdeutlicht werden, die jeweils die Kurve der entsprechenden Funktion darstellen:

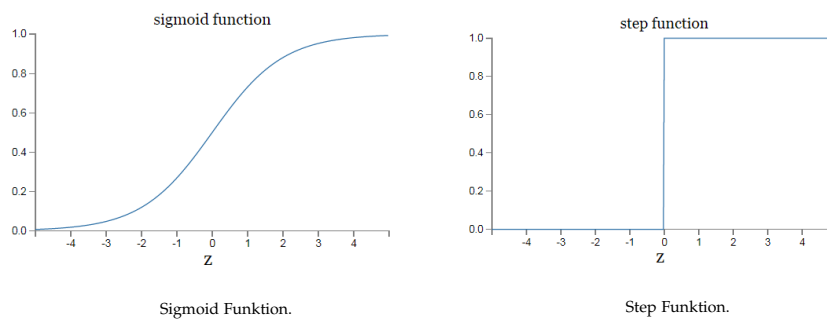


Abbildung 8: Vergleich der Aktivierungsfunktionen *Sigmoid* und *Step-Funktion*

2.2 Arten und Anwendungsgebiete // Architektur

Mit diesen Bestandteilen als Ausgangspunkt können nun tatsächlich konkretere Neuronale Netze und deren Architekturen eingeführt werden. Neuronale Netze bestehen üblicherweise aus mehreren Schichten, den *Layern*. Diese lassen sich grundsätzlich in drei Kategorien aufteilen: Input, Hidden und Output. Neuronale Netze beinhalten für gewöhnlich ein Input-Layer und ein Output-Layer sowie dazwischen beliebig viele Hidden-Layer. Die Form der Input- und Output-Layer ist dabei sehr naheliegend: das Input-Layer hat die gleiche Struktur wie die des Inputs und das Output-Layer hat entsprechend die gleiche Struktur wie der Output.

Angenommen es sollen Bilder der Größe 28×28 Pixel klassifiziert werden und es gibt 10 mögliche Klassen, dann besteht das Input-Layer aus $28 \times 28 = 784$ Neuronen und das Output-Layer aus 10 Neuronen.

Lediglich der Bereich zwischen Input- und Output-Layer - die Hidden-Layer - lässt sich nicht ohne Weiteres aus dem Input oder

dem Output ableiten. Es gibt lediglich Heuristiken, die beim Design der Hidden-Layer angewandt werden können, allerdings keine konkreten Regeln, die befolgt werden müssen. Diese Struktur kann anhand des nachfolgenden Bilds verdeutlicht werden, bei dem - um die Übersichtlichkeit zu wahren - das Input-Layer etwas komprimiert dargestellt wird:

NOTES: python tutorial uh -> softcomputing

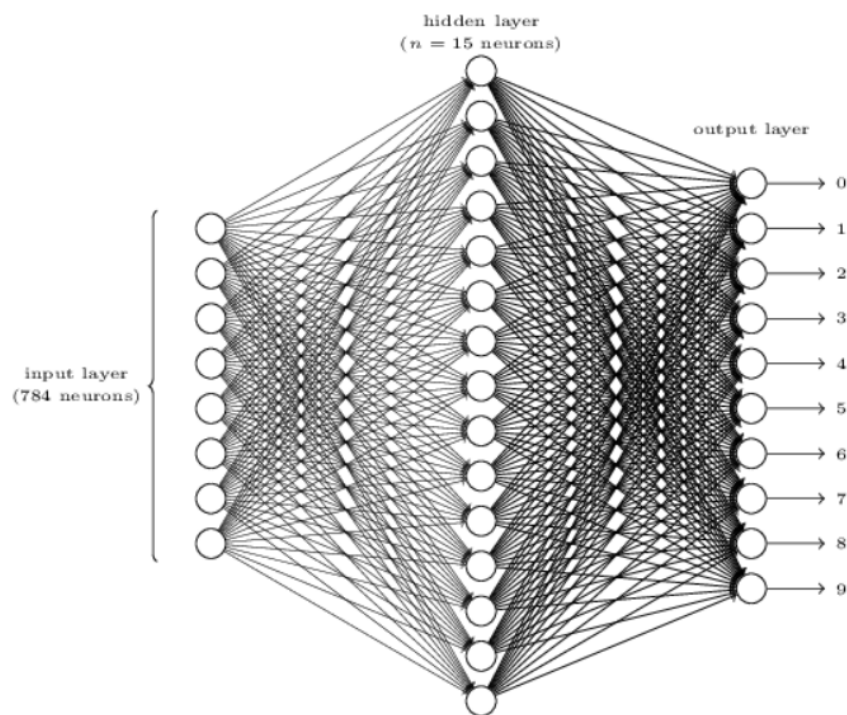


Abbildung 9: Hidden-Layer Darstellung.

2.3 Lernen - Das Anpassen der Gewichte

Das Lernen stellt den zentralen Ansatz von neuronalen Netzes dar. Eng im Zusammenhang mit dem Lernen steht eine Kosten-Funktion, die häufig auch als Verlust-Funktion bezeichnet werden kann. Diese stellt letztlich den Fehler zwischen dem Erwartungswert und dem tatsächlichen Wert, den das neuronale Netz berechnet, dar. Mathema-

tisch betrachtet ist das grundlegende Prinzip des Lernens, diese Funktion zu minimieren, also zu gewährleisten, dass die Abweichungen zwischen Erwartungswert und tatsächlichem Wert möglichst gering sind. Es sind grundsätzlich viele verschiedene Verlust-Funktionen denkbar, eine, die jedoch eine breite Verwendung findet, ist die quadratische Kosten-Funktion - auch als *mean squared error* (MSE) bezeichnet.

$$C(w, b) \equiv \frac{1}{2n} \sum_x \| y(x) - a \|^2$$

Abbildung 10: Mean Squared Error (MSE).

Dabei beschreiben w und b die Gewichte bzw. die Bias des neuronalen Netzes und n stellt die Anzahl der Trainingsdaten dar. Der Vektor a beschreibt den Output des Netzes und $y(x)$ stellt den Erwartungswert zu einem Input x dar. Das Ziel besteht nun darin, die Gewichte und Bias so zu manipulieren, dass die gezeigte Funktion einen möglichst kleinen Wert annimmt.

– Wie Neuronen funktionieren Die Arbeitsweise ist erstaunlich einfach: Immer wenn die Summe der Eingangssignale einen bestimmten Schwellenwert überschreitet, sendet die Zelle ein Ausgangssignal. Bleibt die Eingangserregung unter der Grenze, reagiert die Zelle nicht. Am Ende der axonalen Verzweigungen stellt eine besondere Struktur, die Synapse, den Kontakt zu anderen Neuronen her. Die meisten Synapsen funktionieren so: Je stärker die Erregung im Axon, desto mehr Moleküle einer Überträgersubstanz schüttet die Synapse aus. Der Überträgerstoff (Neurotransmitter) wandert zur Zielzelle. Manche Neurotransmitter erhöhen die elektrische Erregung der angefunkelten Zelle, andere hemmen sie.

Die Netzwerke der Erinnerung Das Netzwerk der Neuronen in der Großhirnrinde (wegen ihrer Form auch „Pyramidenzellen“) ist im Gegensatz zu einem Computer nicht nach einem detaillierten Plan geknüpft, sondern weitgehend zufällig organisiert. Sind miteinander verbundene Zellen gemeinsam aktiv, verstärken sich die Synapsen. Demnach aktiviert das Lernen immer wieder eine Anzahl miteinander verknüpfter Pyramidenzellen. Deren Verbindung verstärkt sich nach und nach, „neuronale Netzwerke“ entstehen. Je öfter sich der synaptische Lernprozess wiederholt, desto leichter lässt sich dieses „Netzwerk“ aktivieren. –

es gibt sie in allen formen und farben. nns zu listen, wäre ein akt der unmöglihekit. im netz finden sich links wie:

The mostly complete chart of Neural Networks, explained <https://towardsdatascience.com/mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>

und liste auf wikipedia https://en.wikipedia.org/wiki/Types_of_artificial_neural_networks

2.4 Überwachtes und Unüberwachtes Lernen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3 VERKEHRSMODELLE

Verkehrsmodelle bestehen zunächst aus dem Streckennetz oder Verkehrsgraph und den Teilnehmern. Natürlich kann es, je nach Anwendungsfall noch Sinn machen, noch andere Einflussfaktoren mit zu berücksichtigen, darauf wird jedoch im Folgenden jedoch verzichtet. Die Fragestellung: Wie lassen sich diese System in datenverarbeitenden Systemen abbilden?

Bei Verkehrsflussproblemen gibt es zwei grundsätzliche Ansätze¹⁴,¹⁵. Einmal die mikroskopischen Modelle¹⁶, die das Gesamtsystem sehr feingranular abbilden. Sie basieren auf individuellem Verhalten und bilden Verkehrsteilnehmer als einzelne Objekte ab. Eine prominente Gruppe an Vertretern dieser Modelle sind die Car-Following models"[6, 17, 57] in denen der Fahrer seine Beschleunigung an die Bedingungen vor ihm anpasst. Dem gegenüber stehen die makroskopischen Modelle¹⁷, die mehr Interesse an Durchschnittsverhalten haben. Sie betrachten etwas wie Verkehrsdichte und Durchschnittsgeschwindigkeit.

Mit dem mikroskopischen Ansatz ist eine sehr präzise Arbeit möglich, die jedoch viel Rechenleistung erfordert, da die Position jedes Objekt in jedem Schritt neu berechnet werden muss. Wohingegen die makroskopischen Ansätze zwar etwas unpräziser sind, dafür, weil sie weniger Details haben, entsprechend günstiger im Bezug auf die benötigte Rechenleistung.

¹⁴ Sven Maerivoet and Bart De Moor, 2008. Traffic Flow Theory. Department of Electrical Engineering ESAT-SCD (SISTA), Katholieke Universiteit Leuven

¹⁵ Springer, 2008. Traffic Flow for 1-D. Pedestrian Dynamics, Feedback Control of Crowd Evacuation

¹⁶ Wikipedia, [Microscopic Traffic Flow Model](#)

¹⁷ Wikipedia, [Macroscopic Traffic Flow Model](#)

Mikroskopische Modelle

In contrast to macroscopic models, microscopic traffic flow models simulate single vehicle-driver units, so the dynamic variables of the models represent microscopic properties like the position and velocity of single vehicles.

We present the well known car-following microscopic traffic flow model. In [93], a 2-D version of this model was used for pedestrian flow in 2-D space. To derive the 1-D model, first assume cars can not pass each other. Then the idea is that a car in 1-D can move and accelerate forward based on two parameters; the headway distance between the current car and the one in front, and their speed difference. Hence, it is called following, where a car from behind follows the one in front, and this is the anisotropic property. This property is also desirable in macroscopic models, since it reflects the actual observed behavior of traffic flow [23]. Suppose the n th car location is $x_n(t)$, then the nonlinear model is given by

The acceleration of the current car depends on the front car speed and location, c is the sensitivity parameter. Integrating the above yields (2.2) Since by the definition of the density (number of cars per unit area)

and the integration constant d_n is chosen such that at jam density p_m , the velocity is zero. Then for steady-state we get

p_m We see that for $p \rightarrow 0$ we get in trouble, but from observations in low traffic densities, car speed is the maximum allowed speed, hence we can assume $v = v_{\max}$, which is the maximum allowed speed.

Makroskopische Modelle

Traffic Flow Theory In this section we will cover the vehicle traffic flow fundamentals for the macroscopic modeling approach. The relation between density, velocity and flow is presented for traffic flow. Then we derive the conservation of vehicles, which is the main governing equation for scalar macroscopic traffic models. Finally, the velocity–density functions that makes the conservation equation a function of only one variable (density) are given.

2.4.1 Flow In this section, we will illustrate the close relationship between the three variables: density, velocity and traffic flow. Suppose there is a road with cars moving with constant velocity v_0 , and constant density ρ_0 such that the distance between the cars is also constant as shown in the Fig. 2.1a. Now let an observer measure the number of cars per unit time τ that pass him (i.e. traffic flow f). In τ time, each car has moved $v_0\tau$ distance, and hence the number of cars that pass the observer in τ time is the number of cars in $v_0\tau$ distance, see Fig. 2.1b. Since the density ρ_0 is the number of cars per unit area and there is $v_0\tau$ distance, then the traffic flow is given by $f = \rho_0 v_0$ (2.5) This is the same equation as in the time varying case, i.e., $f(p, v) = p(x, t)v(x, t)$.

Fig. 2.1. (a) Constant flow of cars; (b) Distance traveled in τ hours for a single car To show this, consider the number of cars that pass point $x = x_0$ in a very small time Δt . In this period of time the cars have not moved far and hence $v(x, t)$, and $p(x, t)$ can be approximated by their constant values at $x = x_0$ and $t = t_0$. Then, the number of cars passing the observer occupy a short distance, and they are approximately equal to $p(x, t)v(x, t)\Delta t$, where the traffic flow is given by (2.6).

2.4.2 Conservation Law The models for traffic, whether they are one-equation or system of equations, are based on the physical principle of conservation. When physical quantities remain the same during some process, these quantities are said to be conserved. Putting this principle into a mathematical representation will make it possible to predict the densities and velocities patterns at future time. In our case, the number of cars in a segment of a highway $[x_1, x_2]$ are our physical quantities, and the process is to keep them fixed (i.e., the number of cars coming in equals the number of cars going out of the segment). The derivation of the conservation law is given in [26, 37], and it is presented here for completion. Consider a stretch of highway on which cars are

– ungern Another type of microscopic models are the Cellular Automata or vehicle hopping which differs from Car-Following in that it is a fully discrete model. It considers the road as a string of cells which are either empty or occupied by one vehicle. One such model is the Stochastic Traffic Cellular Automata, given in [75]. –

DAS HOPFIELD NETZWERK UND DER VERKEHRS- FLUSS (15–20 MINUTEN)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Ampelschaltung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Hopfield Modell

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

RESULTATE (5-6 MINUTEN)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

FAZIT (5-6 MINUTEN)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. technologische weiterentwicklung autonomes fahren

ABBILDUNGSVERZEICHNIS

Abbildung 1	Signalzeitenplan	6
Abbildung 2	Rückstau an einer Kreuzung	7
Abbildung 3	Perzeptron	10
Abbildung 4	Berechnung des Outputs.	11
Abbildung 5	Mehrschichtiges neuronales Netz.	11
Abbildung 6	Berechnung des Outputs bei Verwendung eines Bias.	11
Abbildung 7	Sigmoid-Funktion.	12
Abbildung 8	Vergleich der Aktivierungsfunktionen <i>Sigmoid</i> und <i>Step-Funktion</i>	13
Abbildung 9	Hidden-Layer Darstellung.	14
Abbildung 10	Mean Squared Error (MSE).	15

TABELLENVERZEICHNIS