Maschinelles Lernen und Mathematik im autonomen Fahren

Abbildung 66: Beispiel einer sogenannten semantischen Segmentierung, berechnet von einem tiefen neuronalen Netz. Für jedes Bildpixel wird eine Klassenzugehörigkeit geschätzt, durch eine Zuordnung von Klassen zu Farben entsteht ein solches Segmentierungsbild.

Maschinelles Lernen, insbesondere die Verfahren des Deep Learning basierend auf sogenannten tiefen neuronalen Netzen, werden im autonomen Fahren und seinen Vorstufen hauptsächlich in drei Anwendungsfeldern verwendet:

1 Bei der Umfeldwahrnehmung mittels der Verarbeitung von Kamera und Sensordaten
2 Zum Erlernen und Umsetzen einer Fahrstrategie
3 Zur Überwachung des Innenraums

In diesen Anwendungsfeldern haben die neuen Deep-Learning-Algorithmen, entwickelt in den 2010er Jahren, Vorhersagegenauigkeiten entwickelt die mit vorherigen Methoden unerreichbar schienen. Im Folgenden werden die 3 Anwendungsfelder des maschinellen Lernens genauer beschrieben:

Umfeldwahrnehmung durch die Verarbeitung von Sensordaten

Bei der Umfeldwahrnehmung spielen hauptsächlich drei Sensoren eine große Rolle: Kamera-, LiDAR- und Radar-Sensoren. Exotischere Varianten wie Wärmesensoren sind aktuell weniger verbreitet. Auf Basis der Sensoren werden andere Verkehrsteilnehmer, die Straße, Hindernisse etc. detektiert. Die Gesamtheit der Detektionen wird in ein 3D Umfeldmodell überführt und liefert die Grundlage für die nachfolgende Fahrplanung und Fahrstrategie.

Kamera. Kameradaten werden durch 3 Farbwerte in jedem Pixel repräsentiert, wodurch ein HD Bild mehrere Millionen numerische Werte beinhaltet. Anhand dieser Pixelwerte möchte man Objekte im Bild detektieren und einer Klasse (PKW, Lastwagen, Bus, Mensch,

Fahrrad, etc.) zuordnen. Mathematisch gesehen möchten wir eine Funktion finden, die den gegebenen Pixelwerten Positionen und Klassen aller Objekte im Bild zuordnen. Eine (klassische) händische Modellierung basierend auf selbst erdachten Regeln erreicht hier nicht die gewünschte Detektionsgenauigkeit. Die Methoden des Deep Learnings haben diese Anwendung revolutioniert.

Beim Deep Learning werden im übertragenen Sinne Schablonen von hierarchisch organisierten Funktionen mit einer großen Zahl von freien (lernbaren) Parametern verwendet. Die Parameter werden in einer mathematischen Optimierung daraufhin angepasst, dass sie einen vorgegebenen funktionalen Zusammenhang erlernen (mehr Information dazu befindet sich weiter unten in diesem Abschnitt). So kann ein tiefes neuronales Netz beispielsweise Lernen, jedem Bildpixel seine Klassenzugehörigkeit zuzuordnen.

Dies ist in der Abbildung 66 dargestellt. Dabei wird zur Illustration jeder Klasse eine Farbe zugeordnet. Um eine solche Funktion zu erlernen, bedarf es große Mengen sogenannter Labels. Aktuell werden neuronale Netze für diese Aufgabe mit Größenordnungen von 10.000en von Bildern trainiert, für jedes dieser Bilder haben Menschen die Objekte im Bild mit Kantenzügen umrandet. Die Akquise solcher Labels ist zeitaufwändig und kostspielig, jedoch für eine akkurate Umfeldwahrnehmung derzeit noch unumgänglich.

Radar. Ein weiterer, in der Automobilindustrie weit verbreiteter Sensor ist Radar (**RA**dio **D**etection **A**nd **R**anging). Obwohl der Radar keine Bilder im klassischen Sinne liefert, hat der Sensor mehrere Vorteile: die Qualität der gemessenen Daten hängt nicht von der Sichtbarkeit ab, und so liefert Radar nutzbare Daten auch in Szenarien wo Kamera oft Probleme hat, wie etwa unter schlechter Beleuchtung, Nebel oder im schlechten Wetter. Als weiterer Vorteil gilt, dass man durch Radar nicht nur die Position, sondern auch die Geschwindigkeit des Objekts direkt messen kann. Der Radar funktioniert auf Basis von elektromagnetischen Wellen: ein Signal wird ausgesendet und wenn dieser auf ein Hindernis trifft wird er zurückreflektiert. Dieses Echo wird dann mit dem gesendeten Signal verglichen, um den Zeitversatz zwischen beiden Signalen zu messen. Der Zeitversatz gibt Informationen über die Distanz des reflektierenden Hindernisses.

Mehrere Messungen erlauben sogar, die Distanzänderung über die Zeit, d.h. die Geschwindigkeit zu messen. Durch den Einsatz mehrerer Radar-Antennen nebeneinander ist es zusätzlich möglich, Informationen über den Beobachtungswinkel und Höhe eines Reflektionsobjektes zu extrahieren. Geeignete mathematische Verfahren überführen die empfangenen Zeitsingale in die Positionsdaten der reflektierenden Ziele, sogenannte Detektionspunkte, die eine Grundlage für Erkennung der Objekten (Fußgänger, Autos) bilden. Die Verarbeitung der Radardaten war über die Jahre hinweg durch klassische Signalverarbeitungsmethoden dominiert, doch in letzten Jahren setzten sich die Machine Learning Techniken auch hier durch. Da die Radardaten aber sehr unterschiedlich zu von Kameras aufgenommenen Bildern sind, muss man dafür auch die Deep Learning Technologie anpassen eine Herausforderung an der auch in der Wuppertaler Niederlassung von Aptiv gearbeitet wird. Für die Entwicklung geeigneter Deep Learning so wie auch der klassischen Signalverarbeitung Methoden ist das Verständnis der Mathematik hinter diesen Technologien essenziell.

Lidar. Neben Kamera und Radar ist beim autonomen Fahren auch ein Lidar (**Li**ght **D**etection **A**nd **R**anging) Sensor von großer Bedeutung. Ähnlich wie beim Radar, misst auch Lidar zurückreflektierte Signale: in diesem Fall geht es aber statt einer elektromagnetischer Welle um einen Lichtstrahl, der mit entsprechender Verzögerung reflektiert wird. Diese Zeitverzögerung wird ebenfalls in die Ortsinformation des Reflektionspunktes überführt. Durch mehrere ausgesendete Lichtstrahlen kann eine Vielzahl von Reflektionspunkten generiert werden, die eine drei-dimensionale Punktwolke bilden, in der man (je nach Qualität des Lidars) die Umgebung gut erkennen kann. Die Vielzahl der Reflektionspunkte und die Genauigkeit der Positionsmessung ist einer der klaren Vorteile der Lidarsensoren. Auf anderer Seite, repräsentiert die Verteilung der Punktdaten im dreidimensionalen Raum eine Herausforderung für viele der Deep Learning Methoden, die oft spezifisch für zweidimensionale Bilder entwickelt wurden. Bei der Entwicklung neuer Algorithmen ist die Laufzeit der Algorithmen genauso zu berücksichtigen wie die Qualität der Erkennung: wenn der Algorithmus zu lange braucht, würden die Ergebnisse zu spät vorliegen, um in der Situation angemessen zu reagieren.

Ob Kamera, Radar, oder Lidar: jeder dieser Sensoren bringt mit sich neben den Vorteilen auch einige Nachteile. Gerade das macht ein weiteres Forschungsgebiet wichtig: die Sensorfusion. Mit mehreren unterschiedlichen Sensoren läßt sich das Risiko, dass alle Sensoren in demselben Moment fehlschlagen minimieren.

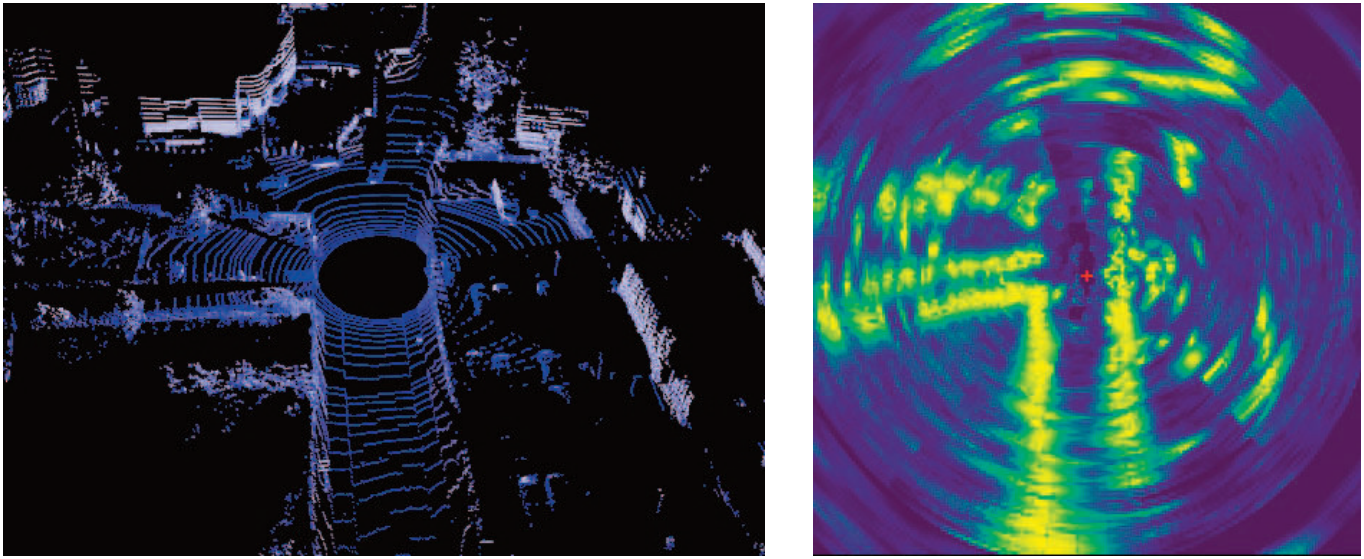

Abbildung 67: Beispiel der Daten aus Lidar (links) und der Daten aus Radar (rechts)

Fahrer-Assistenz und automatisierte Fahrzeug-Steuerung

Anhand der aus Sensordaten extrahierten Informationen über andere Verkehrsteilnehmer (wie z.B. Fußgänger, Fahrräder, Autos) und über die Umgebung, kann das Fahrzeug automatisiert Maßnahmen treffen. Dabei kann es je nach Level der Automatisierung um z.B. eine Warnung an den Fahrer, automatisches Notbremsen oder sogar um volle Steuerung des Fahrzeugs gehen. Vor allem bei komplexeren Aufgaben die sich voller Automatisierung nähern, kann es schwierig sein, das Verhalten des Autos nur durch Regeln programmieren, und ein weiteres Teil von Maschinellem Lernen kommt zum Einsatz: Reinforcement Learning. Bei dieser Methode geht es darum, anhand aller zu dem Zeitpunkt verfügbaren Informationen (Umgebung, Positionen und Bewegungsrichtungen anderer Verkehrsteilnehmer, aber auch aller solchen verfügbaren Daten aus der Vergangenheit) optimale Entscheidung zu treffen. Dabei müssen nicht nur die sofortige, sondern auch die weiterfolgende Konsequenzen dieser Entscheidung in Betracht genommen werden.

Um die Entscheidung als eine mathematische Funktion von allen verfügbaren Informationen zu repräsentieren, greift man wieder oft auf die tiefe Netzwerke zu. Die Konsequenzen dieser Entscheidung werden dann durch einen positiven numerischen Wert bewertet falls die erwünscht sind (wie z.B. das Ankommen ins Ziel), und mit einem negativen Wert falls unerwünscht (wie z.B. verursachen eines Unfalls). Mit Hinsicht auf die Sicherheit trainiert man dieses Netzwerk in der Regel nur in einer Simulation. Während des Trainings wird das Netzwerk mit einer Verkehrssituation konfrontiert und gibt anhand der verfügbaren Informationen eine Empfehlung. Die empfohlene Handlung kann dann in der Simulation ausgeführt werden, muss aber nicht notwendig: mit gewisser Wahrscheinlichkeit trifft das Programm eine andere Entscheidung um die unbekannte Folgen zu untersuchen. Anhand der Folgen werden danach die Parameter des neuronalen Netzwerks so geändert, dass das Netzwerk künftig in ähnlichen Situationen Empfehlungen mit möglichst positiv-bewerteten Konsequenzen liefert.

Eine andere Methode die optimale Steuerung zu lernen heißt Imitation Learning. Bei dieser Methode werden als Daten aufgenommene (nicht-autonome) Fahrten benutzt: während des Trainings bekommt dann das neuronale Netzwerk die aufgenommene Daten aus einem Szenario als Eingang und versucht anhand dieser die aufgenommene Entscheidung des menschlichen Fahrers zu imitieren.

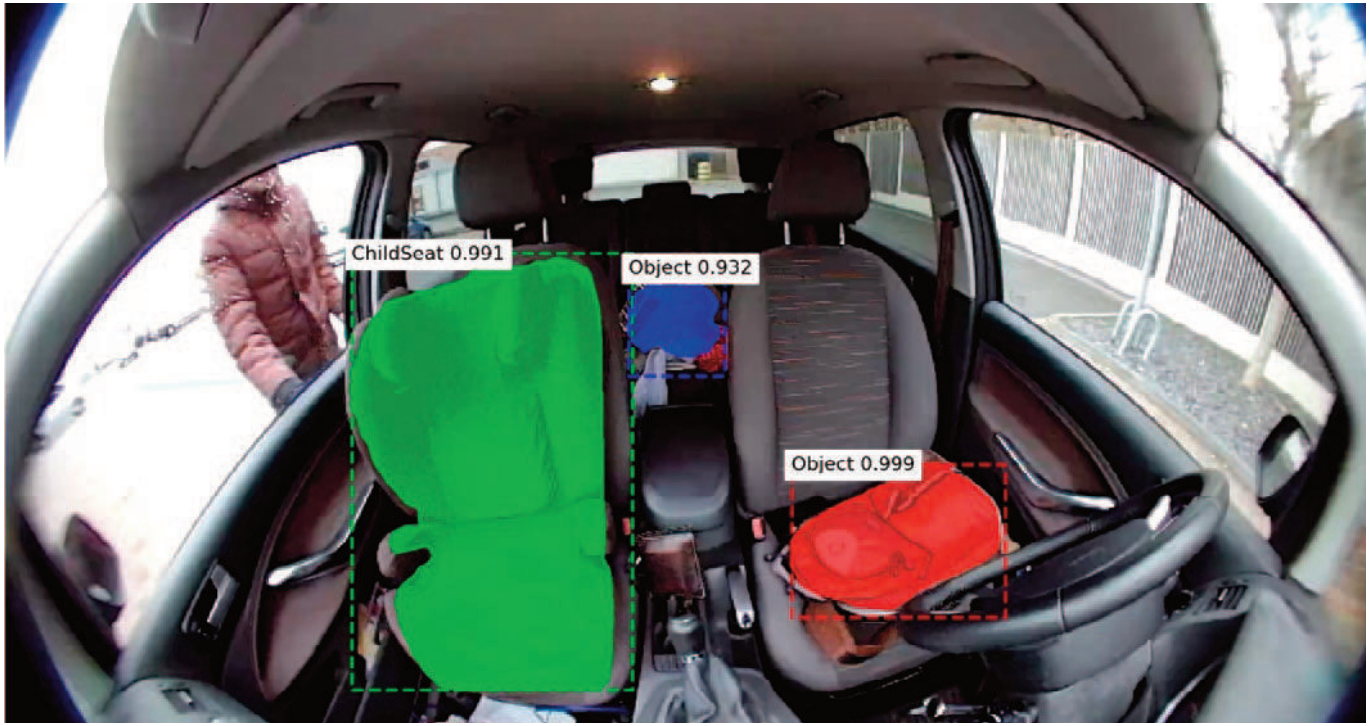

Abbildung 68: Beispiel einer Instanzsegmentierung im Innenraum eines Autos. Zuerst werden Objekte lokalisiert und von Rechtecken umrandet. Anschließend wird die Instanz innerhalb des Rechtecks segmentiert.

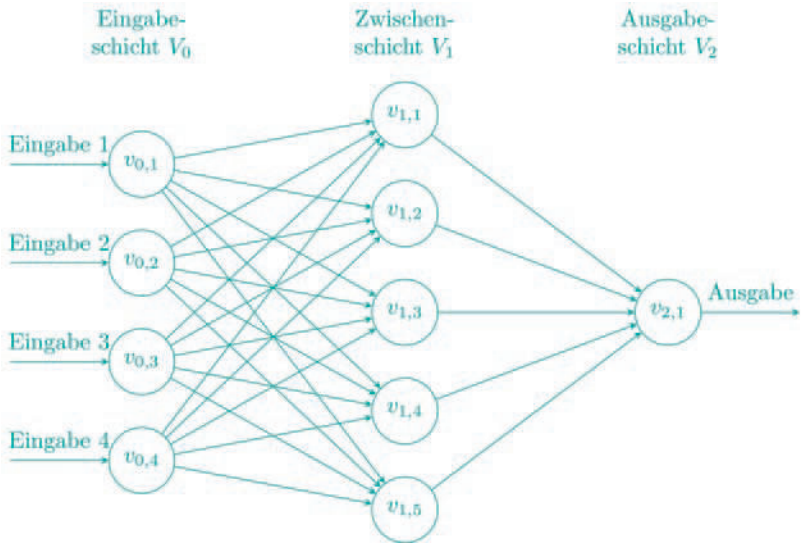

Abbildung 69: Skizze eines Feedforward-Netzes mit 3 Lagen. Tiefe neuronale Netze in der Bildverarbeitung sind im allgemeinen um viele Größenordnungen komplexer.

Überwachung des Innenraums

Ähnlich wie bei der Erkennung des Umfeldes mittels Kameras, ist die Erkennung der Personen und Objekte im Innenraum ebenfalls von großer Bedeutung, sowohl für das automatisierte Fahren als auch für Assistenzsysteme. Dabei kommen im Innenraum ebenfalls Kameras und Deep Learning zum Einsatz, siehe Abbildung 68. Die Detektion von Personen, die Erkennung ihrer Verfassung, die Detektion von Kindersitzen, Kindern und Haustieren sind beispielsweise von höherem Interesse. In Vorstufen des automatisierten Fahrens muss z.B. ermittelt werden, ob der Fahrgast im Fahrersitz in der Verfassung ist, das Steuer zu übernehmen. Auch könnten akut auftretende Symptome einer Erkrankung detektiert werden. Das Fahrzeug kann außerdem darauf Hinweisen, wenn ein Kind oder ein Haustier im Auto zurückgelassen wird. In Mobilitätslösungen, die den ÖPNV durch kleinere autonome Gruppentaxis voraussichtlich in Zukunft ergänzen werden, spielt es auch eine Rolle, die Fahrgäste zu überwachen um beispielsweise Übergriffe zu erkennen.

Erlernen funktionaler Zusammenhänge mittels Deep Learning

Nachdem wir nun viele Beispiele für die Anwendung des Deep Learning im automatisierten Fahren gesehen haben, beleuchten wir noch einmal die Rolle der Mathematik im Deep Learning, denn diese geht über die bloße Approximation funktionaler Zusammenhänge hinaus. In einem neuronalen Netz, der Funktionsschablone des Deep Learnings, finden hauptsächlich arithmetische Operationen (Additionen und Multiplikationen statt) statt. Neuronen bilden die kleinsten Einheiten der neuronalen Netze und erhalten als Eingabe ein Signal in Form einer Zahl von anderen Neuronen, zu denen eine Verbindung in Form einer sogenannten Kante (als Pfeil in der Abbildung 69) besteht. Auf jeder Kante lebt ein sogenanntes Gewicht, ein freier Parameter der neuronalen Netzes, welcher gelernt wird. Das Gewicht wird mit dem Signal multipliziert (also gewichtet) und alle gewichteten Signale werden im Neuron summiert und anschließend wird eine Schwellwertfunktion auf die gewichtete Summe angewandt. Wird ein bestimmte Gesamtsumme überschritten, wir das Neuron aktiviert und leitet ein Signal an die anderen mit ihm verbundenen Neuronen weiter. Diese modellieren ist biologisch inspiriert. Neuronalen Netzen für die beschriebenen Anwendungen sind meistens sogenannte Feedforward-Netze. Bei diesen sind die Neuronen gruppenweise in Lagen organisiert und eine Lage leitet Signale nur an nachfolgende Lagen (also vorwärts) weiter (vgl. Abbildung 69).

Abgesehen davon, dass neuronale Netze nichts weiter als Mathematik sind, spielen verschiedene klassische Disziplinen der Mathematik bei folgenden Fragestellungen eine gewichtige Rolle, hier nennen wir die prominentesten Beispiele:

1. Wie viele lernbare Gewichte werden benötigt, damit ein neuronales Netz theoretisch einen gegebenen funktionalen Zusammenhang zu einer gewünschten Genauigkeit approximieren kann? Die grobe Richtung lautet: Mehr Gewichte entsprechen mehr Kapazität und erlauben, komplexe Zusammenhänge darzustellen. Um diese Frage jedoch genauer und auch quantitativ zu beantworten (welche Anzahl an Neuronalen wird benötigt?), wird immer genauere Theorie entwickelt. Da jedoch auch immer wieder neue Architekturen von neuronalen Netzen entwickelt werden, wird auch neue Theorie nötig.

2. Wenn das neuronale Netz nun die Kapazität in Form von freien Gewichten besitzt, wie findet man die richtigen Gewichte? Optimale Gewichte zu finden ist ein Problem, das nicht in annähernd in akzeptabler Rechenzeit, auch nicht mit den größten Computern der Welt, zu lösen ist. Effiziente Algorithmen (sogenannte Heuristiken), die möglichst gute Lösungen für das Problem der Gewichtsfindung liefern, sind Gegenstand aktueller Forschung.

3. Wie viele Daten braucht man, damit man einen gegebenen funktionalen Zusammenhang zu einer gewünschten Genauigkeit approximieren kann? Grob gilt: mehr Daten erlauben besseres Lernen. Auch hier sind quantitative Antworten von Interesse. Diese Fragestellung und das Finden immer genauerer Antworten ist Gegenstand der Forschung in der Stochastik / Statistik.

4. Wie werden die arithmetischen Operationen in neuronalen Netzen möglichst effizient in Computern umgesetzt? Ingenieure und Informatiker entwickeln dazu geeignete Hardware, Mathematiker und Informatiker entwickeln effiziente Implementierungen (Programme) von neuronalen Netzen und außerdem immer wieder neue Netzwerkarchitekturen, die genauere Vorhersagen machen und/oder weniger Rechenoperationen dafür benötigen.