

Bachelorarbeit

**Deep Reinforcement Learning zur Minderung
von Verspätungen im ÖPNV**

Hendrik Hildebrandt
Januar 2018

Gutachter:

Prof. Dr. Katharina Morik

Dr. Thomas Liebig

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Künstliche Intelligenz
<http://www-ai.cs.tu-dortmund.de>

Inhaltsverzeichnis

1	Einleitung	1
2	Verwandte Arbeiten	4
3	Verkehr	6
4	Grundlagen	8
4.1	Maschinelles Lernen	8
4.2	Tiefgehendes Lernen	9
4.2.1	Neuronale Netze	9
4.2.2	Trainieren von neuronalen Netzen	10
4.3	Bestärkendes Lernen	11
4.3.1	Q-Lernen	12
4.4	Tiefgehendes bestärkendes Lernen	13
5	Algorithmus	15
5.1	Umsetzung	15
5.2	Performance	16
5.3	Skalierbarkeit	17
6	Evaluation	19
6.1	Szenario des Experiments	20
6.2	Ablauf des Experiments	21
6.3	Experiment – Keine Beeinflussung von Ampeln	22
6.4	Experiment – Ampelbeeinflussung bei Busverspätungen	24
6.5	Experiment – Tiefgehendes Bestärkendes Lernen	27
6.6	Weitere Experimente	30
7	Fazit	32
	Abbildungsverzeichnis	33
	Literaturverzeichnis	34

Kapitel 1

Einleitung

Heutzutage ist in vielen größeren Städten der Verkehr sehr dicht [2]. Dieses Problem wird sich in den kommenden Jahren weiter verschlimmern. Denn die Anzahl der Einwohner, welche vom Land in die Stadt ziehen, wird sich weiter erhöhen [3]. In Kombination mit dem Umstand, dass viele Leute ein eigenes Auto besitzen, wird die Anzahl der Fahrzeuge auf eine kleinere Fläche konzentriert. Dies hat ein höheres Verkehrsaufkommen zur Folge und damit ein höheres Staurisiko [24].

Natürlich könnte eine erhöhte Anzahl an Staus in Kauf genommen werden, jedoch ergeben sich dadurch mehrere negative Auswirkungen. An erster Stelle wird dadurch das alltägliche Leben stark beeinflusst. Die Reisezeit für jede Fahrt erhöht sich und es fällt Zeit weg, die entweder zum Arbeiten oder für die Freizeit genutzt werden könnte [4]. Zusätzlich ist die Emission bei wiederholtem Anfahren und Anhalten am größten [13], wodurch die Belastung der Luft in den Städten weiter steigt. Somit entsteht eine erhöhte Gefahr für Menschen und Umwelt. Des Weiteren ist die Wirtschaft wegen der Globalisierung stark auf den Transport von Waren ausgelegt [16]. Durch häufig auftretende Verkehrsbehinderungen auf der gleichen Strecke steigt die Transportzeit für diese. Als Effekt könnten Produktions- und Versorgungsengpässe entstehen. Zusätzlich steigen auch die Transportkosten und letztlich auch der Preis der Güter [31].

Ein Stau entsteht größtenteils wegen eines zu hohem Verkehrsaufkommens [11]. Daher kann eine Verminderung des Staus erreicht werden, indem das Straßennetz erweitert oder es besser ausgenutzt wird. Aufgrund von starkem Platzmangel in größeren Städten ist ein Ausbau oder Neubau von Straßen schwer umsetzbar. Eine mögliche Lösung ist eine Verringerung des Platzes den eine Person bei der Fortbewegung benötigt. Am besten dafür geeignet ist der Einsatz von Schienenfahrzeugen oder Bussen [18]. Der Aufbau und Ausbau von infrastrukturellen Maßnahmen, wie einer Stadtschnellbahn, ist aus Platzgründen schwer umzusetzen. Alternativ könnte eine U-Bahn gebaut werden, jedoch haben Tunnelarbeiten einen hohen finanziellen und planerischen Aufwand. Somit ist die beste Möglichkeit, um das Verkehrsaufkommen zu verringern, die Nutzung von Bussen zu erhöhen.

Um eine Person dazu zu bringen den Bus als Reisemittel zu bevorzugen, muss man diese von den Vorteilen überzeugen und eine hohe Zufriedenheit erreichen. Dies wird zum Teil durch eine größere Zuverlässigkeit erreicht. Hierbei spielt Pünktlichkeit und eine schnellere Taktung eine Rolle [28]. Als Ausgangssituation ist dabei gegeben, dass die Busse auch von einem schlechten Verkehrsfluss betroffen sind. Wenn nun mehr Menschen mit Bussen fahren und sich dadurch die Verkehrsdichte verringert, können die Busse zuverlässiger operieren. In Folge dessen erwägen mehr Leute den Bus zu nutzen, welches wieder die Anzahl der Verkehrsteilnehmer verringert.

Eine mögliche Lösung um die Busse im Falle eines langsamen Verkehrs zu beschleunigen, wäre sie dynamisch über weniger befahrene Straßen zu der nächsten Haltestelle zu führen. Es gibt auch schon mehrere Arbeiten aus der Informatik zu dem Thema. Hierbei wird meistens mit Hilfe von Sensoren die Dichte der Fahrzeuge und die Durchschnittsgeschwindigkeit aufgezeichnet. Anhand der Daten werden dann mit Methoden, wie dem K-Nächstem-Nachbar [10] oder der Strategie Optimierung für Exponentielle Modelle [17], die staubelasteten Bereiche ermittelt. Basierend darauf wird einigen Verkehrsteilnehmern eine neue Strecke vorgeschlagen, wodurch der Verkehr auf mehrere Straßen verteilt wird. Auch wenn in den hier angeführten Arbeiten ein verbesserter Verkehrsfluss erreicht wurde, ist dieser Ansatz nicht uneingeschränkt auf Busse anwendbar. Die Haltestellen haben eine feste Position, sodass der Bus oft nur eine geringe Distanz zwischen ihnen zurücklegt [1]. Daher stellt eine alternative Strecke bei den meisten Bussen einen zu großen Umweg dar, der selten zu einer Zeitersparnis führt.

Ein Ansatz mit einer flexibleren Reaktion auf das Verkehrsgeschehen ist die Veränderung der Phasen von Lichtsignalanlagen je nach Fahrzeugaufkommen [22]. Dies wird aktuell auch schon in der Praxis angewandt, indem Sensoren in der Straße den aktuellen Verkehr erfassen und weitergeben. Ein solches System ist SCATS, welches den Platz zwischen den Fahrzeugen misst und daran die Ampelschaltung optimiert [19]. Durch diese Lösungen soll aber der gesamte Verkehrsfluss verbessert werden und sie können nicht auf einzelne Fahrzeuge beschränkt werden.

Daher wird in dieser Arbeit untersucht, ob eine gute Strategie für das Problem mithilfe von tiefgehendem bestärkendem Lernen¹ gefunden werden kann. Das Ziel ist es, die Schaltung von Lichtsignalanlagen im Hinblick auf die Busverspätung zu optimieren. Hierfür wird ein neuronales Netz für jede Kreuzung mit einer Aneplanlage aufgebaut, welches mit der Eingabe der aktuellen Busverspätungen für diese eine Ampelschaltung berechnet. Auch wenn der Autoverkehr durch die Änderung der Lichtsignalanlagen betroffen ist, steht im Vordergrund die kleinstmögliche Verspätung der Busse bei der gegebenen Verkehrssituation zu erreichen.

Die Arbeit ist wie folgt aufgebaut. Zunächst werden in Kapitel 2 bestehende Forschungen zu dem Thema vorgestellt. Danach wird in Kapitel 3 das Problem noch einmal ver-

¹Übersetzung von Deep Reinforcement Learning

deutlich und Besonderheiten hervorgehoben. In Kapitel 4 werden die Methoden des tiefgehenden und bestärkenden Lernens vorgestellt. Das Kapitel 5 umfasst die Umsetzung des tiefgehenden bestärkenden Lernens anhand der vorliegenden Problemstellung und betrachtet die Skalierbarkeit. Anschließend werden in Kapitel 6 die Ergebnisse evaluiert und in Kapitel 7 ein Fazit erarbeitet.

Kapitel 2

Verwandte Arbeiten

Aufgrund des Einflusses von Verkehr auf das tägliche Leben gibt es mehrere Arbeiten, die sich mit dem Thema beschäftigen. Hierbei wurde sich meist auf die Verbesserung des gesamten Verkehrsflusses durch verschiedene Ampelphasen konzentriert. Einige Arbeiten werden im Folgenden vorgestellt.

In der Arbeit [22] wurde beobachtet, welche Auswirkung eine Veränderung der Ampelphasen auf Verkehr von verschiedener Dichte hat. Es hat sich gezeigt, dass mit einer an den Verkehrsfluss angepassten Lichtsignalanlage sowohl die durchschnittliche Wartezeit, als auch die durchschnittliche Reisezeit niedriger ausfällt. Obwohl hierbei nicht der Zusammenhang von Kreuzungen untereinander untersucht wurde, können diese Ergebnisse als Ansatzpunkt benutzt werden.

Dass die Schaltung der Lichtsignalanlagen auch mit Hilfe eines Multi-Agenten System berechnet werden kann, zeigt die Arbeit [9]. Dafür wurden 5 Arten von Agenten definiert: Ein Kreuzungsagenten, ein Busagenten, ein Sensoragenten, ein Routenagenten und ein Überwachungsagenten.

Der Sensoragent sammelt die Signale der Mobiltelefone auf der Straße und bereinigt diese nach Fahrzeugen, da sich mehrere Mobiltelefone in einem Verkehrsmittel befinden können. Die Daten werden an den Kreuzungsagenten weitergeben und mit der Verspätung der Busse für die Berechnung einer besseren Ampelphase genutzt. Die Verspätung der Busse wird durch Kommunikation zwischen dem Busagenten und dem Routenagenten ermittelt. Der Überwachungsagent überprüft, ob jeder Bus seine Aufgaben korrekt erfüllt oder irgendwelche Fehler auftreten. Die Ergebnisse dieses Ansatzes könnten mit den erarbeiteten Ergebnissen verglichen werden, welches im Rahmen dieser Arbeit nicht erfolgt.

Obwohl es Arbeiten gibt, die Multi-Agenten Systeme mit bestärkenden Lernen [32] oder mit neuronalen Netzen [25] verbindet, um die Ampelschaltung zu optimieren, wird in dieser Arbeit ein zentraler Ansatz genutzt. Denn die notwendigen Kommunikationsmittel für einen Multiagenten System im Straßenverkehr ist noch nicht flächendeckend gegeben. Außerdem würde dadurch die Angriffsfläche kritischer Verkehrsinfrastruktur wie beispielsweise Lichtsignalanlagen erhöht. So könnte bei einem erfolgreichen Angriff eine gefährliche

Ampelphase eingeleitet werden und ein Unfall provoziert werden, der zu einer Gefährdung von Menschenleben führen kann. Also werden hier alle Daten zentral gesammelt, um dann eine Ampelschaltung zu berechnen.

Der Effekt von tiefgehendem bestärkendem Lernen zum Optimieren der Phasen von Lichtsignalanlagen wurde in Arbeit [12] untersucht. Hierbei wurde eine Verbesserung des gesamten Verkehrsflusses angestrebt. Für die Konstruktion des Zustandsraumes wurden jeder Fahrspur drei Vektoren zugewiesen. Der erste Vektor signalisiert, ob ein Fahrzeug vorhanden ist oder nicht. Der zweite Vektor gibt die Geschwindigkeit wieder und der dritte Vektor beinhaltet die aktuelle Lichtsignalphase. Um diese Vektoren zu füllen, wird jede Spur in Stücke einer festen Länge zerlegt und für jedes Stück die benötigten Werte ermittelt. Die Vektoren werden zum einen als boolesche Werte gespeichert zum anderen als Realwerte. Als Aktionen werden die 4 Lichtsignalphasen (NS, OW, NSL, OWL) definiert. Zum einen können alle Spuren in Richtung Norden beziehungsweise Süden (NS) auf grün geschaltet werden. Zum anderen alle Spuren in Richtung Osten und Westen (OW). Zusätzlich gibt es noch die Phasen NSL und OWL, welche die beiden vorherigen Phasen um ein grünes Signal für die Linksabbiegerspur erweitern. Die Belohnung ist die Differenz der aktuellen Verspätung bezüglich der Verspätung ohne Beeinflussung. Zur Darstellung des Agenten wird ein neuronales Netz gebaut, welches den booleschen Vektor und den Echtwertvektor als Eingabe bekommt. Diese werden einzeln in den ersten beiden Schichten auf ihre lokalen Charakteristika untersucht und zusammengefasst. Beide Ergebnisse dienen mit der aktuellen Lichtsignalschaltung als Eingabe für die dritte Schicht, deren Ausgabe komplett vernetzt ist mit der vierten Schicht. Die erzeugte Ausgabe bestimmt die Aktion, die für die Lichtsignalanlage ausgeführt werden soll. Der gleiche Ansatz wird in dieser Arbeit verfolgt, daher können die Ergebnisse zur Orientierung genutzt werden. Jedoch unterscheiden sich in dieser Arbeit die Zustandsräume und der gewählte Aktionsraum.

Kapitel 3

Verkehr

Das Problematische am Bussystem ist die Struktur der Haltestellen. Dadurch wird ein Ort festgelegt, an welchem der Bus zu einer festen Zeit sein muss. Dies hat zur Folge, dass sich eine Verspätung ohne Gegenmaßnahmen auf alle nachfolgenden Haltestellen überträgt. Aber wenn der Bus früher als geplant eine Haltestelle anfährt, muss er die positive Zeit verfallen lassen und die Abfahrtszeit einhalten.

Der unterschiedliche Verkehrsfluss führt dazu, dass die Reisezeit einer Strecke für die gleiche Uhrzeit an verschiedenen Tagen anders ausfällt. Zusätzlich ist es nicht möglich, den Fahrplan spontan zu ändern. Ansonsten könnten Personen, die den Bus nutzen wollen, nicht vernünftig planen, da sich die Informationen zu Abfahrtszeit und Reisedauer ständig ändern könnten. Eine Strategie um die Verspätung aufzuholen wäre, die Haltezeit an den Haltestellen solange zu verringern bis der Bus wieder planmäßig fährt. Dies ist aber nicht unbegrenzt durchführbar, da dort Personen ein- und aussteigen müssen. Außerdem ist so im besten Falle nur die Summe der Wartezeit an allen Haltestellen einzusparen. Sollte die Verspätung größer ausfallen, ist es für den Bus so nicht möglich pünktlich zu kommen.

Als Erweiterung der Idee könnte an einigen Haltestellen bewusst eine größere Pufferzeit für eine Verspätung eingeplant werden. Jedoch ist dadurch die Ankunftszeit nicht garantiert. Es könnte vorkommen, dass es zu einer stärkeren Staubildung kommt als vom Fahrplan einberechnet und sich dadurch der Bus trotzdem verspätet. Außerdem fällt bei weniger Verkehr als angenommen die Haltezeit der Busse unnötig lange aus, wodurch der Bus für die gesamte Strecke mehr Zeit benötigt. Dadurch wird die Nutzung des Busses unwirtschaftlicher und die Kapazitäten werden nicht gut genutzt. Denn hinter dem öffentlichen Nahverkehr agieren immer noch Firmen, bei denen ein guter zuverlässiger Service mit den wirtschaftlichen Aspekten konkurriert. In Deutschland ist über die Grundversorgung geklärt, dass der Nahverkehr pünktlich sein soll und kurze Taktzeiten besitzt. Des Weiteren soll er zu einem angemessenen Preis angeboten werden. Um dies zu gewährleisten, ohne Verluste zu machen, wird versucht, die minimale Anzahl an Bussen einzusetzen.

Der Verkehrsfluss kann in seinen Grundzügen nicht beeinflusst werden. Eine Person entscheidet selbstständig zu welchem Zeitpunkt sie ein Fahrzeug führt und welche Strecke

sie benutzt. Trotzdem muss sie sich an die Straßenverkehrsordnung halten. Vor allem Lichtsignalanlagen sind dabei geeignet, auf einfache Art und Weise den Verkehr zu beeinflussen. Die direkte Beziehung zwischen Ampeln und Straßen ist fest gegeben und auch die Effekte der Phasen. Bei einem grünen Lichtsignal bewegen sich die Fahrzeuge in eine Richtung, die durch die Spur festgelegt ist, auf der sie sich befinden.

Bei den Bussen wissen wir durch den Fahrplan genau welche Haltestellen angefahren werden und welche Strecke benutzt wird. Also ist es möglich, durch Beeinflussung der passenden Lichtsignalanlagen den Bus zu beschleunigen und so eine Verspätung zu verringern oder aufzuholen. In der Realität wird dies auch schon verwendet. Die am großflächigsten eingesetzte Methode ist eine bevorzugte Behandlung der Busse an den Ampelanlage [15]. Dabei meldet sich der Bus automatisch ein paar hundert Meter vor der Lichtsignalanlage bei dem Steuergerät an. Dieses fragt an einem Server an, ob es vom Zeitplan möglich ist, dass dieser Bus eine Priorisierung erfährt. Sollte die Priorisierung erlaubt sein, wird der nächste Zyklus abgeändert. Dabei müssen keine komplizierten Berechnungen gemacht werden oder andere Sensoren vorhanden sein. Sobald der Bus die Kreuzung passiert hat, fällt die Lichtsignalanlage in ihren alten Zyklus zurück.

Zu einem Problem kommt es, wenn mehrere Busse gleichzeitig an einer Kreuzung die Priorisierung beantragen. Es wird grundsätzlich derjenige bevorzugt, der zuerst angefragt hat. Alternativ können für verschiedene Buslinien auch Priorisierungsstufen im System hinterlegt werden. Prinzipiell ist dies eine gute Funktionsweise für einen schnellen Busverkehr. Jedoch liegt es in Deutschland im Ermessen der Kommunen, ob eine Ampel mit einer derartigen Signaltechnik ausgestattet wird. Daher wird in diesen Arbeiten nur ein GPS-Sender im Bus vorausgesetzt, mit welchem die Position und damit etwaige Verspätungen nachgehalten werden.

Kapitel 4

Grundlagen

Die Arbeit baut stark auf Methoden aus dem maschinellen Lernen auf. Insbesondere das tiefgehende und bestärkende Lernen, welches im Folgenden erklärt wird. Anschließend wird die Kombination aus beiden Methoden anhand der Arbeit von Deep Mind erläutert.

4.1 Maschinelles Lernen

Mit Hilfe von maschinellem Lernen kann ein Computer Gesetzmäßigkeiten in Daten erkennen, ohne explizit dafür programmiert worden zu sein [5]. Hierfür werden zunächst Beispieldaten zu einem Thema eingespeist anhand denen das System lernt. Nach diesen Lernprozessen können so Lösungen für andere Eingaben gefunden werden. Beispielsweise soll eine Kamera die Handschrift einer Person erkennen. Dafür liegen Texte in handschriftlicher und codierter Form vor, welche dann beide dem Computer zugeführt werden. Anhand derer werden die Charakteristika der Handschrift extrahiert. Dadurch können weitere neue handschriftliche Texte in die übliche Textcodierung übertragen werden.

Insgesamt verbindet maschinelles Lernen die menschliche Fähigkeit des Lernens mit der Geschwindigkeit einer Maschine. Im Zusammenhang mit dem Verkehr bringt dies viele Vorteile. Am Beispiel der Stadt Luxemburg mit knapp 600.000 Einwohnern wird an dieser Stelle verdeutlicht, dass eine größere Stadt aus verkehrstechnischer Sicht eine große Herausforderung darstellt. So existieren in Luxemburg mehr als 5500 Straßen und über 4000 Kreuzungen mit entsprechend vielen Lichtsignalanlagen [8]. Im Zusammenhang mit etwa 2200 Bussen pro Tag ist abzusehen, dass es sehr aufwändig ist ein solches System mit menschlichem Personal zu beeinflussen. Es müsste kontinuierlich beobachtet werden wann ein Bus sich einer Kreuzung nähert und dann eine passende Schaltung der Lichtsignalanlage gewählt werden. Dabei dürfen keine Flüchtigkeitsfehler auftreten oder Busse übersehen werden. Um dies auszuschließen wird eine Maschine eingesetzt, denn Maschinen arbeiten wesentlich zuverlässiger als Menschen. Ein konventionell programmiertes Verhalten wird in vielen Fällen von seiner Logik eingeschränkt und agiert schlechter als ein lernendes Programm.

4.2 Tiefgehendes Lernen

Das tiefgehende Lernen ist eine Unterart des maschinellen Lernens, bei dem neuronale Netze als Lernbasis benutzt werden [21]. Ein neuronales Netz entsteht durch das Verbinden von Neuronen, welche über die Eingabe von binären Werten funktionieren. Diese werden addiert und gegen einen Grenzwert verglichen. Wenn die Summe kleiner ist als der Grenzwert wird eine 0 ausgegeben, ansonsten eine 1. Damit der Einfluss von Signalen auf ein Neuron bestimmt werden kann, ist jede Eingabe gewichtet. Über die Gewichte wird der Lernprozess realisiert, welches in einem späteren Abschnitt weiter ausgeführt wird. Das Gewicht ist eine reelle Zahl, die mit der binären Eingabe multipliziert wird. Über alle Produkte wird dann wieder die Summe gebildet, welche für die Aktivierungsfunktion als Eingabe genutzt wird. Die Aktivierungsfunktion ersetzt den Grenzwert und bestimmt die Ausgabe des Neurons für die gegebenen Eingaben.

4.2.1 Neuronale Netze

Ein Neuron kann beliebig viele Eingaben und Ausgaben haben, somit kann man sie untereinander beliebig vernetzen. Dies wird genutzt, um neuronale Netze zu bauen. In Abbildung 4.1 ist beispielhaft ein neuronales Netz zu sehen. Dabei ist anzumerken, dass sich neuronale Netze in Größe und Komplexität stark voneinander unterscheiden können. Um die dargestellten Schichten zu erzeugen, wird eine beliebige Anzahl von Neuronen in Schichten zusammengefasst. Des Weiteren werden Schichten untereinander verbunden. Es ist immer eine Eingabeschicht, welche die Ursprungsdaten verarbeitet und eine Ausgabeschicht, die das Ergebnis liefert, gegeben. Dazwischen können beliebig viele andere sogenannte versteckte Schichten eingefügt werden. Die Ausgabe des Neurons wird für jedes damit vernetzte Neuron der nächsten Schicht als Eingabe genutzt [23].

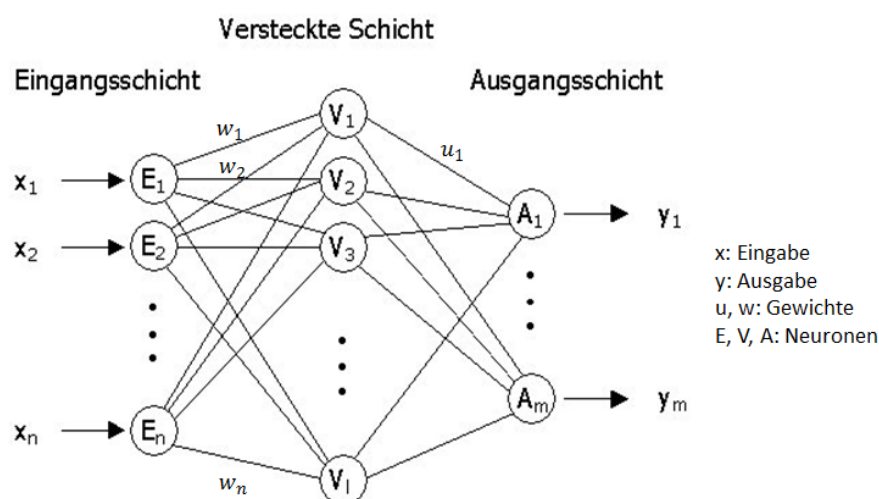


Abbildung 4.1: Neuronales Netz

4.2.2 Trainieren von neuronalen Netzen

Um das neuronale Netz zu trainieren, können jegliche Bestandteile verändert werden. Es können Verbindungen gelöscht werden oder neue Verbindungen erzeugt werden. Es können Neuronen gelöscht und hinzugefügt werden. Außerdem kann der Schwellwert der Neuronen angepasst werden. Am häufigsten werden die Werte der Gewichte geändert. Zum Trainieren von neuronalen Netzen kann überwachtes Lernen eingesetzt werden. Die Gewichte werden anhand von Lerndaten verändert und somit das neuronale Netz trainiert. Die Daten bestehen aus Paaren von Eingaben und Ausgaben. Alle Eingaben der Beispieldaten werden in das neuronale Netz gespeist. Die berechnete Ausgabe wird anhand des bekannten Ergebnisses zu der Eingabe evaluiert. Dies geschieht mithilfe einer Verlustfunktion, welche die Abweichung der Ergebnisse berechnet. Die Verlustfunktion könnte beispielsweise die quadratische Differenz berechnen. Das Ziel ist es über die Berechnung von neuen Gewichten die Verlustfunktion über alle Beispiele zu minimieren. Somit wird der Lerneffekt über die Aktualisierung der Gewichte realisiert.

Eine Methode zum Anpassen der Gewichte ist das statische Gradientenverfahren [7]. Dies ist ein Algorithmus, um Optimierungsprobleme zu lösen. Im Falle von neuronalen Netzen ist das die Minimierung der Verlustfunktion. Die Berechnung läuft nach den Formeln in Abbildung 4.2 statt. Θ stellt die Gewichte in Vektorform dar, wodurch mit dem Index t der Iterationsschritt angezeigt wird. Ein Gradient ∇f spannt einen Vektor über die Dimension der Funktion auf. Dabei wird die übergebene Funktion f partiell für jede Dimensionsvariable abgeleitet. Da die Ableitung die Steigung der Funktion repräsentiert, ist der Gradient ein Vektor in Richtung der größten Steigung.

$$\theta_{t+1} = \theta_t - \gamma * \nabla f(\theta_t)$$

$$\nabla f(w) = \left(\frac{\partial f(w)}{\partial w_1}, \frac{\partial f(w)}{\partial w_w}, \dots, \frac{\partial f(w)}{\partial w_n} \right)$$

Abbildung 4.2: Formel für einen Gradientenschritt

Die übergebene Funktion f ist die Verlustfunktion, also wird die Verlustfunktion nach jedem Gewicht einzeln differenziert. Das Ziel ist es, sich langsam den Gewichten zu nähern, bei dem die Verlustfunktion minimal ist. Dafür wird das Produkt aus Gradient und Lernrate γ von den Gewichten abgezogen. Es wird in Richtung des negativen Gradienten herabgestiegen. Über die Lernrate wird dabei bestimmt, wie stark der Einfluss des Gradienten ist. Hierbei führt eine große Lernrate zu einer starken Veränderung der Gewichte, sodass man sich eventuell von einer guten Lösung entfernt. Eine kleine Lernrate nähert sich dem Optimum, wobei die Anzahl an Iterationen steigt.

Es kann passieren, dass eine Lösung gefunden wird, bei der die Verlustfunktion vermeintlich minimal ist. Dabei könnte es nur ein lokales Minimum sein. Das Gradientenverfahren garantiert nicht, dass ein globales Minimum gefunden wird. Außerdem muss es nicht konvergieren, meistens wird die Modifikation der Gewichte für eine feste Anzahl von Iteration wiederholt und dann abgebrochen. Es wird dann angenommen, dass das Netz ausreichend trainiert ist.

Sollten mehrere Schichten in dem Netz gegeben sein, wird jeder Gewichtsvektor der Schichten einzeln trainiert. Die neuen Gewichte werden erst in das Netz übertragen, wenn die Berechnung aller Gewichte abgeschlossen ist.

4.3 Bestärkendes Lernen

Ein anderer Bereich des maschinellen Lernens ist das bestärkende Lernen [26], bei welchem der Agent mit der Umwelt interagieren kann und dadurch neues Wissen generiert. Die Umwelt wird als eine Menge von Zuständen dargestellt. Es stehen in jedem Zustand gewisse Aktionen zur Verfügung, durch deren Ausführung unterschiedliche Belohnungen ausgegeben und neue Zustände erreicht werden. Das Ziel ist es, die Gesamtbelohnung zu maximieren. In der Abbildung 4.3 ist der beschriebene Prozess bildlich dargestellt. Die Umwelt ist in einem Zustand s_t , in welcher der Agent eine Aktion a durchführt. Diese führt die Umwelt in einen neuen Zustand und es wird eine Belohnung r an den Agenten vergeben. Während der gesamten Zeit sind die möglichen Werte der Variablen in Abhängigkeit von der Zeit t , welche auch als Anzahl der Iteration interpretiert werden kann.

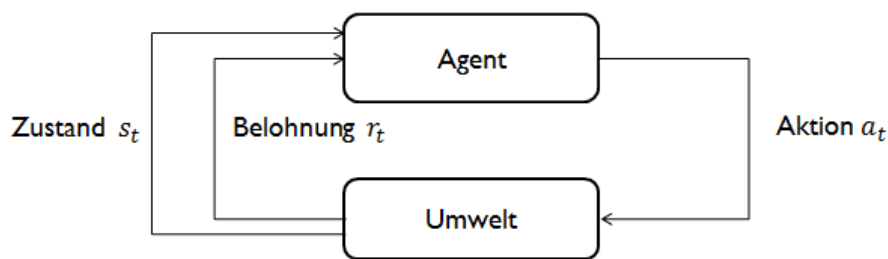


Abbildung 4.3: Prozess des bestärkenden Lernens

Beim Durchführen dieses Ansatzes konkurriert die Wahl der größtmöglichen Belohnung in dem aktuellen Zustand mit einer eventuell größeren zukünftigen Belohnung bei einer anderen Aktion. Dieses Problem ist bekannt als Ausbeutung gegen Erkundung [27]. Das heißt, am Anfang wird ein möglichst großer Lösungsraum gesucht und nach diesen Ergebnissen eine Strategie gebildet. Sobald die Strategie sich nach vorher programmierten Kriterien als ausreichend erweist, wird der Lösungsraum verkleinert und die Strategie optimiert. Der Lösungsraum umfasst die ausgewählten Aktionen die ausgeführt werden. Dadurch wird quasi

ein Baum aufgespannt, der mit jeder ausgewählten Aktion einen neuen Zweig bekommt. Jeder dieser Zweige stellt selber einen solchen Baum dar. Die Schwierigkeit liegt darin, dass möglichst viele Aktionen betrachtet werden sollen. Jedoch wächst dadurch der Baum in der Breite und die Laufzeit, um alle Zweige zu durchlaufen, erhöht sich exponentiell. Dabei ist die Anzahl der Aktionen entscheidend für den Anstieg der Laufzeit. Wenn nur ein Teil des Baumes betrachtet wird, kann es passieren, dass die beste Strategie übersehen wird. Es muss also ein gutes Gleichgewicht zwischen Laufzeit und Optimierung gefunden werden.

Bestärkendes Lernen kann nach zwei Arten durchgeführt werden. Entweder ohne ein Modell oder mit einem Modell. Beim modellbasierten Lernen wird anhand der Daten ein Modell gelernt und daran eine Wertefunktion geplant. Dahingegen ist das Ziel des modellfreien Lernens direkt aus den Daten eine Strategie zu lernen. Ein Modell zu erlernen ist aufwändiger und birgt ein Fehlerrisiko in sich. Ein falsches Modell wirkt sich auf den Lernprozess aus. Es hat aber den Vorteil, dass es bei einer Änderung des Ziels in der gleichen Umwelt weiter genutzt werden kann. Modelle, die verkehrstechnische Zusammenhänge abbilden veraltern sehr schnell durch neue Straßen, Haltestellen und sich verändernde Verkehrsgewohnheiten. Daher wird hier ein modellfreies bestärkendes Lernen gewählt.

4.3.1 Q-Lernen

Ein möglicher Algorithmus des modellfreien Lernens ist Q-Lernen [29]. Hierbei wird eine Q-Funktion verwendet, welche für jeden Zustand und allen möglichen Aktionen ein Tupel des Zustandes mit jeder Aktion bildet. Die Werte werden nach der Formel in Abbildung 4.4 mit der Zeit neu berechnet. Die Variablenbenennung von Aktionen (a, a') und Zuständen (s, s') ist die gleiche wie oben. Die Belohnung ist eine Funktion der Zustände und Aktionen, $R(s, a)$. Die Lernrate wird durch α repräsentiert und bestimmt, wie groß der Einfluss von neuen Informationen ist. Der Wert der Lernrate ist definiert im Bereich von 0 bis 1. Bei einer kleinen Lernrate haben neue Informationen kaum einen Einfluss auf die bisherige Strategie. Um die Auswirkung von zukünftigen Entscheidung zu steuern, existiert der Abzugsfaktor γ . Dieser kann auch Werte zwischen 0 und 1 annehmen. Wenn er nahe 0 ist, haben Belohnungen in den zukünftigen Zuständen kaum einen Einfluss. Während sich die Zustände, Aktionen und Belohnungen ändern, werden die Lernrate und der Abzugsfaktor einmal fest gewählt .

$$(1 - \alpha) * Q(s, a) + \alpha * [R(s, a) + \gamma * \max_{a'} Q(s', a')]$$

Abbildung 4.4: Formel des Q-Lernens

Die Q-Werte werden in einer Tabelle gespeichert die über die Dimension der Aktion und der Zustände aufgespannt wird. Also eine Tabelle mit der Anzahl $|a| \times |s|$ an Einträgen. Da für jeden kleinen Unterschied in der Umwelt ein neuer Zustand erzeugt wird, wird diese Tabelle bei komplexeren Problemen schnell sehr groß. So stellt bei nur einem Fahrzeug jede Kombination von Geschwindigkeit und Position eine Zustandsänderung da. Wenn das auf die Komplexität einer Stadt übertragen wird, ist erkennbar, dass sehr schnell Speicherprobleme auftreten können.

4.4 Tiefgehendes bestärkendes Lernen

Deep Mind ist ein Unternehmen von Google, welches auf Künstliche Intelligenz spezialisiert ist. Im Jahre 2013 haben sie neuronale Netze und bestärkendes Lernen zu einem neuen Ansatz verbunden, das tiefgehende bestärkende Lernen [20]. Ausgangslage war, dass durch tiefgehendes Lernen sehr gut die Information aus vieldimensionalen Daten extrahiert werden kann. Dabei wurde dann untersucht, ob das Trainieren durch bestärkendes Lernen eine gute Strategie findet.

Das Ziel war es, einen Agenten zu trainieren, der auf einem Atari 2600 gewinnen konnte. Der Atari ist eine Spielekonsole mit mehreren Spielen, bei denen 2 Spieler gegeneinander antreten. Dabei sollte der Agent so viele Spiele wie möglich beherrschen und gegen den besten Computergegner des entsprechenden Spieles gewinnen. Als Eingabe wurden nur die Pixel des Bildschirms gewählt und es wurden keine anderen Informationen vorab bereitgestellt. Basierend darauf sollte mit Hilfe von Belohnung und der möglichen Aktionen eine Strategie gefunden werden. Am Ende wurde bei sechs von sieben Spielen ein besseres Ergebnis erzielt als mit bisherigen Ansätzen. Außerdem wurde bei 3 Spielen gegen einen menschlichen Spieler auf Experten-Niveau gewonnen.

Als Agent wird ein tiefes Q-Netzwerk verwendet, welches aus zwei faltbaren und zwei vollständig verbundenen Schichten besteht. Das Bild wird vor der Eingabe in Graustufen umgewandelt und die Pixelanzahl etwas reduziert. Letztendlich werden die Bildpunkte von 4 Bildern auf einmal eingespeist. Die faltbaren Schichten reduzieren die übergebene Matrix und fassen die wichtigen Charakteristika lokal zusammen. Es wird eine Fenstergröße festgelegt die bestimmt, wie viele Werte der Eingabematrix zur Berechnung für einen Wert in der reduzierten Matrix genutzt werden. Zusätzlich wird noch festgelegt, wie viele Stellen das Fenster auf der Matrix weitergeschoben wird, nachdem alle Werte die es umschließt zusammengefasst wurden. Nachdem die beiden faltbaren Schichten die Eingabematrix vereinfacht haben, berechnen die beiden letzten Schichten durch ihre Neuronen die Werte für die Aktion.

Um das Speicherplatzproblem der Tabellen vom Q-Lernen zu lösen, wird die Q-Funktion durch ein neuronales Netz ersetzt. Die Belohnung für eine Aktion in dem aktuellen Zustand wird so immer mit den aktuellen Gewichten neu berechnet. Um den passenden Q-Wert zu

berechnen, wird der Zustand und die Aktion in das neuronale Netz gegeben. Nun müsste bei einem Zustand für jede mögliche Aktion der Q-Wert berechnet werden. Aufgrund der hohen Laufzeit wird dies beschleunigt, indem nur der Zustand an das Netzwerk übergeben wird und ein Vektor mit Q-Werten berechnet wird. Der Vektor hat die gleiche Dimension wie die Anzahl der Aktionen und jeder Q-Wert gehört zu einer Aktion.

Am Anfang werden alle Gewichte des neuronalen Netzes zufällig gesetzt. Es ist wahrscheinlich, dass Zustände, die nacheinander auftreten, sich ähneln. Dadurch ist ein zustandsabhängiges trainieren des Netzes möglich. In Folge dessen könnten Aktionen, die für den Moment gute Ergebnisse bringen, zu stark bestärkt und andere Aktionen vernachlässigt werden. Um solch einem Verhalten entgegen zu wirken und das Netz möglichst allgemein zu halten, wird ein Wiederholungsspeicher implementiert. In diesem wird vor jedem Aktualisierungsschritt ein 4-elementiges Tupel gespeichert. Dieses Tupel umfasst den Ausgangszustand, die gewählte Aktion, die Belohnung und den erreichten Zustand. Für den Lernschritt wird dann eine zufällige Anzahl an Tupel aus dem Wiederholungsspeicher ausgewählt und mit ihnen ein Gradientenschritt auf dem Netz vollzogen. Durch den Wiederholungsspeicher haben alle Arten von vorher aufgetretenen Zuständen einen Einfluss auf den Lernprozess.

Kapitel 5

Algorithmus

5.1 Umsetzung

Als erstes sind die Begriffe des bestärkenden Lernens auf das zu untersuchende Praxisproblem abzubilden. Die Phasen jeder Lichtsignalanlage bilden die auszuführenden Aktionen ab. Die Belohnung berechnet sich aus der Verspätung aller Busse im Vergleich zum Busfahrplan. Die Auswirkung auf den restlichen Verkehr sind erst einmal nicht relevant. Als Zustand dient die Verspätung von Bussen an jeder Haltestelle. Auch wenn die Anzahl der Busse relativ gering ist, wird durch die Anzahl der Haltestellen und aller möglichen Kombinationen an Verspätungen ein sehr großer Zustandsraum aufgespannt.

In dem Deep Mind Ansatz [20] wurden vier Schichten verwendet. Die ersten beiden Schichten dienten dazu, aus den ankommenden Pixeln eine Situation zu extrahieren. Da in dieser Arbeit die Daten über andere Wege aufbereitet werden, sind die vorderen Schichten nicht nötig. Daher haben wir nur eine versteckte Schichten, die komplett vernetzt ist mit der Eingabeschicht. Die Eingabeschicht besteht aus so vielen Neuronen wie die Anzahl aller Busse in dem Simulationslauf. Als Aktivierungsfunktion für die versteckte Schicht wurde ein Rektifizierer verwendet, der nur die positiven Werte auf sich selbst abbildet. Die Ausgabe hat nur ein Neuron und legt die Sigmoidfunktion für die Aktivierung zugrunde.

Es wird für jede Kreuzung mit einer Ampel ein neuronales Netz aufgebaut und die Gewichte nach der Xavier Methode initialisiert. Das hat zum Effekt, dass die Signale auch in tiefen Schichten in einem vernünftigen Rahmen bleiben. Sollten die Gewichte zu klein gewählt sein, dann fällt das Signal mit jeder Schicht ab und ist irgendwann nicht mehr nutzbar. Im umgekehrten Fall, bei zu großen Gewichten, wächst das Signal mit jeder Schicht. Dadurch ist die Aktivierungsfunktion irgendwann gesättigt und die Steigung nimmt ab. Der Gradient nähert sich null und die Gewichte ändern sich kaum noch.

Für jeden verspäteten Bus wird dann in dem Eingabevektor für das entsprechende neuronale Netzwerk ein fester Wert, in diesem Fall eins, gesetzt. Durch die einzigartige ID der Busse wird eine feste Position in dem Eingabevektor festgelegt. Nun wird die Ausgabe berechnet, welche die Wahrscheinlichkeit für eine Aktion darstellt. Da nur eine Ausga-

bemöglichkeit existiert können nur zwei Klassen definiert werden. In diesem Fall entspricht dies zwei Ampelphasen. Zum einen sollen sie so stark komplementär sein, wie es möglich ist. Zum anderen sollten die Hauptverkehrswege durch die Umschaltung profitieren. Beispielsweise für den einfachen Fall, dass 2 Straßen sich kreuzen, wäre die Phase für die Klasse 1 die Richtung Nord bzw. Süd und für Klasse 0 die Richtung Ost bzw. West.

Die gesamte Ausgabe des Netzes wird als Wahrscheinlichkeit angesehen für eine der beiden Ampelphasen. Diese Wahrscheinlichkeit wird mit einer Zufallszahl zwischen 1 und 0 verglichen. Sollte die Wahrscheinlichkeit größer sein wird immer die eine Aktion ausgewählt, ansonsten immer die andere. Durch den Zufallsfaktor werden nicht immer die gleichen Aktionen gewählt. Am Ende eines Beobachtungszeitraums werden die Gewichte dann mit dem Gradientenverfahren modifiziert.

Gegenüber einem Spiel existiert hier kein eindeutiger Sieg, worüber evaluiert werden kann, ob die Strategie erfolgreich war oder nicht. Zusätzlich ist die Zeit für ein Spiel endlich. Dahingegen hat der Verkehr weder einen festgelegten Anfang noch ein festes Ende. In dieser Arbeit werden die Start- und Endpunkte der Simulationsdaten als Anfangs- und Endpunkte gewählt. Im Allgemeinen kann der Analysezeitraum beliebig gewählt werden, beispielsweise tage- oder stundenweise. So ergibt sich ein Endzustand, an dem die Busverspätung evaluiert werden kann. Aber es fehlt eine Grenze anhand welcher die Effektivität der Strategie ermittelt werden kann. Im besten Falle wäre diese 0, jedoch ist das bei einer hohen Verkehrsdichte realistisch nicht erreichbar. Also wird versucht eine Grenze zu ermitteln. Hierfür werden mehrere Durchläufe mit verschiedenen Ampelschaltung simuliert. Die Ergebnissen werden nach ihrer akkumulierten Verspätung sortiert und die Grenze anhand der besten 15 Prozent festgesetzt. Sofern die Verspätung kleiner ausfällt, zählt die Strategie als erfolgreich.

Nach Abschluss eines beobachteten Zeitraums, werden die Gewichte dann mit der Gradientenmethode modifiziert. Bei Verkehr entfalten die Aktionen eine großflächige Wirkung, daher sind die zukünftigen Belohnungen fast genauso wichtig wie die aktuellen. Also wird der Abzugsfaktor sehr nahe bei Eins gewählt.

5.2 Performance

Die benötigte Zeit um im Verkehrsgeschehen einen weiteren Schritt zu simulieren, setzt sich aus dem Aufwand zur Ermittlung der neuen Verkehrssituation und der Berechnung der neuronalen Netze zusammen. Für die Simulation des Verkehrs wird die Software SUMO eingesetzt, welche im Kapitel 6 genauer beschrieben wird. Hierbei verhält sich die Anzahl der Fahrzeuge proportional zu der Laufzeit eines Simulationsschrittes. Andere Größen, wie das Ausmaß des Straßennetzes haben keinen zeitlichen Einfluss. Um die Ausgabe der neuronalen Netze zu ermitteln wird dahingehend immer gleich viel Zeit benötigt. Da sich

die Dimensionen der Eingabevektoren während des gesamten Durchlaufes nicht verändern und auch die Struktur der neuronalen Netze gleich bleibt.

Dies wird in Abbildung 5.1 grafisch aufgezeigt. Der Balken für das Minimum repräsentiert die Zeit, wenn keine Fahrzeuge betrachtet werden und nur die Grundfunktionen der Simulation laufen. Hier wird schon 85 Prozent der Zeit von der Simulationssoftware benötigt. In der Spitze müssen die Bewegung von ungefähr 900 Fahrzeugen ausgeführt werden und die Berechnung der neuronalen Netze macht nur einen sehr kleinen Anteil aus. Im Durchschnitt braucht ein Schritt 13.7 Millisekunden, wovon die maschinelle Berechnung nur 0.6 Millisekunden ausmacht.

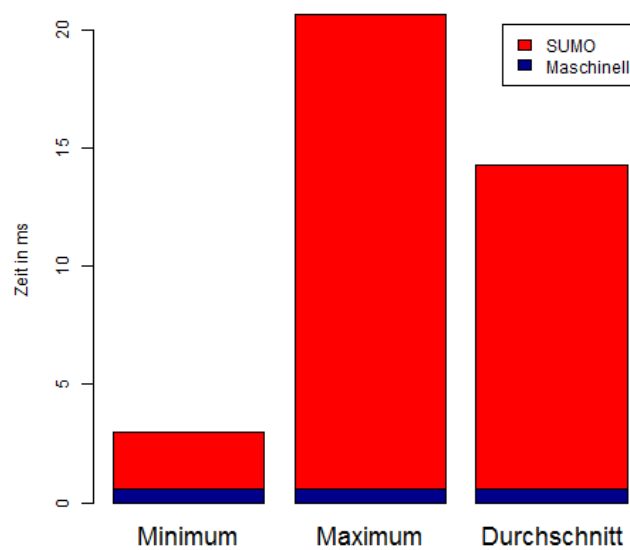


Abbildung 5.1: Zeitaufwand pro Simulationsschritt

5.3 Skalierbarkeit

In dem obigen Ansatz wird für jeden Bus die Verspätung festgehalten. Auch wenn dadurch die Gewichte individuell für jeden Bus geändert werden können, entpuppt sich dies in der Praxis als problematisch. Denn der Eingabevektor der neuronalen Netze muss eine ausreichende Länge aufweisen, um für jeden Bus einen Platz zu haben. Natürlich könnte er ausreichend groß gewählt werden, jedoch erhöht sich dadurch der Berechnungsaufwand für die neuronalen Netze. In Abbildung 5.2 ist die benötigte Zeit, um die Ausgabe der neuronalen Netze zu berechnen, dargestellt. Hierbei ist zu erkennen, dass die Zeit linear zu der Anzahl an Kreuzungen steigt, aber exponentiell mit der Länge des Eingabevektors. Also sollte das Ziel sein, einen möglichst kleinen Eingabevektor zu erreichen. Dafür muss dann zum Start eines Beobachtungszeitraums die Anzahl aller fahrenden Busse bekannt

sein. Diese kann je nach Wochentag und Uhrzeit stark variieren. In Folge dessen müsste die zugrundeliegende Datenstruktur häufig angepasst werden, damit die Verspätungen aller Busse aufgenommen werden können, ohne die Laufzeit unnötig zu verlängern. Eine andere Möglichkeit besteht darin, dass jeder neue Bus dynamisch aufgenommen wird. Dadurch wäre die Länge zwar so klein wie möglich, jedoch führt dies auch zu einer kontinuierlichen Anpassung der neuronalen Netze. Alternativ um eine Verringerung der Länge zu erreichen könnten die Busse nach ihren Buslinien zusammengefasst werden. Dadurch wird aber nicht die Schwankung, nach Wochentagen oder bei der Einführung von neuen Linien, verhindert.

Ein Ansatz zur Problemlösung ist die Orientierung an statischen Größen, wie die Spuranzahl der jeweiligen Kreuzung. So ergibt sich nur selten eine Änderung und zwar nur, wenn an einer Kreuzung eine Straße aus- oder neugebaut wird. Des Weiteren steigt so die Dimension des Eingabevektors selten über 30. Selbst bei einer große Kreuzung mit 4 ankommende Straßen und je 2 Linksabbieger-, 2 Rechtsabbieger- und 3 Geradeausspuren, ergibt sich eine Dimensionalität von 28. Nachts wenn nur wenige Buslinien befahren werden, könnte die Länge des Eingabevektors zwar kleiner sein. Jedoch ist in Abbildung 5.2 zu sehen, dass der Zeitunterschied bei den Dimensionen kleiner als 30 sehr gering ausfällt.

Berechnungsaufwand des Neuronalen Netzes

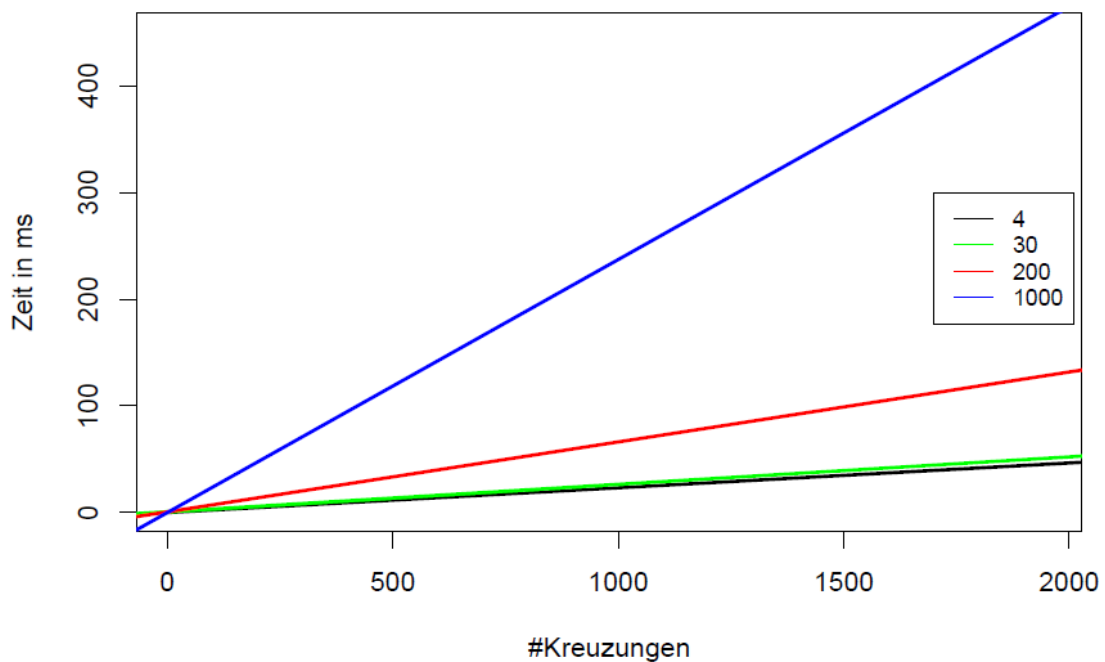


Abbildung 5.2: Aufwand zur Berechnung des Neuronalen Netzes in Abhängigkeit von der Dimensionalität des Eingabevektors

Kapitel 6

Evaluation

In der Arbeit wurde die Open-Source Verkehrssimulation “Simulation of urban Mobility“ (SUMO) verwendet [14]. Es ist eine freie und offene Simulationssoftware des Deutschen Zentrum für Luft- und Raumfahrt. Durch den Einsatz in mehreren Projekten hat sie sich als sehr zuverlässig und performant erwiesen. In dieser ist es möglich, alle infrastrukturellen Bestandteile eines Straßennetzes zu bauen. Das heißt, man kann Straßen mit einer beliebigen Anzahl an Spuren erzeugen, die erlaubten Spurwechsel zwischen kreuzenden Straßen festlegen und Lichtsignalanlagen mit einer frei bestimmbarer Phasenlogik einpflegen.

Für diese Arbeit ist vor allem wichtig, dass Haltestellen eingefügt werden können. Zusätzlich können diese in ihrer Länge verändert werden, sodass auch mehrere Busse gleichzeitig eine Haltestelle belegen können. Der gesamte Verkehr kann nach den eigenen Anforderungen eingespeist werden. Vor dem Start können die Arten von Fahrzeugen mit ihren Eigenschaften wie Beschleunigung, Fahrzeuggröße, Bremsverhalten und vielem mehr festgelegt werden. Nun können diese in beliebiger Menge zu einem frei wählbaren Zeitpunkt in das Straßennetz eingespeist werden. Zusätzlich wird jedem Fahrzeug eine individuelle Strecke zugewiesen, die es abfährt.

Die Simulation kann nun sowohl in grafischer Form, als auch auf der Konsole durchgeführt werden. Der Ablauf der Simulation ist in einzelnen Schritten möglich und kann beeinflusst werden. Dafür wird ein Traffic Control Interface (TraCI) [30] genutzt, welches im Hintergrund einen Server bereitstellt und auf eingehende Verbindungen wartet. Mithilfe des TraCI können Informationen zu dem Status von Fahrzeugen, Lichtsignalanlagen, Straßen und anderen Gegenständen der Simulation abgefragt werden. Darüber hinaus ist eine Einflussnahme auf viele Simulationsobjekte während eines Durchlaufes möglich. Die Route von Fahrzeugen kann geändert werden und ausgewählte Spuren können für bestimmten Verkehr gesperrt werden. Am Wichtigsten ist die Möglichkeit, dass damit Einfluss auf die Phasen der Lichtsignalanlagen genommen werden kann.

6.1 Szenario des Experiments

Als Datengrundlage für die Versuche wird das Straßennetz aus Bologna, einer italienischen Stadt, genutzt [6]. Dieses wurde vom Projekt iTETRIS implementiert, welches von der Europäischen Union mitfinanziert wurde. Für das Verkehrsaufkommen wurden Daten im Zeitraum vom 11.11.2008-13.11.2008 gesammelt. Dieser umfasst die Wochentage Dienstag, Mittwoch und Donnerstag, welche keine Beeinflussung durch den Wochenendverkehr erfahren. Es sollte der durchschnittliche Verkehr in der Woche simuliert werden, wobei Montage durch die Sonntage als alternativen Reisetag beeinflusst werden und an Freitagen ein früherer Feierabendverkehr auftritt. Mit Hilfe von 90 Detektoren wurden Verkehrsdaten aufgenommen. Daraus wurde die Zeit von 8:00 bis 9:00 Uhr, die Stunde mit dem höchsten Verkehrsaufkommen, erhoben (Abb. 6.1) und in SUMO übertragen. In diesem Zeitraum fahren über 140 Busse auf 20 Linien und mehrere hundert Autos. Wegen des dichten Verkehrs ist dieses Szenario sehr gut für diese Arbeit geeignet. Denn durch die vielen Fahrzeuge tritt eine große Menge an Stau auf, welches mit einer Verspätung der Busse einhergeht. Außerdem haben Veränderungen von Lichtsignalanlagen einen größeren Einfluss bei einer starken Verkehrsdichte im Vergleich zu wenig Verkehr. Die Taktung, in der die Busse fahren, beträgt bei mehreren Buslinien nur wenige Minuten. Somit ist der Zeitunterschied bis zwei aufeinanderfolgende Busse einer Linie an der gleichen Haltestelle gehalten haben nicht so groß und die Situationsänderung des Verkehrs fällt nur klein aus. Dadurch sind die Auswirkungen besser zu erkennen.

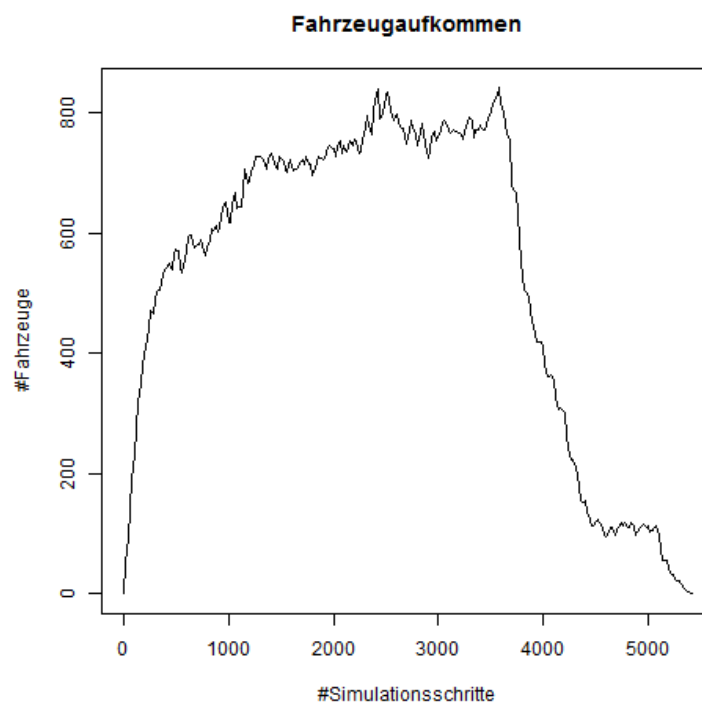


Abbildung 6.1: Anzahl der Fahrzeuge während der Simulation

6.2 Ablauf des Experiments

Als erstes muss ein Busfahrplan erstellt werden. Hierfür werden die Busse ohne Verkehr durch das Straßennetz geschickt und an jeder Haltestelle der Zeitstempel mit der entsprechenden Station gespeichert. Daraus kann bei anderen Simulationen die Verspätung der Busse berechnet werden. Zusätzlich wird bei jedem Wechsel zwischen Straßen der Zeitpunkt festgehalten, um mehr Messpunkte zu haben und die Lichtsignalanlagen effektiver schalten zu können. Im Allgemeinen soll der Bus sowohl pünktlich ankommen als auch rechtzeitig wieder abfahren. Prinzipiell kann man über eine Verkürzung der Haltezeit die Verspätung zu einem gewissen Grad verringern. In unserem Fall wollen wir die Verspätung nur mithilfe der Ampelschaltung steuern und setzen daher die Haltezeit auf einen Simulationsschritt fest. Das betrachtete Straßennetz ist in Abbildung 6.2 dargestellt. Zusätzlich sind die Buslinien eingezeichnet, welche im Weiteren betrachtet werden.

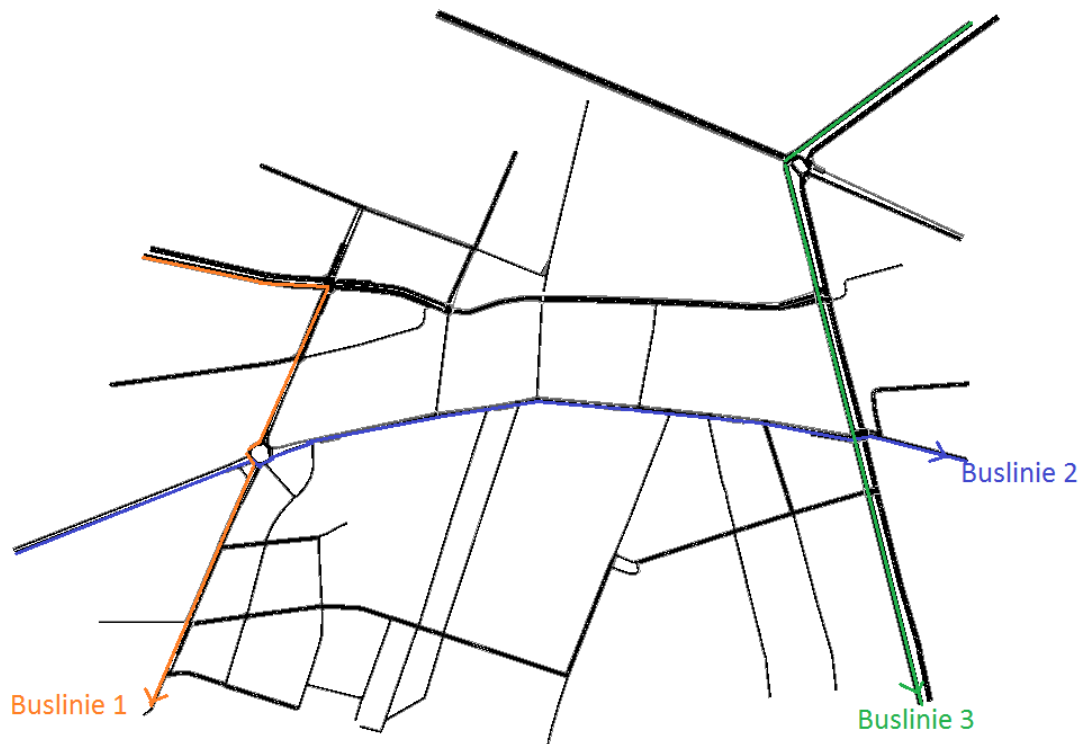


Abbildung 6.2: Straßennetz von Bologna in SUMO mit den betrachteten Buslinien

6.3 Experiment – Keine Beeinflussung von Ampeln

Das oben beschriebene Szenario wird zuerst ohne Veränderung der Lichtsignalanlage gestartet. In Abbildung 6.1 ist die Anzahl der Fahrzeuge in Abhängigkeit zu den Simulationsschritten dargestellt. Dazu kann die Verspätung der eingezeichneten Buslinien in den Abbildungen 6.3, 6.4 und 6.5 nachverfolgt werden. In den Diagrammen wird durch jede Linie das Anfahren aller Haltestellen des jeweiligen Busses repräsentiert. Die steile Flanke am Ende entsteht dadurch, dass der Bus nach der letzten Haltestelle mit einer Verspätung von 0 registriert wird. An der ersten Haltestelle ist jeder Bus pünktlich, auch wenn es in den Diagrammen schwer erkennbar ist. Der nahezu senkrechte Anstieg ergibt sich wenn nach der ersten Haltestelle in kurzer Zeit eine starke Verspätung angesammelt wird. Alle hier betrachteten Buslinien bauen mit der Zeit eine Verspätung auf, aber es gibt auch Buslinien die jede Haltestelle pünktlich anfahren. Auch wenn hier nicht eingegriffen wird um die Verspätung der Busse zu verringern, kann es passieren, dass während einer Fahrt der Bus etwas pünktlicher wird. Der Grund liegt darin, dass durch die Verzögerung andere Ampelphasen erreicht werden, als bei den Fahrten nach Fahrplan und diese zu einer leichten Beschleunigung führen können. Dieser Effekt fällt aber nur klein aus und tritt zufällig auf.

Für die Verspätung der Linie 1 (Abb. 6.3) ist über die gesamte Simulation ein ähnliches Verhalten erkennbar. Nach der ersten Haltestelle ist das Verkehrsaufkommen auf der Strecke sehr hoch und es baut sich in sehr kurzer Zeit eine hohe Verspätung auf. Dieses Wachstum wird erst gestoppt, aber mit den darauf folgenden Haltestellen steigt die Unpünktlichkeit noch ein wenig mehr. Über die letzten Haltestellen findet keine große Veränderung der Verspätung mehr statt.

Auch bei der Linie 2 (Abb. 6.4) wird im Laufe jeder Fahrt eine Verspätung angesammelt. Am Anfang fällt diese nur klein aus und kann durch Glück mit den Ampelphasen wieder aufgeholt werden. Jedoch steigt die Verspätung bei einer höheren Verkehrsdichte stark an. Bei den Fahrten um die Simulationsschritte 2000 und 3000 wird die Verspätung mit den letzten Haltestellen leicht abgebaut, während bei der fünften und siebten Fahrt die Verspätung immer weiter zunimmt.

Bei der Buslinie 3 (Abb. 6.5) wird zu keinem Zeitpunkt die Abweichung vom Fahrplan wieder aufgeholt, sondern immer weiter erhöht. Am Anfang befinden sich nur wenige Fahrzeuge auf der Strecke von dieser Linie, sodass für die erste und dritte Fahrt nur eine kleine Verspätung auftritt. Hier fährt sogar der zweite Bus komplett pünktlich. Für die letzten Fahrten hat sich das Verkehrsaufkommen auf der Strecke erhöht und es wird eine wachsende Verspätung über die Haltestellen angesammelt.

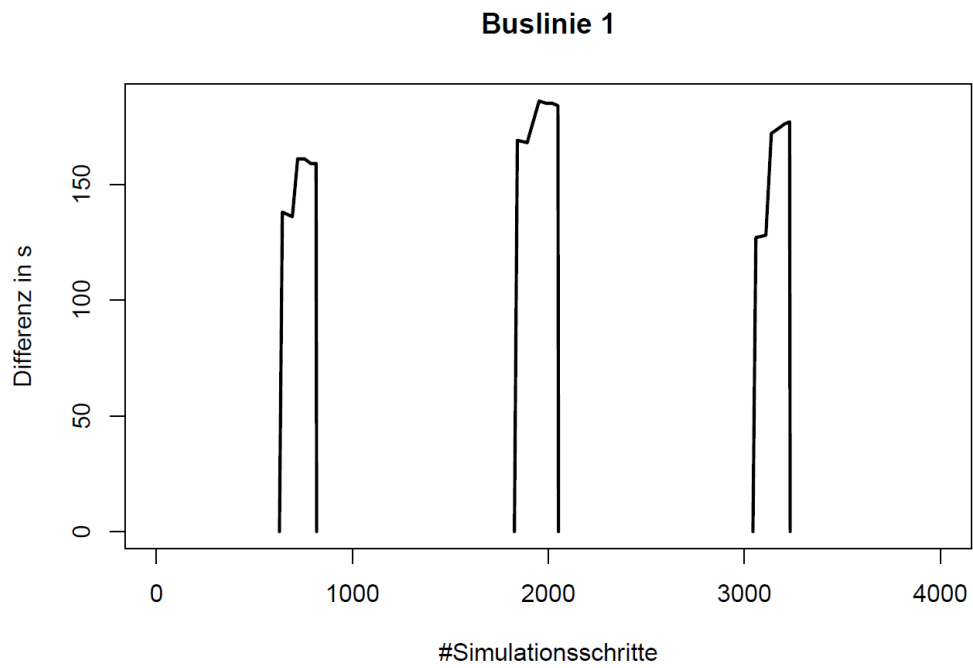


Abbildung 6.3: Akkumulierte Verspätung der Buslinie 1 an den Haltestellen

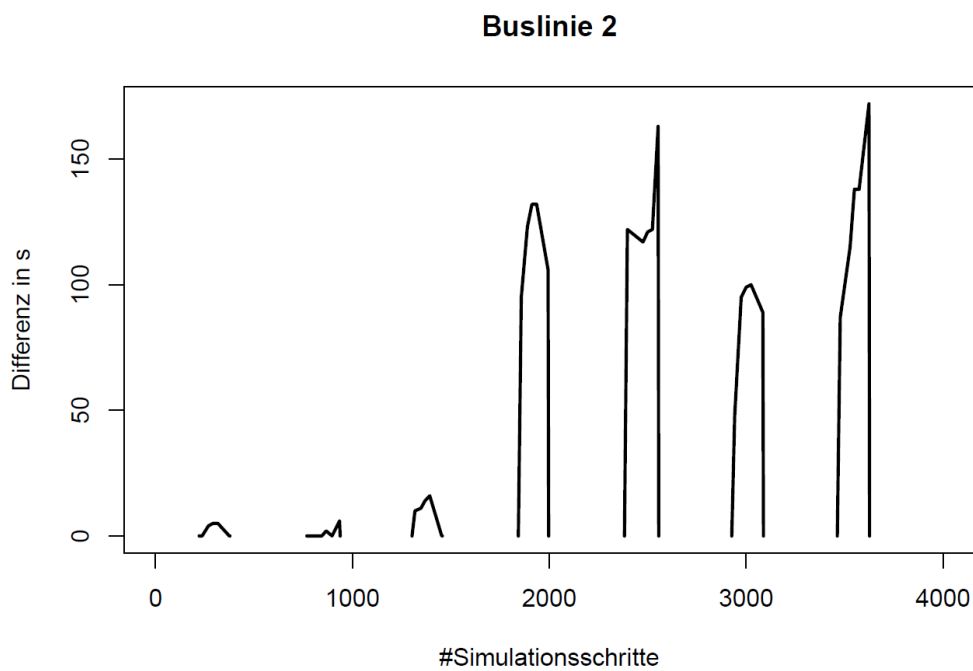


Abbildung 6.4: Akkumulierte Verspätung der Buslinie 2 an den Haltestellen

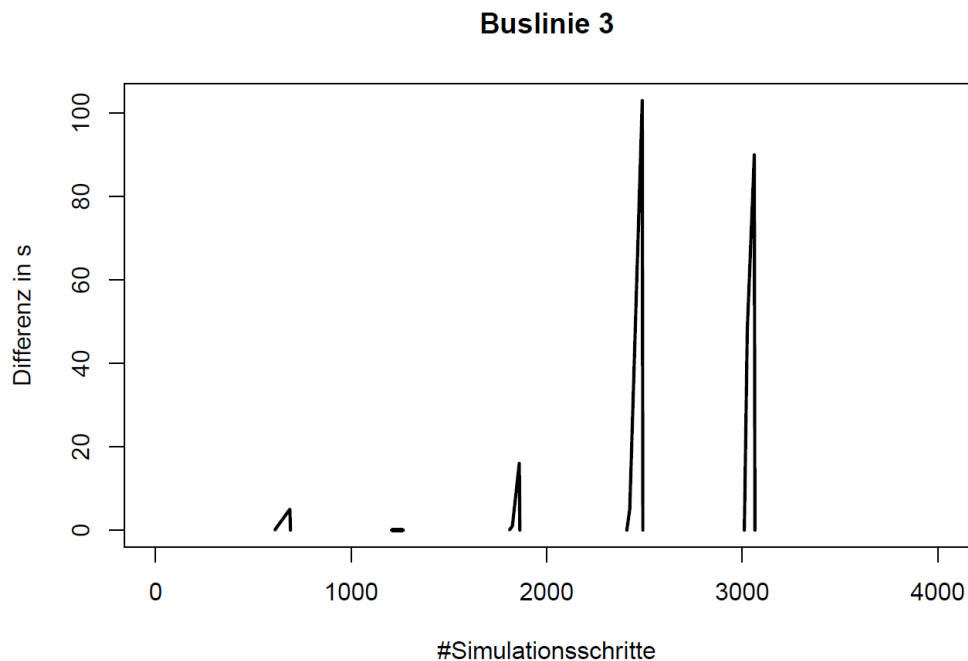


Abbildung 6.5: Akkumulierte Verspätung der Buslinie 3 an den Haltestellen

6.4 Experiment – Ampelbeeinflussung bei Busverspätungen

Für den nächsten Ansatz wird eine einfache Priorisierung gewählt. Sobald für einen Bus eine Verspätung entdeckt wird, verändert die nächste Lichtsignalanlage auf der Busstrecke ihre Phase. Solange ein Bus priorisiert wird, ist eine neue Anfrage gesperrt, bis der Bus die Ampel passiert hat oder eine festgelegte Zeit abgelaufen ist. Hierbei ist entscheidend, dass der Bus, welcher die Priorisierung zuerst anfragt, eine Beschleunigung erfährt. So kann es vorkommen, dass Busse auf einer anderen Strecke benachteiligt werden.

Bei der ersten Linie (Abb. 6.6) ist eine positive Auswirkung durch die Priorisierung an den Lichtsignalanlagen zu erreichen. Zu jedem Zeitpunkt fällt die Verspätung geringer aus, als vorher. Dieser Effekt ist darauf zurückzuführen, dass kein anderer Bus zu den Zeitpunkten an den Kreuzungen eine Priorisierung benötigt und so diese Linie ein grünes Signal erhält, wenn sie es benötigt.

Für die Linie 2 ist ein gegenteiliger Effekt erkennbar (Abb. 6.7). Am Anfang ist der Unterschied zwischen den beiden Methoden eher klein. Die erste Haltestelle jeder Fahrt wird weiterhin pünktlich angefahren, aber danach steigt die Verspätung noch höher als bei dem Durchlauf ohne Beeinflussung der Ampelanlage. Der Grund hierfür liegt in den Auswirkung auf den gesamten Verkehrsflusses durch die anderen Ampelphasen und einem daraus resultierendem verändertem Stauverhalten. Nach den ersten Haltestellen, wird der Bus an den Kreuzungen priorisiert und die Verspätung kann abgebaut werden.

In Abbildung 6.8 sind für die Linie 3 gegenteilige Effekte der priorisierte Ampelschaltung bezüglich der Pünktlichkeit gegenüber des anderen Ansatzes erkennbar. Zuerst fällt die Verspätung bei der ersten Fahrt minimal geringer aus. Die zweite Fahrt weicht nun an einer Haltestelle sehr leicht von ihrem Fahrplan ab. Eine Beschleunigung des Busses ist bei der Fahrt um den Simulationsschritt 2500 erreicht worden, bei welchem die Verspätung kleiner ausfällt. Sowohl in der dritten, als auch bei der letzte Fahrt wird das Problem des priorisierten Ansatzes aufgezeigt. Denn bei diesen Fahrten nimmt die Verzögerung über alle Haltestellen zu, was darauf zurückzuführen ist, dass andere Busse die Kreuzung blockieren und so die Linie 3 über einen längeren Zeitraum warten muss.

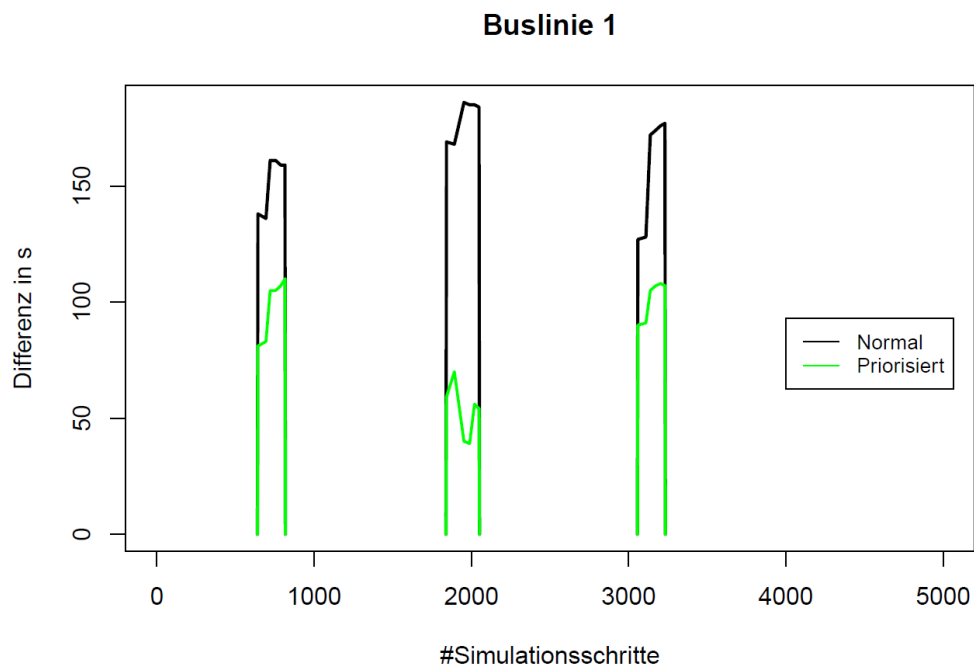


Abbildung 6.6: Vergleich des priorisierten und des normalen Ansatzes anhand der Linie 1

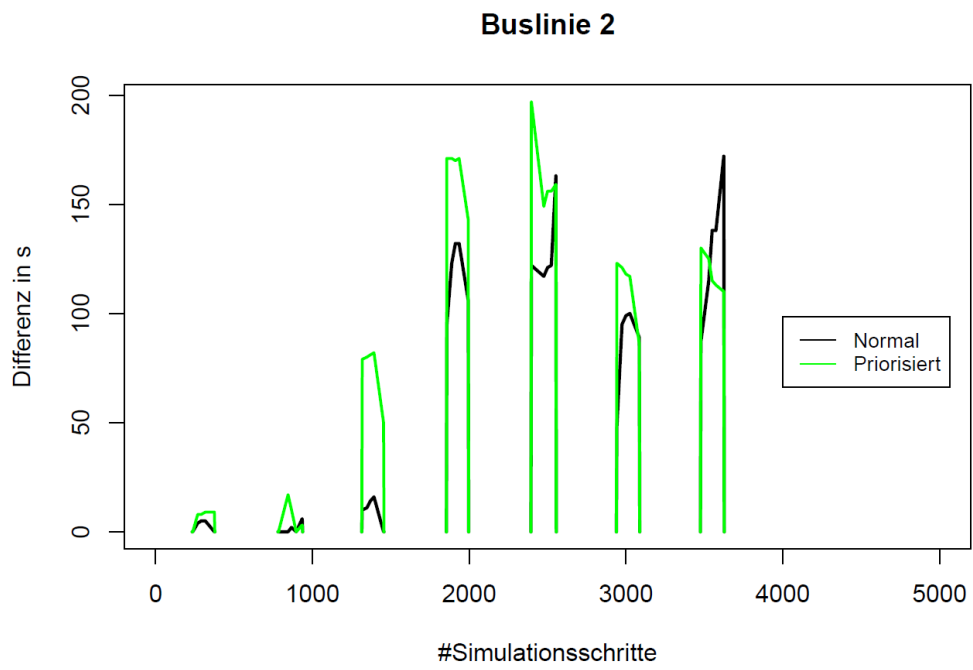


Abbildung 6.7: Vergleich des priorisierten und des normalen Ansatzes anhand der Linie 2

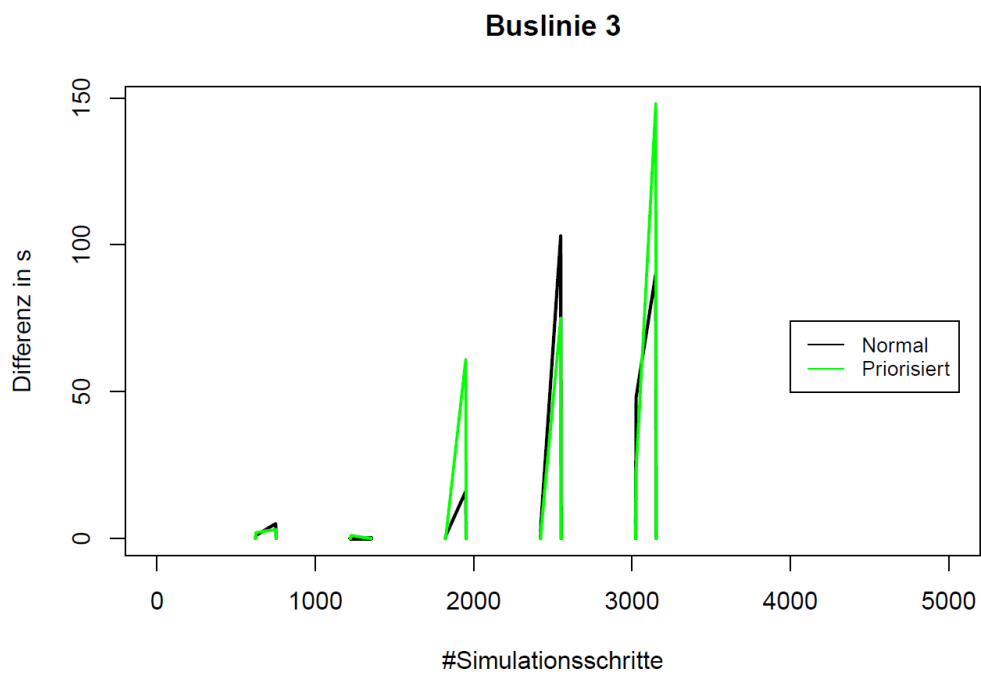


Abbildung 6.8: Vergleich des priorisierten und des normalen Ansatzes anhand der Linie 3

Es ist in Abbildung 6.9 zu sehen, dass eine Verminderung der Verspätung erreicht werden kann, sofern die Ampel gerade keinen anderen Bus bevorzugt. Mit steigendem Verkehr entsteht so aber ein höheres Stauaufkommen in dem Straßennetz. Außerdem nimmt die Verspätung bei einer konkurrierenden Blockierung der Lichtsignalanlage stark zu. Die Abbildung zeigt, dass bei wenig Verkehr im Durchschnitt eine Verbesserung der Pünktlichkeit erreicht wird. Mit einer steigenden Anzahl an Fahrzeugen überwiegen jedoch die negativen Effekte und die Verspätung steigt sehr viel stärker als bei dem anderen Durchlauf.

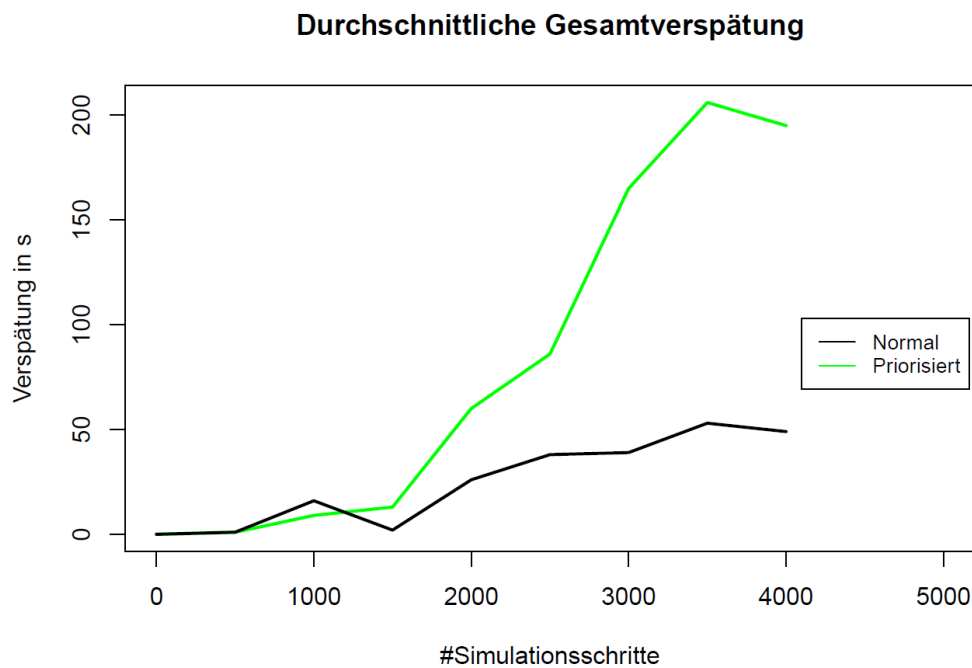


Abbildung 6.9: Vergleich des priorisierten und des normalen Ansatzes mit der durchschnittlichen Verspätung

6.5 Experiment – Tiefgehendes Bestärkendes Lernen

Nun wird die Simulation mit dem Algorithmus gestartet der in Kapitel 5 erklärt wurde. Zur Evaluation der Ergebnisse werden die beiden vorherigen Simulationen genutzt und die erste Haltestelle jeder Fahrt wird weiterhin pünktlich angefahren.

Für die Linie 1 (Abb 6.10) ist eine starke Verschlechterung gegenüber dem priorisierten Ansatz erkennbar. Im Vergleich mit der Simulation ohne Priorisierung existieren sowohl Verschlechterung als auch erhebliche Verbesserung. Beispielsweise läuft die erste Fahrt sehr ähnlich zu dem Bus ohne Veränderung der Ampelphasen ab. Die Verspätung der zweiten Fahrt liegt zwischen den beiden vorherigen Experimenten und baut die anfänglichen Verspätung mithilfe einer vorteilhaften Ampelschaltung im Laufe der Fahrt ab. Auch die letzte

Busstrecke agiert ähnlich wie die normalen Bussfahrt. Die Verspätung nach der ersten Haltestelle fällt höher aus, wächst dafür im weiteren Verlauf langsamer.

Dahingegen ist für die Linie 2 (Abb 6.11) durchgehend eine erhebliche Verbesserung gegenüber den beiden anderen Ansätzen erreicht worden. Hauptsächlich wird der Verkehrsfluss verändert und die Abweichung vom Fahrplan während der ersten Haltestellen steigt nicht so hoch. Zusätzlich wird diese aber auch durch die ausgewählten Ampelschaltungen bei den meisten Fahrten weiter verringert.

Die Verbesserung der Linie 2 geht teilweise zu Lasten der dritten Buslinie (Abb. 6.12). Am Anfang unterscheidet sich das Verhalten aller Methoden nicht stark. Doch die letzten beiden Fahrten werden durch die Kreuzung, an der sie mit der Linie 2 konkurrieren, sowohl durch schlechtere Ampelschaltung als auch ein höheres Verkehrsaufkommen benachteiligt und erfahren so eine höhere Verspätung.

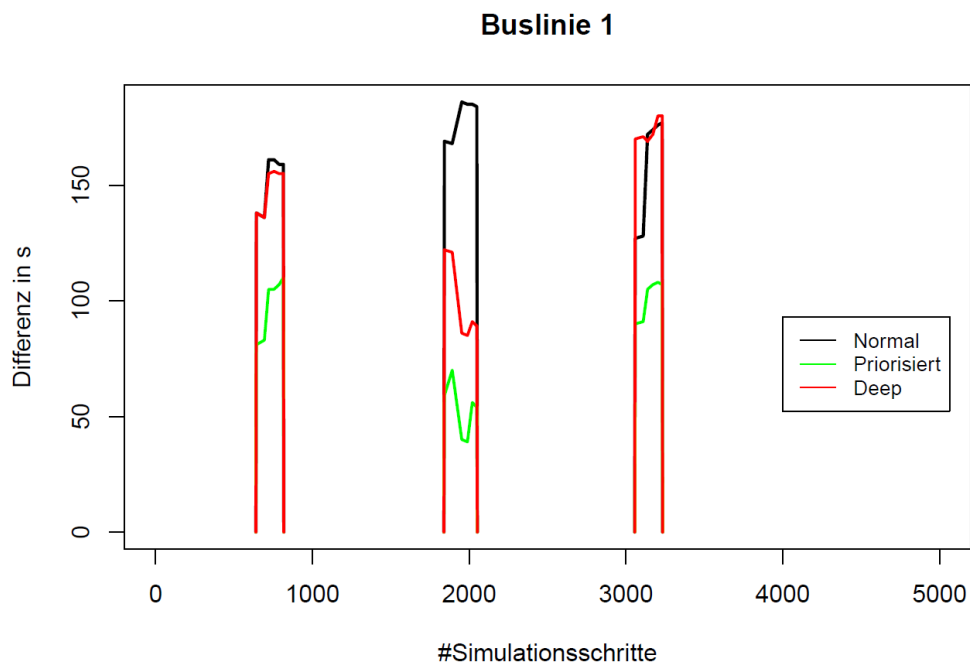


Abbildung 6.10: Methodenvergleich anhand der Linie 1

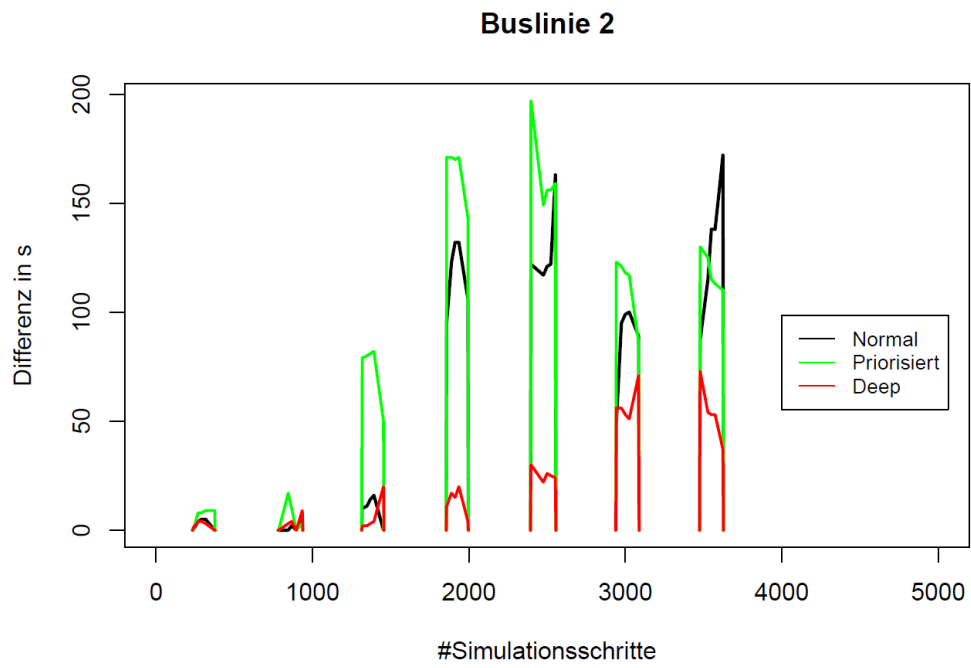


Abbildung 6.11: Methodenvergleich anhand der Linie 2

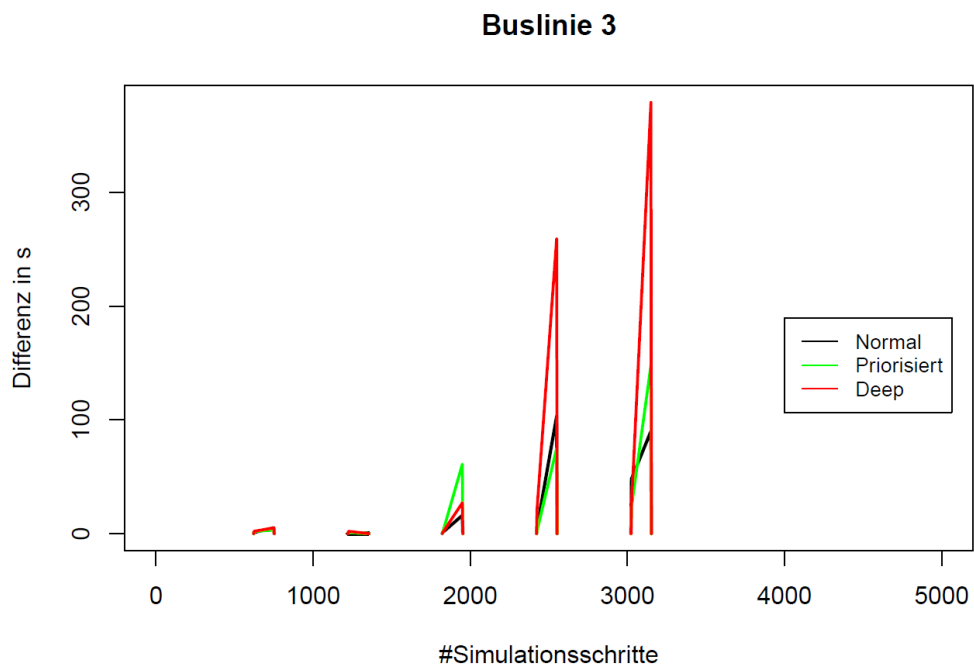


Abbildung 6.12: Methodenvergleich anhand der Linie 3

Obwohl die priorisierte Methode die Verspätung der Busse erhöht hat, sieht man in Abbildung 6.13, dass über die Beeinflussung der Lichtsignalanlagen auch eine Verbesserung herbeigeführt werden kann. Die Methode mit tiefgehendem bestärkendem Lernen liegt bis zum Simulationsschritt 1500 zwischen der priorisierten Methode und dem Ansatz ohne Beeinflussung. Im weiteren Verlauf erreicht sie dann die beste Durchschnittsverspätung über alle Busse. Der Grund für die höhere Pünktlichkeit liegt darin, dass die Methode sich merkt welche Auswirkungen das Schalten der Lichtsignalanlage hat. Mit diesen Informationen kann eine Ampelschaltung entwickelt werden, die eventuell einzelne Busse nicht so stark beschleunigt wie es möglich wäre. Stattdessen wird auch mal eine höhere Verspätung akzeptiert, damit insgesamt die Leistung aller Busse verbessert werden kann.

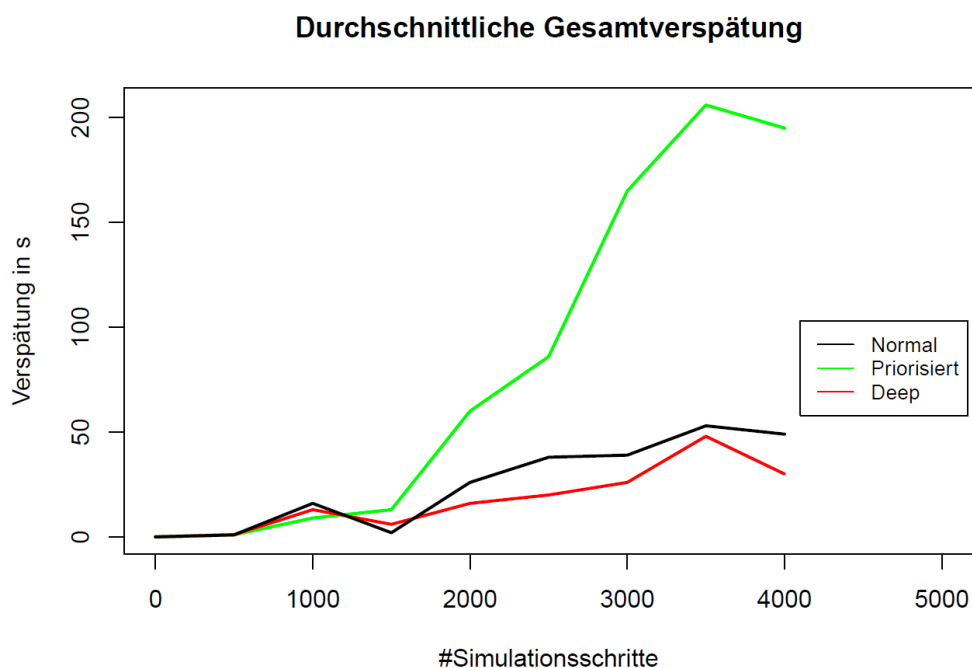


Abbildung 6.13: Methodenvergleich anhand der Gesamtverspätung

6.6 Weitere Experimente

Es kann untersucht werden, ob eine andere Merkmalsextraktion in einer Ampelschaltung mit einer kürzeren Verspätung resultiert. Dazu könnten die Straßen in mehrere gleich große Streckenabschnitte aufgeteilt werden und die verspäteten Busse in diesen Abschnitten erkannt werden. Mithilfe von faltenden neuronalen Netzen können die Daten dann weiterverarbeitet werden und dann als Basis zu Berechnung der Ampelphasen dienen.

Desweiteren kann durch das Einbeziehen von mehr Sensordaten ein genaueres Bild des Verkehrsgeschehens erreicht werden. So können weitere Aspekte in die Entscheidungsfindung für die Priorisierung eines Busses einbezogen werden. Beispielsweise stehen zwei

verspätete Busse an der gleichen Kreuzung und benötigen zwei konkurrierende Ampelschaltungen für ihre Weiterfahrt. Auf der ausstehenden Busroute des einen Busses sind nur noch wenig befahrene Straßen, sodass dort das Herausfahren der Verspätung sehr wahrscheinlich ist. Der andere Bus hat dahingegen viel Verkehr auf der restlichen Strecke. Auch wenn die aktuelle Höhe der Verspätung ausschlaggebend für die Entscheidung zwischen den Bussen ist, können solche Informationen auch einbezogen werden.

Kapitel 7

Fazit

Durch die Experimente hat sich gezeigt, dass die erzielte Ampelschaltung durch tiefgehendes bestärktes Lernen einen positiven und negativen Effekt auf die Buslinien erzielt. Auch wenn lokal die gesamte Verspätung höher ausfiel, wurde insgesamt das beste Gesamtergebnis erzielt.

Es muss aber beobachtet werden, welche Effekte in anderen Straßennetzen auftreten. Insbesondere, ob es zu Nachteilen auf bestimmten Straßen kommt. Beispielsweise an Kreuzungen oder Einmündungen bei der eine Strecke mit hoher Busfrequenz langführt und dadurch eventuell kreuzende Spuren zu stark benachteiligt werden. Dazu müssten aber geeignete Datensätze aufgebaut werden. Es existiert noch eine Simulation mit dem Verkehr von Luxemburg, jedoch treten dort nur wenige Staus und Verspätungen auf. Momentan werden Daten in Köln erfasst, die aktuell aber noch keinen öffentlichen Personennahverkehr betrachten.

Am wichtigsten ist es nachzuhalten, ob die Verbesserung in der simulierten Umgebung auf die Realität übertragbar ist. Hierbei ist zu beachten, dass es zu Unfällen kommen kann und Baustellen auftreten können, die den Verkehrsfluss einschränken. Auch hat das Verhalten der Passagiere einen entscheidenden Einfluss auf die Haltezeit eines Busses und führt eventuell zu einer Verspätung. Zusätzlich ist der Zusammenhang zwischen stark genutzten Umsteigemöglichkeiten nicht dargestellt, bei welchen eine Verspätung viele Passagiere betrifft. Auch der zeitkritische Aspekt spielt in einem realen Straßennetz eine wichtige Rolle. Die Berechnung der Ampelphasen muss abgeschlossen sein bevor der Bus sich an der entsprechenden Kreuzung befindet. Hierbei hat die Rate mit der die Verspätung erkannt und die Ampelphasen berechnet werden den größten Einfluss.

Diese Arbeit hat gezeigt, dass tiefgehendes bestärkendes Lernen auf das Verkehrsgeschehen angewendet werden kann. Eventuell können mit diesem Ansatz auch Asuweichrouten für Fahrzeuge in einem staubelasteten Verkehrsnetz ermittelt werden und so eine Entlastung erreicht werden.

Abbildungsverzeichnis

4.1	Neuronales Netz	9
4.2	Formel für einen Gradientenschritt	10
4.3	Prozess des bestärkenden Lernens	11
4.4	Formel des Q-Lernens	12
5.1	Zeitaufwand pro Simulationsschritt	17
5.2	Aufwand zur Berechnung des Neuronalen Netzes in Abhängigkeit von der Dimensionalität des Eingabevektors	18
6.1	Anzahl der Fahrzeuge während der Simulation	20
6.2	Straßennetz von Bologna in SUMO mit den betrachteten Buslinien	21
6.3	Akkumulierte Verspätung der Buslinie 1 an den Haltestellen	23
6.4	Akkumulierte Verspätung der Buslinie 2 an den Haltestellen	23
6.5	Akkumulierte Verspätung der Buslinie 3 an den Haltestellen	24
6.6	Vergleich des priorisierten und des normalen Ansatzes anhand der Linie 1	25
6.7	Vergleich des priorisierten und des normalen Ansatzes anhand der Linie 2	26
6.8	Vergleich des priorisierten und des normalen Ansatzes anhand der Linie 3	26
6.9	Vergleich des priorisierten und des normalen Ansatzes mit der durchschnittlichen Verspätung	27
6.10	Methodenvergleich anhand der Linie 1	28
6.11	Methodenvergleich anhand der Linie 2	29
6.12	Methodenvergleich anhand der Linie 3	29
6.13	Methodenvergleich anhand der Gesamtverspätung	30

Literaturverzeichnis

- [1] Busnetzplan Paris, URL: <http://bit.ly/2h2T1si> (aufgerufen am 27.12.2017)
- [2] TomTom Traffic Index, URL: <http://bit.ly/2naylR5> (aufgerufen am 27.12.2017)
- [3] Abteilung für wirtschaftliche und soziale Angelegenheiten der Vereinten Nationen (HRSG.) , World Urbanization Prospects: The 2014 Revision, URL: <http://bit.ly/2bcPLDc> (aufgerufen am 27.12.2017)
- [4] Abteilung für wirtschaftliche und strategische Analysen des U.S Ministerium für Transport(HRSG.) , Assessing the full cost of congestion on surface transportation systems and reducing them through pricing, 2009, URL: <http://bit.ly/2h4E4Wz> (aufgerufen am 27.12.2017)
- [5] J. Bell, *Machine learning: hands-on for developers and technical professionals*, John Wiley & Sons, 2014.
- [6] L. Bieker et al. , Traffic simulation for all: a real world traffic scenario from the city of Bologna, In *Simulation of Urban Mobility Conference 2014*, Seiten 19-26, 2014
- [7] I.N. Bronstein und K. A. Semendjaew, *Handbuch der Mathematik I*, Springer Verlag, 2013, Auflage: 3, 2012
- [8] L. Codeca, R. Frank, T. Engel, Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research scenario, In *2015 IEEE Vehicular Networking Conference (VNC)*, Seiten 1-8, 2015
- [9] F. Derakhshan and N. Shahpasandi, Design and implementation of an urban traffic control system for public transport using multi-agent systems, In *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 16 No.10, Seiten 70-77, 2016
- [10] A.M de Souza et al., Real-time path planning to prevent traffic jam through an intelligent transportation system, In *2016 IEEE Symposium on Computers and Communication (ISCC)*, Seiten 726-731, 2016

- [11] J. Geistesfeldt, J. Lohoff, Stausituation auf den Autobahnen in Nordrhein-Westfalen, 2011, URL: <http://bit.ly/2y5J4hV> (aufgerufen am 27.12.2017)
- [12] W. Genders and S. Razavi, Using a deep reinforcement learning agent for traffic signal control, arXiv Vorabdruck, *arXiv:1611.01142*, 2016
- [13] M. Kleinebrahm, H. Steven, Landesanstalt für Umwelt, Messung und Naturschutz Baden-Württemberg (Hrsg.) , PEMS-Messungen an drei Euro 6-Diesel-Pkw auf Streckenführungen in Stuttgart und München sowie auf Außerortsstrecken, 2015, URL: <http://bit.ly/2wqnSB8> (aufgerufen am 27.12.2017)
- [14] D. Krajzewicz, G. Hertkorn, P. Wagner, Simulation of urban mobility, In *2002 Middle East Symposium on Simulation and Modelling (MESM)*, Seiten 183-187, 2002
- [15] O. Krettek, Gegenwartiger Stand der Telematikanwendungen im Öffentlichen Nahverkehr der Bundesrepublik Deutschland, In *Archives of Transport System Telematics*, Vol.4 Heft 1, Seiten 49-56, 2011
- [16] B. Lenz, *Produktion, Distribution, Konsum*, Springer-Verlag, 2010
- [17] T. Liebig und M. Sotzny, On avoiding traffic jams with dynamic self-organizing trip planning, In *13th International Conference on Spatial Information Theory (COSIT 2017)*, Seiten 17:1-17:12, 2017
- [18] T. A. Litman und E. Doherty, Transportation cost and benefit analysis, Victoria Institut für Verkehrspolitik, 2009, URL: <http://bit.ly/1n1fO1I> (aufgerufen am 27.12.2017)
- [19] B. McCann, A review of SCATS operation and deployment in Dublin, Technischer Bericht, Stadtrat Dublin, 2014
- [20] V. Mnih et al., Playing atari with deep reinforcement learning, arXiv Vorabdruck, arXiv:1312.5602
- [21] S. Ohlsson, Introduction to deep learning using R, Apress, 2017
- [22] K. Rademacher, *The influence of sensor-based intelligent traffic light control on traffic flow in dublin*, Dissertation, Dublin Institut für Technologie, doi:10.21427/D7J89B, 2017
- [23] A. Scherer, Neuronale Netze: Prinzipien und Anwendungen, Springer Verlag, 1998
- [24] M. Schubert et al., Verkehrsverflechtungsprognose 2030 Schlussbericht, 2014, URL: <http://bit.ly/1qMLiek> (aufgerufen am 27.12.2017)
- [25] D. Srinivasan, M. C. Choy, R. L. Cheu, Neural networks for real-time traffic signal control, In *IEEE Transaction on Intelligent Transportation Systems*, vol. 7 no. 3, Seiten 261-272, 2007

- [26] R. S. Sutton und A. G. Barto, *Reinforcement learning: An introduction*, MIT Press Cambridge, 1998
- [27] S.B. Thrun, Efficient exploration in reinforcement learning, Technischer Bericht, Carnegie Mellon Universität, 1992
- [28] I. Utzmann, Beeinflussungsfaktoren für Verkehrsverhalten, In *Wuppertaler Verkehrsstunde 2003*, Seiten 18-25, URL: <http://bit.ly/2h5dzjY> (aufgerufen am 27.12.2017)
- [29] C. Watkins, Q-Learning, In *Machine Learning*, Vol. 8 Heft 3, Seiten 279–292, 1992
- [30] A. Wegener, et al. , TraCI: An interface for coupling Road traffic and network Simulators, In *11th Communications and Networking Simulation Symposium (CNS)*, Seiten 155-13, 2008
- [31] G. Weisbrod, D. Vary, G.Treyz, Economic implications of congestion, National Co-operativ Highway Research Program Report 463, 2001, URL: <http://bit.ly/2y0IZuS> (aufgerufen am 27.12.2017)
- [32] M. Wiering, Multi-agent reinforcement learning for traffic light control , In *Seventeenth International Conference on Machine Learning (ICML)*, Seiten 1151 - 1158, 2000

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 4. Januar 2018

Hendrik Hildebrandt