Seminar

“Neueste Trends in Big Data Analytics”

Thema:
Learning Human Body Movement

Lennart Kordt

Betreuer: Christian Hovy

Abgabedatum: 31.03.2018

Inhaltsverzeichnis

1. Einführung

Einem Roboter – menschliche – Bewegungen beizubringen ohne im Besitz von programmiertechnischen Kenntnissen zu sein, ist ein Ansatz das komplexe und komplizierte Feld der Robotik der breiten Masse nahezubringen.
Einem Roboter soll also eine einzelne Bewegung oder mehrere, zum Lösen einer bestimmten Aufgabe notwendige Bewegungen beigebracht werden. Der Roboter soll seine aktuelle Umgebung wahrnehmen und anhand ihrer Eigenschaften Aktionen auswählen, um die ihm gestellte Aufgabe zu lösen. Um das zu verstehen, müssen wir die Dinge aus Sicht des Roboters betrachten.
Die Welt des Roboters besteht aus Action-State-Paaren[1], die der Roboter aus Demonstrationen (D) ableitet. Ein State ist hier der aktuelle, noch unbekannte Zustand (S) der Welt, in der sich der Roboter befindet. Wird eine Demonstration auf den State durchgeführt, wird er zu einem bekannten State (Z). Zu jedem State existiert ein Pool aus Actions (A), die auf den jeweiligen State anwendbar sind. Dieses Mapping aus State und Action ist der Grundsatz der Entscheidungsfindung der meisten Roboter. Durch in Bezug setzen von State und Action wird eine sogenannte Policy (π) erlernt. Der Roboter nutzt diese Policy um die ihm gestellten Aufgaben zu lösen.

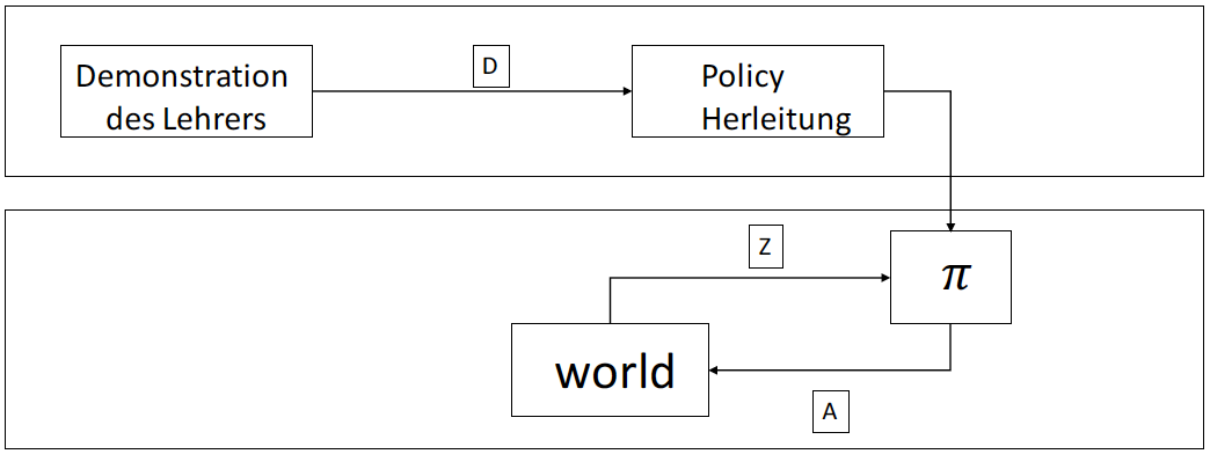

Abb. 1 LfD Policy Herleitung und Ausführung

In dieser Arbeit sollen im Kontext von *Learning from Demonstration* (LfD) verschiedene Ansätze des Mappings und der Herleitung einer Policy bei Robotern zum Ausführen von menschlichen Bewegungen beschrieben werden.

[1] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

2. Learning from Demonstration

Das Hauptprinzip des LfD ist, dass Menschen Robotern neue Aufgaben ohne Programmierung beibringen können.[2] Betrachten wir zum Beispiel einen Haushaltsroboter, dem ein Besitzer beibringen möchte, Orangensaft zum Frühstück zuzubereiten. Die Aufgabe selbst besteht offensichtlich aus mehreren Teilaufgaben, wie das Entsaften der Orange, das Wegwerfen Orangenschalen in den Müll und das Gießen der Flüssigkeit in ein Glas. Zu beachten ist hier, dass der Roboter höchstwahrscheinlich niemals an zwei verschiedenen Tagen die exakt gleiche Ausgangssituation, also die exakt gleiche Anordnung von Orange und Utensilien, auffinden wird. In einem herkömmlichen Programmierszenario müsste ein menschlicher Programmierer im Voraus eine Robotersteuerung entwerfen und programmieren, die in der Lage ist, auf jede Situation zu reagieren, der der Roboter ausgesetzt sein könnte, egal wie unwahrscheinlich diese auch sein mag. Dieser Prozess kann beinhalten, dass die Aufgabe in unzählige verschiedene Schritte unterteilt und jeder Schritt gründlich getestet wird. Wenn nach dem Einsatz des Roboters Fehler oder neue Umstände auftreten, muss möglicherweise der gesamte kostspielige Prozess wiederholt werden, und der Roboter wird zurückgerufen oder außer Betrieb gesetzt, während er repariert wird. Im Gegensatz dazu ermöglicht LfD dem Besitzer, den Roboter zu programmieren, indem er ihm einfach zeigt, wie er die Aufgabe ausführt.[3] Es ist also keine Codierung erforderlich. Sollten Fehler auftreten, muss der Besitzer, in diesem Fall also auch der Lehrer, nur mehr Demonstrationen bereitstellen. Es ist also nicht nötig professionelle Hilfe anzufordern. LfD zielt daher darauf ab, Robotern die Möglichkeit zu geben, zu lernen, was es bedeutet, eine Aufgabe auszuführen, indem sie mehrere Demonstrationen verallgemeinern.[4]

[2] https://en.wikipedia.org/wiki/Programming_by_demonstration

[3] Vgl. Christopher G Atkeson and Stefan Schaal College of Computing Georgia Institute of Technology, Robot Learning From Demonstration

[4] Vgl. http://www.scholarpedia.org/article/Robot_learning_by_demonstration

3. Exkurs: Machine Learning

Das Feld des Machine Learnings befasst sich grundlegend mit der Frage, wie Computerprogramme automatisch aus Erfahrungen lernen und sich somit verbessern. Programme, die auf Machine Learning basieren, sollen mithilfe der gesammelten Erfahrungen selbstständig Lösungen für neue und unbekannte Probleme finden.[5] Noch tiefer gehende Konzepte, wie zum Beispiel das Deep Learning, verwenden dem menschlichen Gehirn nachempfundene neuronale Netzwerke um möglichst viele elementare Verknüpfungen innerhalb einer Problemlösung zu erstellen.[6] Diese Verknüpfungen helfen dann bei unbekannten Problemen indem sie auf diese Probleme assoziiert werden.

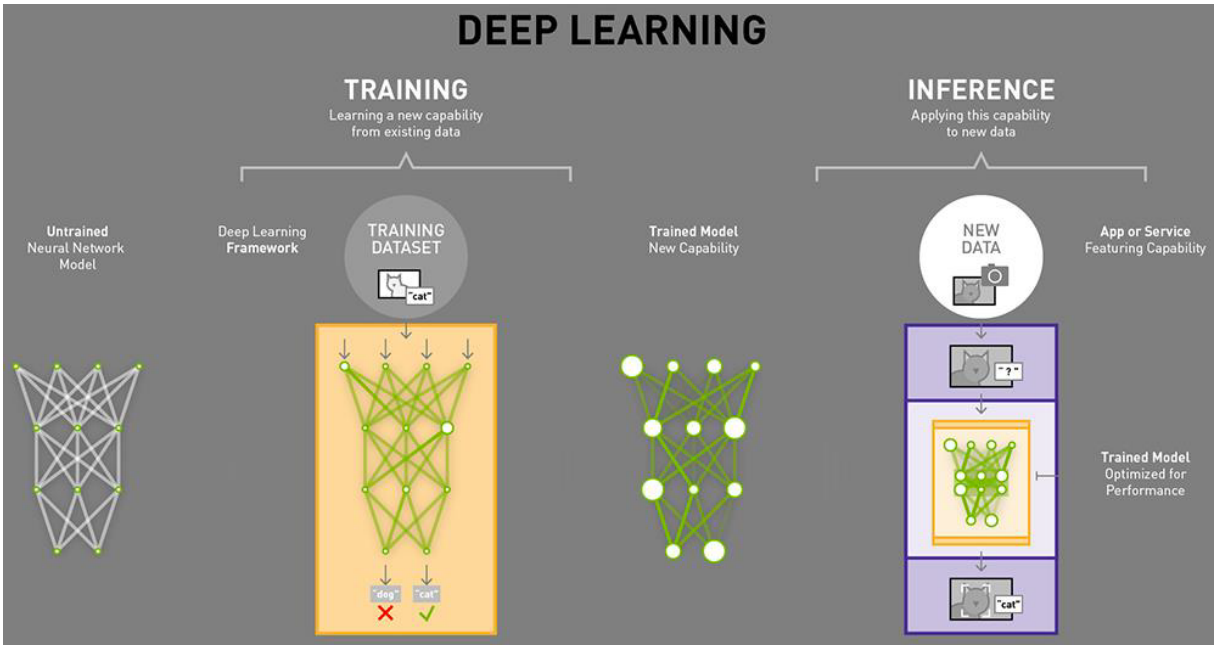

Abb. 1: Training and Inference of NNs, Nvidia Corporation

In Abbildung 1 wird ein in dem Sinne frisches neuronales Netzwerk auf die Aufgabe spezialisiert, Bilder von Katzen als solche zu erkennen. Dem Netzwerk werden Bilder von Katzen zur Verfügung gestellt um ein Datenset aufzubauen. Die Merkmale einer Katze werden vom Netzwerk in Relation gestellt und so gespeichert. Wenn dem Netzwerk nun ein Bild einer bisher unbekannten Katze gegeben wird, soll es die Katze auch als Katze erkennen können. Dazu untersucht das Netzwerk die Merkmale der unbekannten Katze. Wenn diese Merkmale in ausreichender Weise ähnlich zu bereits bekannten und gespeicherten Merkmalen passen, wird das Netzwerk die richtige Antwort geben können. Zusätzlich zum erfolgreichen Lösen der Aufgabe hat das Netzwerk sein Datenset in Bezug auf Katzen erweitert und somit automatisch dazugelernt.

[5] Vgl. https://de.ryte.com/wiki/Machine_Learning
[6] Vgl. https://de.wikipedia.org/wiki/Deep_Learning

4. Aufbau des Datensets

Da ein Roboter meist deutlich komplexere Aufgaben als das Erkennen einer Katze zu bewältigen hat, braucht er auch differenziertere Möglichkeiten Daten aufzunehmen und sich so eine Policy herzuleiten. Es ist hier zwischen einer Demonstration und einer Imitation zu unterscheiden. Während die Bewegungsausführung des Roboters bei der Demonstration zeitgleich zu denen des Lehrers erfolgt, speichert der Roboter die Bewegungen des Lehrers im Falle einer Imitation und reproduziert sie erst später. Zum besseren Verständnis sei gesagt, dass der Begriff „Demonstration“ hier zwei Bedeutungen bzw. Anwendungsgebiete hat. Zum einen beschreibt er namensgebend das gesamte Konzept von *Learning from Demonstration* an sich. Zum anderen wird der Begriff in diesem Abschnitt auch für die Ansätze der *Teleoperation* und *Shadowing* genutzt. Die Demonstration dabei ist das Ausführen der Bewegungen. Hier ist zu bemerken, dass der Imitation des Roboters immer eine Demonstration der Bewegung vorhergeht.

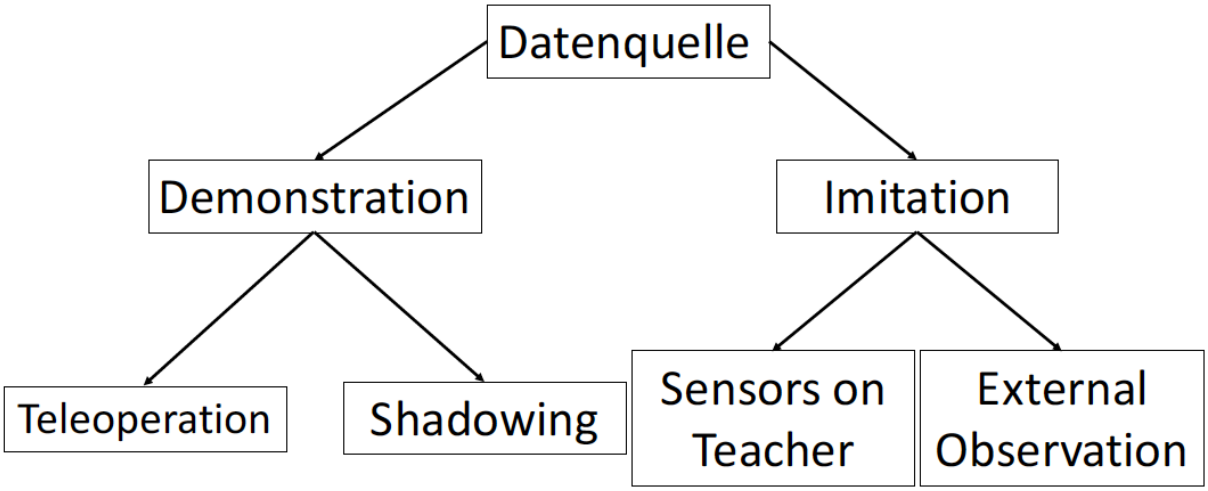

Abb. 3: Demonstration und Imitation[7]

4.1. Demonstration

Die für den Lehrer einfachste Möglichkeit dem Roboter die anwendbaren Aktionen zu zeigen ist, den Roboter per *Teleoperation* direkt durch die Bewegung zu führen. Der Roboter wird hier vom Lehrer gesteuert und speichert bei der Ausführung der Bewegungen die genauen Abläufe, die Veränderung der Winkel in den Gelenken sowie den eigenen Start- und Endzustand. Die Steuerung kann über einen Joystick, über Sprachsteuerung oder, auch wenn Teleoperation

[7] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009),doi:10.1016/j.robot.2008.10.024

eigentlich „Arbeiten auf Distanz“ bedeutet, über ein direktes Führen durch die Bewegungsabläufe geschehen. Der Roboter zeichnet die Daten also über eigene Sensoren auf.
Wenn der Lehrer dem Roboter eine Bewegung demonstriert und der Roboter sie simultan selbstständig versucht auszuführen, spricht man vom *Shadowing*. Offensichtlich ist hier ein zusätzlicher Algorithmus zur aktiven Aufzeichnung und Reproduktion der Daten notwendig. Über das Shadowing war es zum Beispiel möglich einem Roboter bestimmte Armbewegungen beizubringen, die für das Schachspiel nötig sind. [8]

4.2. Imitation

Wie auch im Falle der Demonstration sind hier zwei Ansätze zu beschreiben. Zum einen *Sensors on Teacher*, also auf dem Lehrer angebrachte Sensoren. Der Roboter bekommt seine Informationen also über die Analyse der Bewegung der Sensoren. Der Vorteil dieser Methode ist, dass der Lehrer so die geforderten Bewegungen sehr präzise darstellen kann. Der Nachteil ist jedoch, dass diese Methode sehr aufwendig ist. So braucht man sehr spezielle und kostspielige Sensoren, deren Einsatzmöglichkeiten aufgrund der hohen Spezialisierung begrenzt sind.
Zum anderen existiert bei der Imitation das Konzept der *External Observation*.[9] Der Lehrer trägt hier keine Sensoren auf seinem Körper, was dazu führt, dass er die Bewegungen von sich aus äußerst präzise durchführen muss um einen hochwertigen Lernerfolg beim Roboter zu bewirken. Der Roboter guckt hier von außen auf die Vorführung und trägt typischerweise die Kameras direkt am eigenen Körper.
Es ist schnell ersichtlich, dass sich die beiden Konzepte der Imitation gut kombinieren lassen. So lässt sich zum Beispiel Geld bei der Beschaffung der Sensoren sparen, wenn man diese nur für extrem genaue Bewegungsabläufe nutzt und andere, gröbere Bewegungen über das Konzept der External Observation zu vermitteln. [10]

[8] M. Ogino, H. Toichi, Y. Yoshikawa, M. Asada, Interaction rule learning with a human partner based on an imitation faculty with a simple visuomotor mapping, in: The Social Mechanisms of Robot Programming by Demonstration, Robotics and Autonomous Systems 54 (5) (2006) 414_418 (special issue).
[9] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024
[10] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

5. Herleitung der Policy

Da wir jetzt wissen, wie der Roboter die notwendigen State-Action-Paare aufbaut, stellt sich nun die Frage, wie er aus diesen Daten eine Policy herleiten kann. Das kann ganz einfach eine Approximation des State-Action-Mappings, also eine neue *Mapping Function*, oder ein erstelltes Model der Dynamiken der Welt des Roboters, also ein *System Model*, sein. Außerdem ist es möglich eine Sequenz von Aktionen zu planen, in dem der Roboter die aktuellen sowie die gewünschten Zustände nach Durchführung der Aktion betrachtet. Er plant also, welche Aktionen nötig sind um die Welt von Zustand A in Zustand B zu versetzen. (*Plans*)

5.1. Mapping Function

In diesem Konzept wird ein Algorithmus verwendet, der eine Funktion berechnet, um einen Zustand auf potentielle Aktionen zu mappen.[11]

$$f(): Z \rightarrow A$$

Das Ziel dieses Algorithmus ist es, die noch unbekannte Policy des Lehrers zu reproduzieren und in so weit zu generalisieren, dass auch gültige Lösungen für ähnliche Probleme, aber im Training noch nicht demonstrierte Probleme gefunden werden können.
Die Details der Approximation der Funktion werden von vielen Faktoren beeinflusst. Dazu gehört, ob die Statuseingabe und Aktionsausgabe kontinuierlich passieren, ob die Funktion noch vor oder auch während der Ausführung zu approximieren ist, oder ob dies überhaupt machbar oder wünschenswert ist um das komplette Datenset über den gesamten Prozess des Lernens erhalten bleiben soll.

5.2. System Model

In diesem Konzept werden die demonstrierten Daten genutzt um ein Model der Dynamiken der Welt und eine mögliche Reward-Funktion nach dem Konzept des *Reinforcement Learnings* zu erstellen. RL basiert darauf, dass der Lernende, in unserem Fall der Roboter, selbstständig eine Reward-Funktion erstellt und mithilfe der bereits gewonnenen Information in Form von State-Action-Paaren versucht, das Ergebnis dieser Reward-Funktion zu maximieren.[12] Hier existiert jedoch der Unterschied, dass die Reward-Funktion nicht nur vom Roboter selbst

[11] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024
[12] Vgl. https://de.wikipedia.org/wiki/Best%C3%A4rkendes_Lernen

erstellt wird, sondern auch vom Lehrer durch die Demonstrationen aufgestellt werden kann.

5.2.1. Erstellte Reward-Funktion

Im Bereich der LfD-Techniken stellt der Lehrer meist selbst die Reward-Funktion auf. Diese Funktionen sind häufig sehr spärlich gehalten. Das bedeutet, dass der Funktionswert, abgesehen von einigen wenigen (gewünschten) States, gleich Null ist. Das hilft dabei, den Roboter von sinnlosen Erkundungen abzuhalten und sich auf die Erfüllung der Aufgabe konzentrieren zu lassen.[13] Die Demonstration mithilfe von Teleoperation kann hier genutzt werden um dem Roboter die States zu zeigen, in denen der Funktionswert einen positiven Betrag hat. [14]

5.2.2. Erlernte Reward-Funktion

Den Roboter in einer realen Welt selbst die Reward-Funktion erlernen zu lassen kann sehr schnell sehr ausufern. Die Menge an möglichen States in der realen Welt ist unheimlich groß. Es ist dem Roboter somit nahezu unmöglich, sich auf alle Eventualitäten einzustellen und diese bei der Berechnung der Reward-Funktion miteinzubeziehen. Dieser Ansatz wird daher nur bei elementaren Aufgaben, die so klein sind, dass sich als Lehrer das Erstellen einer Reward-Funktion nicht lohnt, angewandt.[15]

5.3. Plans

In diesem Konzept werden die demonstrierten Daten dazu genutzt, Regeln über Auswirkungen von Aktionen abzuleiten. Eine Aktion wird über zwei verschiedenen States abgebildet: Zum einen die Pre-Condition, also der State, der erreicht sein muss, um die gewünschte Aktion ausführen zu können und, zum anderen die Post-Condition, also der State, der durch die Ausführung der Aktion erreicht werden soll.[16] Es ist also möglich die Planung rückwärts zu gestalten. Der Roboter kennt den gewünschten State nach Ausführung der Aktionen und sucht nun nach den passenden State-Action-Paaren, um den aktuellen State in den gewünschten State zu transformieren.

---

[13] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

[14] E. Oliveira, L. Nunes, Learning by exchanging advice, in: R. Khosla, N. Ichalkaranje, L. Jain (Eds.), Design of Intelligent Multi-Agent Systems, Springer, New York, NY, USA, 2004 (Chapter 9).

[15] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

[16] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

6. Limitationen des Datensets

Roboter sind von Natur aus an die Informationen im Datenset gebunden. Damit einhergehend ist das Ergebnis der Ausführungen des Roboters streng von der Qualität der Daten abhängig. Reicht die zur Verfügung gestellte Menge an Information nicht aus, ist es sehr wahrscheinlich, dass die Ausführung des Roboters nicht den Wünschen des Lehrers entspricht. Es ist von zwei möglichen Gründen für die schlechte Performance des Roboters auszugehen: ein nicht-demonstrierter State und schlechte Datenqualität.

6.1. Nicht-demonstrierter State

Der Zugang zu Daten und Informationen limitiert alle LfD Policies. Das liegt zumeist am Lehrer selbst, da er nicht in der Lage ist, alle möglichen anwendbaren Aktionen auf jeden möglichen State zu demonstrieren. Um die gegebene Aufgabe trotzdem lösen zu können, gibt es verschiedene Möglichkeiten mit dem nicht-demonstrierten State umzugehen.

Zum einen kann bereits gewonnene und gespeicherte Information genutzt werden um bekannte und demonstrierte States zu generalisieren und so eine Lösung für das unbekannte Problem zu finden. Die genaue Art der Generalisierung unterscheidet sich hier insofern, ob der Roboter den ähnlichsten bekannten State sucht um anwendbare Aktionen auszuwählen oder Merkmale des unbekannten States in vielen bekannten States versucht wiederzufinden und so aus Aktionen einen Durchschnitt bildet.

Zum anderen kann der Roboter erkenntlich machen, dass die gegebene Menge an Informationen für eine erfolgreiche Ausführung der Aufgabe nicht ausreicht und so neuerliche Demonstrationen von Seiten des Lehrers fordern. Diese Methode wird vom Roboter meistens dann gewählt, wenn er auf einen komplett neuen State trifft und keine Möglichkeit sieht, bekannte States zu generalisieren und eine neue Policy aus bekannten Policies herzuleiten. [17]

6.2. Schlechte Datenqualität

Die Qualität einer erlernten Policy hängt stark von der Qualität der bereitgestellten Demonstrationen ab. Generell ist aus Sicht des Roboters davon auszugehen, dass die Demonstration von einem Experten durchgeführt wurde. Trotzdem kann es natürlich dazu kommen, dass eine Demonstration nicht richtig oder nicht ausführlich genug durchgeführt wurde und der Roboter so an

[17] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

schlechte oder nicht vollständige Informationen kommt. Auch hier gibt es zwei Ansätze mit schlechter Datenqualität umzugehen. Sind dem Roboter verschiedene Ansätze zum Lösen eines Problems zur Verfügung gestellt worden, sollte er die Möglichkeit besitzen, die schlechten Ansätze aus seinem Speicher zu löschen und sich so auf die erfolgsversprechenden Ansätze zu beschränken.
Gegensätzlich läuft der Ansatz schlechte Erfahrungen zu speichern und in die Herleitung einer Policy einfließen zu lassen. Hier ist es hilfreich als Lehrer Feedback zur Verfügung zu stellen um dem Roboter die Auswirkungen seiner Aktionen zu verdeutlichen. Auf einen unbekannten State wird der Roboter so nur Aktionen anwenden, die bereits ein positives Feedback erwirkt haben. [18]

7. Zusammenfassung

In dieser Arbeit wurden Techniken vorgestellt, einem Roboter mit Konzepten aus *Learning from Demonstration* menschliche Bewegungen beizubringen. Diese Konzepte bringen eine intuitive Kommunikationsmethode zwischen Lehrer und Roboter so wie offenen Kontrollalgorithmen mit sich. Besonders für Menschen ohne Erfahrungen auf dem Gebiet der Robotik sind die Ansätze hilfreich und leicht zu verstehen. Die Techniken lassen sich grob in zwei Phasen unterteilen:
Die erste Phase besteht aus dem Aufnehmen und des Akquirierens der Demonstrationsbeispiele. Hier wurde beschrieben, wie die Demonstration ausgeführt und gespeichert wurde. Nämlich *Teleoperation*, *Shadowing*, *Sensors on Teacher* und *External Observation*. Hier wurde zwischen Ausführung einer Demonstration und Imitation von Seiten des Roboters unterschieden. In der zweiten Phase ging es um die Herleitung einer Policy sowie um die Limitationen und deren Überwältigung bei der Arbeit mit Robotern unter den gegebenen Umständen. Die vorgestellten Konzepte reichen über *Mapping Function* und *System Model* zu *Plans*.

[18] Vgl. B.D. Argall, et al., A survey of robot learning from demonstration, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2008.10.024

8. Literaturverzeichnis

B.D. Argall, et al., A survey of robot learning fromdemonstration, Robotics and Autonomous Systems (2009)

Baris Akgun, et al., Keyframe-based Learning from Demonstration Method and Evaluation

Aude Billard and Daniel Grollman (2013), Scholarpedia, 8(12):3824.

https://koroibot-motion-database.humanoids.kit.edu/list/motions/, 20.03.2018

http://rll.berkeley.edu/deeprlcourse/, 20.03.2018

Stefan Schaal, Learning From Demonstration

A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programmingby demonstration," in Springer handbook of robotics. Springer, 2008, pp. 1371–1394.

Jangwon Lee, A survey of robot learning fromdemonstrations for Human-Robot Collaboration (2017)

Training andInferenceofNNs, NvidiaCorporation

https://www.forbes.com/sites/aarontilley/2017/09/19/ai-startup-invents-trick-for-robots-tomore-efficiently-teach-themselves-complex-tasks/#17b0cd2a15fe, 20.03.2018

https://de.wikipedia.org/wiki/Best%C3%A4rkendes_Lernen, 20.03.2018

J.A. Clouse, On integrating apprentice learning and reinforcement learning.

Ph.D. Thesis, University of Massachusetts, Department of Computer Science,

1996.