Versuch und Irrtum

Eine überraschend große Zahl an Studien in der Psychologie, die Lernen untersuchen, setzen Ratten in ein Labyrinth. Das Labyrinth steht meist auf einem großen Tisch in einem Labor und hat undurchsichtige Wände, ist aber nach oben hin offen, sodass die Tiere sehen können, wo die Regale, die Lampen oder die Fenster im Raum sind. Dadurch verlieren die Ratten nicht komplett die Orientierung. Wenn ich durch eine Stadt laufe, orientiere ich mich zum einen daran, wie die Häuser aussehen, aber es ist auch immer gut, wenn durch die Straßenschluchten oder über den Häuserdächern Kirchtürme oder Hochhäuser zu sehen sind. Und komme ich in eine neue Stadt, komme ich mir manchmal vor wie eine Ratte in einem Labyrinth. Insbesondere, wenn ich mal wieder mein Handy vergessen habe, hungrig am Bahnhof ankomme und jetzt das einzige Restaurant suche, das noch nach 22 Uhr eine warme Küche hat. Das ist die Situation, in der die Ratte ist. Sie hofft, dass irgendwo in diesem Labyrinth etwas zu essen versteckt ist (aber man kann es nicht riechen und einfach nur der Nase folgen).

Beim ersten Mal, wenn die Ratte in das Labyrinth gesetzt wird, kann sie nichts anderes tun, als durch das Labyrinth zu laufen, bis sie irgendwann zufällig auf das Futter stößt. Wird die Ratte am nächsten Tag wieder in das Labyrinth gesetzt, dann könnte man meinen, dass sie vielleicht direkt wieder zu der Stelle läuft, wo es am Vortag etwas zu essen gab. Wahrscheinlich würde sie das auch gerne tun. Aber so wie auch ich nicht mehr weiß, wie ich vom Bahnhof zum einzigen offenen Restaurant gekommen bin und mich beim nächsten Besuch wieder verlaufe, läuft auch die Ratte nicht auf dem direkten Weg zum Futter, sondern irrt umher. Über mehrere Tage, in denen die Ratte immer wieder in das Labyrinth gesetzt wird und immer wieder an der gleichen Stelle Futter findet, wird das Herumirren allerdings immer weniger und die

Ratte läuft irgendwann auf dem kürzesten Weg vom Startpunkt zum Zielpunkt. Die Ratte hat offenbar gelernt.

Im Kapitel über Suchalgorithmen hatte ich beschrieben, wie man mit einer Karte den kürzesten Weg von einem Startpunkt zu einem Zielpunkt findet, indem man eine Heuristik benutzt. Abbildung 9 zeigt nochmal die Netzkarte mit den Shuttlerouten zwischen verschiedenen Planeten. Shuttles fliegen entlang der schwarzen (nicht der grauen) Linien und die Zahl auf jeder Linie zeigt an, wie viele Monate die Reise zwischen zwei Planeten dauert. Möchte man von Alderaan nach Endor fliegen, ist die kürzeste Route über Felucia, Corellia und Dagobah. Eine gute Heuristik ist, immer zunächst den Planeten anzufliegen, der dem Ziel am nächsten ist, weshalb von Alderaan aus Felucia vielversprechender aussieht als Bespin. Die Suche nach dem kürzesten Weg nach Endor ist die gleiche Art von Problem, welches die Ratten in ihrem Labyrinth haben. Die Ratten im Labyrinth können aber nicht die gleiche Heuristik nutzen, weil sie gar nicht wissen, wo das Ziel ist. Und selbst, wenn sie es nach einigen Versuchen wüssten, haben sie keine Karte, auf der sie den Abstand zum Ziel leicht messen könnten. Eine vielversprechende Hypothese ist aber, dass die Ratten über die Zeit eine immer bessere Heuristik lernen, und daher immer weniger umherirren.

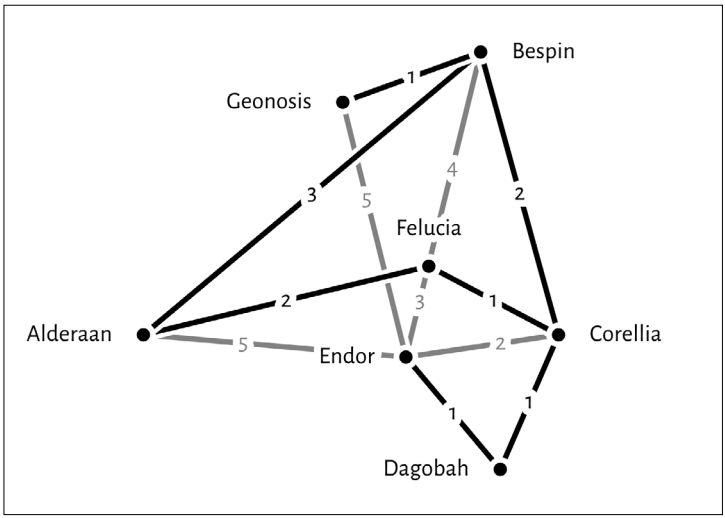

Abb. 9: Fast dieselbe Planetenkarte wie zuvor

In Abbildung 9 ist auf den grauen Linien mit den grauen Zahlen verzeichnet, wie lange die Reise von jedem Planeten aus nach Endor dauert, wenn man die kürzeste Route nimmt, die möglich ist. Das ist die beste Heuristik, denn es ist nicht nur eine Abschätzung durch die Luftlinie, sondern die exakte Lösung (darin unterscheidet sich die Abbildung von dem Beispiel aus dem früheren Kapitel).

Mit dieser perfekten Heuristik biegt man niemals falsch ab. Wollen wir beispielsweise von Bespin nach Endor reisen, haben wir die Wahl, von dort nach Geonosis, Alderaan oder Corellia zu fliegen. Die Zahlen auf den grauen Linien zeigen an, dass es von Geonosis oder Alderaan fünf Monate Lebenszeit kostet, um nach Endor zu kommen. Von Corellia aus nur zwei Monate. Also fliegen wir nach Corellia. Obwohl Felucia der Luftlinie nach näher an Endor liegt als Dagobah, fliegt man besser nach Dagobah, weil es von dort nur noch einen Monat Lebenszeit nach Endor kostet. Mithilfe dieser Heuristik muss man nicht mehr langwierig nach der besten Route suchen, sondern man fliegt immer einfach zu dem Planeten weiter, der die kleineren Lebenszeitkosten hat.

Damit wir diese perfekte Heuristik nutzen können, muss aber jemand vorher die Kosten berechnet und sie auf der Karte notiert haben. Man kann die Länge der kürzesten Route nach Endor für jeden Planeten berechnen, indem man mit der Rechnung am Ziel anfängt (im Kapitel zu Suchalgorithmen hatten wir immer am Start angefangen). Wenn man in Endor ist, dann ist man am Ziel. Um da hinzukommen, muss man vorher in Dagobah gewesen sein, was einen Monat vom Ziel entfernt ist. Da Dagobah einen Monat von Corellia entfernt ist, braucht man von Corellia zwei Monate nach Endor, und so weiter. So können wir mit Papier und Bleistift die bestmögliche Heuristik bestimmen, aber wie können die Ratten lernen, sich entsprechend zu verhalten?

Wie Verhalten verstärkt wird

Eine alte Idee der Psychologie besagt, dass Tiere – und Menschen manchmal auch – aus Versuch und Irrtum lernen. Im Jahr 1898 setzte Edward Thorndike Katzen in einen verschlossenen Käfig, aus dem sie sich selber befreien konnten, indem sie einen Mechanismus betätigten. Er beobachtete, dass die Tiere viele verschiedene Sachen machen, bis sie irgendwann den richtigen Mechanismus finden. Er beobachtete auch, dass die Tiere, wenn sie wiederholt in den gleichen Käfig gesetzt

werden, sich nach und nach immer schneller befreien. Die Katzen in seinem Experiment scheinen nicht die Situation zu evaluieren und durch Nachdenken zu einer Lösung zu kommen, die sie immer wieder anwenden können. Sie probieren vielmehr zufällig verschiedene Verhaltensweisen aus, um ihr Ziel zu erreichen. Hat ein Verhalten zum gewünschten Erfolg geführt, dann wird dieses Verhalten verstärkt. Damit ist gemeint, dass die Wahrscheinlichkeit, in der Zukunft in der gleichen Situation wieder das gleiche Verhalten zu zeigen, ansteigt. Das nennt man in der Psychologie auch das ›Gesetz der Wirkung‹.[1]

Psychologinnen und Psychologen haben im 20. Jahrhundert Ratten deshalb so oft in Labyrinthe gesetzt, weil sie herausfinden wollten, wie diese Verstärkung von Verhalten durch Erfolg im Detail funktioniert. In einem Labyrinth hängt der Erfolg oder Misserfolg von einer ganzen Kette von Entscheidungen ab – jede Weggabelung kann die Ratte entweder näher ans Ziel bringen oder weiter davon entfernen. Eine wichtige theoretische Frage ist daher, wie die einzelnen Entscheidungen verstärkt werden, wenn unklar ist, was jede einzelne Entscheidung in der Entscheidungskette genau zum Erfolg beigetragen hat. Manche der Entscheidungen haben vielleicht zu Umwegen geführt und es wäre schlecht, diese so stark zu verstärken, dass das Tier danach immer den Umweg nimmt und deshalb vielleicht die Abkürzung nie entdeckt. Eine Theorie, die erklären kann, wie Tiere in einem Labyrinth lernen, sollte – so die Hoffnung – auch Hinweise darauf geben, wie Tiere jedes andere komplexe Verhalten lernen, das aus einer längeren Abfolge von Entscheidungen und Handlungen besteht. Inspiriert von dieser psychologischen Forschung entwickelte sich ein Teilgebiet des maschinellen Lernens, das man ›verstärkendes Lernen‹ nennt.[2] Erkenntnisse im maschinellen Lernen haben wiederum zu neuen Experimenten in Hirnforschung und Psychologie geführt. Neben künstlichen neuronalen Netzen ist das verstärkende Lernen ein weiteres Beispiel dafür, wie sich psychologische und neurowissenschaftliche Grundlagenforschung und KI-Forschung gegenseitig befruchtet haben.[3]

1 Siehe Thorndike (1898).

2 Auf Englisch ›reinforcement learning‹.

3 Das Standardwerk zu verstärkendem Lernen ist Sutton & Barto (2018).

Ein Lernalgorithmus, der Lernen in Computern und Ratten beschreibt, ist das sogenannte ›Q-Lernen‹.[4] Stellen wir uns eine Ratte vor, die durch ein Labyrinth läuft, das wie unsere Planetennetzkarte aussieht. Die Ratte startet in Alderaan und das Futter ist am Ende des Labyrinths in Endor versteckt. Die Ratte will möglichst schnell zum Futter kommen. Jeder Gang verursacht Kosten (auch für Ratten ist Zeit Geld) und die Ratte will diese Kosten minimieren. Die Ratte wird immer wieder in das Labyrinth gesetzt und der Algorithmus soll beschreiben, was die Ratte jedes Mal macht und wie sich das Verhalten der Ratte durch Lernen ändert. Hinter dem Algorithmus steckt die Idee, für alle Planeten zu lernen, welche Kosten und Folgekosten jede der möglichen Entscheidung bis zum Ziel verursachen wird.[5]

Biegt die Ratte in Alderaan rechts nach Felucia ab, sind die Gesamtkosten bis zum Ziel in Endor 5, sofern das Tier ab Felucia den kürzesten Weg nimmt. Da das die Kosten für den kürzesten Weg sind, ist das die Zahl, die in Abbildung 9 auf der grauen Linie von Alderaan nach Endor steht. Es ist außerdem die Summe der Kosten für die Strecke von Alderaan nach Felucia und der Kosten des kürzesten Weges von Felucia nach Endor: 5=2+3. Denn so hatten wir die Kosten des kürzesten Weges ja von hinten anfangend berechnet. Die Ratte muss also lernen, dass wenn sie in Alderaan rechts abbiegt, die Endkosten bestenfalls 5 sein werden. Biegt die Ratte allerdings in Alderaan links nach Bespin ab, dann sind die Endkosten bestenfalls 7, denn die Kosten von Alderaan nach Bespin sind 3 und die Kosten des kürzesten Weges von Bespin nach Endor sind 4 und 7=3+4. Das muss die Ratte auch lernen. Sobald sie beides gut genug gelernt hat, wird sie sich in Alderaan immer für das Rechtsabbiegen entscheiden, weil das die geringeren Endkosten verursacht.

Im Algorithmus wird nun angenommen, dass die Ratte für jede Abbiegung, die sie in Alderaan oder jedem anderen Planeten nehmen kann, eine Schätzung dafür hat, wie hoch die Kosten am Ende sein werden, wenn sie sich für diese Abbiegung entscheidet – die Ratte hat also eine Heuristik, die ihr wie beim Topfschlagen ansagt, wo es wärmer und käl-

4 ›Q-Learning‹ auf Englisch. Der Algorithmus wurde zuerst in der Doktorarbeit von Watkins (1989) beschrieben und die zugehörige Theorie ist gut ausgearbeitet (Watkins & Dayan, 1992).

5 Statt der Kosten wird beim Q-Lernen traditionell die Güte der Entscheidungen gelernt. Das macht aber keinen Unterschied, denn die zwei sind bis auf das Vorzeichen identisch: Kosten haben einfach eine negative Güte. Die Güte wird üblicherweise mit dem Buchstaben ›Q‹ für ›quality‹ abgekürzt, daher der Name des Lernalgorithmus.

ter wird. Wenn der Algorithmus in einem Computer läuft, werden diese Schätzungen einfach als Zahlen gespeichert (den sogenannten ›Q-Werten‹). Bei der Ratte stellt man sich aber besser vor, dass sie ein mehr oder weniger gutes Gefühl mit den einzelnen Abbiegungen assoziiert. Ein gutes Gefühl signalisiert der Ratte, dass sie auf dem richtigen Weg ist, ein schlechtes, dass sie sich auf dem Holzweg befindet.

Wird die Ratte das erste Mal in das Labyrinth gesetzt, kann sie noch keine vernünftige Schätzung für die Kosten der einzelnen Abbiegungen haben. Meist wird angenommen, dass die Ratte mit optimistischen Schätzungen beginnt und dass jede neue Abbiegung für sie daher potenziell interessant ist. Die Erfahrungen, die die Ratte nach der Entscheidung für eine Abbiegung macht, fließen in zukünftige Schätzungen ein und verbessern sie. Wenn die Ratte nun zufällig von Alderaan nach Felucia läuft, erfährt sie, dass die Kosten für diesen Gang 2 sind. Sie hat eine Schätzung dafür, wie teuer die beste Entscheidung in Felucia am Ende sein wird, und sie nutzt diese Schätzung zusammen mit den gerade direkt erfahrenen Kosten von 2, um die Schätzung der Endkosten für die gerade in Alderaan getroffene Entscheidung zu aktualisieren. Durch die direkte Erfahrung wird die Schätzung der Endkosten realistischer, selbst wenn die Schätzung für Felucia vielleicht noch nicht gut war. In Felucia trifft die Ratte wieder eine Entscheidung und aktualisiert dann die Schätzung der Endkosten für diese Entscheidung, und auch diese Schätzung wird dadurch realistischer. Und so geht es weiter, bis die Ratte durch Zufall in Endor landet und vom Versuchsleiter wieder an den Anfang gesetzt wird. Im nächsten Durchgang wiederholt sich der ganze Vorgang und so werden die Schätzungen der Endkosten für jede Entscheidung immer realistischer. Solange die Schätzungen noch nicht verlässlich sind, ist das Verhalten des Algorithmus mehr oder weniger zufällig und die Ratte erkundet das ganze Labyrinth. Je besser die Schätzungen werden, desto mehr trifft der Algorithmus die richtigen Entscheidungen und die Ratte läuft auf dem kürzesten Weg zum Futter.

Lernen ist mehr als Verstärkung

Bevor hier der falsche Eindruck entsteht, dass dieser Algorithmus perfekt erklären könnte, wie Ratten lernen: Verstärkendes Lernen ist nur ein kleiner, aber gut verstandener Teil einer vollständigen Erklärung.

Es gibt mehrere Phänomene, die nicht leicht alleine durch Versuch und Irrtum erklärt werden können. Sagen wir, unsere Ratte hat nun in unserem Planetenexperiment durch zufälliges Umherirren und Verstärkung von erfolgreichem Verhalten gelernt, wie sie schnellstmöglich von Alderaan nach Endor kommt. Die Ratte wird wieder in Alderaan ausgesetzt, aber diesmal gibt es zwei neue Gänge, die wir über Nacht an das Labyrinth angebaut haben. Der eine Gang geht von Alderaan direkt nach Geonosis und der andere direkt nach Endor. Die Ratte nimmt, ohne zu zögern, den neuen Gang nach Endor, wo das Futter ist. Da beide Gänge neu sind, sollte der Algorithmus beide zufällig erkunden, aber die Ratte ist offensichtlich schlauer als der Algorithmus, weil sie weiß, in welcher Richtung das Ziel liegt. Der Algorithmus kann auch nicht erklären, warum Ratten, die zunächst mehrere Male das Labyrinth erkundet haben, ohne dass es irgendwo im Labyrinth Futter gab, später schneller lernen, wie sie zur Futterquelle in Endor finden. Und wenn es auf einmal kein Futter mehr in Endor gibt, sondern in Geonosis, können die Ratten sich auch erstaunlich schnell umstellen. Diese Phänomene lassen sich nicht dadurch erklären, dass die Ratte jede Abbiegung mit einem guten oder schlechten Gefühl assoziiert hat. Die Ratte muss eine Karte des Labyrinths im Kopf haben, die es ihr erlaubt, bei einer Änderung des Ziels schnell umzuplanen.[6]

Bei Menschen und Menschenaffen ist die Vorstellung, dass sie Probleme nur durch Versuch und Irrtum lösen, ohnehin lächerlich. In einer berühmten (und ebenfalls sehr alten) Studie hat Wolfgang Köhler im Jahr 1921 Schimpansen verschiedene Probleme lösen lassen, um an Bananen zu kommen. In einem Versuch hing eine Banane an einer Schnur von der Decke, sodass der Affe sie nicht erreichen konnte. Neben vielen anderen Gegenständen standen im Gehege auch mehrere Kisten herum, die der Affe nicht kannte. Er hatte auch noch nie vorher dieses spezielle Problem gelöst. Nach einer Weile, in der der Affe wohl nachdachte, stapelte er die Kisten so übereinander, dass er die Banane erreichen konnte. Köhler beobachtete kein zufälliges Ausprobieren, sondern zielgerichtetes Handeln – ganz im Gegensatz zu den Beobachtungen, die Thorndike bei Katzen gemacht hatte. Die Affen lernen nicht langsam über viele Versuche sich zielgerichtet zu verhalten, sondern scheinen direkt aus Einsicht zu handeln.[7]

6 Siehe Tolman (1948).

7 Siehe Köhler (1921).

Wie wir bereits im Kapitel über Suchalgorithmen und im Kapitel über Schach gesehen hatten, lassen sich viele verschiedene Probleme als Suchprobleme verstehen. Schach hat eine große Zahl an Zuständen (alle möglichen Konfigurationen der Spielfiguren) und jeder Zug ist eine Reise von einem Zustand zu einem anderen. Genauso wie wir eine Netzkarte für Shuttlereisen zwischen Planeten erstellen können, können wir auch eine Karte der möglichen Zustände im Schach erstellen. Ein Spieler will möglichst schnell einen Zustand finden, in dem er gewinnt. Mit der kleinen Komplikation, dass es einen Gegenspieler gibt, der das verhindern möchte. Schach ist trotzdem nur ein großes Labyrinth, in dem irgendwo Futter versteckt ist. Weil aber der Suchraum im Schach nicht nur groß, sondern mit 10^{43} Zuständen riesig ist, funktionieren Suchalgorithmen nur, wenn ihnen eine Heuristik hilft abzuschätzen, wie gut mögliche Züge wohl am Ende sein werden. Nur weil mehrere Großmeister IBM geholfen hatten, eine gute Heuristik zu entwickeln, konnte Deep Blue gegen Garri Kasparow gewinnen. Hätte Deep Blue diese Starthilfe durch menschliche Intelligenz nicht gehabt, hätte Kasparow nicht verloren. Computer können aber – so wie Ratten – durch verstärkendes Lernen selbständig Heuristiken lernen. Können Computer daher auch ohne Einsicht – rein durch Versuch und Irrtum – lernen, Schach zu spielen? Kann man so ein lernendes Schachprogramm entwickeln, das nicht mehr auf Starthilfe durch die menschliche Einsicht in das Spiel angewiesen ist?

Computer lernen Menschen zu imitieren

Das Brettspiel Go ist in Asien ähnlich beliebt wie in Europa Schach. Das quadratische Brett besteht aus 19x19 Feldern, die am Anfang des Spiels leer sind. Die zwei Spieler legen abwechselnd schwarze und weiße Spielsteine auf das Brett und versuchen den Gegner zu umzingeln. Der Suchraum bei diesem Spiel ist mit 10^{170} Zuständen deutlich größer als beim Schach.[8] Die Anzahl der Atome im beobachtbaren Universum

8 Jedes Feld kann leer sein oder mit einem schwarzen oder weißen Stein belegt sein. Für jedes Feld gibt es also drei Möglichkeiten. Bei einem Brett mit 19·19=361 Feldern gibt es daher $3^{361} \approx 10^{172}$ Möglichkeiten, die Steine auf das Brett zu legen. Wenn man berücksichtigt, dass nicht alle diese Zustände in einem Spiel legal auftreten können, kommt man auf etwa 10^{170} Spielzustände (Tromp & Farnebäck, 2016).

wird im Vergleich dazu auf ›nur‹ 10^{80} geschätzt (plus minus ein paar Größenordnungen). Der Ansatz, Experten eine Heuristik entwickeln und dann ein KI-Programm nach guten Zügen suchen zu lassen, der bei Schach so erfolgreich war, ist bei Go gescheitert. Die Heuristiken für Go waren nicht gut genug für einen so riesigen Suchraum. Wie beim Schach gibt es Großmeisterinnen und Großmeister, die dieses Spiel extrem gut spielen. Und wie beim Schach ist auch ihr Wissen zu großen Teilen implizit. Die Go-Experten schaffen es nicht, explizit genug zu erklären, wie sie spielen, um einen Computer entsprechend zu programmieren. Ein neuer Ansatz musste her, um Computern Go beizubringen. Im Jahr 2016 hat dann ein Computerprogramm namens AlphaGo der Firma DeepMind (die zu Google gehört) in einem öffentlichkeitswirksam inszenierten Spiel einen der besten Go-Spieler der Welt, Lee Sedol, geschlagen. Dieses Programm hat seine Suchheuristik mit verstärkendem Lernen gelernt.[9]

Verstärkendes Lernen lernt aus Erfolg. Damit das funktioniert, muss der Lernalgorithmus auch manchmal Erfolg haben. Anfangs probiert das Programm nur zufällig Züge aus. Wenn der Suchraum riesig ist, ist es extrem unwahrscheinlich, dass man durch Zufall zum Ziel gelangt. Und falls so eine zufällige Suchstrategie bei Go gewinnt, tut sie das nur, weil der Gegner noch schlechter gespielt hat. So lernt man nicht gut zu spielen. Damit ein Computerprogramm in großen Suchräumen das Ziel oft genug findet, um eine Heuristik lernen zu können, braucht es eine Heuristik, die die Suche leitet. Eine klassische Henne-Ei-Situation.

Wie ist es DeepMind also gelungen, einem Lernalgorithmus die nötige Starthilfe zu geben, wenn selbst sehr gute Go-Spieler nur implizit wissen, wie ihre Heuristik aussieht? Im Internet sind eine große Zahl an Go-Partien mit allen Zügen dokumentiert. (Was würden KI-Forscherinnen und -Forscher nur ohne die großen Datenmengen im Internet tun?) Statt selber zu spielen, schaut sich der Lernalgorithmus eine Heuristik aus den Zügen von anderen Spielern ab! Der Algorithmus lernt so zunächst, erfolgreiche menschliche Spieler zu imitieren. Danach lässt man das Computerprogramm gegen sich selber spielen – und zwar richtig lange. Die Heuristik verbessert sich dadurch weiter. Und mit genügend Übung spielt das Computerprogramm dann besser als Lee Sedol.

9 Das Programm ist in dem Artikel von Silver et al. (2016) beschrieben.

Eine weitere nötige Zutat für diesen Erfolg der KI-Forschung war die Kombination von verstärkendem Lernen mit künstlichen neuronalen Netzen. Es ist ausgeschlossen, dass der Lernalgorithmus für jeden möglichen Zustand des Spiels seine aktuelle Schätzung dafür speichert, wie gut er ist, denn diese Tabelle wäre deutlich größer als die Anzahl der Atome im beobachtbaren Universum. Gute menschliche Spieler erkennen auf dem Brett bestimmte Muster. Diese Muster sind Teil des impliziten Wissens, das Expertinnen und Experten besitzen und das ihnen erlaubt, extrem gut zu spielen. Nicht jedes Detail einer Stellung ist wichtig, um abschätzen zu können, ob eine Stellung gut oder schlecht ist. Vielmehr reichen menschlichen Experten nur wenige Merkmale einer Stellung, um mit einer Heuristik eine Abschätzung zu machen. Die Experten erkennen bestimmte Muster, die sie mit einem guten oder schlechten Gefühl assoziieren. Wir haben im Kapitel über künstliche neuronale Netze gesehen, dass diese gut darin sind, Muster zu lernen. Daher liegt es nahe, die Heuristik mit einem neuronalen Netz zu berechnen, das seine Mustererkennung selbständig an das Problem anpassen kann.[10]

Dass die Menschheit sich nun auch in Go den Computern geschlagen geben musste, war nach Schach ein weiterer Meilenstein der KI-Entwicklung. Solange Maschinen durch verstärkendes Lernen nur Schach und Go lernen, könnte man meinen, KI wäre reine Spielerei. Ein gutes Gegenbeispiel ist StarCraft, in dem KI-Systeme mittlerweile auch sehr gut geworden sind.[11] Das Computerspiel simuliert ein Kriegsszenario. Kriegsspiele werden seit jeher im Militär genutzt, um verschiedene Szenarien theoretisch durchzuspielen (das war ja auch der Zweck des KI-Programmes WOPR im Film *War Games*). In StarCraft müssen die Spieler verschiedene Rohstoffe finden und abbauen, ihre Wirtschaft managen, Waffen entwickeln und produzieren sowie eine Armee aufbauen. Bei einem Angriff steuern sie die einzelnen militärischen Einheiten und müssen dabei darauf achten, ihre Langzeitstrategie und ihre Wirtschaft ebenso wenig zu vernachlässigen wie die Rüstungsproduktion. Der Reiz des Spieles ist unter anderem, dass es recht schnell abläuft und viel gleichzeitig passiert. Daher muss man ziemlich

10 Tesauro (1992) konnte zeigen, dass die Idee, verstärkendes Lernen und neuronale Netze zu kombinieren, tatsächlich für praktische, nicht-triviale Probleme funktionieren kann. Er zeigte das am Beispiel von Backgammon.

11 Siehe Vinyals et al. (2019).

lange üben, bis man schnell genug reagieren kann. Dieser Aspekt ist für Computer, die viel besser multitasken können als wir, trivial – im Gegensatz zur strategischen Planung und der flexiblen Anpassung an sich ständig ändernde Situationen. Darüber hinaus gibt es nicht nur Gegenspieler, die sich unberechenbar verhalten können, sondern auch Teampartner, mit denen man zusammenarbeiten muss. Um in diesem Spiel Erfolg zu haben, braucht ein KI-System eine ganze Reihe von Fähigkeiten, die KI-Systeme auch in der wirklichen Welt brauchen, wenn sie in Zukunft immer mehr Aufgaben für uns autonom erledigen sollen. Eine Aufgabe wie Autofahren lässt sich zum Beispiel in einem Computerspiel simulieren. Und in dem Maße, wie Computerspiele inzwischen die Komplexität von Alltagsproblemen erreichen, sind KI-Programme, die Computerspiele spielen, wirklich keine Spielerei mehr.

Computer lernen fast von alleine

Trotz aller Erfolge mit verstärkendem Lernen fuchste die KI-Forscherinnen und -Forscher lange noch, dass auch bei AlphaGo menschliche Spieler dem Computer Starthilfe geben mussten. Nicht mehr ganz so explizit wie bei Deep Blue, aber AlphaGo hatte immer noch Zugriff auf eine große Datenbank an Go-Partien, die von Menschen gespielt wurden. Das implizite Wissen der menschlichen Spieler konnte so von AlphaGo genutzt werden. Kann man nicht auch ein Computerprogramm entwickeln, das ohne diese menschliche Starthilfe auskommt? Ein Programm, das tabula rasa startet? Der Nachfolger von AlphaGo heißt AlphaZero und fängt wirklich bei null an. AlphaZero spielt von Anfang an gegen sich selber und lernt so immer bessere Heuristiken. Am Anfang spielt das Programm nämlich nur schlecht, aber da es gegen sich selber spielt, gewinnt und verliert es bei jedem Spiel. Entscheidungen, die zum Sieg geführt haben, werden verstärkt. So lernt das Programm, nach und nach immer besser zu spielen. Um Go zu lernen, hat AlphaZero 140 Millionen Spiele gegen sich selber gespielt. Um Schach zu lernen, waren es nur 44 Millionen.[12]

Ohne enorm viel Rechenleistung wäre das nicht möglich gewesen. Mehr Rechenpower und mehr Zeit als jemals ein Mensch zur Verfügung haben wird, um Go oder Schach zu lernen. Nehmen wir an, dass

12 AlphaZero ist in dem Artikel von Silver et al. (2018) beschrieben.

ein Schachspiel grob eine Stunde dauert. Ein Jahr hat etwa 50 Wochen, bei 5 Arbeitstagen mit jeweils 8 Stunden, kommt man im Jahr auf 2.000 Arbeitsstunden. Jemand, der 50 Jahre arbeitet, kommt in seinem ganzen Arbeitsleben auf 100.000 Arbeitsstunden. Ein Mensch kann in seinem Leben also vielleicht mit viel Anstrengung 100.000 Spiele spielen. AlphaZero hat demnach ungefähr so viele Spiele gespielt wie 440 Menschen, die ihr ganzes Arbeitsleben nur Schach spielen.

Im Vergleich zu Computern lernen Menschen also erstaunlich schnell. So beeindruckend es ist, dass AlphaZero Schach von null auf lernt – Menschen lernen anders. Sie bekommen durchaus Starthilfe von wohlmeinenden Mitmenschen. Ein guter Trainer erklärt Eröffnungen und Strategien und beginnt mit grundlegenden Übungen und Begriffen. Und hat man keinen Trainer, gibt es Lehrbücher, die die Übungen und das Wissen gut strukturieren. Selbst wenn viel Wissen beim Schach implizit ist, gibt es trotzdem auch viel explizites Wissen, das in Büchern steht. Man kann erstaunlich gut Schach lernen, indem man Bücher liest. Trotzdem muss man auch regelmäßig selbst spielen und so das Spiel üben. Spielen kann man gegen andere Spieler im Verein oder gegen einen Computer, der verschiedene Schwierigkeitseinstellungen hat. Üben ist aber nicht nur ein zufälliges Ausprobieren von möglichen Handlungen (so wie es beim verstärkenden Lernen passiert). Erfolgreiches Training ist nicht nur Versuch und Irrtum, sondern es ist eine überlegte und systematisch strukturierte Aktivität. Zum Training gehört auch, dass man Standardsituationen wiederholt, dass man bewusst an seinen Schwächen arbeitet und dass man viele Schachpartien analysiert. Insbesondere alte Partien des nächsten Gegners.

Da die Maschinen anders lernen als wir Menschen, ist auch ihr Verhalten nicht unbedingt menschlich. Aber weil die Maschinen lernen, uns zu imitieren, wird es manche Ähnlichkeiten geben. Trotzdem (und ich weiß, ich wiederhole mich) sollten wir den Maschinen nicht vorschnell menschliche Intelligenz zuschreiben. Ihre Intelligenz ist anders. Während der Partie, die AlphaGo gegen Lee Sedol gewann, beschrieb ein Kommentator einen der Züge so: »Das ist kein menschlicher Zug. Ich habe noch nie einen Menschen einen solchen Zug spielen sehen.«[13]

13 Der Kommentator war der europäische Go-Champion Fan Hui, der zuvor selber schon gegen AlphaGo verloren hatte (Metz, 2016).

Das Besondere am AlphaZero-Algorithmus ist, dass fast identische Computerprogramme selbständig verschiedene Aufgaben lernen können. Sehr ähnliche Programme können fast ohne menschliche Hilfe Backgammon, Go, Schach oder jedes andere Spiel lernen. Der menschliche Entwickler muss immer noch die Ein- und Ausgaben definieren, die spielspezifisch sind, aber er muss kein Experte mehr für das jeweilige Spiel sein. Am Ende kann ein Programm, das Go gelernt hat, nicht auch Schach spielen, weil die Spielbretter, -steine und -regeln andere sind. Für jedes Spiel muss man zwar immer noch ein eigenes Programm aufsetzen, aber diese Technologie erlaubt es, KI-Systeme für viele verschiedene Anwendungen relativ automatisch zu entwickeln. Das ist praktisch, aber noch lange keine Allgemeine Künstliche Intelligenz (AKI). Der Mensch mit seiner Intelligenz lernt nicht nur eine Aufgabe zu erledigen, sondern kann viele verschiedene Aufgaben bearbeiten. Oftmals sogar ohne viel Übung. Wahrscheinlich muss man sich für AKI noch mehr Tricks beim Menschen abschauen.[14] Unser Verhalten ist halt doch komplexer als das von Ratten in einem Labyrinth – und selbst das haben Wissenschaftlerinnen und Wissenschaftler noch nicht vollständig verstanden.

14 Siehe Lake, Ullman, Tenenbaum & Gershman (2017).