**Zur Diskussion der Vermittlung von Verfahren des Maschinellen Lernens im Informatikunterricht**

von Dr. Annika Eickhoff-Schachtebeck und Prof. Dr. Kerstin Strecker

Mai 2023

Ein Bild, das Text, ClipArt enthält.

Automatisch generierte Beschreibung

# **Inhaltsverzeichnis**

[Inhaltsverzeichnis 2](#_Toc143583881)

[Abbildungsverzeichnis 3](#_Toc143583882)

[1 Zusammenfassung 4](#_Toc143583883)

[2 Einleitung 4](#_Toc143583884)

[3 Beispiele konkreter Verfahren 6](#_Toc143583885)

[3. 1 k- nächste Nachbarn: didaktisiertes Verfahren und Lernziele 6](#_Toc143583886)

[3. 2 Entscheidungsbäume: didaktisiertes Verfahren und Lernziele 9](#_Toc143583887)

[3. 3 Neuronale Netze: didaktisiertes Verfahren und Lernziele 12](#_Toc143583888)

[4 Zusammenfassung der Lernziele konkreter Verfahren und Fazit 18](#_Toc143583889)

[Bildquellenverzeichnis 19](#_Toc143583890)

# Abbildungsverzeichnis

[Abbildung 1 - Mantelgröße für Blacky gesucht 8](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168782)

[Abbildung 2 - Trainingsdaten Eisverkauf 9](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168783)

[Abbildung 3 - Trainingsdaten Ernteentscheidung 10](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168784)

[Abbildung 4 - Erstellung eines Entscheidungsbaums 11](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168785)

[Abbildung 5 - Erstellung eines Entscheidungsbaums 11](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168786)

[Abbildung 6 - Erstellung eines Entscheidungsbaums 11](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168787)

[Abbildung 7 - Ausreißer in den Daten 12](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168788)

[Abbildung 8 - Erkunden der Funktionsweise eines Neurons 14](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168789)

[Abbildung 9 - Erkunden der Berechnung der Ausgabe des Neurons 14](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168790)

[Abbildung 10 - Wahl einer Schriftfarbe bei dunklem und hellem Hintergrund 15](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168791)

[Abbildung 11 - Scratch-Simulation zur automatischen Wahl einer Schriftfarbe 15](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168792)

[Abbildung 12 - Skript zum Ändern des Gewichts 16](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168793)

[Abbildung 13 - Lernalgorithmen 17](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168794)

[Abbildung 14 - Scratch-Simulation zur automatischen Fensteröffnung mit einem Neuron 17](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168795)

[Abbildung 15 - Scratch-Simulation zur automatischen Fensteröffnung mit einem neuronalen Netz 18](file:///K:\EKV\GSV\04_Praktikanten\Bartzke\Margraf\03_Landingpage_Informatik\Artikel_Autorinnen\Art_ML_im_Informatikunterricht_.docx#_Toc143168796)

Zur Diskussion der Vermittlung von Verfahren des Maschinellen Lernens im Informatikunterricht

Dr. Annika Eickhoff-Schachtebeck Mai 2023

Prof. Dr. Kerstin Strecker

# 1 Zusammenfassung

Viele Unterrichtsmaterialien, die zurzeit im Themenbereich „maschinelles Lernen“ für den Informatikunterricht entstehen, vermitteln konkrete Verfahren. Wir wollen mit diesem Artikel einen Beitrag liefern, der exemplarisch an einigen Verfahren des maschinellen Lernens aufzeigt, welche Lernziele mit den jeweiligen, didaktisch reduzierten, Verfahren im Unterricht verfolgt werden sollten. Gleichzeitig begründen wir unsere Auffassung, dass im Informatikunterricht nur einige wenige Verfahren des maschinellen Lernens thematisiert werden sollten und dass im Unterricht weniger das wiederholte Anwenden des zugrundeliegenden Konzepts dieser Verfahren notwendig ist als vielmehr die daraus abgeleitete informatisch begründete Reflexion.

# 2 Einleitung

Die didaktisch-methodische Aufbereitung von Konzepten des maschinellen Lernens für den Unterricht ist notwendig, weil diesen für Forschung und Lernen, Wirtschaft, sozialem Leben, Kultur und Politik eine immer stärkere Bedeutung zugesprochen wird. Kenntnisse, Fertigkeiten und Erfahrungen auf dem Gebiet des maschinellen Lernens müssen folglich vermittelt werden, um den späteren mündigen Bürgerinnen und Bürgern eine verständige Teilhabe am gesellschaftlichen und politischen Diskurs zu ermöglichen.

Es ist Aufgabe der Fachdidaktik, allgemeinbildende Konzepte des Themengebiets zu identifizieren, die bisher im Informatikunterricht keine oder zu wenig Beachtung finden und übertragbar sind in dem Sinn, dass sie längerfristig relevant bleiben und auch nach Beendigung der Schule Schülerinnen und Schülern ermöglichen, mit diesem informatischen Hintergrundwissen Chancen und Risiken der dann im Fokus stehenden Beispiele reflektieren zu können. Eines dieser Themen im Zusammenhang mit maschinellem Lernen ist sicher der Unterschied zwischen Korrelation und Kausalität, um ein Beispiel zu nennen.

Im Zusammenhang mit der Didaktisierung des Themengebiets „maschinelles Lernen“ ist die Vermittlung von konkreten Verfahren in diesem Bereich, wie z.B. Entscheidungsbäumen, natürlich ebenfalls sehr sinnvoll. Auch wir werden im Folgenden Beispiele dazu vorstellen. Es kann aber im Informatikunterricht nicht darum gehen, sämtliche vorhandenen Verfahren und Algorithmen des maschinellen Lernens rezeptartig anwenden zu können. Vielmehr müssen wir die Lernziele definieren, die hinter der Kenntnis der einzelnen Verfahren stehen. Ein Vergleich mit dem Unterrichtsthema „Sortieralgorithmen“ soll dies veranschaulichen. Man kann fünf verschiedene Sortieralgorithmen vermitteln oder 25. Doch was ist der Mehrwert dabei, noch einen weiteren Sortieralgorithmus zu thematisieren, wenn man bereits sortieren kann? Warum soll man im Bereich des maschinellen Lernens ein siebtes Verfahren kennenlernen, wenn man schon sechs Verfahren kennt? Und warum sechs und nicht fünf?

Stellen wir die Frage, warum man überhaupt mehrere Sortieralgorithmen thematisiert. Dahinter steht meist das Lernziel, dass dasselbe Problem („Sortieren“) auf unterschiedliche Weise gelöst werden kann und Algorithmen nie alternativlos sind. Oder aber es steht das Lernziel dahinter, Algorithmen hinsichtlich ihrer Komplexität zu bewerten und den Unterschied zwischen Quicksort und BubbleSort experimentell oder analytisch zu erfassen. Dann ergibt sich aus den Lernzielen die Begründung, mehrere, aber nicht beliebige oder beliebig viele Sortieralgorithmen zu behandeln. Wenn wir mehrere Verfahren im Bereich des Maschinellen Lernens thematisieren, dann müssen wir zunächst ein Äquivalent zu den dahinterstehenden Lernzielen analog dem Bereich der Sortieralgorithmen finden, um zeigen zu können, warum wir das Lernziel mit einem konkreten Verfahren besser erreichen als mit einem anderen. Was sind dann allgemeinbildende Lernziele bei der Vermittlung von Verfahren des Maschinellen Lernens? Hier wird sicherlich eine Entmystifizierung des Begriffs der „künstlichen Intelligenz“ angestrebt und eine Auseinandersetzung mit der Frage, was eigentlich „Lernen“ im Kontext des maschinellen Lernens bedeutet. Ein weiteres Lernziel wird die Unterscheidung zwischen regelbasierten und datenbasierten Systemen sein sowie das Wissen um die Notwendigkeit und Bedeutung von Trainingsdaten. Möglicherweise soll auch das Problem der Überanpassung oder der Unterschied zwischen Ausreißer und Mustern in Daten verdeutlicht werden. Auch eine vertiefte Auseinandersetzung mit Daten wie beispielsweise die Notwendigkeit der Aufbereitung von Daten oder ein Wissen darum, dass Entscheidungen von Verfahren des maschinellen Lernens immer auf Daten aus der Vergangenheit beruhen, kann Ziel im Informatikunterricht sein.

Eine weitere Frage, die wir stellen, lautet: wo liegt der Schwerpunkt bei der Thematisierung eines bestimmten Verfahrens des maschinellen Lernens im Informatikunterricht? Übertragen auf das Unterrichtsthema „Sortieralgorithmen“ entspricht dies der Frage: welchen Mehrwert hat es, BubbleSort auf immer wieder neue unsortierte Zahlenfolgen anzuwenden? Die Antwort lautet, es gibt keinen Mehrwert, ebenso wenig, wie immer wieder neue Entscheidungsbäume zu zeichnen. Sicher muss man ein Verfahren auch anwenden können. Dann aber, um die intendierten Lernziele zu erreichen oder weil daraus weitere Lernziele folgen, beim Sortieren beispielsweise, um die lokale, beschränkte Sicht eines Algorithmus auf zwei Zahlen der globalen Sicht auf die gesamte Zahlenfolge beim Menschen gegenüberstellen zu können.

Unsere letzte Frage lautet: Welche weiteren Faktoren sollten bei der Auswahl von Verfahren des Maschinellen Lernens für den Informatikunterricht eine Rolle spielen? Auch hier ist ein Vergleich mit dem Unterrichtsthema „Sortieralgorithmen“ möglich: Wenn die Schülerinnen und Schüler selbst den Wunsch nach einer Sortierung haben (z.B. in einer highscore-Liste), dann sind sie selbst in der Lage, einen Sortieralgorithmus zu erfinden. Haben wir uns nicht in den vergangenen Jahren in der Informatikdidaktik erarbeitet, wie wichtig entdeckendes, handlungsorientiertes Lernen, offene Aufgaben und Kreativität für die Kompetenz des algorithmischen Problemlösens sind? Diese Zeit sollten wir nicht unnötig einschränken, indem wir uns zu viel mit vorgegebenen, konkreten Verfahren beschäftigen. Im Bereich des maschinellen Lernens geht es jedoch im Bereich der Thematisierung eines Verfahrens oft eher darum, ein ganz konkretes Verfahren zu vermitteln, äquivalent zu beispielsweise Quicksort. Quicksort erfindet sicher kein Lernender spontan selbst. Die Konsequenz für die Auswahl eines Verfahrens und die Umsetzung im Unterricht ist eine methodische: trotz der Vermittlung konkreter Verfahren des maschinellen Lernens entdeckendes Lernen zu ermöglichen. Für die gewählten Verfahren müssen daher auch verschiedene Repräsentationsebenen und Zugänge im Sinne der Differenzierung für heterogene Lerngruppen erarbeitet und verschiedene Altersstufen mit ihrem unterschiedlichen Vorwissen angesprochen werden.

Wir hoffen, für die folgende Auswahl konkreter Verfahren des überwachten Lernens die Frage nach den dahinterstehenden Lernzielen beantworten zu können. Gleichzeitig versuchen wir die Frage zu beantworten, in welchem Umfang das konkrete Verfahren dazu didaktisch reduziert werden muss und welche methodische Vorgehensweise sich eignet.

# 3 Beispiele konkreter Verfahren

## 3. 1 k- nächste Nachbarn: didaktisiertes Verfahren und Lernziele

Wir beginnen mit dem Verfahren k-nächste Nachbarn. In dem Lernszenario aus (Brandt, Eickhoff-Schachtebeck & Strecker 2022a) geht es um die Suche der passenden Mantelgröße für den eigenen Hund (Datenpunkt B in Abbildung 1). Die Käufe zufriedener Kunden (Trainingsdaten) sind in einer Grafik abgebildet. Ohne Kenntnis des Verfahrens k-nächste Nachbarn sollen die Lernenden begründen, für welche Mantelgröße sie sich entscheiden würden (bzw. die Besitzer der Hunde F(ifi), R(ocky) und N(ala)). Ein Hund wird über einen Datenpunkt mit den Merkmalen Rückenlänge (x-Achse) und Schulterhöhe (y-Achse) abgebildet. Die zugehörige Mantelgröße wird über die Farbe des Datenpunktes bestimmt (Abbildung 1). Die Beschränkung auf drei Merkmale ermöglicht die Darstellung der Daten in einem zweidimensionalen Koordinatensystem und so einen enaktiven Zugang. Diese Didaktisierung ergibt sich also aufgrund der gewählten Methodik.

Es wird gehofft, dass sich eine Strategie unter den Ideen der Lernenden findet, die dem Verfahren k-nächste Nachbarn entspricht, nämlich sich für die Mantelgröße zu entscheiden, welche der Mehrheit der z.B. fünf nächsten Datenpunkte entspricht. In diesem Fall würde die Wahl für den eigenen Hund Blacky auf S fallen (Fifi: S, Rocky: M und Nala: L).

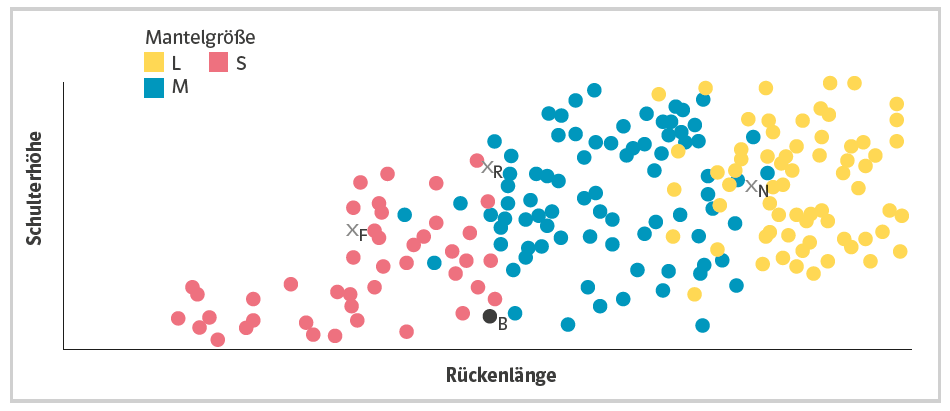


Abbildung 1 - Mantelgröße für Blacky gesucht

So entdecken die Lernenden selbst das Grundprinzip des Verfahrens k-nächste Nachbarn. Gleichzeitig kann man aber auch über das Verfahren an sich reflektieren:

* Vielleicht ist Größe M für Blacky besser geeignet. Das Gewicht des Hundes wurde ja gar nicht berücksichtigt und könnte bei der Wahl der Mantelgröße ebenfalls eine Rolle spielen. Dahinter steht die Erkenntnis, dass die Daten im Beispiel auf drei Merkmale reduziert wurden (Rückenlänge des Hundes, Schulterhöhe des Hundes und die gewählte Mantelgröße). Die Frage ist, ob durch die Reduktion nicht relevante Merkmale unberücksichtigt geblieben sind, bzw. wie man überhaupt relevante Merkmale identifiziert.
* Durch die Reduktion auf zwei Merkmale des Hundes sind die Trainingsdaten vielleicht ungeeignet oder ungenügend.
* Bereits klassifizierte Trainingsdaten müssen in ausreichender Zahl vorliegen.
* Um den nächsten Nachbarn zu bestimmen, wurde der euklidische Abstand gewählt. Das ist aber nur möglich, wenn auf den Achsen Zahlenwerte vorliegen. Mindestens muss es eine Ordnung auf den Datenwerten geben. Auch müssen die Einheiten an den Achsen stimmig zueinander sein, bzw. die Daten normiert vorliegen. Was wäre beim Merkmal „Beruf“ der Abstand zwischen einem Feuerwehrmann und einer Polizistin?
* Welche Zahl wählt man für k? Würden wir k=3 wählen, würden wir für Nala im obigen Beispiel die Mantelgröße M vorhersagen, im Fall k=5 die Größe L. Dahinter steht eigentlich die Frage: Wann kann man von Mustern in Daten sprechen und was sind Ausreißer? Informatisch gesehen bedeutet eine zu große Wahl von k eine Unteranpassung.
* Statt Regeln anzugeben wie: Wenn die Rückenlänge im Bereich a bis b und die Schulterhöhe im Bereich c bis d, dann wähle Mantelgröße x, haben wir es hier nicht mit einer regelbasierten Problemlösestrategie zu tun. Basierend auf ausreichend vielen Trainingsdaten, die klassifiziert sind, treffen wir mit der Methode k-nächste Nachbarn eine Entscheidung für neue, ähnliche Daten.

Viele dieser Reflexionen können auch mit anderen Verfahren des maschinellen Lernens sehr gut thematisiert werden. Ganz besonders eindrucksvoll wird, unserer Meinung nach, aber am Verfahren k-nächste Nachbarn, dass Daten normiert werden müssen und zueinander stimmig sein, bzw. dieselben Einheiten verwenden müssen.

Dazu gehört auch, dass Zahlenwerte gefunden werden müssen, mit denen die Merkmale der Daten beschrieben werden können. Und das ist bei manchen relevanten Merkmalen von Daten gar nicht möglich, wie Beruf, Freundschaft, Eleganz, …

In (Brandt, Eickhoff-Schachtebeck & Strecker 2022a) findet sich noch ein zweites Beispiel. Basierend auf Erfahrungen wurden für die Vorhersage des Eisverkaufs die Trainingsdaten aus Abbildung 2 protokolliert:

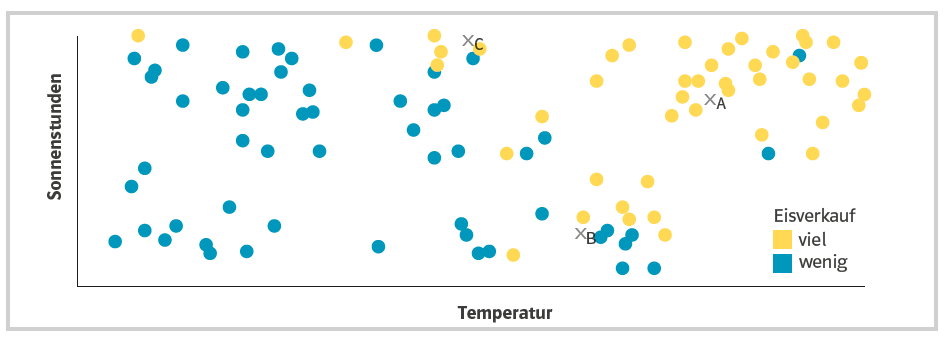


Abbildung 2 - Trainingsdaten Eisverkauf

Für die Wetterprognosen A, B und C soll mithilfe des Verfahrens 3-nächste Nachbarn der zu erwartende Eisverkauf klassifiziert werden. Doch wieso kann eigentlich hier der Abstand bestimmt werden, wenn Sonnenstunden und Temperatur ganz andere Einheiten haben? Einfach weil es funktioniert und die Muster der Trainingsdaten den Schluss nahelegen, dass das Verfahren k-nächste Nachbarn anwendbar ist. Das zeigt das eher „handwerkliche“, auf Erfahrungen beruhende Vorgehen im Bereich Data Science.

Ein weiteres Kriterium das Verfahren k-nächste Nachbarn zu thematisieren, unabhängig von den dahinterstehenden Lernzielen, eher auf methodischer Ebene, ist die Einfachheit des Verfahrens. Wie oben gezeigt, ist es in geeigneten Lernszenarien möglich, dass das Verfahren von den Lernenden selbst entdeckt wird.

## 3. 2 Entscheidungsbäume: didaktisiertes Verfahren und Lernziele

In dem Lernszenario zu den Abbildungen 3-6 aus (Brandt, Eickhoff-Schachtebeck & Strecker 2022b) geht es darum, dass eine Software zukünftig automatisiert entscheiden soll, ob geernet oder noch gewartet werden soll. Dazu sind die Entscheidungen eines erfahrenen Landwirts in der Vergangenheit protokolliert worden. Aus diesen Trainingsdaten erstellt eine Software automatisch einen Entscheidungsbaum, der dann für zukünftige Entscheidungen genutzt wird. Die Daten beinhalten drei Merkmale: die aktuelle Regenwahrscheinlichkeit, die aktuelle Feuchte des Getreides und die Entscheidung des Landwirts zur Ernte. Weizen ist unter einer Feuchte von 14,5% trocken und erntereif. Droht Regen, ist es aber manchmal besser nicht ganz trockenes Getreide einzufahren (auch wenn dies beim Verkauf Abzüge bedeuten könnte), als noch größere Einbußen durch den eventuellen Regen in Kauf zu nehmen. Die Trainingsdaten sind in Abbildung 3 gegeben. Abbildungen 4 bis 6 zeigen die Erstellung eines Entscheidungsbaums per Hand, obwohl die Konstruktion des Baums eigentlich von einer Software automatisiert durchgeführt wird. Die Beschränkung auf drei Merkmale ermöglicht die Darstellung der Daten in einem zweidimensionalen Koordinatensystem und so einen enaktiven Zugang, bei dem das Verfahren der rekursiven Partitionierung angewendet werden kann. Diese Didaktisierung ergibt sich also aufgrund der gewählten Methodik.

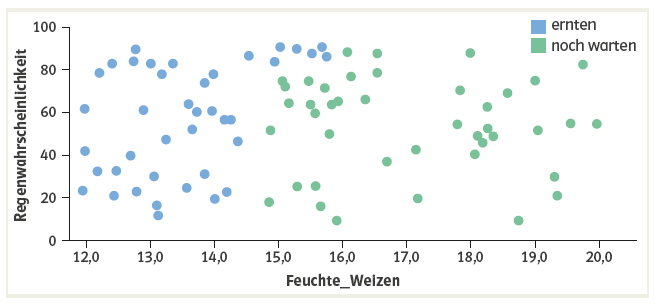


Abbildung 3 - Trainingsdaten Ernteentscheidung



Abbildung 4 - Erstellung eines Entscheidungsbaums

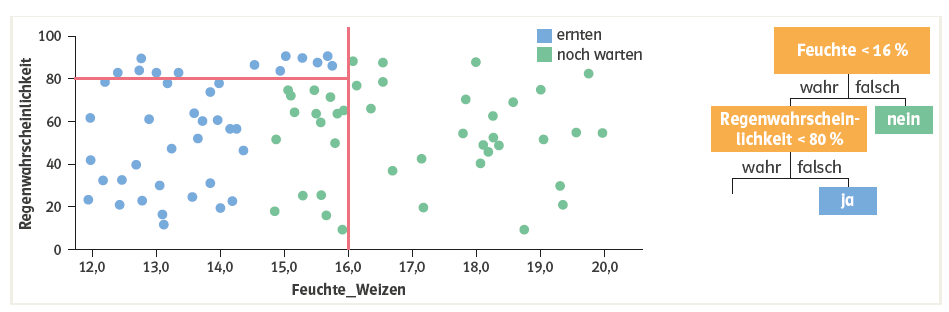


Abbildung 5 - Erstellung eines Entscheidungsbaums

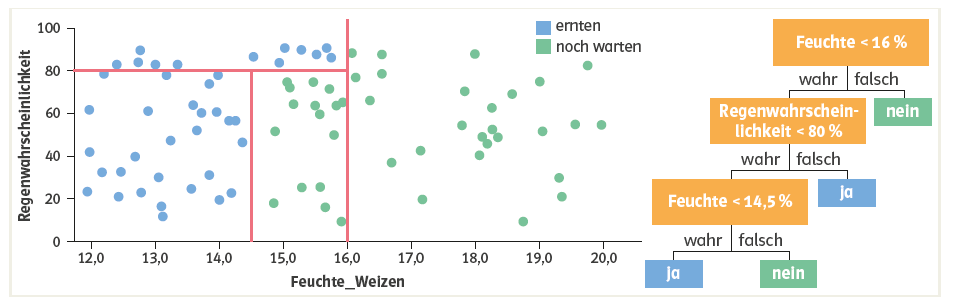


Abbildung 6 - Erstellung eines Entscheidungsbaums

Basierend auf den Trainingsdaten, die das Erfahrungswissen des Landwirts abbilden, werden nach und nach vertikale oder horizontale Bereiche ausgewählt, die Datenpunkte gleicher Farbe vereinen. Mithilfe des Entscheidungsbaums würde für die, nicht in den Trainingsdaten abgebildete, Situation einer Feuchte des Weizens von 15% und einer Regenwahrscheinlichkeit von 40% die Entscheidung „noch warten“ getroffen.

Nun gilt es die Frage zu beantworten, was denn an diesem Verfahren besser gelernt werden kann als am Verfahren k-nächste Nachbarn, bzw. warum noch ein weiteres Verfahren hier sinnvoll ist:

* Die Bedeutung der Trainingsdaten kann hier ebenso gut gezeigt werden, wie beim Verfahren k-nächste Nachbarn
* Auch die Beschränkung auf wenige Merkmale und sich daraus ergebene Probleme kann hier gut thematisiert werden, ebenso aber auch beim Verfahren k-nächste Nachbarn.
* Entscheidungsbäume benötigen weniger eine Vereinheitlichung an den Achsen, da die Merkmale nicht in dem Sinne miteinander kombiniert werden wie beim euklidischen Abstand.

Darüber hinausgehende Lernziele, die mit dem Verfahren k-nächste Nachbarn nicht so unmittelbar thematisiert werden können, werden deutlich, wenn man sich überlegt, was passieren würde, wenn es einen Ausreißer in den Daten gibt, bzw. (um bei unserer Geschichte zu bleiben) wenn der Mähdrescher an einem Tag nicht funktioniert hat und diese Entscheidung ebenfalls in die Trainingsdaten mit aufgenommen wurde (Abbildung 7).

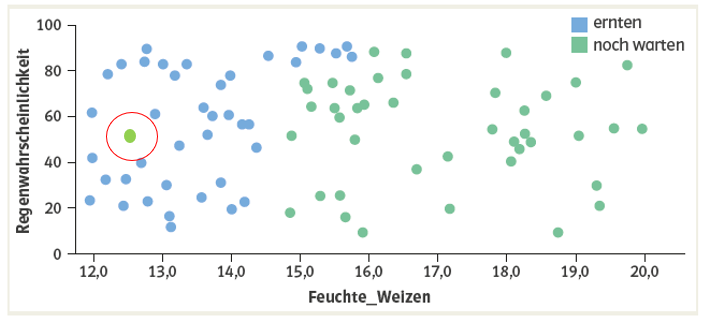


Abbildung 7 - Ausreißer in den Daten

* Der Entscheidungsbaum würde in dem Szenario in Abbildung 7 unnötig tief werden. Wir sprechen von Überanpassung oder overfitting. Dieser Themenkomplex kann bei Entscheidungsbäumen besonders gut thematisiert werden. Lernende könnten den Algorithmus daraufhin dahingehend anpassen, dass eine Toleranz eingebaut wird, z.B. es werden horizontale oder vertikale Linien eingezeichnet, die Daten gleicher Farbe vereinen, wobei z.B. 95% der Daten dieselbe Farbe haben müssen.
* Das Verfahren k-nächste Nachbarn ist von sich aus bereits tolerant gegenüber Ausreißern in den Daten. Das Verfahren der rekursiven Partitionierung in Reinform würde Ausreißer in den Daten als Muster interpretieren. Die Frage ist, was sind Ausreißer und was Muster in den Daten und: kann man das überhaupt zweifelsfrei entscheiden?
* Dadurch wird sofort deutlich, dass die Vorhersage im individuellen Einzelfall falsch sein kann.
* Zunächst erscheint das Verfahren regelbasiert, da man einen Entscheidungsbaum ja auch mit Hilfe von vielen geschachtelten Bedingungen oder einer Datenstruktur „Baum“ sehr gut angeben kann. Im Bereich Algorithmik ist dies vielleicht sogar im Unterricht bereits geschehen. Deshalb ist es hier von entscheidender Bedeutung, dass den Lernenden transparent wird, dass die Software den Entscheidungsbaum automatisiert generiert, und zwar auf Basis von Trainingsdaten und nicht der Programmierer oder die Programmiererin. Thematisiert man dies, wird der Unterschied zwischen einer regelbasierten und einer datenbasierten Strategie greifbar.
* Weiterhin wird bei diesem Verfahren sehr deutlich, dass die Ergebnisse und Vorhersagen des Verfahrens aus Daten der Vergangenheit berechnet werden, vergleichbar mit dem Modellierungskreislauf der Mathematik. Wird in einem nassen Jahr z.B. Weizen ab einer Feuchte von 15% toleriert, dann liefert das Verfahren ungeeignete Ergebnisse.

## 3. 3 Neuronale Netze: didaktisiertes Verfahren und Lernziele

Das Themengebiet maschinelles Lernen ohne neuronale Netze zu thematisieren, erscheint in der Übermächtigkeit des Verfahrens in der Realität kaum denkbar. Da ein einzelnes Neuron nicht so mächtig ist wie ein neuronales Netz, sollte mindestens die Funktionalität eines kleinen Netzes thematisiert und verstanden werden. Wir skizzieren an dieser Stelle das virtuell-enaktive Entdecken der Funktionsweise neuronaler Netze aus (Brandt, Eickhoff-Schachtebeck & Strecker 2022b) und verweisen zur näheren Erläuterung der hier verwendeten didaktischen Reduktion neuronaler Netze auf Strecker (2020 und 2021).

In all unseren folgenden Beispielen verwenden wir die Methode des virtuell-enaktiven Erkundens. Wir präsentieren jeweils eine interaktive Simulation einzelner Neuronen oder kleiner neuronaler Netze, in der die Lernenden aktiv Eingaben verändern und/oder Parameter einstellen können. Durch Beobachtung der Ergebnisse und Reaktionen kann sich so zumindest in Teilen die Funktionsweise erschlossen werden. Gleichzeitig sind diese interaktiven Simulationen in Scratch implementiert. Dadurch kann auch der Programmcode von den Lernenden nachvollzogen werden, der in den Beispielen sehr einfach und wenig umfangreich ist. Auch dies hilft, sich die Funktionsweise vollständig zu erschließen. Durch die Bereitstellung einer Implementierung wird gehofft, dass bei den Lernenden zumindest das Gefühl entsteht, einer Implementierung grundsätzlich „gewachsen“ zu sein. Mindestens wird aber deutlich, dass zur Implementierung neuronaler Netze nur algorithmische Grundbausteine von Scratch verwendet werden und keine „Magie“. Das Verfahren ist also algorithmisch beschreibbar.

An einem einzelnen Neuron kann zunächst gut thematisiert werden, wie sich die Ausgabe aus Eingaben, Gewichten und Schwellenwert berechnet, wenn wir folgende Didaktisierung zugrunde legen: Das Neuron feuert (Ausgabe dabei wird von dem Programmierer oder der Programmiererin festgelegt), wenn die Summe der Produkte aus Eingabe und dem zugehörigen Kantengewicht einen Schwellenwert erreicht bzw. überschreitet. Sonst feuert das Neuron nicht (Ausgabe dabei wird von dem Programmierer oder der Programmiererin festgelegt.).

Im ersten Beispiel eines Neurons geht es um den Kontext, dass eine Gärtnerei ihre Freilandpflanzen automatisch bewässern möchte. Es gibt drei verschiedene Gärtnertypen:

Die erste Gärtnerin hat folgende Anforderung: Es soll immer dann bewässert werden, wenn es sehr unwahrscheinlich ist, dass es regnen wird. Egal wie trocken der Boden ist. Die neu gepflanzten Pflanzen brauchen viel Wasser.

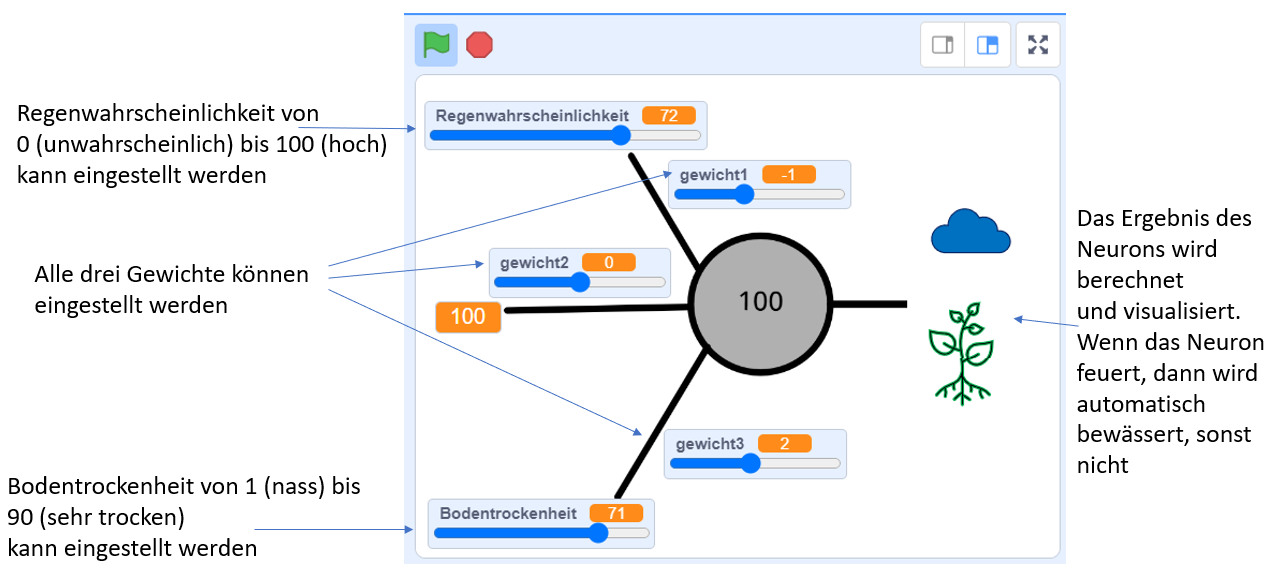


Abbildung 8 - Erkunden der Funktionsweise eines Neurons

Der zweite Gärtner wünscht sich von der automatischen Bewässerung folgendes: Da Wasser gespart werden soll, soll immer nur dann bewässert werden, wenn es sehr unwahrscheinlich ist, dass es regnen wird und gleichzeitig der Boden sehr trocken ist. Und der dritte Gärtner hat diese Wünsche: Es soll immer dann bewässert werden, wenn der Boden sehr trocken ist. Auf die Wettervorhersage verlässt er sich nicht.

Mit folgender Simulation in Scratch kann durch Experimente und dem Erkunden des Programmcodes zur Berechnung der Ausgabe die Funktionsweise eines Neurons nachvollzogen werden. Gleichzeitig wird deutlich, dass die Funktionsweise des Neurons von der Wahl der Gewichte abhängt, denn bei unterschiedlicher Wahl der Gewichte reagiert die Bewässerungsanlage auf dieselben Eingaben jeweils verschieden. Die Gewichte werden so gewählt, dass die Funktionalität den jeweiligen Wünschen der Gärtner:innen entspricht.



Abbildung 9 - Erkunden der Berechnung der Ausgabe des Neurons

In einem zweiten Schritt muss transparent gemacht werden, dass das Einstellen der Gewichte nicht händisch passiert, sondern die Gewichte automatisch in einer Trainingsphase angepasst werden. Dafür ist ein Fehlersignal eines Trainers oder einer Trainerin notwendig, wenn die Gewichte noch nicht optimal eingestellt sind.

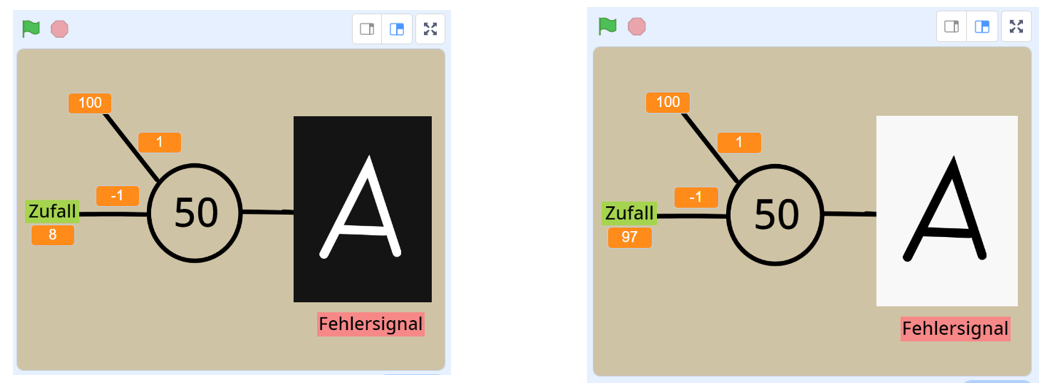


Abbildung 10 - Wahl einer Schriftfarbe bei dunklem und hellem Hintergrund

Unser nächstes Beispiel hat folgenden Kontext: Für dunklen Hintergrund eignet sich helle Schriftfarbe, für hellen Hintergrund dunkle Schriftfarbe. Bei Grautönen dazwischen können die Empfindungen individuell unterschiedlich sein.

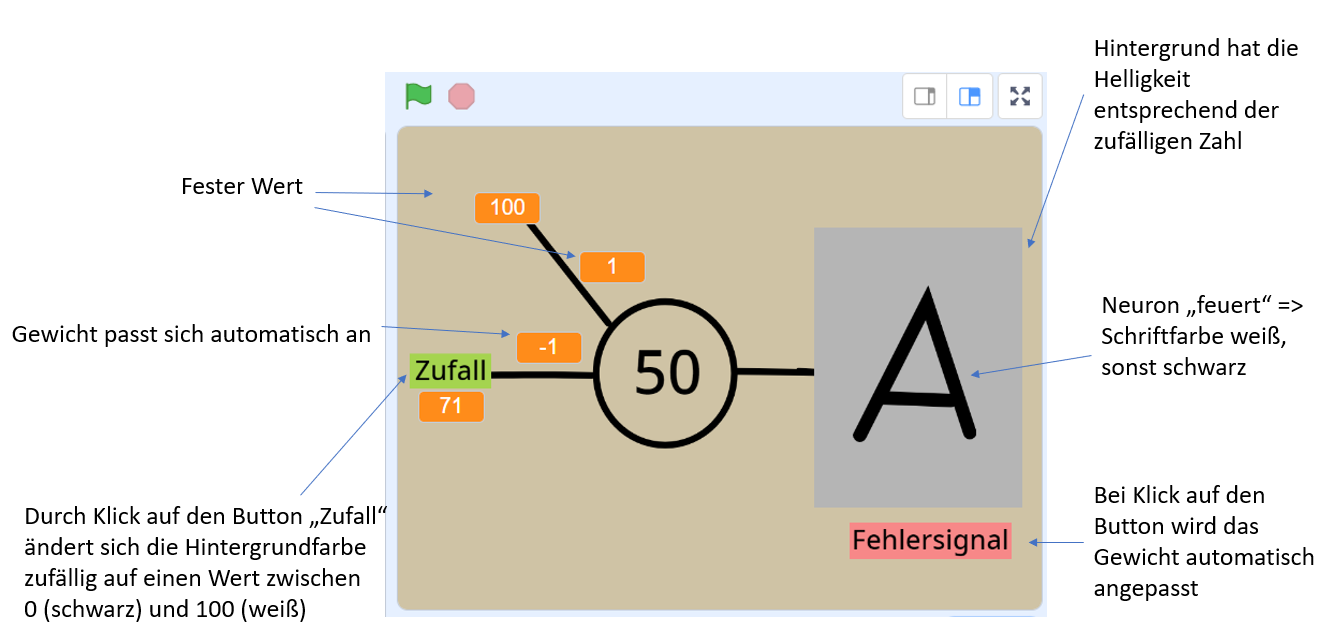


Abbildung 11 - Scratch-Simulation zur automatischen Wahl einer Schriftfarbe

Auch hierzu gibt es eine Simulation:

Die Änderung des Gewichts beläuft sich dabei auf folgenden Lernalgorithmus und findet sich im Skript zum Button „Fehlersignal“:

In den weiteren Beispielen reduzieren wir uns auf die binären Eingaben 1 und -1. Wir verwenden für die folgenden Beispiele zwei Eingaben mit jeweils den beiden möglichen genannten Werten. Weiterhin verwenden wir ein neuronales Netz, für das für alle möglichen Szenarien passende Gewichte gefunden werden können und das dennoch in dem Sinn reduziert ist, dass die letzte Schicht feste Kantengewichte verwendet, um keinen backpropagation-Algorithmus anwenden zu müssen (siehe (Strecker 2020), (Strecker 2021)).



Abbildung 12 - Skript zum Ändern des Gewichts

Für unsere nächsten beiden Beispiel verwenden wir folgenden Kontext: Es geht um eine automatische Fensteröffnung im Smart-Home. Es gibt zwei Sensoren: ein Sensor liefert einen Wert, der angibt, ob es draußen laut (S=1) oder leise (S=-1) ist und ein Sensor liefert einen Wert, der angibt, ob es draußen hell (L=1) oder dunkel (L=-1) ist. Das neuronale Netz, bzw. ein Neuron liefert den Wert 1, wenn es feuert und sonst den Wert -1. Die Ausgabe des Netzes bewirkt ein automatisches Öffnen des Fensters (wenn das Neuron der letzten Schicht feuert) oder ein Schließen des Fensters (wenn das Neuron der letzten Schicht nicht feuert). Die Begründung für die Implementierung mit einem Verfahren des maschinellen Lernens, entgegen der mit individuellen regelbasierten Algorithmen für jeden der Bewohner:innen, ist folgende: Vier Bewohner des Hauses wollen die automatische Fensteröffnung in ihrem Schlafzimmer installieren, aber jeder der Bewohner:innen hat andere Wünsche, unter welchen Bedingungen sich das Fenster öffnen oder schließen soll. Implementiert man das System mit einem neuronalen Netz, kann jeder der Bewohner:innen es den eigenen Wünschen entsprechend trainieren und so liefert derselbe Algorithmus (durch jeweils unterschiedliche Gewichte nach der Trainingsphase) bei jedem/jeder Bewohner/in andere Ergebnisse. Funktioniert die automatische Fensteröffnung noch nicht wie gewünscht, wird das Fenster „manuell“ mit Klick auf das Fenster in der Simulation geöffnet oder geschlossen. Diese „manuelle“ Betätigung des Fensters ist das Fehlersignal, woraus geschlossen werden kann, wie die Gewichte verringert oder erhöht werden müssen. Die Wünsche der Bewohner:innen lauten:

Onkel Andreas: „Ich arbeite Nachtschicht. Mein Fenster soll nachts zu sein, damit keine Einbrecher einsteigen. Tagsüber schlafe ich. Da soll das Fenster nur offen sein, wenn es draußen leise ist.“ (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Oma Emmi: „Mein Fenster soll nachts offen sein, wenn es leise draußen ist, damit ich gut schlafen kann. Bei Lautstärke kann ich nicht schlafen. Tagsüber soll das Fenster offen sein, wenn die Enkel im Garten spielen und es laut ist, damit ich sie hören kann. Ist es tagsüber leise, soll das Fenster zu bleiben.“ (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Lin: „Ich kann nur schlafen, wenn das Fenster nachts offen und es leise ist. Bei Lärm soll das Fenster nachts zu bleiben. Tagsüber soll das Fenster zum Lüften aber immer offen stehen, egal wie laut oder leise es ist. (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Papa Erik: „Ich bin tagsüber immer bei der Arbeit und daher soll das Fenster tagsüber zu sein. Nachts soll es aber zum Schlafen offen sein, egal wie laut oder leise es ist.“ (Brandt, Eickhoff-Schachtebeck & Strecker 2022b)

Wir verweisen an dieser Stelle darauf, dass die Anforderungen von Oma Emmi einem NOT-XOR entsprechen und damit nur mit einem neuronalen Netz implementiert werden können, nicht mit einem einzelnen Neuron.

Für ein einzelnes Neuron verwenden wir zur Anpassung der Gewichte die folgenden Lernalgorithmen, mit einer Lernrate von 0,5. (Eine Erläuterung, wieso diese funktionieren, findet sich in (Strecker 2020, Strecker 2021)).



Abbildung 13 - Lernalgorithmen

Die Simulation stellt sich den Lernenden wie folgt dar:

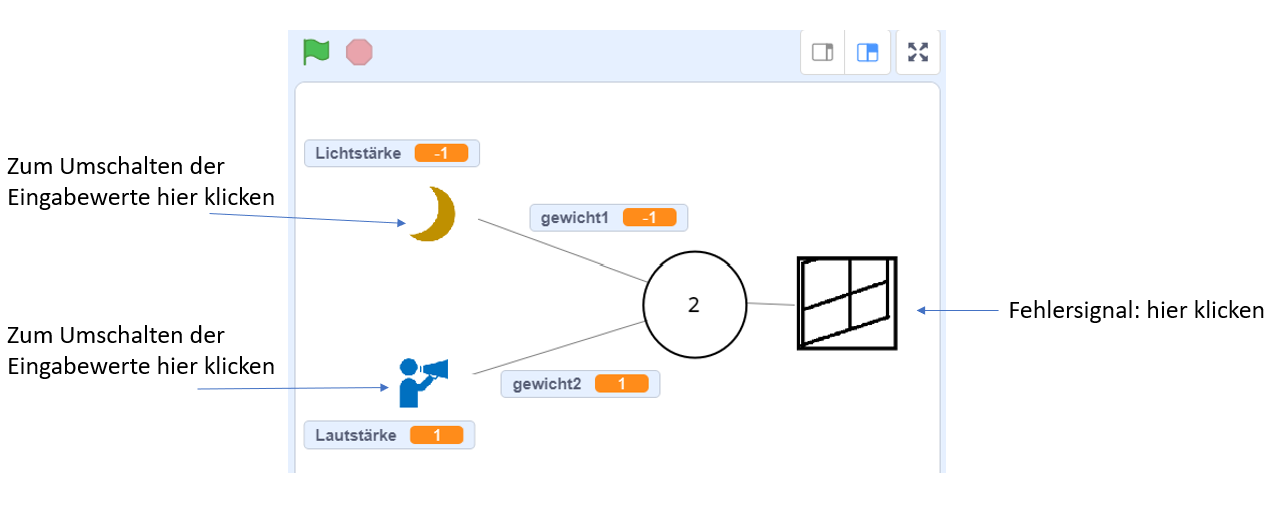


Abbildung 14 - Scratch-Simulation zur automatischen Fensteröffnung mit einem Neuron

Da ein einzelnes Neuron nicht mächtig genug ist, um z.B. das Fenster so zu steuern, wie Oma Emmi sich das wünscht, muss ein neuronales Netz verwendet werden. Die Simulation stellt sich den Schülerinnen und Schülern wie folgt dar:

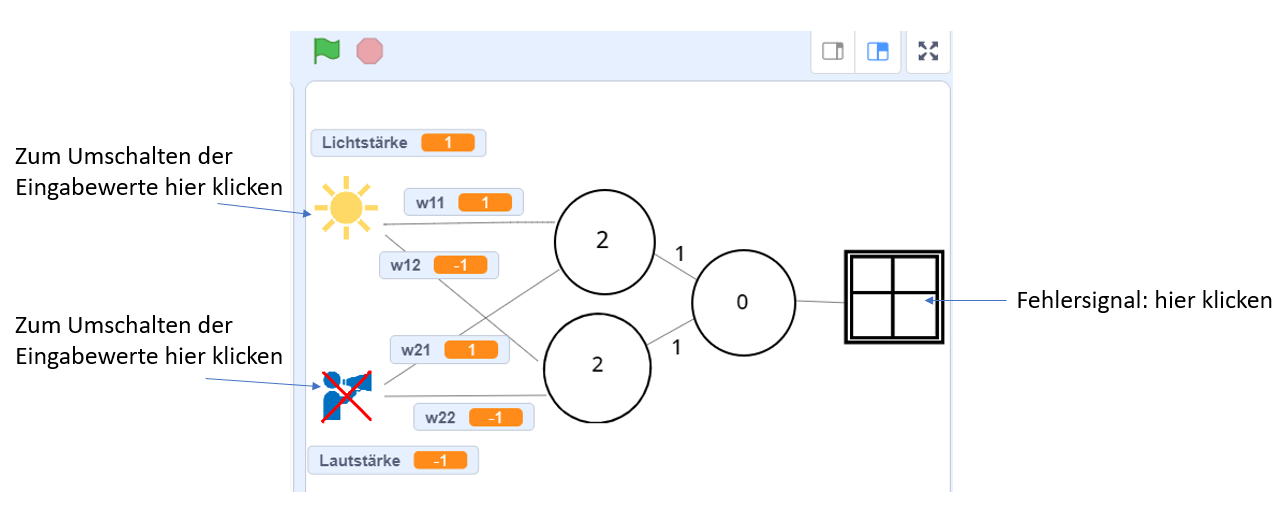


Abbildung 15 - Scratch-Simulation zur automatischen Fensteröffnung mit einem neuronalen Netz

Durch die festen Gewichte in der letzten Schicht kann mit einfachen Mitteln nachvollzogen werden, welche Gewichte bei einem Fehlersignal wie verändert werden müssen. Gleichzeitig können für alle denkbaren Szenarien und Wünsche der Bewohner:innen passende Gewichte automatisch „gelernt“ oder eingestellt werden. Die theoretischen Hintergründe werden in Strecker (2020 und 2021) beschrieben.

Das Erkunden der Funktionsweise eines didaktisch reduzierten neuronalen Netzes ermöglicht folgende Reflexion und hat dementsprechend folgende Lernziele:

* Die Trainingsphase wird hier sehr gut greifbar. Während bei den anderen Verfahren die Trainingsdaten bereits vorlagen, wird das Netz hier interaktiv trainiert, bzw. die Parameter während der Laufzeit angepasst.

Während bei den anderen beiden Verfahren die Trainingsphase und die Anwendungsphase deutlich getrennt waren, sieht man hier, dass Trainings- und Anwendungsphase theoretisch auch miteinander verzahnt ablaufen können.

* Man sieht weiterhin sehr gut, dass die Trainingsdaten die Funktionalität bestimmen. Da die Gewichte aufgrund der Trainingsdaten angepasst werden und die Wahl der Gewichte die Funktionalität des Netzes bestimmt, bedeutet das in der Konsequenz folgendes: wenn in der Trainingsphase ungenügende oder ungeeignete Daten verwendet werden, ist die Vorhersage des Netzes für ähnliche Daten schlecht.
* Es wird deutlich, dass das „Lernen“ eines Neurons dem automatischen Anpassen der Gewichte entspricht.
* Es wird ebenfalls deutlich, dass neuronale Netze nur diejenigen der Aufgaben sehr gut lösen können, für die sie trainiert wurden.
* Ein virtuell-enaktives Erkunden des Verfahrens neuronale Netze zeigt deutlich, dass das Verfahren algorithmisch beschreibbar ist, also nur algorithmische Grundbausteine verwendet werden und keine „Magie“.
* Skaliert man das neuronale Netz und spricht von 100 Milliarden Neuronen, dann wird deutlich, dass das Ergebnis eines Netzes für den Menschen nicht nachzurechnen ist. Es ist also nicht nachvollziehbar, wie das Netz zu seinem Ergebnis kommt. Auch das kann zu ethischen Dilemmata führen.

# 4 Zusammenfassung der Lernziele konkreter Verfahren und Fazit

In der Einleitung haben wir gefordert, dass sich die Auswahl konkreter Verfahren daraus bestimmt, welche dahinterstehenden allgemeinbildenden Lernziele abgedeckt werden. Wir fassen diese Lernziele zusammen und nennen dahinter die Verfahren, mit denen die Lernziele unserer Meinung nach besonders gut erreicht werden können:

* Entmystifizierung des Begriffs der künstlichen Intelligenz (alle, insbesondere neuronale Netze)
* „Was heißt Lernen?“ (neuronale Netze)
* Entscheidung beruht auf der Vergangenheit (Vergleichbar mit Modellierungskreislauf Mathematik) (Entscheidungsbäume)
* Unterschied Ausreißer und Muster (Entscheidungsbäume im Vergleich zu k-nächste Nachbarn)
* Ergebnisse können im individuellen Einzelfall falsch sein (k-nächste Nachbarn, Entscheidungsbäume)
* Unterscheidung regelbasierte und datenbasierte Systeme (k-nächste Nachbarn)
* Überanpassung (Entscheidungsbäume)
* Unteranpassung (k-nächste Nachbarn)
* Notwendigkeit von Trainingsdaten (alle)
* Trainingsdaten bestimmen die Ergebnisse (alle, insbesondere neuronale Netze)
* Notwendigkeit der Datenaufbereitung (Normalisierung, Ordnung auf Daten, Daten müssen zueinander passen) (k-nächste Nachbarn)
* Geeignete Merkmale der Daten identifizieren (alle)

In der Hoffnung mit dieser Aufzählung begründet zu haben, warum wir die genannten Verfahren des überwachten Lernens ausgewählt haben und in Ermangelung weiterer allgemeinbildender Lernziele im Bereich konkreter Verfahren, die mit diesen drei Verfahren nicht geeignet thematisiert werden können, wollen wir noch einen zweiten Punkt ins Auge fassen. Im Unterricht muss sehr deutlich werden, dass es nicht primär darum geht, für wiederholte Anwendungsfälle Entscheidungsbäume zu erstellen oder das Verfahren k-nächste Nachbarn anzuwenden. Sind die intendierten Lernziele bei der Vermittlung eines Verfahrens erreicht, können die erworbenen Kompetenzen zur Reflexion aktueller Systeme basierend auf informatischen Hintergrundwissen eingesetzt werden. Die fachlich begründete Reflexion mit informatischem Hintergrundwissen ist damit das entscheidende Lernziel, wenn es darum gehen soll, dass die Lernenden als mündige Bürgerinnen und Bürger Chancen und Risiken der Verfahren des maschinellen Lernens in Ansätzen reflektieren können sollen. Ist die Reflexionsebene abgekoppelt von der fachlichen Analyse, kann ein tiefgreifendes Verständnis für übertragbare Konzepte nicht erworben werden und eine Entmystifizierung nicht gelingen.

* Brandt, Y., Eickhoff-Schachtebeck, A. und Strecker, K. (2022a): „*Schulbuch starkeSeiten Informatik Jahrgang 9/10 Differenzierende Ausgabe Niedersachsen*“, Klett-Verlag, ISBN: 9783120075707
* Brandt, Y., Eickhoff-Schachtebeck, A. und Strecker, K. (2022b): „*Schulbuch starkeSeiten Informatik Jahrgang 9/10 Gymnasium Niedersachsen*“, Klett-Verlag, ISBN: 9783120075721
* Strecker, K. (2022): „*Virtuell-enaktives Erkunden von Verfahren des Maschinellen Lernens*“ in Marco Thomas, Michael Weigend (Hrsg.): „Inklusion mit Informatik“, 10. Münsteraner Workshop zur Schulinformatik, Books on Demand, ISBN: 9783756227235
* Strecker, K., (2021). *Physical Computing und neuronale Netze - Ein Konzept zur Kombination der Themengebiete Physical Computing und neuronale Netze*.   LOG IN: Vol. 41, No. 2. Berlin: LOG IN Verlag. (S. 65-76).
* Strecker, K. (2020): “*Realisation of Smart Home applications with Lego Mindstorms and neural network*” International Conference on Informatics in School: Situation, Evaluation, Problems, <http://ceur-ws.org/Vol-2755/>

# Bildquellenverzeichnis

S. 4.1, 5.2, 6.3, 6.4, 7.5, 7.6, 8.7: Media Office GmbH, Kornwestheim

S. 10.8, 10.9, 11.10, 11.11, 12.12, 13.13, 13.14, 14.15: Scratch/CC-BY-SA-4.0 Creative Commons. Scratch ist ein Projekt der Scratch Foundation und der Lifelong Kindergarten Group am MIT Media Lab. Es steht kostenlos unter https://scratch.mit.edu zur Verfügung.