

Spiele KI

Seminararbeit

Ein Vergleich von Lernprinzipien in Spielen

Ein Vergleich von Lernprinzipien in Spielen

Inhaltsverzeichnis

Vorwort	3
Einleitung	3
Faultier	3
Kolkrahe	3
Der Vergleich	4
Lernen – Was ist das?	5
Maschinelles Lernen	5
Definition	5
Warum Lernen?	5
Anpassung – Was ist das?	6
Maschinelle Anpassung	6
Definition	6
Warum Anpassen?	6
Allgemeines Lernverfahren	7
Aufbau	7
Bestimmung der Trainingserfahrung	7
Bestimmung der Zielfunktion	8
Bestimmung der Repräsentation	9
Bestimmung des Lernalgorithmus	10
Ablauf	11
Lernstrategien	12
Deduktives Lernen	12
Induktives Lernen	13
Variationen von Lernverfahren	14
Überwachtes Lernen	14
Unüberwachtes Lernen	14
Bekräftigungslernen	15
Anpassungsformen	16
Indirekte Anpassung	16
Direkte Anpassung	16
Anpassung durch Optimierung	17
Anpassung durch Verstärkung (reinforcement)	17
Anpassung durch Imitation	17
Hinweise für die Praxis	18
Wissensbasis	18
Lokales Optimum	18
Abhängigkeiten	18
Overfitting	19
Explore vs. Exploit	19
Fazit	20
Quellenangabe	21

Vorwort

Einleitung

Das Gebiet der „Künstlichen Intelligenz“ hat sich über Jahrzehnte beständig weiterentwickelt und es wurden immer neue Wege gefunden, ein Computersystem „intelligent“ zu gestalten.

Gerade im Bereich der Spieleentwicklung werden sehr hohe Anforderungen an das künstliche System gestellt, um möglichst menschengleiches Verhalten zu simulieren.

Im Fokus der „Intelligenz“ stehen Verfahren und Algorithmen, die dem Computer das Lernen ermöglichen.

Es existiert eine Vielzahl von Lernalgorithmen, die unterschiedliche Ansätze verfolgen und alle ihre Eigenheiten besitzen.

Oftmals findet hierbei eine Anlehnung an die Natur statt:



Faultier

Das Faultier ist eines der langsamsten Tiere der Welt.

Es bewegt sich mit einer durchschnittlichen Geschwindigkeit von rund 0,27 km/h, das entspricht 7,5 cm/s.

Es hält sich meist in den Ästen des Cecropia-Baumes auf, wo ebenfalls die aggressiven Azteka-Ameisen leben. Diese erkennen das Faultier aufgrund seiner Langsamkeit jedoch nicht als Feind an.

Die Anpassung an seine Umwelt erfolgte hier durch evolutionäres Lernen.

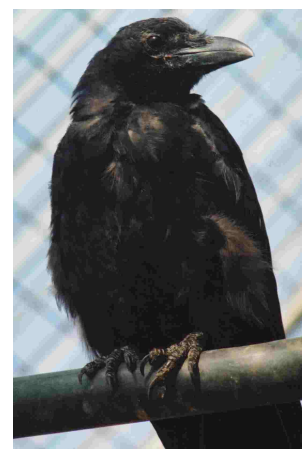
Kolkrabe

Der Kolkrabe überraschte die Wissenschaft durch seine außerordentliche Intelligenz.

Er ist dazu fähig komplexes Werkzeug zu bauen, um beispielsweise an Nahrung zu gelangen, wie viele wissenschaftliche Versuche gezeigt haben.

Er kann sich außerdem in seine Artgenossen hineinversetzen, um sich so z.B. vor Mundraub zu schützen.

Das Lernverhalten des Raben erfolgt hier durch Beobachtung und logische Schlussfolgerung.



Der Vergleich

Wie ist es nun möglich diese unterschiedlichen Arten des Lernens zu vergleichen?

Wenn man zwei so grundverschieden Lernverfahren betrachtet, scheint es keine sichtbaren Zusammenhänge zu geben. Es ist, als ob Äpfel mit Birnen verglichen werden, besser noch, Äpfel mit Aprikosen.

Es besteht natürlich die Möglichkeit, jede der Früchte einzeln zu beschreiben und die Besonderheiten hervorzuheben, was in einer endlos langen Liste von Eigenschaften enden würde.

Eine weitaus effektivere und übersichtlichere Lösung besteht in der Abstraktion der Früchte oder Tiere – oder der Lernalgorithmen.

In diesem Fall lassen sich Äpfel als Kernfrucht und Aprikosen als Steinfrucht leicht unterscheiden und die wesentlichen Merkmale herausarbeiten.

Im Folgenden werden nun die Lernalgorithmen abstrahiert und auf eine gemeinsame Basis gebracht.

Vorher muss jedoch erst einmal definiert werden, was Lernen und Anpassung in Computerprogrammen eigentlich bedeutet.

Zu guter Letzt erfolgt der eigentliche Vergleich zwischen den Lernmethoden anhand der Umsetzung in die Praxis.

Lernen – Was ist das?

Maschinelles Lernen

Definition

Das Wort "Lernen" geht auf die gotische Bezeichnung für "ich weiß" (lais) und das indogermanische Wort für "gehen" (lis) zurück. Die Herkunft des Wortes deutet bereits darauf hin, dass Lernen ein Prozess ist, bei dem man einen Weg zurücklegt und dabei zu Wissen gelangt.

Es existieren mehrere Auffassungen, was unter maschinellern Lernen verstanden werden kann, jedoch liefert „Wikipedia“ eine der sinnvollsten Definitionen:

Maschinelles Lernen ist ein Oberbegriff für die „künstliche“ Generierung von Wissen aus Erfahrung:

Ein künstliches System lernt aus Beispielen und kann nach Beendigung der Lernphase verallgemeinern.

D. h. es lernt nicht einfach die Beispiele auswendig, sondern es „erkennt“ Gesetzmäßigkeiten in den Lerndaten.

So kann das System auch unbekannte Daten beurteilen.

- Wikipedia

Lernen, im Sinne der künstlichen Intelligenz, ist also nicht das einfache Speichern von Daten, das Programmieren von Lösungen und insbesondere auch nicht die Anpassung an neue Zustände der Welt.

Warum Lernen?

Ohne Lernen keine Lernverfahren.

Es ist jedoch nur dann sinnvoll, wenn eine spätere Anpassung des lernenden Systems an die „Umwelt“ erfolgt – dies bildet die Grundlage für spätere Entscheidungsprozesse.

Oftmals muss gelernt werden, wenn die Parameter für die Lösung eines Problems zu komplex sind oder wenn die Eingabe von Daten von Hand zu aufwändig wird (datamining).

Ein weitaus wichtigerer Aspekt ist jedoch die Nutzbarmachung von Erfahrungen und die damit zusammenhängende Möglichkeit der dynamischen Anpassung an sich ändernde Umgebungen.

Anpassung – Was ist das?

Maschinelle Anpassung

Definition

Die Anpassung ist im Vergleich zum „Lernen“ sehr viel einfacher zu definieren, da der Begriff nicht viel Spielraum lässt.

Eine einfache und sinnvolle Definition der Anpassung im Allgemeinen:

Anpassung bedeutet die Adaption, beziehungsweise die Einstellung oder die Angleichung eines Objektes oder Zustandes an einen anderen.

- Wikipedia

Bei der „maschinellen Anpassung“ insbesondere bezogen auf Computerspiele, kann zusätzlich noch definiert werden, was unter Objekten bzw. Zuständen zu verstehen ist:

Objekte entsprechen hierbei dem Agenten bzw. dem Spieler in einer virtuellen Welt.

Die Zustände beinhalten das Gesamtsystem bzw. die virtuelle Welt, in dem sich die Agenten bzw. Spieler aufhalten.

Anpassung bedeutet also auch, dass Informationen über die Veränderung vorhanden sein müssen. Grundvoraussetzung ist daher eine Wissensbasis, die entweder direkt einprogrammiert oder gelernt werden kann.

Warum Anpassen?

Der Hauptgrund für die Anpassung besteht in der Nutzbarmachung von Erfahrungen.

Dies ermöglicht die Modellierung von adaptivem Verhalten, welches besonders in Spielen zu einer erhöhten Immersion des Benutzers führt und für mehr Spielspaß sorgt.

Außerdem können Systeme auf diese Art und Weise ihre Leistungsfähigkeit steigern.

Allgemeines Lernverfahren

Aufbau

Das abstrakte Modell eines konkreten Lernverfahrens besteht immer aus den gleichen Komponenten. So kann man einen schematischen Aufbau definieren, der für jedes Design durchlaufen werden muss:

Bestimmung der Trainingserfahrung

Es wird die Art der Datengrundlage bestimmt, d.h. in welcher Form die Daten vorliegen, wie z.B. als Datenbank, -sammlung, Expertenwissen oder beim Spiel mit einem Experten.

Zusätzlich kann bestimmt werden, ob die Daten vorhanden sind, ob sie erst generiert werden müssen oder ob die Daten bei Benutzung erzeugt werden.

Es wird in zwei Kategorien unterschieden:

➔ Offline Training

Die Daten liegen zur Entwicklungszeit vor und werden zum Trainieren benutzt.

➔ Online Training

Die Daten werden erst bei Benutzung, d.h. zur Laufzeit generiert und für das Training verwendet.

Zudem spielt der Lernzeitpunkt ebenfalls eine wichtige Rolle. Es gibt drei verschiedene Varianten:

➔ Learning by pattern

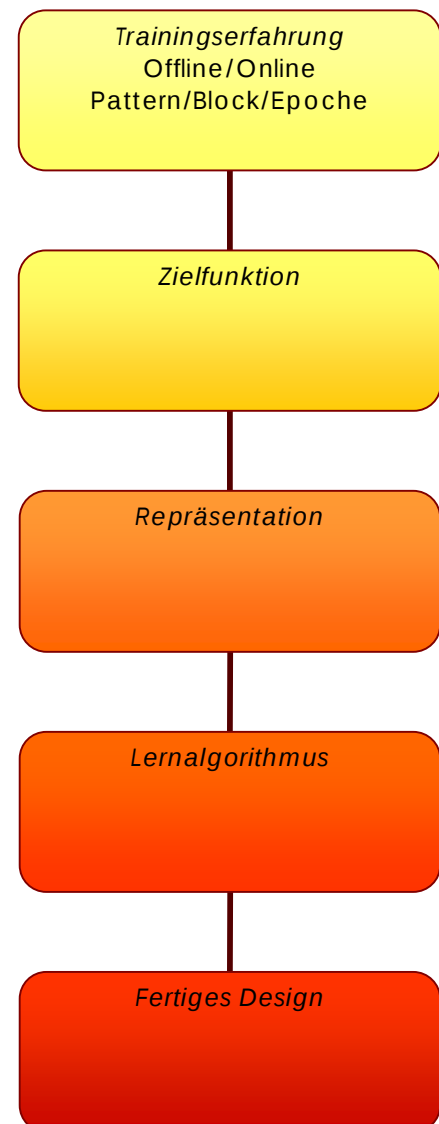
Die Trainingsdaten werden nach jedem Schritt ausgewertet.

➔ Learning by block

Erst nach einigen Schritten erfolgt die Auswertung.

➔ Learning by epoche

Gelernt wird nach einem kompletten Satz von Eingaben.



Bestimmung der Zielfunktion

Aus den vorhandenen Trainingsdaten muss nun eine Zielfunktion gebildet werden, die das Rohwissen in eine Wissensrepräsentation des Systems umwandelt.

Die Wahl der Trainingsdaten und Problemstellung beeinflusst die Bestimmung der Zielfunktion.

In den meisten Fällen wird hier das Problem der Mustererkennung definiert, d.h. die Funktion, um vorliegenden Daten zu klassifizieren und/oder zu bewerten.

Diese Funktion wird aus den Trainingsdaten durch z.B. Verallgemeinerung oder Extrapolation rekonstruiert.

In den anderen Fällen werden die Daten direkt genutzt, um die Funktion abzubilden.

Je nachdem wie die Zielfunktion bestimmt wird, ergeben sich zwei Arten des „Lernens“:

➡ Induktives Lernen

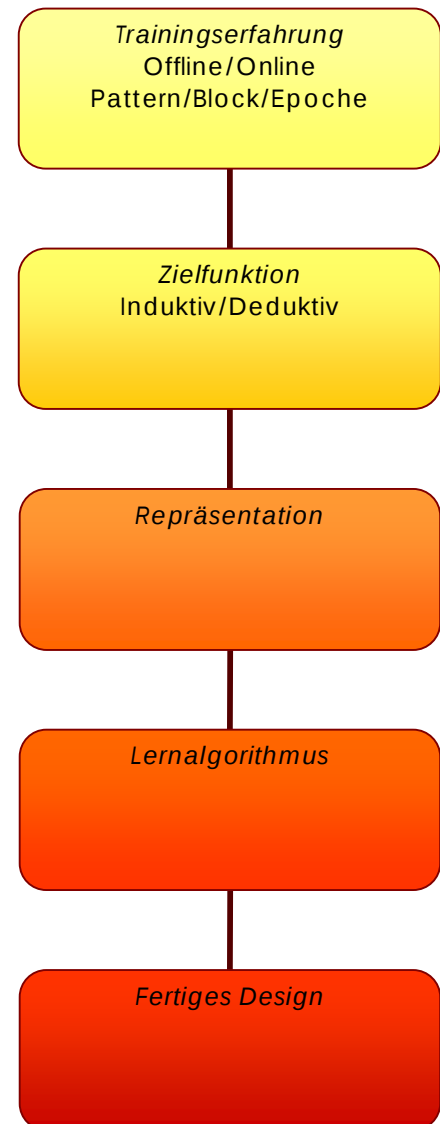
Es wird gelernt, indem von dem Speziellen auf das Allgemeine geschlussfolgert wird.

‡ Mustererkennung/Klassifizierung

➡ Deduktives Lernen

Es wird gelernt, indem von dem Allgemeinen auf das Spezielle (Fall) geschlussfolgert wird.

‡ Logik/Beweis



Bestimmung der Repräsentation

Die Repräsentation bestimmt in welcher Form die Zielfunktion bzw. die Ergebnisse der Zielfunktion abgebildet werden. Sie definiert die Wissensbasis und schränkt damit die Wahl des Lernalgorithmus stark ein.

Die meisten Repräsentationen bilden das Wissen in komprimierter Form ab, um es effizienter nutzen zu können.

Genauso wie es bei den Menschen ein implizites und ein explizites Wissen gibt, so gibt es die Unterscheidung auch in der Computerrepräsentation:

➔ Symbolisch

Das Wissen wird in verständlicher Form gespeichert, entspricht also dem expliziten Wissen.

D.h. für außenstehende Personen ist die Darstellung verständlich.

Z.B.: Regeln, Entscheidungsbäume, logische Formeln

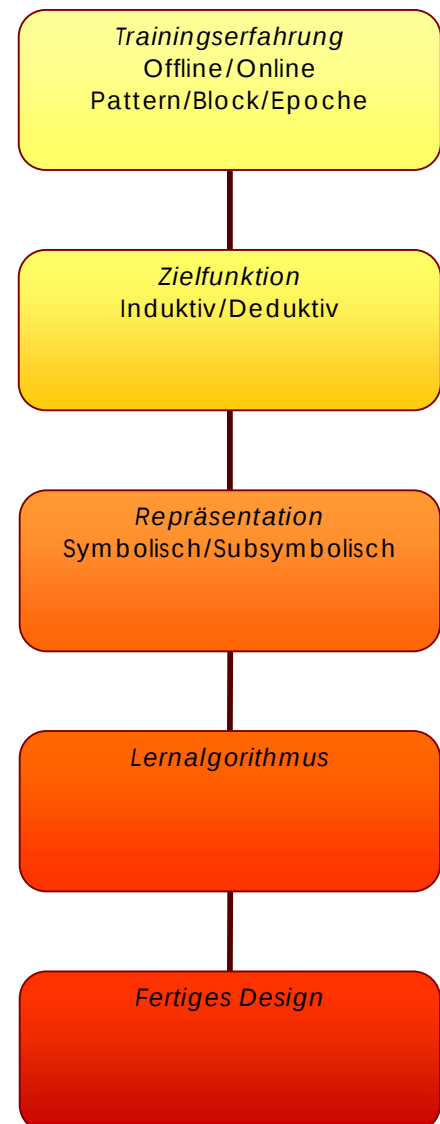
➔ Subsymbolisch

Das Wissen wird in abstrakter Form gespeichert, entspricht also dem impliziten Wissen (Können).

D.h. für außenstehende Personen ist die Darstellung nicht verständlich.

Der Vorteil dabei ist jedoch, dass die subsymbolische Abspeicherung meist leistungsfähiger ist.

Z.B.: Statistiken, Neuronale Netze, Genstrings



Bestimmung des Lernalgorithmus

Der Lernalgorithmus an sich ist der eigentliche Kern des ganzen Verfahrens.

Der Lernalgorithmus bestimmt die Form der Anpassung und bewertet gegebenenfalls das gelernte Wissen, um sich so selbst zu verbessern. Alternativ kann von dem Trainer ein Feedback zurückgegeben werden, um das Gelernte selbst zu bewerten.

Hier ergeben sich hauptsächlich zwei Möglichkeiten des Lernens:

➔ Überwachtes Lernen

Das Resultat kann direkt bewertet werden, da ein zu erwartendes Ergebnis vorliegt.

➔ Unüberwachtes Lernen

Das Resultat kann nicht bewertet werden, da keine zu erwartenden Ergebnisse vorliegen.

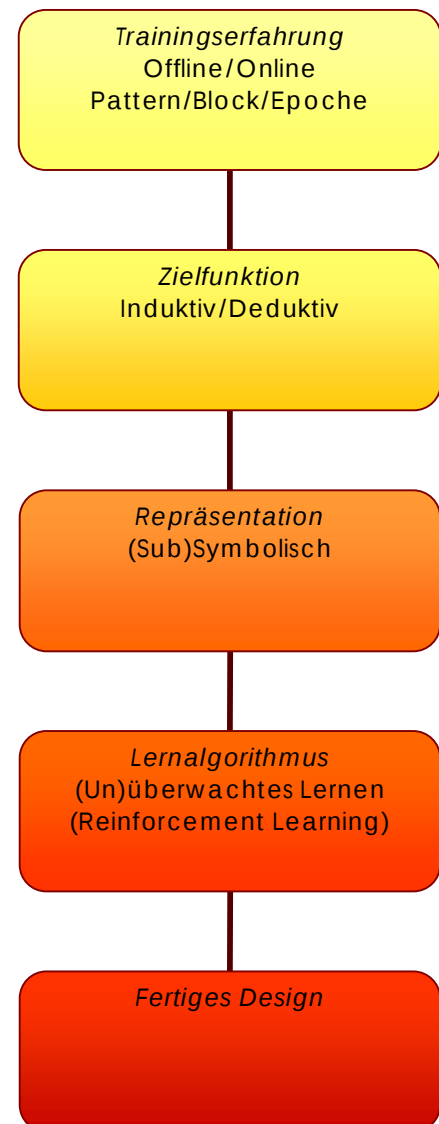
Eine Fitnessfunktion ist notwendig, um den Grad des Erfolges festzustellen.

➔ Reinforcement Learning

Unterstütztes Lernen ist eine Mischung aus den obigen Lernstrategien und wird manchmal auch als halbüberwachtes Lernen bezeichnet.

Das Resultat wird in diesem Fall zwar bewertet, jedoch nur indirekt, da lediglich eine Belohnung/Bestrafung erfolgt, ohne dass ein erwartetes Ergebnis vorliegt.

Es wird also der Zusammenhang zwischen Aktionen und Auswirkung gelernt.



Ablauf

Ist ein fertiges Design vorhanden, so durchläuft das Verfahren bei seiner Anwendung verschiedene Stadien. Je nach Wahl der Lernkomponenten können Schritte des Ablaufs entfallen oder sich in komplexere Module unterteilen.

Der schematische Ablauf bleibt jedoch immer gleich:

- Trainingsdaten werden erhoben oder generiert.
- Initialisierung der Zielfunktion
- Trainingsdaten werden eingegeben (iterativ).
 - Trainingsdatum wird eingegeben.
 - Die Zielfunktion wird gegebenenfalls gebildet.
 - Wissen wird intern gespeichert.
 - Lernalgorithmus wird angewendet.
 - Ergebnis wird bewertet.
 - Zielfunktion wird optimiert.
- Testdaten werden erhoben oder generiert.
- Testdaten werden eingegeben (iterativ).
- Überprüfen der Ergebnisse.
- Bei Fehlern wird der ganze Prozess wiederholt.

Wenn das Lernverfahren fertig gestellt ist, dann führt das Ergebnis der Zielfunktion zu einer Anpassung des Systems.

Welche Art der Anpassung vorhanden ist, hängt jedoch von der Implementierung des Programmierers ab. Je nach Lernverfahren eignen sich bestimmte Anpassungsmethoden besser als andere.

Lernstrategien

Im Vordergrund der Lernstrategien steht der Aufbau einer breiten Wissensbasis bzw. eines Regelwerks, anhand derer sich zukünftige Ereignisse einordnen lassen.

Wenn möglich sollte dieses Wissen in sich konsistent sein, d.h. richtige Eingaben sollten nicht zu dem falschen Ergebnis führen. Zudem müssen Lernstrategien in Computersystem schnell arbeiten, da es sonst nicht praktikabel ist sie zu verwenden.

Daraus ergeben sich zwei verschiedenen Ansätze, die jeweils den einen oder anderen Aspekt mehr unterstützen.

Deduktives Lernen

Das deduktive Lernen wird oftmals als erklärungsbasiertes Lernen bezeichnet. Wie der Name schon sagt, handelt es sich hierbei um eine Strategie, um eine Erklärung für ein gegebenes Trainingsbeispiel zu erzeugen.

Diese logische Schlussfolgerung wird aus dem Hintergrundwissen des Systems abgeleitet. Die Regel bzw. Erklärung wird dann verallgemeinert, sodass nicht nur das gegebene Beispiel von der generalisierten Regel erfasst wird, sondern auch jedes andere, welches sich nach dem gleichen Schema ableiten lässt.

Es wird also aufgrund eines Spezialfalls, unter Einbezug des Vorwissens, eine allgemeine Regel aufgestellt, um nun mit deren Hilfe ähnliche Spezialfälle erklären zu können.

Gegeben ist also ein Tripel (*In*, *f*, *Erwartet_Out*), wobei „*In*“ ein Beispiel darstellt, „*f*“ die Regel und „*Erwartet_Out*“ das erwartete Resultat („*f*“ könnte auch anhand „*In*“ und „*Erwartet_Out*“ aufgestellt werden).

Nun wird die Regel „*f*“ zu „*f**“ verallgemeinert, sodass für ein ähnliches Beispiel „*In**“ ebenfalls die Regel „*f**“ greift und somit zum Resultat „*Erwartet_Out*“ führt.

Ein Beispiel: „Ein intelligenter Höhlenmensch brät seine erbeutete Eidechse an einem angespitzten Stock über einem Lagerfeuer, um sich seine Finger nicht am Feuer zu verbrennen. Seine weniger intelligenten Genossen, die zu diesem Zweck bislang ihre Finger benutzen, beobachten ihn dabei.“

Aus dieser Beobachtung und ihrem Hintergrundwissen können sie ableiten, dass man eine Eidechse rösten kann, ohne sich die Finger dabei zu verbrennen, indem man einen dünnen, spitzen Stock benutzt.

Durch eine Generalisierung kommen sie zu dem Schluss, dass sich jedes Kleintier mit einem dünnen, langen, festen, spitzen Gegenstand gefahrlos über einem Feuer rösten lässt.

Einer der Genossen hat nun eine Ratte und hat jetzt gelernt, dass er auch diese mit einem anderen Spieß rösten kann.“

Die deduktive Lernstrategie hat damit also den Anspruch logisch korrekt und vollständig zu sein, da sich alle Regeln direkt ableiten lassen.

Dies gilt jedoch nur, wenn das Hintergrundwissen (Wissensdomäne) korrekt und vollständig ist. Alle abgeleiteten Schlussfolgerungen sind somit auch beweisbar richtig.

Der Nachteil dabei ist, dass die Abgrenzung einer Wissensdomäne manchmal sehr schwierig und aufwändig sein kann. Dadurch ist das Anwendungsgebiet dieser Lernstrategie nur auf die vorgegebene Wissensgrundlage beschränkt und somit nicht auf neue Situationen übertragbar.

Induktives Lernen

Das induktive Lernen basiert, im Gegensatz zu dem deduktiven Ansatz, auf dem Aufstellen von Hypothesen. Es begründet sich, wie der Name schon sagt, auf der Induktion, d.h. auf der Schlussfolgerung vom Besonderen auf das Allgemeine.

Diese Art der Regelgenerierung ist formal nicht korrekt, da sie eine korrekte Prämisse voraussetzt, die nicht in jedem Fall gegeben ist. Daher beruht die Strategie auch darauf, möglichst viele Beispiele zu nutzen, um eine Hypothese aufzustellen. Je mehr positive Beispiele vorhanden, desto sicherer ist auch die Hypothese.

Es wird also aufgrund von vielen allgemeinen Beispielen eine spezielle Regel aufgestellt, um mit deren Hilfe ähnliche Beispiele erklären zu können.

Gegeben ist also ein Tupel (In, Erwartet_Out).

Die Regel „f“ wird rekonstruiert anhand vieler ähnlicher „In“ Vektoren, die dasselbe „Erwartet_Out“ ergeben, sodass für ein neues ähnliches Beispiel „In*“ ebenfalls die spezielle Regel „f“ greift und somit zum Resultat „Erwartet_Out“ führt.

Ein Beispiel:

Sonne	Wind	Temperatur	Wasser	Sport
Scheint	Still	Warm	Kalt	JA
Scheint	Sturm	Warm	Warm	NEIN
Scheint	Still	Warm	Warm	JA

Aus den Beispielen kann nun die Regel erstellt werden, dass nur dann Sport getrieben wird, wenn gilt: „In“ = {Scheint, Still, Warm, ?}

Bei einem neuen Beispiel „In*“ = {Scheint, Still, Kalt, Heiß} kann nun darauf geschlossen werden, dass an diesem Tag kein Sport getrieben wird, da die „Temperatur = Kalt“ ist.

Die induktive Methode ist formal nicht ganz korrekt, erlaubt aber eine flexiblere Gestaltung der Regeln. Außerdem muss lediglich eine Zielkategorisierung vorhanden sein statt einer vollständigen Domäentheorie.

Variationen von Lernverfahren

Das Prinzip der Lernverfahren ist es, Wissen zu generieren, um damit eine Möglichkeit bereit zu stellen, das System an gegebene Zustände anzupassen.

Dabei muss beachtet werden, dass jedes Lernverfahren ein Ziel verfolgen muss. Es muss gewährleistet werden, dass der Lernprozess zum gewünschten Erfolg führt.

Deshalb ist es wichtig, dass der Lernprozess eine stetige Rückmeldung über die Qualität des gelernten Wissens erhält.

Diese Rückkopplung wird im Allgemeinen als Feedback bezeichnet und kann verschiedene Ausprägungen annehmen, die im Folgenden erklärt werden.

Überwachtes Lernen

Beim überwachten Lernen (englisch: supervised learning) gibt es eine höhere Instanz, auch „Lehrer“ genannt, die den Lernprozess direkt bewertet.

Das Feedback wird durch den Lehrer als Person oder durch das System selbst übermittelt. Es besteht ebenfalls die Möglichkeit dieses über die Trainingsdaten einzugeben. Die Daten geben für eine Auswahl an Problemen die korrekte Lösung vor und können somit abgeglichen werden (siehe Induktives Lernen/Deduktives Lernen).

Oftmals wird die Differenz zu dem gewünschten Wert und dem tatsächlichen Wert als Anhaltspunkt genommen, um die Güte des Lernprozesses zu bestimmen.

Unüberwachtes Lernen

Unüberwachten Lernen (englisch: unsupervised learning) bezeichnet ein Lernverfahren ohne Überprüfung von außen. Die Zielwerte sind im Voraus nicht bekannt und somit kann auch kein Feedback durch den „Lehrer“ gegeben werden.

Bei diesem Lernverfahren wird aufgrund der Beziehung zwischen den Eingabedaten gelernt. Außerdem existieren keine von dem Programmierer vorgenommenen Klassifizierungen, d.h. das unüberwachte Lernen beinhaltet eine besondere Art des induktiven Lernens.

Gegeben sind hier nur Beispiele „In“, die in Beziehung gesetzt werden. Das Verfahren ähnelt einem Clusteringverfahren, welches die Daten in Kategorien einteilt, die sich durch charakteristische Muster unterscheiden. Auf der gefundenen Unterteilung kann nun die Zielfunktion gebildet werden.

Um eine solche Einordnung überhaupt zu finden, muss ein Algorithmus existieren, der die aktuelle Einteilung bewertet, also internes Feedback gibt. Diese Funktion wird meistens Fitnessfunktion oder Optimierungsfunktion genannt. Sie bestimmt die Güte des Ergebnisses.

Die Schwierigkeit diese Methode besteht in der Kontrolle des Lernens.

Meist kann erst nach langer Trainingszeit (siehe learning by epoche) eine Beobachtung des Verhaltens und damit eine Bewertung vorgenommen werden.

Das unüberwachte Lernen eignet sich besonders beim Vorliegen von unübersichtlichen oder komplexen Parametern....

Der große Vorteil ergibt sich jedoch durch den sehr flexiblen Einsatz bei unübersichtlichen Parametern. Zusätzlich werden neue Verhaltensweisen entdeckt, die nicht durch den Programmierer bestimmt wurden.

Bekräftigungslernen

Eigentlich ist das Bekräftigungslernen (englisch: reinforcement learning) nur eine Mischart der oben genannten Variationen. Es ist daher auch unter dem Synonym „halbüberwachtes Lernen“ bekannt.

Bei diesem Lernverfahren wird das Feedback ähnlich wie bei dem überwachten Lernen vermittelt, jedoch wird statt der korrekten Lösung nur eine Benotung, im Sinne einer Belohnung oder einer Bestrafung, übergeben.

Dies hat zur Folge, dass die Beziehung zwischen Aktion und Auswirkung gelernt wird. Wenn kein externer Lehrer vorhanden ist, muss auch hier eine Funktion, wie bei dem unüberwachten Verfahren, die „Benotung“ übernehmen.

Häufig findet diese Modell Anwendung in der Agentenarchitektur, die sich am sinnvollsten dafür eignet.

Anpassungsformen

Anpassung ist kein direkter Bestandteil von Lernverfahren, obwohl sie dies oft implizieren, denn Lernverfahren ohne Anpassung macht das Lernen überflüssig.

Für den Benutzer von Computerspielen steht die Anpassung des NPC an erster Stelle. Ihm ist es egal auf welche Art und Weise das System lernt oder ob es überhaupt lernt. Wichtig ist, dass sich der NPC möglichst gut an das menschliche Verhalten anpasst.

Daher ist die einfachste Handhabe eine Anpassung vorzunehmen, das Vortäuschen des Lernens. Wann immer möglich und für den Spieler nicht sichtbar sollte diese Methode angewandt werden.

Wenn jedoch komplexere Verhaltensmuster von Nöten sind, dann sollte doch auf „echte“ Anpassungsarten zurückgegriffen werden.

In der Regel werden zwei Anpassungsformen unterschieden:

Indirekte Anpassung

Dieser Form der Anpassung ist die nächst höhere Stufe nach dem Vortäuschen. Anstatt nur die aktuellen Weltzustände als Orientierung zu nehmen, werden Statistiken über die Welt aufgestellt und gespeichert. Sie bilden die Datengrundlage für den Lernprozess.

Durch die indirekte Anpassung wird der aktuelle Spieler und die Welt mit einbezogen, was zu einer deutlich flexibleren Anpassung führt.

Die indirekte Anpassung bietet sich für Lernverfahren an, die noch einfach zu handhaben sind, aber dennoch eine Vielzahl von Verhaltensmöglichkeiten bieten sollen. Die Information, auf denen das Lernverhalten basiert, muss trotz allem vom Programmierer vorgegeben werden, sodass dieses Verfahren noch einfach zu testen ist.

Als Nachteil anzuführen ist die Tatsache, dass die Verhaltensmöglichkeiten trotz allem eingeschränkt sind und keine charakterliche Anpassung des NPC möglich ist.

Ansonsten ist dies das meistgenutzte, effizienteste Verfahren, welches sehr realistische Anpassungsmöglichkeiten bietet.

Direkte Anpassung

Die direkte Anpassung ergibt sich meist aus dem unüberwachten Lernverfahren, da jeweils die Einteilung der Kategorien und Güte der Abbildung optimiert werden.

Dies entspricht neuem Verhalten und charakterlichen Änderungen des NPCs und bietet somit die flexibelste Art der dynamischen Anpassung.

Die Adaption wird direkt über Parameter des Charakters gesteuert und verbessert.

Den Verhalten sind hier keine Grenzen durch den Programmierer gesetzt, was einerseits positiv ist, da sich der NPC auf jede erdenkliche Situation einstellen kann, sich andererseits aber auch negativ auf das Verhalten auswirkt, wenn nicht jede Situation bedacht wird und somit chaotisches Verhalten auch „gültig“ wäre.

Das bedeutet, dass die Randbedingungen und das Grundwissen von dem Programmierer im Vorfeld implementiert werden sollten, um dem System möglichst wenig Spielraum für eine „falsche“ Entwicklung zu geben.

Zudem ergeben sich drei Variationen der direkten Anpassung, die sich in dem Feedbacksignal unterscheiden:

Anpassung durch Optimierung

Dies ist die herkömmliche Methode, um direkte Anpassung umzusetzen. Die Grundidee ist die Bewertung des Verhaltens aufgrund einer Funktion, die versucht die optimalen Parameter für das Maximum der Verhaltensweise zu berechnen.

Oftmals werden damit genetische und evolutionäre Algorithmen in Verbindung gebracht.

Anpassung durch Verstärkung (reinforcement)

Beim reinforcement learning ergibt sich die Anpassung indirekt aus Reward-Feedback. Hier wird, wie schon erwähnt (siehe Variation von Lernen - reinforcement learning), die Auswirkung einer Aktion betrachtet und diese im Sinne einer Benotung beurteilt.

Es ist ein System, welches aus Fehlern lernt und damit eine strategische Anpassung vornimmt. Hierfür eignen sich insbesondere Agentenarchitekturen oder das sogenannte Q-Learning.

Anpassung durch Imitation

Bei der Imitation werden Referenzdaten benutzt, um sich diesen lernend anzunähern.

Dafür wird beispielsweise das Verhalten eines Menschen in einem Spiel aufgenommen und für die Bewertung des Lernprozesses verwendet.

Das Feedback wird also durch die Referenzdaten erzeugt.

Ein Beispiel wären Neuronale Netze, die oft trainiert werden, um menschliches Verhalten zu imitieren.

Hinweise für die Praxis

Die folgenden Hinweise beziehen sich hauptsächlich auf die direkten Anpassungsformen, sollten jedoch auch bei indirekter Anpassung beachtet werden.

Wissensbasis

Bei der Implementierung sollte darauf geachtet werden, dass eine genügend große Wissensbasis zur Verfügung steht.

Andernfalls würde das Lernverfahren eine zu langsame Anpassung hervorbringen, zufälliges Verhalten erzeugen und in keiner Weise kontrollierbar sein.

Daher sollten sinnvolle Begrenzungen vorgegeben und eventuell ein indirektes Lernverfahren vorgeschoben werden.

Lokales Optimum

Bei einer Verbesserung des Verhaltens durch eine Optimierungsfunktion sollte darauf geachtet werden, dass diese möglichst monoton ist. Die Monotonie sorgt dafür, dass das Verhalten ständig weiter verfeinert wird, kann aber zum „Overfitting“ führen.

Falls ein lokales Optimum vorhanden ist, so führt es zu einer Stagnation des Anpassungsprozesses, d.h. obwohl das Verhalten zu dem Zeitpunkt optimal erscheint, so kann es in einem größerem Umfeld betrachtet nur suboptimal sein.

Abhängigkeiten

Bei allen Lernverfahren muss darauf geachtet werden, dass die Adaption an ein bestimmtes Verhalten nicht in Wechselwirkung mit einem anderen erlernten Verhalten steht. Wenn möglich müssen diese unabhängig voneinander sein.

Da Abhängigkeiten zwischen den Lernzielen nicht immer leicht zu erkennen sind, sollten die Lernverfahren auch nur an den Ecken angewandt werden, an denen sie wirkliche Vorteile mit sich bringen. Der sparsame Einsatz dieser Methoden erleichtert die Kontrolle über das Gesamtsystem und führt oft zu einer höheren Effizienz.

Overfitting

Overfitting ist ein Phänomen, welches sich aus zu langen oder allzu speziellen Lernprozessen ergibt. Hier wird zwar ein optimales Verhalten gefunden, jedoch nur für spezielle Situationen, sodass das Verhalten bei kleinster Änderung der Situation nicht mehr anwendbar ist.

Vermieden werden kann das Overfitting durch Anpassung der Lerndauer und durch geschickte Wahl der Optimierungsparameter, d.h. zustandsunabhängigen Parametern.

Explore vs. Exploit

In der Praxis sollte stets darauf geachtet werden, welches Ziel das Lernverfahren anstrebt. Einerseits kann neues Verhalten gelernt oder andererseits altes Verhalten optimiert werden.

Die Kunst besteht nun darin, ein geeignetes Mittel zwischen diesen zwei Wegen zu finden, sodass weder zu wenige „gute“ Verhalten auftreten, noch zu viele „Schlechte“.

Oftmals kann dies nur aus Erfahrung oder durch Ausprobieren richtig abgewogen werden.

Fazit

In der Welt der künstlichen Intelligenz gibt es eine Reihe unterschiedlicher Lernprinzipien und noch mehr Lernalgorithmen. Die Unterscheidung ist meist nicht ganz eindeutig, da viele Prinzipien miteinander verbunden werden können.

Letzten Endes variieren die Lernprinzipien hauptsächlich nur in ihrer Komplexität.

Da die Kontrolle über lernende Systeme ein sehr wichtiger Aspekt (gerade in Bezug auf Computerspiele) ist, werden meist immer noch indirekte Lernverfahren bevorzugt.

Inzwischen gibt es aber auch einige Ansätze der Implementierung direkter Verfahren, die jedoch noch schwer zu handhaben sind, wobei sie dem menschlichen Spieler wohl am nächsten kommen.

Letztendlich zählt für den Computerspieler jedoch nur das Resultat – „Hauptsache die Anpassung ist realistisch“. Ihn interessiert nicht, welche Lernprinzipien hinter einem Spiel stecken. Die Anpassung im Spiel wird, wie beim „reinforcement learning“, nur mit „gut“ oder „schlecht“ bewertet.

Daher kann man sagen, dass Komplexität nicht zwangsläufig vorteilhaft sein muss.

Also sollte, wenn die Möglichkeit besteht, auf die einfacheren Lernverfahren zurückgegriffen werden und nur dann, wenn es wirklich notwendig erscheint, sollten die komplexeren Verfahren Anwendung finden.

Trotz Allem wird das direkte Lernverfahren zur Laufzeit, also online, das Ziel der Spieleindustrie sein, da damit das dynamischste, individuellste und menschlichste Verhalten erreicht werden kann.

Quellenangabe

Kolkrabe (Photo)

<http://members.home.nl/rjvandoorn/asielkraai.jpg>

Wikipedia – Faultiere (Photo)

<http://de.wikipedia.org/wiki/Faultier>

Tierlobby - Faultiere

<http://www.tierlobby.de/rubriken/Tiergarten/faultiere/faultiere.htm>

Wikipedia – Anpassung

<http://de.wikipedia.org/wiki/Anpassung>

Wikipedia – Maschinelles Lernen

http://de.wikipedia.org/wiki/Maschinelles_Lernen

University Utrecht - Department of Information and Computing Sciences

www.cs.uu.nl/docs/vakken/gag/Adaptive%20Game%20AI%20Lecture.pdf

Association for the Advancement of Artificial Intelligence – Machine Learning

<http://www.aaai.org/AITopics/html/machine.html>

Uni Bremen - Maschinelles Lernen in Computerspielen

<http://www.informatik.uni-bremen.de/~jplath/WS05/Seminar/Folien2.pdf>

Uni Heidelberg - Einführung in die KI

<http://www.cl.uni-heidelberg.de/kurs/ss05/einfki/KI-2005-Teil11.pdf>

Uni Ulm - Einführung in die KI

<http://www.informatik.uni-ulm.de/ki/Edu/Vorlesungen/GdKI/WS0203/r12a.pdf>

Delft University of Technology - Improving adaptive game AI with evolutionary learnings

www.kbs.twi.tudelft.nl/docs/MSc/2004/Ponsen_Marc/thesis.pdf