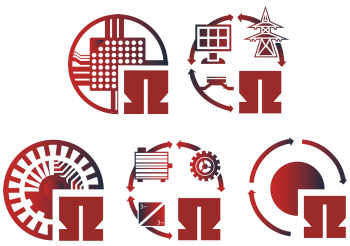
ELSYS Note



Reinforcement Learning

In diesem ELSYS Note werden Grundlagen zum Reinforcement Learning (RL) dargestellt. Es erfolgt eine Abgrenzung zu anderen Verfahren und das einem RL-System zugrundeliegende Prinzip wird erläutert. Außerdem werden Aufbau und Funktionsweise der beim Deep RL verwendeten neuronalen Netze erklärt. Abschließend wird kurz auf den Einsatz von RL in der elektrischen Antriebsregelung eingegangen.

Einleitung

Der Begriff künstliche Intelligenz (KI) taucht heutzutage in fast allen Bereichen des Lebens auf, auch im Bereich der elektrischen Antriebstechnik, Leistungselektronik und Regelungstechnik. KI umfasst verschiedenartige Ansätze, um Maschinen und technische Systeme, die im Stande sind Probleme intelligent zu lösen, zu erschaffen. Hierbei existiert allerdings keine allgemeingültige Definition des Begriffs „Intelligenz“. Beim Ansatz des maschinellen Lernens erfolgt die Problemlösung durch Algorithmen. Diese verarbeiten Beispieldaten, was als Lernen bezeichnet wird, und erstellen basierend hierauf ein generalisiertes Modell, mit dem im Anschluss reale Probleme gelöst werden können. Die am weitesten verbreiteten Verfahren des maschinellen Lernens sind das Supervised Learning, das Unsupervised Learning und das Reinforcement Learning (RL).

Abgrenzung

Beim Supervised Learning benötigt der Algorithmus im Lernprozess einen ausreichend großen Satz an Datenpaaren, die aus Eingangs- und korrekt zugeordneter Ausgangsvariable bestehen. Basierend auf diesem Trainingsdatensatz generiert der Algorithmus eine Approximation der reellen Funktion, aus der der Datensatz generiert wurde. Durch diese Funktion werden nach dem Training Ausgangsvariablen für beliebige Eingangsvariablen abgeschätzt.

Beim Unsupervised Learning werden ebenfalls Datensätze für den Lernprozess benötigt, die jedoch ausschließlich aus Eingangswerten bestehen. Der Algorithmus versucht in den Daten Muster zu erkennen.

Beim RL wird kein zuvor erstellter Datensatz benötigt. Die für den Lernprozess benötigten Daten werden schrittweise produziert, indem das System mit seiner Umgebung interagiert.

Die Einordnung der Verfahren ist in Abbildung 1 grafisch dargestellt.

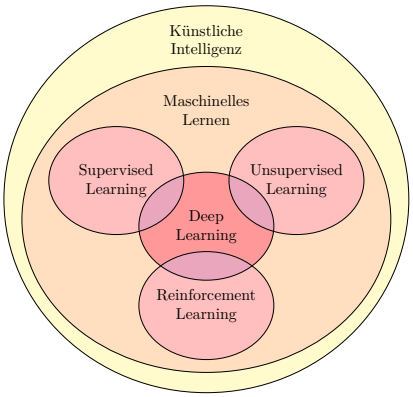


Abb. 1 Einordnung von maschinellem Lernen und KI

Aufbau eines RL-Systems

Die zwei Hauptkomponenten eines RL-Systems sind, wie in Abbildung 2 dargestellt, der Agent und seine Umgebung. Der Agent besteht aus programmierter Software. Das Kernelement des Agenten ist eine Handlungsstrategie, die den vorliegenden Zuständen Wahrscheinlichkeiten zuordnet, mit denen mögliche Aktionen ausgewählt werden. Die Umgebung umfasst alle Bestandteile des Systems außerhalb des Agenten.

Im Lernprozess wählt der Agent ausgehend von dem Zustand $s(t)$, in dem sich das System zum Zeitpunkt t befindet, basierend auf der aktuellen Handlungsstrategie eine Aktion $a(t)$ aus. Diese liegt je nach Anwendung in einem definierten Bereich und kann diskret oder kontinuierlich sein. Daraufhin gerät die Umgebung in einen neuen Zustand $s(t+1)$, der wiederum vom Agenten beobachtet wird.

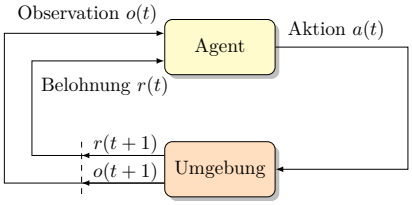


Abb. 2 Aufbau eines RL-Systems

Der Zustand $s(t)$ beschreibt hierbei die gesamte Umgebung des Systems. Die Zustandsinformationen, die tatsächlich dem Agenten zugeführt werden, werden Observationen $o(t)$ genannt. Zusätzlich erhält der Agent eine Belohnung $r(t)$, die die gewählte Aktion und den daraus resultierenden Systemzustand bewertet. Die Belohnung ist eine Zahl, die aus einer festgelegten Belohnungsfunktion berechnet wird.

Basierend hierauf wird die Handlungsstrategie des Agenten in jedem Lernschritt angepasst. Das Ziel des Agenten ist es, die optimale Handlungsstrategie zu entwickeln, die für jeden Zustand, in dem sich das System befindet, die bestmögliche Aktion ausgibt, und dadurch die über alle Zeitschritte kumulierte Belohnung, den sogenannten Return $G(t)$, zu maximieren.

Bei der Formulierung eines RL-Problems muss beachtet werden, dass das System die Markov-Eigenschaft innehat. Diese besagt, dass der Folgezustand ausschließlich vom aktuellen Zustand und den in diesem Zustand möglichen Aktionen abhängt.

In der Realität sind die Voraussetzungen für das Finden der optimalen Handlungsstrategie, wie das Vorhandensein ausreichender Rechenleistung oder das Aufweisen der Markov-Eigenschaft, meistens nicht vollständig erfüllt. Deshalb verwenden viele RL-Konzepte Näherungsverfahren, um die im Optimierungsprozess entstehenden nichtlinearen Gleichungssysteme zu lösen.

Deep RL

Deep Learning bezeichnet eine Methode des maschinellen Lernens, in der künstliche, tiefe neuronale Netzen eingesetzt werden, um komplexe, hochdimensionale Probleme zu lösen. Wie in Abbildung 1 dargestellt, ist eine Kombination aus Deep Learning und den verschiedenen Verfahren des maschinellen Lernens möglich. So wird beim Deep RL klassisches RL mit Deep Learning verknüpft. Durch die Abbildung der Handlungsstrategie im RL-Agenten durch ein neuronales Netz kann hochdimensionales, interaktives Lernen realisiert werden.

Künstliche neuronale Netze stellen parametrisierte Funktionen dar, die aus Eingängen Ausgangswerte berechnen. Sie bestehen aus mehreren Schichten mit einer Vielzahl an miteinander verknüpften Neuronen, wie beispielhaft in Abbildung 3 dargestellt. Schichten, die zwischen Ein- und Ausgangsschicht liegen, heißen versteckte Schichten. Netze, die mehrere versteckte Schichten aufweisen, werden als tief bezeichnet.

Die Anzahl der Schichten sowie der Neuronen je Schicht unterscheidet sich je nach Anwendungsfall. Mit steigender Anzahl an Neuronen oder Schichten können komplexere Funktionen abgebildet werden. Gleichzeitig erhöht sich allerdings das Risiko einer verringerten Verallgemeinerungsfähigkeit des Netzes.

In Abbildung 4 werden die Berechnungen innerhalb eines Neurons aufgezeigt. Die Eingangssignale x_1 bis x_N werden mit den zugehörigen Gewichten w_i multipliziert. Im Anschluss wird die Summe aus diesen Produkten und dem Bias b gebildet. Auf diese Summe wird abschließend eine, häufig nichtlineare, Aktivierungsfunktion σ angewendet. Der Ausgang der Aktivierungsfunktion entspricht dem Ausgang $\hat{y}$ des Neurons.

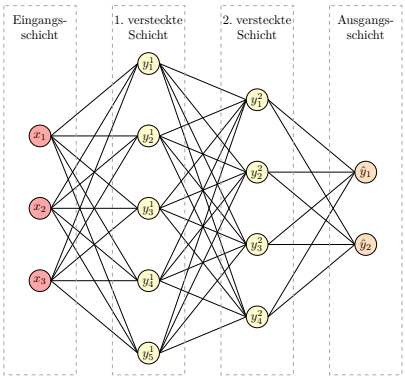


Abb. 3 Beispielhafter Aufbau eines neuronalen Netzes mit zwei versteckten Schichten

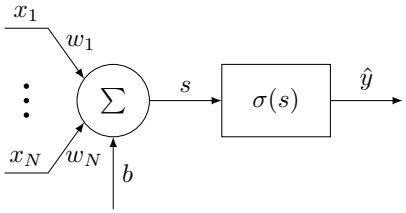


Abb. 4 Funktionales Modell eines Neurons

Mathematisch lässt sich dieser Zusammenhang wie folgt formulieren:

$$\hat{y} = \sigma\left(\sum_{i=1}^{N} w_i \cdot x_i + b\right) \quad (1)$$

Die Gewichte w_i und der Bias b sind hierbei die Parameter des neuronalen Netzes, die im Training, genauer gesagt Lernprozess iterativ optimiert werden. Bei mehrschichtigen Netzen erfolgt die Optimierung durch Anwendung des Backpropagation-Algorithmus. Bei diesem wird zunächst in einer Vorwärtsphase der Eingangsschicht des neuronalen Netzes Eingangswerte übergeben. Daraufhin erfolgt von der Eingangsschicht über alle versteckten Schichten hinweg, bis hin zur Ausgangsschicht, in jedem Neuron die in Gleichung (1) dargestellte Berechnung, wodurch letztendlich in der Ausgangsschicht die Bestimmung der Ausgangswerte des neuronalen Netzes erfolgt. In der anschließenden Rückwärtsphase werden die Netzparameter optimiert, indem der Ausgangsfehler zwischen dem tatsächlichen und dem durch das neuronale Netz abgeschätzten Wert bestimmt wird.

RL in der elektrischen Antriebstechnik

Auch in der elektrischen Antriebstechnik und -regelung wird Deep RL zunehmend eingesetzt. So wird das Forschungsfeld der RL-basierten Regelung auch am Institut ELSYS aktiv vorangetrieben. Beispielsweise wird am Einsatz von RL-basierten Stromreglern für permanenterregte Synchronmaschinen geforscht.