Materialien zur Vorlesung Lernen
Reinforcement - Lernen

Ralf Der

January 18, 2007

Die bisher behandelten Lernsituationen des uberwachten Lernens gingen alle davon aus daß der Lehrer zu jedem Input x die Soll-Werte für den Output des lernenden Systems zur Verfügung stellt. Dies entspricht in keiner Weise der Situation in der sich z. B. ein Lebewesen befindet das lernen muß, sich in einer ihm mehr oder weniger unbekannten Umgebung zu behaupten. In dieser Situation gibt es selten einen Lehrer der dem lernenden System das Soll Verhalten vorgeben kann. Die einzige Information die dem Lerner zur Verfügung steht ist die oft erst nach langer Zeit erfolgende Reaktion der Umgebung in Form von Belohnung oder Bestrafung für richtiges oder falsches Verhalten Der Lerner deutet die Belohnung als Bestätigung (*reinforcement*) seines Verhaltens und entwickelt Strategien um sein Reinforcement zu vergrößern.

Eine erste Formulierung des RLs wurde von Thorndike unter der Bezeichnung *the law of effects* aus dem Studium des Lernverhaltens von Tieren abgeleitet: "Animals responses to environment in a given situation are adapted such that the probability for finding satisfaction if this situation occurrs again is enhanced". Die *satisfaction* kann als die Belohnung oder Bestätigung (Reinforcement) des Lebewesens durch seine Umgebung aufgrund eines sinnvollen Verhaltens gedeutet werden. Ziel der Adaption des Lebewesens ist also die Verbesserung seiner Chancen aus jeder Situation heraus in Zukunft ein möglichst großes Reinforcement zu realisieren.

Diese Form des Lernens kann als eine Art des uberwachten Lernens verstanden werden allerdings bei sehr unspezifischer Information durch den Lehrer.

1 Realisierung

Wir betrachten speziell das sog. Q–Lernen nach Watkins (1992).

- Gegeben eine Menge S von diskreten Zuständen. Zustand ist beispielsweise der Ort eines Roboters in einer Gitterwelt.
- Menge A von diskreten Aktionen. Beispiel fahre nach Norden, Süden, ...

- Über den Zuständen $x \in S$ ist eine Funktion $r(x)$ definiert, die in jedem Zustand das Reinforcement bemisst. Meist ist $r(x) = 0$ und nur in bestimmten Zielzuständen ist $r(x) > 0$ (Belohung) oder < 0 (Bestrafung).

- Die Policy-Funktion π: Legt das Verhalten des Agenten fest. Benennt zu jedem Zustand x die Aktion a

$$a = \pi(x)$$

  In der Praxis zwei Möglichkeiten:

  - Deterministische Policy: Tabelle die jedem x genau ein a zuordnet.
  - Stochastische Policy. Varianten:
    1. Die Aktion a aus der Tabelle wird nur mit Wahrscheinlichkeit p ausgeführt. Mit der komplementären Wahrscheinlichkeit $1 - p$ wird eine der restlichen Aktionen ausgewählt.
    2. Man weist über eine Wertfunktion Q jedem Zustands-Aktionspaar (x, a) einen Wert $Q(x, a)$ zu und orientiert die Wahrscheinlichkeit für a an $Q(x, a)$. Meist verwendeter Ansatz über sog. Boltzmannfaktor

$$p_a(x) = \frac{e^{Q(x,a)/T}}{\sum_{a'} e^{Q(x,a')/T}} \tag{1}$$

       Diskussion: Es gilt

$$\sum_a p_a(x) = 1$$

       und T ist die sog. Temperatur. Für $T \to \infty$ alle Aktionen gelichwahrscheinlich, für $T = 0$ deterministisches Verhalten, d.h. $p_{a^*} = 1$ für die beste Aktion

$$a^* = \arg\max_a Q(x, a)$$

- Die Q Funktion wird am Anfang mit Null (oder random) initialisiert und in jedem Zeitschritt nach fogendem Algorithmus aktualisiert (Update-Regel).

2 Algorithmus des Q-Lernens:

1. Registriere aktuellen Zustand. Dieser sei x. (Beispiel x = aktuelle Position des Roboters)

2. Im Zustand x: Wähle Aktion a mit Wahrscheinlichkeit p_a (nach Formel 1).

3. Führe Aktion a aus. (Beispiel gehe nach links). Registriere neuen Zustand. Dieser sei y.

4. Führe update des Q Wertes für das Zustands-Aktionspaar (x, a) aus:

$$\Delta Q(x, a) = \varepsilon (r(y) + \gamma E(y) - Q(x, a))$$

wobei $0 << \gamma < 1$ und

$$E(y) = \max_a Q(x, a) = Q(x, a^*)$$

ist die sog. Wertfunktion.

5. Ersetze $x \leftarrow y$. Gehe zu 2.

Die **Parameter** ε und T werden im Laufe der Zeit "abgekühlt", d.h. nach 0 gefahren.

2.1 Konvergenz

Der Algorithmus ist konvergiert, wenn $\Delta Q(x, a) = 0$ für alle ZA-Paare (x, a). Dann gelten folgende Beziehungen:

$$Q(x, a) = r(y) + \gamma E(y) \tag{2}$$

oder

$$Q(x, a) = r(y) + \gamma Q(y, a^*) \tag{3}$$

mit y ist Folgezustand von x unter der Aktion a. Daraus kann man ableiten, dass im konvergierten Zustand sich ein Wertgebirge entwickelt, wobei der Wert eines Zustandes exponentiell mit dem Abstand zum Zielzustand x^{ziel} (falls es nur einen gibt, d.h. nur an x^{Ziel} ist $r > 0$) abnimmt.

2.2 Bedeutung der Wertfunktion

Betrachte Zeittakt $t = 0, 1, 2, ...$Zu jeder Zeit t ist System in Zustand $x_t \in S$. Sei Algorithmus konvergiert und wir nehmen immer die beste Aktion, d.h. $Q(x, a) = E(x)$. Damit folgt aus 2 bzw. 3

$$E(x_t) = r(x_{t+1}) + \gamma E(x_{t+1})$$

Durch fortgesetzte Iteration dieser Beziehung

$$E(x_t) = r(x_{t+1}) + \gamma (r(x_{t+2}) + \gamma E(x_{t+2}))$$

bekommt man

$$E(x_t) = \sum_{k=1}^{\infty} \gamma^{k-1} r(x_{t+k})$$

Wert eines Zustande = diskontierte Summe der zukünftigen Reinforcements die man unter der optimalen Strategie erzielen kann.