CCA – COMPETENCE CENTRE
HTL Anichstraße

# Q-Learning im Nim-Spiel [2, 2]

## Regeln und Lernziel

Beim Nim-Spiel mit zwei Stapeln zu je zwei Objekten gilt: Wer den letzten Stein nimmt, verliert. Ziel des lernenden Agenten ist es also, den Gegner dazu zu bringen, den letzten Zug zu machen.

## Q-Learning Überblick

Q-Learning basiert auf der Q-Tabelle, in der jedem Zustand-Aktions-Paar ein Wert zugewiesen wird. Diese Werte werden mit folgender Formel aktualisiert:

$$Q(s, a) \leftarrow Q(s, a) + \alpha * (\text{Belohnung} + \gamma * \max_{a'} Q(s', a') - Q(s, a))$$

## Initiale Q-Tabelle (alle Werte = 0)

Zustand: [2, 2]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| **(0,1)** | 0.0 |
| **(0,2)** | 0.0 |
| **(1,1)** | 0.0 |
| **(1,2)** | 0.0 |

### *Spiel 1 – Niederlage (Agent macht letzten Zug)*

1. [2,2] → (0,2) → [0,2] (Agent)
2. [0,2] → (1,1) → [0,1] (Random)
3. [0,1] → (1,1) → [0,0] (Agent verliert)

Letzte Aktion: [0,1], (1,1) → Q = 0 + 0.5 * (-1 - 0) = -0.5

Rückpropagation: [0,2], (1,1) → Q = 0 + 0.5 * (0 + 0.9 * -0.5) = -0.225

Rückpropagation: [2,2], (0,2) → Q = 0 + 0.5 * (0 + 0.9 * -0.225) = -0.101

### *Spiel 2 – Sieg (Gegner macht letzten Zug)*

1. [2,2] → (1,2) → [2,0] (Random)
2. [2,0] → (0,1) → [1,0] (Agent)
3. [1,0] → (1,1) → [0,0] (Random verliert)

Letzte Aktion: [2,0], (0,1) → Q = 0 + 0.5 * (1 - 0) = 0.5

Rückpropagation: [2,2], (1,2) → Q = 0 + 0.5 * (0 + 0.9 * 0.5) = 0.225

## Q-Tabelle nach 2 Spielen

Zustand: [2,2]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| **(0,1)** | 0.000 |
| **(0,2)** | -0.101 |
| **(1,1)** | 0.000 |
| **(1,2)** | 0.225 |

Zustand: [0,2]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| **(1,1)** | -0.225 |
| **(1,2)** | 0.000 |

Zustand: [0,1]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| **(1,1)** | -0.500 |

Zustand: [2,0]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| **(0,1)** | 0.500 |

## Fazit

Der Agent lernt aus den Ergebnissen ganzer Spiele. Macht er den letzten Zug, erhält er eine negative Belohnung. Wenn der Gegner das Spiel beendet, wird eine positive Belohnung zurückpropagiert. Dadurch passt sich die Q-Tabelle dynamisch an und hilft dem Agenten, strategisch zu spielen.

Nach nur zwei Spielen spiegelt die Q-Tabelle noch **kein intelligentes Verhalten** wider. Die Werte beruhen auf wenigen Erfahrungen und können zu falschen Entscheidungen führen. Zum Beispiel wirkt die Aktion (1,2) im Zustand [2,2] optimal, obwohl sie oft zum Verlust führt.

## Q-Tabelle nach 100 Spielen

Zustand: [2,2]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| (0,1) | 0.65 |
| (0,2) | -0.55 |
| (1,1) | 0.65 |
| (1,2) | -0.45 |

Zustand: [0,2]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| (1,1) | -0.95 |
| (1,2) | -0.85 |

Zustand: [0,1]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| (1,1) | -1.00 |

Zustand: [2,0]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| (0,1) | 0.95 |

Zustand: [1,0]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| (0,1) | -1.00 |

Zustand: [1,1]

| Action (Stapel, Anzahl) | Q-Wert |
|---|---|
| (0,1) | 0.90 |
| (1,1) | 0.90 |

Nach vielen Spielen hat der Agent durch Belohnungen und Rückpropagation gelernt, welche Züge zu sicheren Siegen führen und welche vermieden werden sollten. Die Q-Tabelle zeigt nun deutlich **bessere bzw. intelligentere Entscheidungen**.