Uczenie ze wzmocnieniem — aplikacje

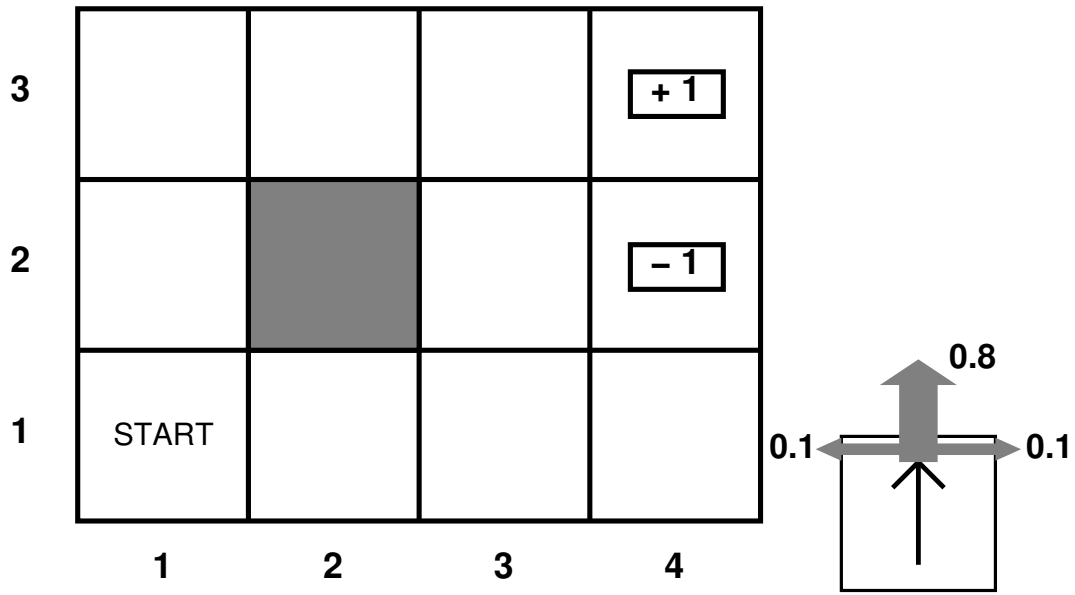
Na podstawie: AIMA ch21 oraz Reinforcement Learning (Sutton i Barto)

Wojciech Jaśkowski

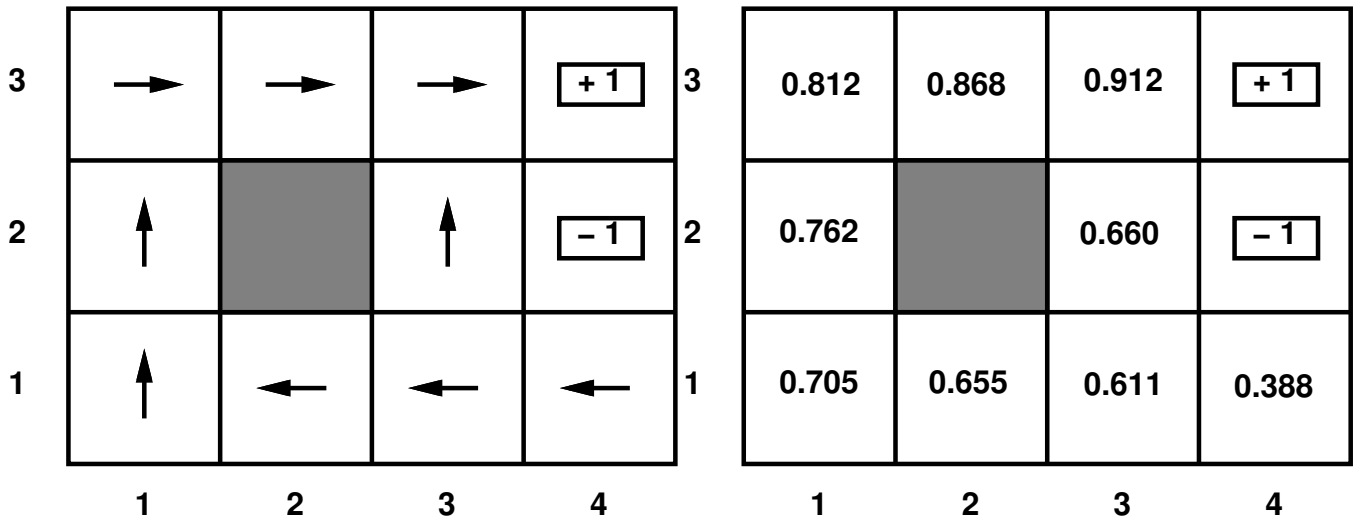
Instytut Informatyki,
Politechnika Poznańska

22 maja 2013

Problem decyzyjny Markova

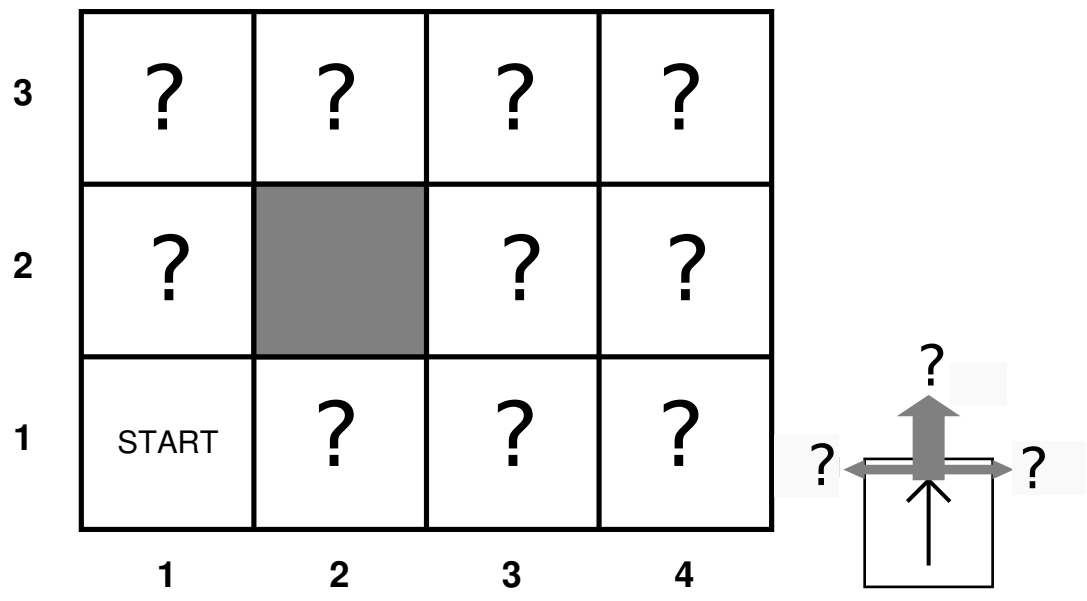


Rozwiązanie problemu decyzyjnego Markova



nieznany MDP

brak f. nagrody i modelu przejść



Uczenie ze wzmocnieniem

Uczenie:

- **pasywne** $\rightarrow$ ocena użyteczności danej polityki π
- **aktywne** $\rightarrow$ znalezienie optymalnej polityki π
  - eksploracja!

Rodzaje uczenia

1. **Uczenie nadzorowane** (nauczyciel)
   1. **klasyfikacja**: *atrybuty* $\rightarrow$ *[znana!] klasa decyzyjna*
   2. **regresja**: *atrybuty* $\rightarrow$ *[znana!] wartość*
2. **Uczenie nienadzorowane** (brak nauczyciela)
   1. atrybuty $\rightarrow$ [*nieznana*!] wartość/klasa
3. **Uczenie ze wzmocnieniem** (krytyk)
   1. akcja $\rightarrow$ wzmocnienie (kara / nagroda)

Podejścia do uczenia ze wzmocnieniem

Równanie Bellman'a:

$$U(s) = R(s) + \gamma \max_{a \in A} \sum_{s'} U(s')P(s'|s,a)$$

Trzy podejścia:

1. **agent odruchowy** (*policy search*)
   - uczy się polityki $\pi : S \to A$
2. **agent z funkcją użyteczności**
   - uczy się f. użyteczności $U(s)$
   - przykłady: **adaptatywne programowanie dynamiczne** (ADP), **uczenie różnicowe** (TDL)
3. **agent z funkcją Q**
   - uczy się funkcji $Q(s,a)$
   - przykład: **Q-learning**

Który agent potrzebuje modelu świata? [zadanie 1]

Reguły uczenia

Uczenie różnicowe

$$U^\pi(s) \leftarrow U^\pi(s) + \alpha\left(R(s) + \gamma U^\pi(s') - U^\pi(s)\right)$$

- α — współczynnik uczenia

Reguła modyfikacji Q-Learning

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Liczba stanów

- ADP działa rozsądnie dla problemów wielkości rzędu 10000 stanów.
  - tryktrak: 10^{20}
  - szachy: 10^{40}
- Nie da się explicite rozważać tylu stanów

Aproksymator funkcji

- Inna reprezentacja niż tablica (Q lub U). Np. liniowa kombinacja jakichś cech $f_1, \ldots, f_n$:

$$\hat{U}_\theta(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \cdots + \theta_n f_n(s)$$

  - Uczymy się tylko parametrów θ.

Przykład

$$\hat{U}_\theta(s) = \theta_1 l\,pionków(s) + \theta_2 l\,figur\,w\,centrum(s) + \theta_3 hetman?(s) + \theta_4 szach?(s)$$

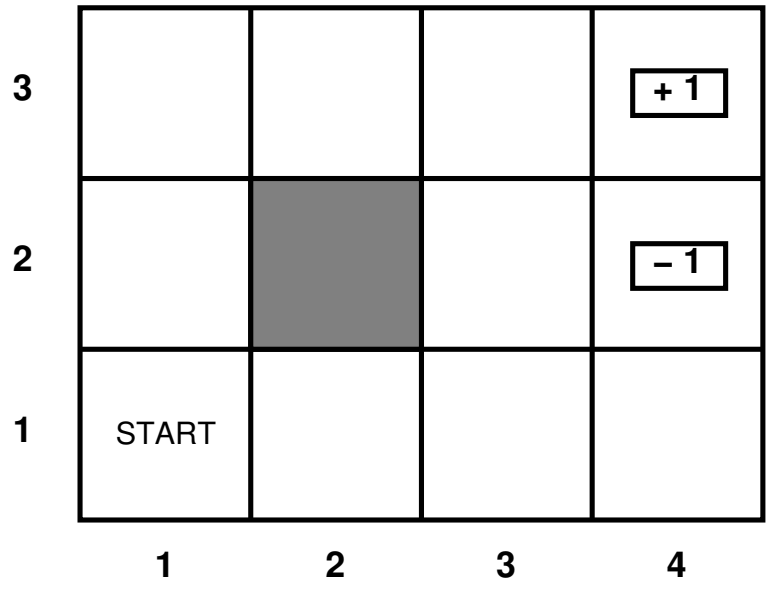
- 10^{40} stanów $\rightarrow$ 6 parametrów

Aproksymator funkcji

- Aproksymator funkcji musi być łatwo obliczalny.
- Cechy:
  1. kompresja (mała liczba stanów)
  2. **możliwość uogólniania wiedzy** (stany odwiedzone vs. nieodwiedzone)
     1. Przykład: co 10^{12} stan $\rightarrow$ „mistrzowski” gracz w tryktraka
- Kompromis: wielkość przestrzeni (jakoś aproksymacji) vs. czas nauki

Reguła Widrow-Hoff'a

Bezpośrednia estymacja użyteczności



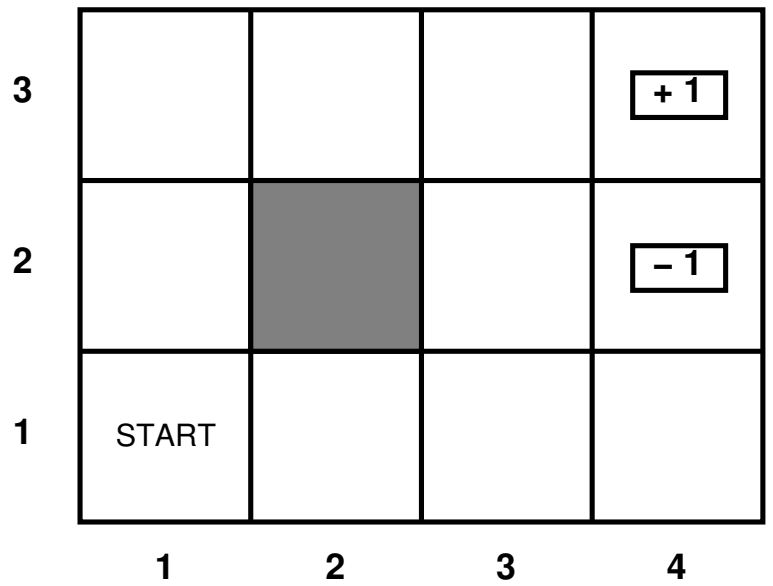
- Przykład. Dla naszego świata 4×3, niech:

$$\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y$$

- $(\theta_0, \theta_1, \theta_2) = (0.5, 0.2, 0.1) \Longrightarrow$ [zadanie 2]
- Wykonaliśmy przebieg od stanu $(1, 1)$ i otrzymaliśmy sumaryczne wzmocnienie $u(1, 1) = 0.4$.
- **Wniosek**: $\hat{U}_\theta(1, 1) = 0.8$ to za dużo.

Reguła Widrow-Hoff'a

Bezpośrednia estymacja użyteczności



- Przykład. Dla naszego świata 4×3, niech:

$$\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y$$

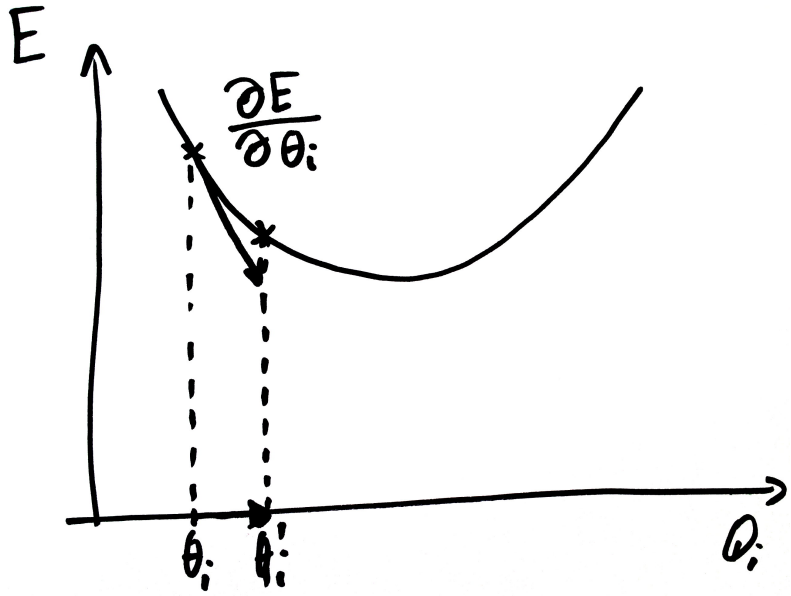
- $(\theta_0, \theta_1, \theta_2) = (0.5, 0.2, 0.1) \Longrightarrow$ [zadanie 2]
- Wykonaliśmy przebieg od stanu $(1, 1)$ i otrzymaliśmy sumaryczne wzmocnienie $u(1, 1) = 0.4$.
- **Wniosek**: $\hat{U}_\theta(1, 1) = 0.8$ to za dużo.

Reguła Widrow-Hoff'a

Bezpośrednia estymacja użyteczności

Definiujemy funkcję błędu:

$$E(s) = \frac{1}{2}\left(\hat{U}_\theta(s) - u(s)\right)^2$$



- Minimalizujemy błąd zgodnie z gradientem:

$$\theta_i \leftarrow \theta_i - \alpha\frac{\partial E(s)}{\partial \theta_i} = \theta_i - \alpha\frac{\partial\left(\frac{1}{2}\left(\hat{U}_\theta(s) - u(s)\right)^2\right)}{\partial\theta_i}$$

$$= \theta_i + \alpha\left(u(s) - \hat{U}_\theta(s)\right)\frac{\partial\hat{U}_\theta(s)}{\partial\theta_i}$$

Reguła Widrow-Hoff'a

Bezpośrednia estymacja użyteczności

$$\theta_i \leftarrow \theta_i + \alpha \left(u(s) - \hat{U}_\theta(s) \right) \frac{\partial \hat{U}_\theta(s)}{\partial \theta_i}$$

U nas:

$$\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y,$$

więc w naszym przykładzie:

$$\theta_0 \leftarrow \theta_0 + \alpha(u(s) - \hat{U}_\theta(s))$$
$$\theta_1 \leftarrow \theta_1 + \alpha(u(s) - \hat{U}_\theta(s))x$$
$$\theta_2 \leftarrow \theta_2 + \alpha(u(s) - \hat{U}_\theta(s))y$$

Przykład

Bezpośrednia estymacja użyteczności

$$\theta_0 \leftarrow \theta_0 + \alpha(u(s) - \hat{U}_\theta(s))$$
$$\theta_1 \leftarrow \theta_1 + \alpha(u(s) - \hat{U}_\theta(s))x$$
$$\theta_2 \leftarrow \theta_2 + \alpha(u(s) - \hat{U}_\theta(s))y$$

Niech:

- $(\theta_0, \theta_1, \theta_2) = (0.5, 0.2, 0.1)$
- $u(1,1) = 0.4$

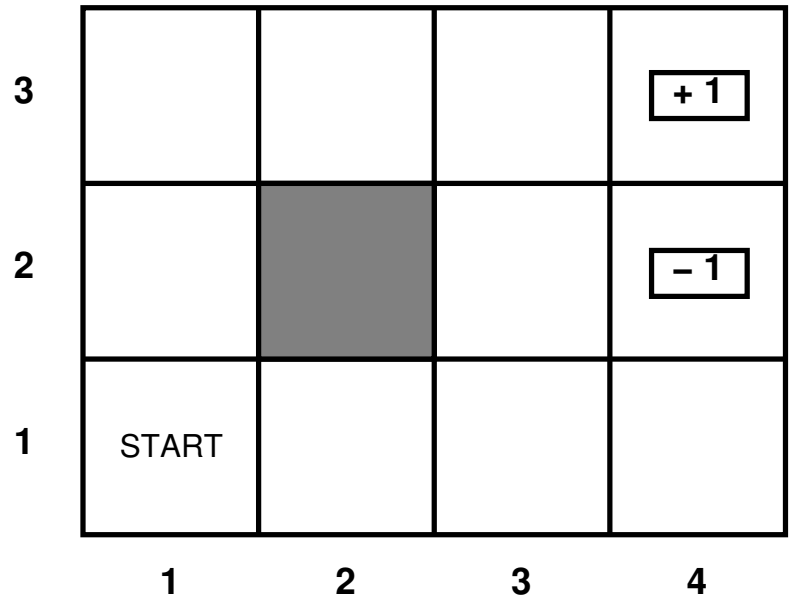
Pytania:

1. Ile będą wynosić parametry $(\theta_0, \theta_1, \theta_2)$ po aktualizacji $(\alpha = 0.25)$? [zadanie 3]
2. Ile wyniesie $\hat{U}_\theta(1,1)$ po aktualizacji parametrów?[zadanie 4]
3. Chcieliśmy, aby $\hat{U}_\theta(1,1)$ się zmieniło. **Czy zmieniło się także** $\hat{U}_\theta(1,2)$? [zadanie 5]

Przykład

Generalizacja

Agent uczy się szybciej z aproksymatorem funkcji, bo może **generalizować**.



- Jeśli aproksymator funkcji ma postać

$$\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y,$$

  to szybciej dla świata 10×10 z nagrodą $+1$ w polu $(10, 10)$.
- A co by było, gdyby $+1$ było w polu $(5, 5)$?[zadanie 6]
- Możemy dodać do $\hat{U}_\theta(x, y)$ składnik $\theta_3 f_3$, gdzie

$$f_3 = \sqrt{(x-5)^2 + (y-5)^2}$$

Uczenie różnicowe

Wersja oryginalna

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha \left(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s) \right)$$

Z aproksymatorem funkcji

$$\theta_i \leftarrow \theta_i + \alpha \left(R(s) + \gamma \hat{U}_{\theta}(s') - \hat{U}_{\theta}(s) \right) \frac{\partial \hat{U}_{\theta}(s)}{\partial \theta_i}$$

Q-learning

Wersja oryginalna

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

Z aproksymatorem funkcji

$$\theta_i \leftarrow \theta_i + \alpha \left(R(s) + \gamma \max_{a'} \hat{Q}_\theta(s',a') - \hat{Q}_\theta(s,a) \right) \frac{\partial \hat{Q}_\theta(s,a)}{\partial \theta_i}$$

Szukanie polityki (ang. *policy search*)

- Polityka $\pi : S \to A$
- Chcemy reprezentować π nie dla każdego stanu, ale w sposób bardziej zwięzły (np. zestaw parametrów θ)
- Np. możemy reprezentować politykę π jako zestaw aproksymatorów funkcji Q:

$$\pi(s) = \max_a \hat{Q}_\theta(s, a),$$

gdzie $\hat{Q}_\theta$ jest np. sumą jakichś funkcji ważoną parametrami θ (*vide* poprzednia sekcja)

- **Szukanie polityki** = dostosowuj θ, tak aby poprawiać działanie π.
  - Czyli: ucz się funkcji $\hat{Q}_\theta$.
  - Czy to jest to samo, co Q-learning?[zadanie 7]

Reprezentacja polityki

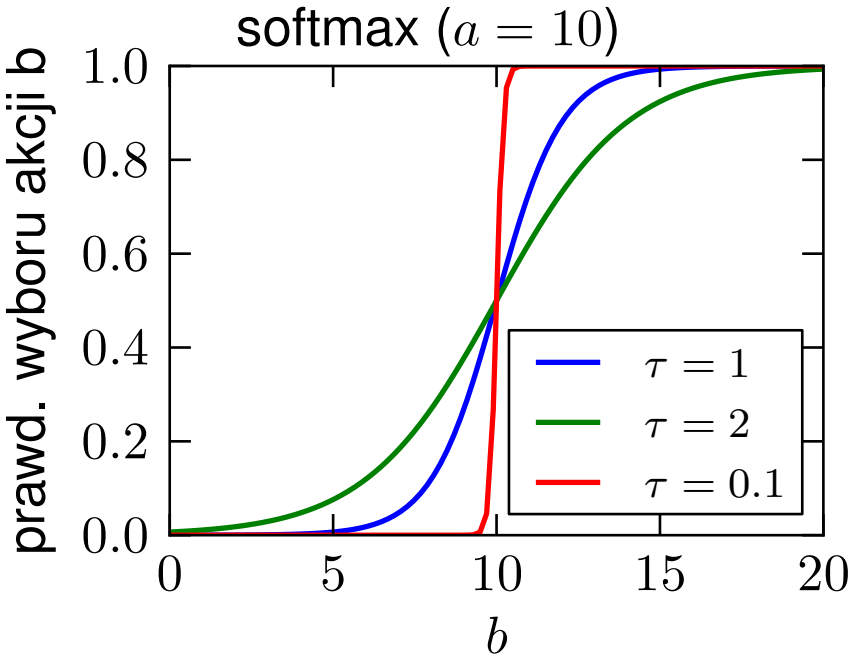
$$\pi(s) = \max_a \hat{Q}_\theta(s, a)$$

- W **Q-learning'u** (z aproksymatorem funkcji) szukamy $\hat{Q}_\theta$, które jest możliwie bliskie Q^*.
- W **szukaniu polityki** szukamy θ, które powoduje, że π „działa dobrze".
  - Przykład: Czy $\hat{Q}_\theta(s, a) = Q^*(s, a)/10$ jest optymalnym rozwiązaniem?[zadanie 8]
- Problem: $\pi(s)$ jest nieciągłą funkcją parametrów θ, jeśli akcje są dyskretne
  - czasem minimalna zmiana w θ może spowodować, że $\pi(s)$ „przeskoczy" z jednej akcji na inną.
    - dlatego **uczenie gradientowe** π nie jest możliwe.

Polityka stochastyczna

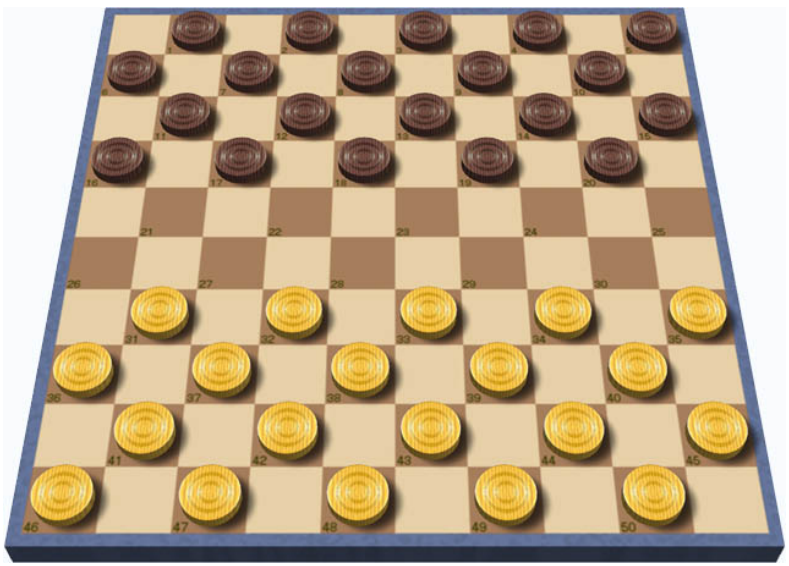
- Dlatego używa się **polityki stochastycznej** $\pi_\theta(s, a)$, reprezentującej prawd. wybrania akcji a w stanie s.
- Reprezentacja z użyciem **funkcji softmax**:

$$\pi_\theta(s, a) = e^{\hat{Q}_\theta(s,a)/\tau} / \sum_{a'} e^{\hat{Q}_\theta(s,a')/\tau}$$



- Zaleta: różniczkowalna

Warcaby (Artur Samuel, 1959)



- liniowa aproksymator funkcji: 16 cech
- wariant uczenia różnicowego (TDL)

Tryktak (Gerry Tesauro, 1992)

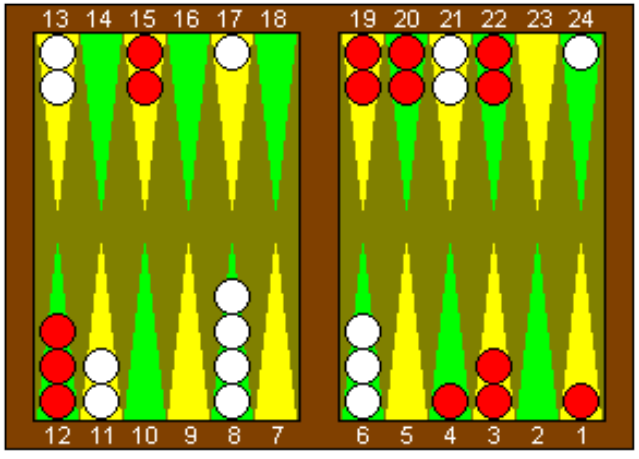


Figure 3. A complex situation where TD-Gammon's positional judgment is apparently superior to traditional expert thinking. White is to play 4-4. The obvious human play is 8-4*, 8-4, 11-7, 11-7. (The asterisk denotes that an opponent checker has been hit.) However, TD-Gammon's choice is the surprising 8-4*, 8-4, 21-17, 21-17! TD-Gammon's analysis of the two plays is given in Table 3.

- TD-Gammon: wcześniej: uczenie ze wzmocnieniem było tylko „teoretyczną ciekawostką”
- Teraz: ¿2000 cytowań
- Poziom mistrzowski

Tryktak (Gerry Tesauro, 1992)

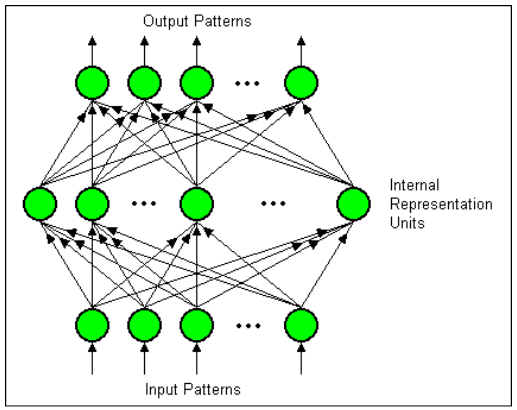
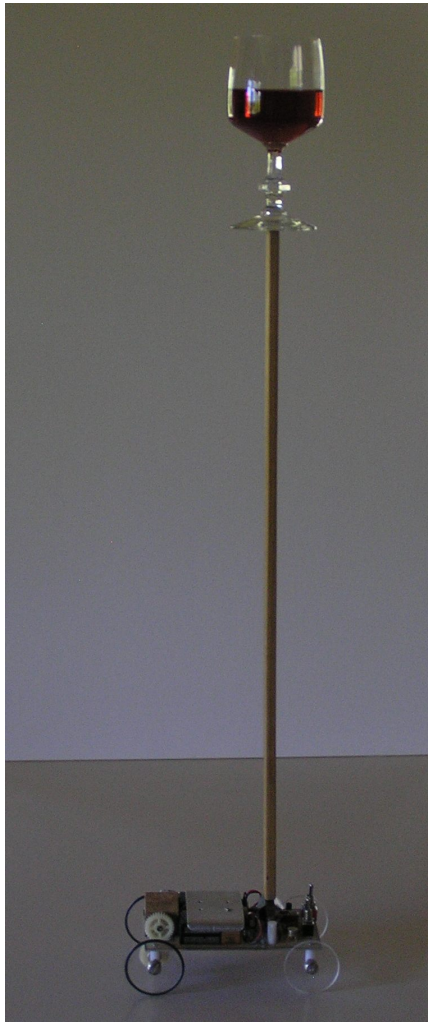


Figure 1. An illustration of the multilayer perception architecture used in TD-Gammon's neural network. This architecture is also used in the popular backpropagation learning procedure. Figure reproduced from [9].

- Początkowo: uczył sieć neuronową reprezentującą $Q(s, a)$ za pomocą przykładów od ekspertów $\rightarrow$ żmudne, słaby program
- Potem: gra z samym sobą (ang. ***self-play***)
- Uczenie różnicowe (TDL), kara/nagroda: ostatni stan
- Wejście (cechy): 24 wartości („surowy” stan planszy) + 40 węzłów w warstwie ukrytej
- 200,000 gier uczących (2 tygodnie uczenia)

Balansowanie tyczką / odwrócone wahadło (Michie, Chambers, 1968)

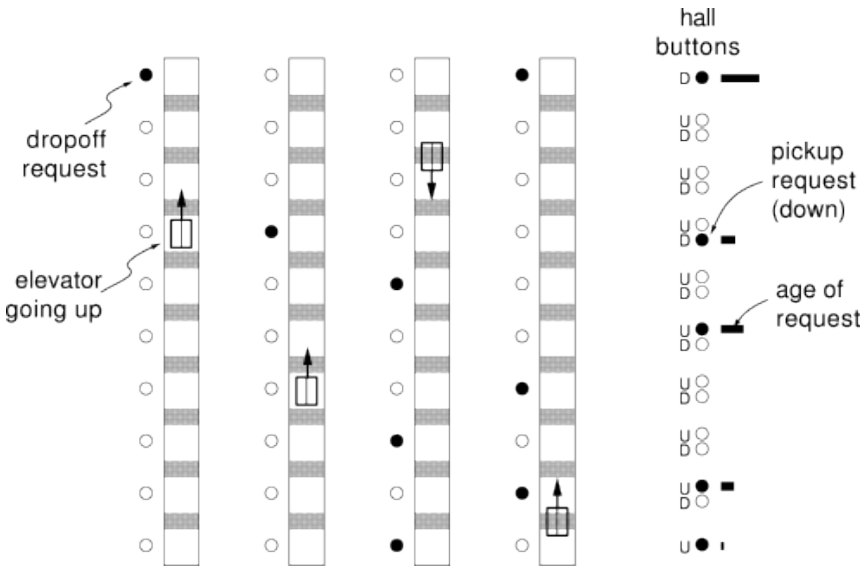
ang. pole balanding / inverted pendulum



- Problem ciągły
- Co jest stanem?[zadanie 9]
- Jakie akcje są możliwe?
- Algorytm Boxes:
  - Dyskretyzacja w „pudełka”
  - Potrzeba jedynie 30 prób uczących, aby balansować przez godzinę
  - Bez symulatora
  - Negatywne wzmocnienie za ostatni (s, a) przed upadkiem.
- Dwie tyczki, Podwójna tyczka, Potrójna tyczka, UAV

Sterowanie dźwigami wind (Crites i Barto, 1996)

ang. elevator dispatching problem



Źródło: http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node111.html

- 4 windy, 10 pięter, przestrzeń stanów: ca. 10^{22} stanów.
- Przestrzeń akcji?
  - Pewne uproszczenia: każda winda osobno: **Multi Agent Reinforcement Learning**
- Q-learning
- Stan reprezentowany przez sieć neuronową: 47 wejść, 20 węzłów ukrytych i 2 wyjścia