Zaawansowane Metody Inteligencji Obliczeniowej
Wykład 7: Actor critic

Michał Kempka Marek Wydmuch Bartosz Wieloch

11 kwietnia 2022

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

"Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)",
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20

Plan wykładu

Uczenie polityki

- Do tej pory omawiane algorytmy działały wg. schematu:
  - wyznacz wartość stanu lub akcji
  - na podstawie tych wartości wyznacz politykę (akcje do wykonania)
- Teraz omówimy metody które będą uczyły się polityki

  $$\pi(a|s,\theta) = P(A_t = a|S_t = s, \theta_t = \theta)$$

  gdzie θ to wektor parametrów polityki
- Funkcja wartości $\hat{v}(s, \mathbf{w})$ może być wykorzystywana do nauki parametrów polityki θ ale **nie jest potrzebna do wyboru akcji**.

Optymalizacja polityki

- Omawiane metody nauki polityki będą bazowały na gradiencie oceny wydajności działania polityki $J(\theta)$
- Będziemy maksymalizowali ocenę

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$

  gdzie $\widehat{\nabla J(\theta_t)}$ to stochastyczna estymata gradientu $J(\theta_t)$
- Metody gradientu polityki (ang. policy gradient methods)
- Jeśli dodatkowo metoda uczy się $\hat{v}(s, \mathbf{w})$ — metoda aktor-krytyk (ang. actor-critic)
  - „aktor” — polityka
  - „krytyk” — funkcja wartości

Parametryzowanie polityki

W metodzie gradientu polityki, polityka $\pi(a|s,\theta)$ może być parametryzowana dowolnie ale:

- musi być różniczkowalna (czyli istnieje $\nabla\pi(a|s,\theta)$ oraz ma zawsze skończone wartości)
- nie powinna być deterministyczna, tzn. $0 < \pi(a|s,\theta) < 1$ (zapewnienie eksploracji)

Zaleta parametryzowania polityki:

- dla części problemów prostsza funkcja niż funkcja wartości akcji,
- ale nie dla wszystkich...

Funkcja preferencji akcji soft-max

- Dla dyskretnej (i niezbyt dużej) przestrzeni akcji często stosuje się funkcję preferencji $h(s, a, \theta) \in \mathbb{R}$
- Akcja z największą wartością funkcji preferencji dostaje najwyższe prawdopodobieństwo wyboru, np. zgodnie z funkcją **soft-max**:

$$\pi(a|s, \theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}$$

- Funkcja preferencji $h(s, a, \theta)$ może być parametryzowana dowolnie, np.:
  - (głębokie) sieci neuronowe (θ – wektor wszystkich wag sieci)
  - funkcja liniowa:

$$h(s, a, \theta) = \theta^T \mathbf{x}(s, a)$$

  $\mathbf{x}(s, a)$ — wektor cech zależny od **stanu** środowiska **i akcji**

Zalety soft-max

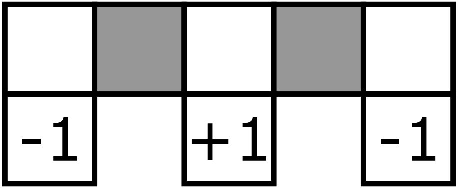
Zalety parametryzowania polityki funkcją preferencji akcji soft-max:

- polityka może dążyć do polityki deterministycznej (gdy taka jest właśnie optymalna)
- umożliwia wybór akcji z dowolnymi prawdopodobieństwami (gdy optymalna polityka jest stochastyczna, np.:
  - blefowanie w pokerze,
  - kamień-papier-nożyce

Przykład: kamień-papier-nożyce

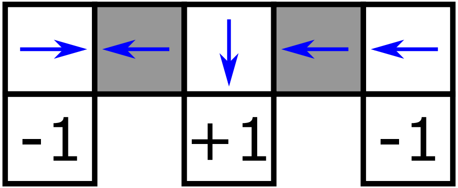
- Gra dla 2 osób:
  - kamień wygrywa z nożycami
  - papier wygrywa z kamieniem
  - nożyce wygrywają z papierem
- Wersja iteracyjna — gramy wielokrotnie:
  - polityka deterministyczna jest łatwa do pokonania
  - optymalna jest polityka losowa (rozkład jednorodny)

Przykład: Aliased Gridworld



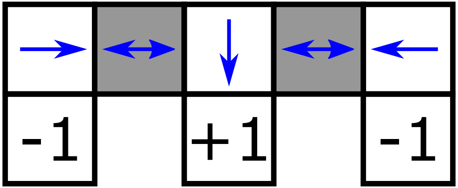
- Cechy opisujące stan: czy ściana w kierunkach N, E, S, W

Przykład: Aliased Gridworld



- Szare pola (stany) są nierozróżnialne
- Polityka deterministyczna:
  - albo kierunek W na obu polach
  - albo kierunek E na obu polach
- W obu przypadkach agent utyka i nie zdobywa nagrody
- Agent z funkcją wartości uczy się polityki prawie deterministycznej:
  - np. zachłanna lub ϵ-zachłanna
- dlatego krąży w labiryncie bardzo długo

Przykład: Aliased Gridworld



- Optymalna polityka — ruch losowy w szarym stanie:
  - $\pi_\theta(\acute{s}ciana\,NS, id\acute{z}\,E) = 0.5$
  - $\pi_\theta(\acute{s}ciana\,NS, id\acute{z}\,W) = 0.5$
- Z dużym prawdopodobieństwem cel osiągnięty w kilku krokach
- Agent z polityką może nauczyć się optymalnej stochastycznej polityki

Porównanie

Porównanie **parametryzowania polityki** funkcją preferencji akcji soft-max:

- metoda ϵ-zachłanna:
  zawsze pewne prawdopodobieństwo ϵ wyboru losowej akcji (vs. zbieganie do polityki deterministycznej)
- soft-max'owy rozkład bazujący na wartości akcji:
  estymaty wartości akcji zbiega do prawdziwej wartości $q(s, a)$ co przekłada się po zastosowaniu soft-max na konkretne (różne od 0 i 1) prawdopodobieństwa (vs. zbieganie do polityki deterministycznej)
- metoda ϵ-zachłanna:
  nagłe zmiany akcji dla dowolnie małych zmian estymaty $q(s, a)$ jeśli (vs. łagodne zmiany prawdopodobieństwa akcji)

Twierdzenie o gradiencie polityki

Dla uproszczenia notacji założymy:

- każdy epizod rozpoczyna się w tym samym stanie s_0
- współczynnik dyskontowy $\gamma = 1$

Dla problemów epizodycznych ocenę polityki zdefiniujemy jako

$$J(\theta) = v_{\pi_\theta}(s_0)$$

Problem: ocena zależy od wyboru akcji oraz rozkładu stanów.

- wpływ polityki na rozkład stanów jest zależny od środowiska (zazwyczaj nam nieznany...)

Twierdzenie o gradiencie polityki

Ratunek — twierdzenie o gradiencie polityki (problem epizodyczny):

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\theta)$$

gdzie μ to rozkład stanów przy stosowaniu polityki π.

- nie wiąże się z różniczkowaniem rozkładu stanów

Stochastyczny wzrost gradientu

Schemat metody gradientowej wymaga:

- zdobywanie takich próbek doświadczenia (wynikających z interakcji ze środowiskiem)
- aby wartość oczekiwana była **proporcjonalna** do faktycznego gradientu miary oceny polityki $J(\theta)$
- wystarczy proporcjonalność — mamy arbitralną stałą α (rozmiar kroku / prędkość uczenia)
- twierdzenie o gradiencie polityki — daje dokładnie wyrażenie które spełnia ten warunek

Potrzebujemy tylko metody próbkowania której wartość oczekiwana będzie równa prawej stronie twierdzenia.

Stochastyczny wzrost gradientu

- Twierdzenie o gradiencie polityki

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\theta)$$

- suma po stanach ważona częstością występowania danego stanu przy stosowaniu polityki π

$$\begin{aligned} \nabla J(\theta) &\propto \textstyle\sum_s \mu(s) \left(\sum_a q_\pi(s,a) \nabla \pi(a|s,\theta)\right) \\ &= \mathbb{E}_\pi \left[\textstyle\sum_a q_\pi(S_t,a) \nabla \pi(a|S_t,\theta)\right] \end{aligned}$$

- metoda stochastycznego wzrostu gradientu (po wszystkich akcjach, ang. *all-actions method*):

$$\theta_{t+1} = \theta_t + \alpha \sum_a \hat{q}(S_t, a, \boldsymbol{w}) \nabla \pi(a|S_t, \boldsymbol{\theta})$$

gdzie $\hat{q}(S_t, a, \boldsymbol{w})$ — uczony aproksymator q_π

Plan wykładu

1. Uczenie polityki

2. REINFORCE

3. Actor-Critic

REINFORCE (Willams, 1992)

- zamiast sumy po akcjach wprowadzamy próbkowanie

$$
\begin{aligned}
\nabla J(\theta) &\propto \mathbb{E}_\pi\left[\sum_a q_\pi(S_t, a)\nabla\pi(a|S_t,\theta)\right] \\
&= \mathbb{E}_\pi\left[\sum_a \pi(a|S_t,\theta)\left(q_\pi(S_t, a)\frac{\nabla\pi(a|S_t,\theta)}{\pi(a|S_t,\theta)}\right)\right] \\
&= \mathbb{E}_\pi\left[q_\pi(S_t, A_t)\frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}\right] \\
&= \mathbb{E}_\pi\left[G_t\frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}\right]
\end{aligned}
$$

- G_t — oczekiwany zysk: $\mathbb{E}_\pi\left[G_t|S_t, A_t\right] = q_\pi(S_t, A_t)$

REINFORCE — aktualizacja wag

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}$$

- zmiany parametrów w kierunku zwiększenia prawdopodobieństwa wyboru akcji A_t
- proporcjonalnie do zysku — większy ruch w kierunku akcji które zapewniają większy zysk
- odwrotnie proporcjonalnie do prawdopodobieństwa danej akcji — aby częste akcje nie zdominowały aktualizacji wag
- metoda Monte Carlo — zysk znamy po zakończeniu epizodu

REINFORCE — algorytm

Wejście: różniczkowalna, parametryzowana polityka $\pi(a|s,\theta)$

1. Zainicjalizuj parametry polityki (np. $\theta = \mathbf{0}$)
2. Powtarzaj dla każdego epizodu:
   - wygeneruj epizod zgodnie z polityką π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
   - powtarzaj dla $t = 0, 1, 2, \ldots, T-1$:
     $G = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
     $\theta = \theta + \alpha\gamma^t G \nabla \ln \pi(A_t|S_t,\theta)$

Uwagi:

- $\nabla \ln x = \frac{\nabla x}{x}$
- $\nabla \ln \pi(A_t|S_t,\theta)$ — różne nazwy: *eligibility vector*, *score function*
- uwzględnia współczynnik dyskontowy γ

Podsumowanie

REINFORCE ma dobre teoretyczne właściwości dot. zbieżności:

- spodziewana aktualizacja wag w kierunku gradientu oceny polityki
- zapewnia poprawę dla odpowiednio małego α
- zbiega do lokalnego optimum (przy odpowiednio malejącym α)

Niestety (jak wszystkie metody MC) może mieć dużą wariancję i w konsekwencji wolno się uczyć.

Wartość referencyjna (ang. baseline)

Uogólnione twierdzenie o gradiencie polityki:

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a (q_\pi(s,a) - b(s)) \nabla \pi(a|s,\theta)$$

$b(s)$ — wartość referencyjna, dowolna funkcja niezależna od akcji a (nawet zmienna losowa):

$$\textstyle\sum_a b(s) \nabla \pi(a|s,\theta) = b(s) \nabla \sum_a \pi(a|s,\theta) = b(s) \nabla 1 = 0$$

- wartość referencyjna nie wpływa na wartość oczekiwaną, ale
- ma duży wpływ na wariancję
  - stan z dużą wartością dla wszystkich akcji — $b(s)$ powinno być duże aby rozróżnić trochę lepsze od trochę gorszych (nadal wysoko ocenianych) akcji
  - stan z małymi wartościami akcji — $b(s)$ powinno być niskie

REINFORCE z wartością referencyjną

- $b(s) = \hat{v}(S_t, \mathbf{w})$ — również uczona metodą MC

Wejście: różniczkowalna, parametryzowana polityka $\pi(a|s, \theta)$ oraz funkcja wartości stanu $\hat{v}(s, \mathbf{w})$

1. Zainicjalizuj parametry polityki i funkcji wartości (np. $\theta = \mathbf{w} = \mathbf{0}$)
2. Powtarzaj dla każdego epizodu:
   - wygeneruj epizod zgodnie z polityką π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
   - powtarzaj dla $t = 0, 1, 2, \ldots, T-1$:
     $G = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
     $\delta = G - \hat{v}(S_t, \mathbf{w})$
     $\mathbf{w} = \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S_t, \mathbf{w})$
     $\theta = \theta + \alpha^{\theta} \gamma^t \delta \nabla \ln \pi(A_t|S_t, \theta)$

Plan wykładu

1. Uczenie polityki
2. REINFORCE
3. Actor-Critic

Actor-Critic

- Cel: redukcja wariancji gradientu w porównaniu z REINFORCE (zwanego również w literaturze: ang. vanilia policy gradient) — bardzo duża różnice pomiędzy epizodami (np. raz nagroda 75pkt, a raz 1350pkt)
- Metody typu Actor-Critic składają się z dwóch komponentów (modeli) mogących opcjonalnie współdzielić parametry:
  - krytyk: aktualizuje parametry $\mathbf{w}$ funkcji wartości stanu lub akcji (zależnie od algorytmu)
  - aktor: aktualizuje parametry θ w kierunku wskazanym przez krytyka

Actor-Critic

- REINFORCE (metoda MC):

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}$$

- Actor-Critic (metoda TD):

$$\begin{aligned} \theta_{t+1} &= \theta_t + \alpha \left(R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}) - \hat{v}(S_t, \mathbf{w}) \right) \tfrac{\nabla \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \\ &= \theta_t + \alpha \delta_t \tfrac{\nabla \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \end{aligned}$$

- uczenie funkcji wartości stanu $\hat{v}(S_t, \mathbf{w})$ przy pomocy TD(0)

Warianty Actor-Critic

- $\theta_{t+1} = \theta_t + \alpha \nabla \ln \pi(A_t|S_t, \theta_t) G_t$ — REINFORCE

- $\theta_{t+1} = \theta_t + \alpha \nabla \ln \pi(A_t|S_t, \theta_t) \delta_t$ — TD Actor-Critic
- $\theta_{t+1} = \theta_t + \alpha \nabla \ln \pi(A_t|S_t, \theta_t) \hat{q}(s, a)$ — Q Actor-Critic
- $\theta_{t+1} = \theta_t + \alpha \nabla \ln \pi(A_t|S_t, \theta_t) A(s, a)$ — Advantage Actor-Critic
  gdzie funkcja przewagi:
  $A(s, a) = \hat{q}(s, a) - \hat{v}(s)$

TD Actor-Critic — algorytm

Wejście: różniczkowalna, parametryzowana polityka $\pi(a|s,\theta)$ oraz funkcja wartości stanu $\hat{v}(s,\mathbf{w})$

1. Zainicjalizuj parametry polityki i funkcji wartości (np. $\theta = \mathbf{w} = \mathbf{0}$)
2. Powtarzaj dla każdego epizodu:
   Zainicjalizuj S (pierwszy stan)
   $I = 1$
   Powtarzaj dopóki S nie jest stanem terminalnym:
   1. Wybierz akcję A zgodnie z π
   2. Wykonaj akcję A i zaobserwuj S' oraz R
   3. $\delta = R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$
   4. $\mathbf{w} = \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S,\mathbf{w})$
   5. $\theta = \theta + \alpha^{\theta} I\delta\nabla\ln\pi(A|S,\theta)$
   6. $I = \gamma I$; $S = S'$

Q Actor-Critic — algorytm

Wejście: różniczkowalna, parametryzowana polityka $\pi(a|s,\theta)$ oraz funkcja wartości akcji $\hat{q}(s,a,\mathbf{w})$

1. Zainicjalizuj parametry polityki i funkcji wartości (np. $\theta = \mathbf{w} = \mathbf{0}$)
2. Powtarzaj dla każdego epizodu:
   Zainicjalizuj S (pierwszy stan)
   Wybierz akcję A ze stanu S używając polityki π
   Dla każdego kroku danego epizodu:
   1. wykonaj akcję A, zaobserwuj nagrodę R i kolejny stan S'
   2. wybierz akcję A' ze stanu S' zgodnie z π
   3. $\theta = \theta + \alpha^{\theta}\hat{q}(S,A,\mathbf{w})\nabla \ln \pi(A|S,\theta)$
   4. $\delta = R + \gamma\hat{q}(S',A',\mathbf{w}) - \hat{q}(S,A,\mathbf{w})$
   5. $\mathbf{w} = \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{q}(S,A,\mathbf{w})$
   6. $A = A'$; $S = S'$

Funkcja przewagi

- Zmniejsza wariancję w porównaniu z funkcją wartości akcji (poprzez odjęcie wartości referencyjnej będącej funkcją wartości stanu)
  $A(s, a) = \hat{q}(s, a) - \hat{v}(s)$
- Krytyk może estymować $A(s, a)$ estymując zarówno funkcję wartości stanu jak i funkcję wartości akcji

$$\begin{aligned} \hat{v}(s) &\approx V(s) \\ \hat{q}(s, a) &\approx Q(s, a) \\ A(s, a) &= \hat{v}(s) - \hat{q}(s, a) \end{aligned}$$

  aktualizując obie funkcję zgodnie z TD (dwa zestawy parametrów!)

Estymacja funkcja przewagi

- Dla rzeczywistej funkcji użyteczności $V(s)$ błąd TD wynosi:

$$\delta = r + \gamma V(s') - V(s)$$

- błąd ten jest nieobciążoną estymatą funkcji przewagi!

$$\begin{aligned}
\mathbb{E}_\pi[\delta] &= \mathbb{E}_\pi[r + \gamma V(s') - V(s)] \\
&= \mathbb{E}_\pi[r + \gamma V(s')] - \mathbb{E}_\pi[V(s)] \\
&= Q(s,a) - V(s) \\
&= A(s,a)
\end{aligned}$$

- stąd możemy użyć błędu TD do obliczenia gradientu polityki:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \ln \pi_\theta(s,a) \delta^{\pi_\theta}]$$

- W praktyce stosuje się przybliżony błąd TD:

$$\delta_v = r + \gamma V(s') - V(s)$$

(jeden zestaw parametrów)

A3C, A2C

- Advantage Actor-Critic ma kilka wariantów, najpopularniejsze to A3C i A2C
- Asynchronous Advantage Actor-Critic (A3C)
  - metoda zaprojektowana na zrównoleglone uczenie:
  - wiele wątków uczy się równocześnie (wiele agentów+wiele środowisk)
  - każdy wątek akumuluje gradienty parametrów
  - asynchroniczna synchronizacja lokalnych parametrów z globalnymi (aktualizując globalne parametry według zakumulowanych gradientów)
- Advantage Actor-Critic (A2C)
  - synchroniczna wersja A3C
  - wszystkie wątki rozpoczynając prace mają tę samą politykę

A3C vs A2C

A3C vs A2C

Soft Actor-Critic (SAC)

- Wykorzystuje miarę entropii polityki do zwiększenia eksploracji
- Trzy główne komponenty SAC:
  - architektura actor-critic
  - off-policy – wykorzystanie wcześniej zebranego doświadczenia
  - maksymalizacja entropii (zwiększa stabilność oraz eksplorację)
- Cel:

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{\pi}[r(s_t, a_t) + \alpha H(\pi_\theta(\cdot|s_t))]$$

  gdzie H to miara entropii
- Maksymalizacja przetargu między spodziewanym zyskiem a losowością polityki

Bibliografia

[1] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, third edition.

[2] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

"Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)",
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20