Zaawansowane Metody Inteligencji Obliczeniowej

Lab 11: Ciągła Przestrzeń Akcji

Michał Kempka Marek Wydmuch

20 maja 2021

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

1 Actor-Critic dla ciągłej przestrzeni akcji

Algorytm 1: Pseudokod dla algorytmu One-step Actor-Critic

Inicjalizacja:
Różniczkowalna polityka $\pi(a|s,\theta)$
Różniczkowalna funkcja wartości stanu $v(s,\boldsymbol{w})$
Zainicjalizowane wagi $\boldsymbol{w} \in \mathbb{R}^d_{\boldsymbol{w}}$ i $\boldsymbol{\theta} \in \mathbb{R}^d_{\theta}$
Zadana szybkość uczenia $\alpha_{\boldsymbol{\theta}}, \alpha_{\boldsymbol{w}} \in \mathcal{R}^+$
repeat
    if S *nieustawiony lub terminalny* **then**
        Rozpocznij nowy epizod
        $S \leftarrow S_0$
    $A \sim \pi(\cdot|S,\theta)$
    Wykonaj akcję A, zaobserwuj nagrodę R i następnik S'
    $\delta \leftarrow R + \gamma v(S',\boldsymbol{w}) - v(S,\boldsymbol{w})$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_{\boldsymbol{w}} \delta \nabla v(S,\boldsymbol{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \delta \nabla \ln \pi(A|S,\boldsymbol{\theta})$
    $S \leftarrow S'$
until *warunek stopu*;

Na poprzednich zajęciach poznaliśmy metody gradientu polityki lub inaczej aproksymacji polityki, jedną z metod z tej rodziny jest algorytm Actor-Crtic (Algorytm 1). Metody te oferują prosty sposób na radzenie sobie z dużymi przestrzeniami akcji, włączając to przestrzenie ciągłe, gdzie liczba możliwych akcji jest nieskończona. W wypadku ciągłej przestrzeni stanów zamiast uczyć się prawdopodobieństw dla każdej z akcji, możemy uczyć się parametrów rozkładu prawdopodobieństwa. Na przykład, przestrzenią akcji mogą być liczby rzeczywiste, a akcję będziemy wybierać z rozkładu normalnego. Funkcja gęstości rozkładu normalnego jest zdefiniowana w następujący sposób:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)},$$

gdzie μ i σ są średnią i odchyleniem standardowym rozkładu normalnego. Sparametryzowana polityka oparta na funkcji gęstości rozkładu normalnego jest więc następująca:

$$\pi(a|s,\boldsymbol{\theta}) = \frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}} e^{\left(-\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2}\right)},$$

gdzie $\mu : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}, \sigma : \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}^+$. Dlatego podzielimy nasz wektor parametrów na dwie części, każda odpowiadająca za parametryzację jedynie jednej z tych funkcji: $\boldsymbol{\theta} = [\boldsymbol{\theta}^\mu, \boldsymbol{\theta}^\sigma]$.
Pytanie: Reguła aktualizacji dla w algorytmie Actor-Critic jest następująca:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\delta_t \frac{\nabla\pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \tag{1}$$

$$= \boldsymbol{\theta}_t + \alpha\delta_t \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}). \tag{2}$$

Wyznacz regułę aktualizacji dla polityki opartej o sparametryzowaną funkcję gęstości rozkładu normalnego.

2 Deep Deterministic Policy Gradient (DDPG)

Algorytm 2: Pseudokod dla algorytmu Deep Deterministic Policy Gradient

Inicjalizacja:
Różniczkowalne parametryczne funkcje rzeczywiste (sieci) $Q(s, a|\boldsymbol{\theta}^Q)$ i $\mu(s|\boldsymbol{\theta}^\mu)$
Powtórzone parametry dla 'target network': $\boldsymbol{\theta}^{Q'} \leftarrow \boldsymbol{\theta}^Q, \theta^{\mu'} \leftarrow \boldsymbol{\theta}^\mu$ Zainicjalizuj pamięć $\mathcal{M} = \emptyset$
Parametr I określające co ile kroków uczymy się z pamięci
Parametr J określający ile doświadczeń wybieramy z pamięci
Parametr $\tau \in (0, 1)$ odpowiadający za szybkość aktualizacji target network

steps $\leftarrow 0$
repeat
    steps $\leftarrow$ steps $+1$
    if S *nieustawiony lub terminalny* **then**
        Rozpocznij nowy epizod
        $S \leftarrow S_0$
    Wybierz A według dowolnej polityki (np. zgodnej z μ poszerzonej o losowość)
    Wykonaj akcję A, zaobserwuj nagrodę R i następnik S'
    $\mathcal{M} \leftarrow \mathcal{M} \cup \langle S, A, R, S' \rangle$
    $S \leftarrow S'$
    if *steps* $\%I = 0$ **then**
        $g \leftarrow 0$
        for $j \leftarrow 1, \ldots, J$ **do**
            $\langle S, A, R, S' \rangle \leftarrow$ losowo wybrane z $\mathcal{M}$
            $g^Q \leftarrow g^Q + \frac{1}{2}\nabla_{\boldsymbol{\theta}^Q}\left(Q(S, A|\boldsymbol{\theta}^Q) - (R + \gamma Q'(S', \mu'(S'|\boldsymbol{\theta}^{\mu'})|\boldsymbol{\theta}^{Q'}))\right)^2$
            $g^\mu \leftarrow g^\mu + \nabla_{\boldsymbol{\theta}^\mu} Q(S, \mu(s|\boldsymbol{\theta}^\mu)|\boldsymbol{\theta}^Q)$
        $\boldsymbol{\theta}^Q \leftarrow \boldsymbol{\theta}^Q - \frac{1}{J}\alpha g^Q$
        $\boldsymbol{\theta}^\mu \leftarrow \boldsymbol{\theta}^\mu + \frac{1}{J}\alpha g^\mu$
    $\boldsymbol{\theta}^{Q'} \leftarrow \tau\boldsymbol{\theta}^Q + (1 - \tau)\boldsymbol{\theta}^{Q'}$
    $\boldsymbol{\theta}^{\mu'} \leftarrow \tau\boldsymbol{\theta}^\mu + (1 - \tau)\boldsymbol{\theta}^{\mu'}$
until *warunek stopu*;

Deep Deterministic Policy Gradient (DDPG) [1] to podejście, które łączy metodę Actor-Critic z metodą Deep Q-Learning. Pozwala nam użyć ciągłych akcji, jest **off-policy**, **model-free** i w naturalny sposób może korzystać ze wszelkich (większości?) rozszerzeń metody DQN. W praktyce możemy nawet zapomnieć o fakcie powiązania DDPG z metodami z rodziny policy gradient i organicznie wymyślić ją wychodząc od DQN.

2.1 Ciągłe akcje w DQN

W normalnym DQN, wejście dla sieci jest stanem, a wyjście wektorem Q długości przestrzeni akcji. W przypadku DDPG jest to niemożliwe ze względu na fakt, że zbiór akcji jest niepoliczalny (lub prawdopodobnie zaporowo duży w zdyskretyzowaniu). Możemy więc użyć zatem naszej ciągłej akcji (wektora liczb rzeczywistych) jako części **wejścia** do sieci. Zamiast $Q_{DQN} : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|}$ uzyskujemy $Q_{DDPG} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

Niestety przy takiej formulacji pojawia się pewien problem: jak wybrać akcję maksymalizującą Q? Stwórzmy funkcję $\mu(S) : \mathcal{S} \to \mathbb{R}^d$ (gdzie d to wymiarowość akcji), która zwraca nam **najlepszą akcję** dla danego stanu (preferencyjnie wyrocznia). Możemy teraz użyć $\mu(S')$ w klasycznej aktualizacji Q-learningu pozbywając się maxa: $\max_{a' \in \mathcal{A}} Q(S', a') \rightarrow Q(S', \mu(S'|\boldsymbol{\theta}^{\mu}))$.

Założenie, że μ jest wyrocznią nie jest niestety bardzo pomocne w praktyce więc określmy, że μ będzie funkcją parametryczną z parametrami $\boldsymbol{\theta}^{\mu}$. Z uwagi na fakt, że **zwracanie najlepszej akcji** jest tożsame z **maksymalizacją Q** możemy sformułować podproblem optymalizacyjny maksymalizacji Q ze względu na $\boldsymbol{\theta}^{\mu}$ (średnio dla wszystkich możliwych stanów):

$$\max_{\boldsymbol{\theta}^{\mu}} \mathbb{E}_{s \in \mathcal{S}}[Q(s, \mu(s|\boldsymbol{\theta}^{\mu})] \tag{3}$$

Powyższy podproblem możemy rozwiązać (rozwiązywać) na różne sposoby, Ale naturalnym, prostym i przyjemnym rozwiązaniem jest użycie SGD z przeciwnym znakiem - **gradient ascent** i przeprowadzanie optymalizacji równolegle z klasyczną aktualizacją funkcji Q. Zakładając, że funkcja Q jest także parametryzowana, powiedzmy przez $\boldsymbol{\theta}^{Q}$, przy napotkaniu pojedynczej krotki $\{S, A, S', R\}$możemy zastosować dwie do pewnego stopnia niezależne aktualizacje:

$$\boldsymbol{\theta}^{Q} \leftarrow \boldsymbol{\theta}^{Q} - \alpha \frac{1}{2} \nabla_{\boldsymbol{\theta}^{Q}} \Big(Q(S, A|\boldsymbol{\theta}^{Q}) - (R + \gamma Q(S', \mu(S'|\boldsymbol{\theta}^{\mu})|\boldsymbol{\theta}^{Q})) \Big)^{2} \tag{4}$$

$$\boldsymbol{\theta}^{\mu} \leftarrow \boldsymbol{\theta}^{\mu} + \alpha \nabla_{\boldsymbol{\theta}^{\mu}} Q(S, \mu(S|\boldsymbol{\theta}^{\mu})|\boldsymbol{\theta}^{Q}) \tag{5}$$

Pseudokod 2 prezentuje ideę DDPG bardziej szczegółowo i z użyciem standardowych ulepszeń z DQN (experience replay, target network)

2.2 Architektura sieci i parametry

Jak widać z równań 4 mamy dwa zestawy parametrów, które mogą być użyte do zaimplementowania dwóch osobnych sieci, które będą aktualizowane niezależnie. Bardzo popularne jednak jest, by część sieci **odpowiedzialna za przetwarzanie stanów** była współdzielona.

2.3 Nawiązanie do Actor-Critic

Jak wspomniano wcześniej, DDPG możemy traktować jako metodę z rodziny Actor-Critic gdzie krytyk uczy się $Q(S, a)$, a aktor polityki $\mu(s)$.

2.4 Eksploracja

Z uwagi na fakt, że DDPG, podobnie jak Q-learning jest metodą off-policy, nasza polityka może być teoretycznie dowolna. Typowym jednak jest by stosować eksplorację poprzez wprowadzenie szumu Gaussowskiego (rozkład normalny) skupionego na 0 z progresywnie malejącą wariancją. Można jednak zastosować inny model eksploracji bez większych teoretycznych reperkusji.

2.5 Powolna aktualizacja sieci

Jak widać w pseudokodzie 2 (ostatnie linijki) w DDPG mamy podobnie jak w DQN do czynienia z 'zamrożonymi' sieciami (target network), które w tym przypadku są aktualizowane na bieżąco, lecz tylko częściowo. Nie jest to kluczowa cześć algorytmu, lecz warto zwrócić uwagę na istnienie takiej alternatywy.

2.6 Rozkłady akcji

Do tej pory, mówiąc o akcjach ciągłych, domniemywaliśmy, że chodzi o wektroy liczb rzeczywistych. W ogólności nie musi tak być (chociaż jest to raczej rzadziej poruszany temat). Na akcje mogą być narzucone najróżniejsze ograniczenia np, mają być dodatnie, z pewnego zakresu $[a, b]$, tylko liczby naturalne. W tym przypadku należy użyć odpowiednich funkcji aktywacji (np. sigmoid, softplus) dla wyjść sieci i rozkładów prawdopodobieństw z odpowiednimi funkcjami gęstości (np. ucięty rozkład normalny, rozkład Bernoulliego)

Literatura

[1] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In Bengio, Y. and LeCun, Y., editors, *ICLR*.

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

[3] 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