Autoreferat

1 Dane kontaktowe

Imię i nazwisko	Paweł Wawrzyński
Adres	ul. Rzemieślnicza 3, 05-090 Falenty Nowe
Email	P.Wawrzynski@elka.pw.edu.pl
Strona WWW	http://staff.elka.pw.edu.pl/∼pwawrzyn/

2 Posiadane dyplomy i stopnie naukowe

- Doktor nauk technicznych w zakresie informatyki. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych. Tytuł rozprawy: „Intensive Reinforcement Learning" (2005). Promotor: prof. nzw. dr hab. inż. Andrzej Pacut.
- Magister ekonomii. Uniwersytet Warszawski, Wydział Nauk Ekonomicznych. Tytuł pracy: „Szoki i instytucje a średniookresowe dostosowania na rynku pracy" (2004). Promotor: prof. UW dr hab. Mieczysław Socha.
- Magister inżynier w zakresie informatyki. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych. Tytuł pracy: „Wyznaczanie strategii optymalnej. Połączenie metod klasycznych i metod wywodzących się ze sztucznej inteligencji" (2001). Promotor: dr inż. Paweł Cichosz.

3 Informacja o zatrudnieniu

Od 2006 roku adiunkt w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej.

4 Wykaz publikacji będących podstawą wniosku habilitacyjnego

Poniższe publikacje wchodzą w skład osiągnięcia, o którym mowa w art. 16 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki. Przy każdej pozycji podany jest Impact Factor (IF) czasopisma (jeśli niezerowy) oraz liczba cytowań (jeśli niezerowa) według Web of Science. Pod każdą pozycją określony jest również indywidualny wkład wnioskodawcy w autorstwo publikacji. Pozostały dorobek wnioskodawcy przedstawiony jest w osobnych załącznikach.

[1] **P.Wawrzyński**, "Real-Time Reinforcement Learning by Sequential Actor-Critics and Experience Replay," Neural Networks 22, pp. 1484-1497, Elsevier, 2009.
IF=2, 11 cytowań
Wkład własny: 100%

[2] **P.Wawrzyński**, A.Pacut, "Balanced Importance Sampling Estimation," Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU), Paris, July 2-7, 2006, pp. 66–73.

Wkład własny: 70%, zaproponowanie formy estymatora będącego przedmiotem pracy, współudział w badaniu jego własności, przygotowanie eksperymentów numerycznych i współudział w opracowywaniu tekstu.

[3] **P.Wawrzyński**, A.K.Tanwani, „Autonomous Reinforcement Learning with Experience Replay," Neural Networks 41, pp. 156-167, Elsevier, 2013.
IF=2, 4 cytowania
Wkład własny: 70%, zaprojektowanie algorytmu i opis jego własności, opiniowanie eksperymentów obliczeniowych, współudział w przygotowaniu artykułu.

[4] **P.Wawrzyński**, B.Papis, "Fixed point method for autonomous on-line neural network training," Neurocomputing 74, pp. 2893-2905, Elsevier, 2011.
IF=1.4, 2 cytowania
Wkład własny: 70%, pomysł i zaprojektowanie algorytmu, dyskusja eksperymentów obliczeniowych, przygotowanie artykułu.

[5] **P.Wawrzyński**, "Fixed point method of step-size estimation for on-line neural network training," International Joint Conference on Neural Networks, Barcelona, Spain, 2010.
Wkład własny: 100%

[6] **P.Wawrzyński**, "Reinforcement Learning in Fine Time Discretization," Lecture Notes in Computer Science 4431, pp. 470-479, Springer-Verlag, 2007.
1 cytowanie
Wkład własny: 100%

[7] **P.Wawrzyński**, „Control policy with autocorrelated noise in reinforcement learning for robotics," International Journal of Machine Learning and Computing, Vol. 5, No. 2, pp. 91-95, IACSIT Press, 2014.
Wkład własny: 100%

[8] **P.Wawrzyński**, „Reinforcement Learning with Experience Replay for Model-Free Humanoid Walking Optimization," International Journal of Humanoid Robotics, World Scientific, Vol. 11, Vol. 11, No. 3, pp. 1450024, 2014.
IF=0.8.
Wkład własny: 100%

[9] **P.Wawrzyński**, "Autonomous Reinforcement Learning with Experience Replay for Humanoid Gait Optimization," Proceedings of the International Neural Network Society Winter Conference (INNS-WC2012), pp. 205-211, Procedia, 2012.
2 cytowania
Wkład własny: 100%

5 Opis cyklu publikacji będących podstawą wniosku habilitacyjnego „Wzbogacenie metodyki uczenia się przez wzmacnianie umożliwiające jej stosowanie w sterowaniu"

Prezentowane tu badania naukowe stanowią odpowiedź na następujące, dalekosiężne wyzwanie. W ramach współczesnego paradygmatu automatyki, urządzenia techniczne (w szczególności roboty) są sterowane przez systemy skonstruowane przez ludzi. Konstrukcje takie opierają się na

modelach dynamiki tych obiektów. Dlatego paradygmat ten jest ograniczony przez złożoność systemu, dla którego człowiek jest w stanie zbudować model dynamiki. To ograniczenie może być pokonane w ramach następującego, alternatywnego paradygmatu: Systemy sterowania poprawiają swoje działanie w jego trakcie, podejmując działania próbne, których konsekwencji nie są w stanie dokładnie przewidzieć.

Powyżej opisany alternatywny paradygmat jest rozwijany w dziedzinie pn. uczenie maszynowe, a w szczególności jej gałęzi pn. uczenie się przez wzmacnianie. Mimo trwającego kilka dekad postępu w tych obszarach, uczące się maszyny nie są rozpowszechnione. W ramach przedstawianych tu badań atakowane były trzy problemy stanowiące najistotniejszą przeszkodę:

– zbyt mała szybkość działania algorytmów uczenia się przez wzmacnianie,

– niedostateczna autonomia tych algorytmów, rozumiana jako niezależność od parametrów, które projektant wyznacza metodą prób-i-błędów,

– niemożność zastosowania tych algorytmów w warunkach gęstej dyskretyzacji czasu, typowej dla zastosowań w sterowaniu.

Metodyka uczenia się przez wzmacnianie rozbudowana o elementy stanowiące rozwiązanie powyższych problemów, stała się dostatecznie efektywna, aby można ją było zastosować do trudnego problemu robotycznego. W ramach przedstawionych tu prac została ona zastosowana do optymalizacji chodu robota humanoidalnego.

Uczenie się przez wzmacnianie

W dziedzinie uczenia się przez wzmacnianie [10] rozważane jest zagadnienie decydenta, który w dyskretnej chwili czasu $t = 1, 2, \ldots$ znajduje się w pewnym *stanie* $x_t \in X$ i na podstawie tego stanu podejmuje *decyzję* $u_t \in U$. Następnie, środowisko przenosi decydenta do następnego stanu x_{t+1} i wypłaca mu *nagrodę* $r_t \in R$, przy czym x_{t+1} jest losowane z rozkładu warunkowego $P_x(\cdot|x_t, u_t)$, zaś nagroda jest wartością funkcji $\bar{r}(x_{t+1}, u_t)$. Rodzina rozkładów P_x jak i funkcja $\bar{r}$ są nieznane. Interakcja decydenta ze środowiskiem może być podzielona na niezależne *epizody*, które są arbitralnie przerywane przez środowisko lub kończą się ponieważ decydent osiąga cel sterowania, albo wręcz przeciwnie – powoduje sytuację awaryjną. W takich wypadkach decydent jest, bez żadnej decyzji, przeniesiony do stanu wybranego losowo i rozpoczyna się nowy epizod.

W ogólności zadanie decydenta polega na tym, aby zoptymalizować reaktywną politykę decyzyjną, czyli *nauczyć się* dopasowywać decyzje do stanów w taki sposób, aby w każdej chwili spodziewać się możliwie najwyższej sumy zdyskontowanych nagród w przyszłości. Formalnie, polityka decyzyjna π to rozkład prawdopodobieństwa decyzji parametryzowany przez stany. Jej jakość określa się w dwóch etapach. *Funkcja wartości* $V^\pi(x)$ przypisuje stanowi x wartość oczekiwaną sumy zdyskontowanych, z dyskontem $\gamma \in (0, 1)$, nagród do zdobycia począwszy od stanu x, przy stosowaniu polityki π:

$$V^\pi(x) = E\left(\sum_{i\geq 0} \gamma^i r_{t+i} \middle| x_t = x, \pi \text{ określa } u_{t+i}\right) \tag{1}$$

Jakość polityki określa wskaźnik jakości będący w istocie wartością oczekiwaną funkcji wartości w rozkładzie stacjonarnym stanów, η^π, wynikającym z polityki π:

$$J(\pi) = \int_X V^\pi(x) \mathrm{d}\eta^\pi(x). \tag{2}$$

Celem uczenia się przez wzmacnianie jest maksymalizacja powyższego wskaźnika jakości ze względu na π, przy nieznajomości P_x ani $\bar{r}$, jedynie na podstawie danych z interakcji decydenta ze swoim otoczeniem. Efektywność algorytmów uczenia się ocenia się ze względu na tempo zbiegania do maksimum J w czasie tej interakcji.

Powyższy model w następujący sposób stosuje się w automatyce i robotyce. Decydent jest sterownikiem urządzenia (robota), decyzjami są sterowania, stanem decydenta jest zestaw informacji obejmujący stan urządzenia, jego otoczenia, a także wskazujących bieżący cel sterowania. Nagroda jest pewną miarą tego, w jak dużym stopniu jest w danej chwili realizowany cel sterowania. Zadaniem uczącego się sterownika jest nauczyć się tak sterować urządzeniem, aby w każdej chwili móc spodziewać się wysokich przyszłych nagród za realizację celu sterowania.

Szybkość algorytmów uczenia się przez wzmacnianie

Klasyczne algorytmy uczenia się ze wzmacnianiem, takie jak Q-Learning [11] i metody Aktor-Krytyk [12,13,14,15] działają w taki sposób, że w miarę jak postępuje proces decyzyjny (sterowania), kolejne stany, decyzje i nagrody są wykorzystywane do dokonywania poprawek parametrów polityki decyzyjnej, zatem próbka danych z procesu decyzyjnego jest przetworzona tylko raz. Taki sekwencyjny sposób przetwarzania informacji powoduje, że oparte o nie algorytmy są nieefektywne w tym sensie, że proces sterowania musi toczyć się bardzo długo, aby na jego podstawie polityka sterowania została zoptymalizowana.

W artykule [1] pokazałem, że każdy sekwencyjny algorytm uczenia się przez wzmacnianie, spełniający pewne typowe warunki, w tym metoda typu Aktor-Krytyk, może być przekonstruowany do następującego schematu: Dane z procesu decyzyjnego są zapamiętywane w bazie danych, stamtąd pobierane przez równoległy proces obliczeniowy, który traktuje je *niemal* tak, jakby opisywały wydarzenia, które właśnie miały miejsce i wykorzystuje je do dokonywania poprawek polityki decyzyjnej. Proces obliczeniowy może przetwarzać dane szybciej, niż one napływają, więc przetwarzać te same próbki wielokrotnie, co prowadzi do poprawy efektywności. W ramach tego podejścia rozwiązałem szereg nietrywialnych problemów, w szczególności taki, że dane z wcześniejszych etapów procesu decyzyjnego zostały wygenerowane przez inną niż obecna politykę decyzyjną, zatem bezpośrednio nie mogą posłużyć jako źródło wskazówek jak poprawić obecną politykę.

Algorytm zmodyfikowany wg zaproponowanego rozwiązania okazuje się nawet piećdziesięciokrotnie bardziej efektywny niż oryginał w tym sensie, że potrzebuje tyleżkrotnie krótszego procesu decyzyjnego do osiągnięcia polityki decyzyjnej tej samej jakości. Przez wiele kolejnych lat algorytm opisany w [1] był najszybszą metodą nieposługującą się modelem dynamiki obiektu, optymalizującą sterowania obiektu pn. *Cart-Pole Swing-Up*. Jest to wózek z luźno zwisającym wahadłem; celem sterowania jest tak ruszać wózkiem, aby podrzucić wahadło do góry i stabilizować je w pionie.

Artykuł [1] wprowadza koncepcję *obciętych* $TD(\lambda)$*–estymatorów*[1] przyszłych nagród, stanowiących ogólne rozwiązanie problemu estymacji kierunku poprawy bieżącej polityki decyzyjnej na podstawie danych o procesie sterowania zebranych wcześniej, kiedy stosowana była

[1] Mają one następującą postać:

$$\frac{\mathrm{d}\ln\pi(u_i;\theta)}{\mathrm{d}\theta}\sum_{k=0}^{K}(\gamma\lambda)^k\left(r_{i+j}+\gamma\bar{V}(x_{i+j+1};\upsilon)-\bar{V}(x_{i+j};\upsilon)\right)\min\left\{\prod_{j=0}^{k}\frac{\pi(u_{i+j};\theta)}{\pi_{i+j}},b\right\}. \tag{3}$$

inna polityka sterowania. Efektywność algorytmu uczenia się przedstawionego w pracy [1] została pokazana na symulowanym obiekcie pod nazwą pół-gepard, który specjalnie opracowałem na potrzeby tych badań. Pół-gepard jest płaskim łańcuchem kinematycznym o sześciu stopniach swobody, o kształcie tułowia z dwiema kończynami. Zadanie sterowania nim polega na tym, aby biegł jak najszybciej do przodu. W czasie, kiedy wydawany był artykuł [1], algorytmy w dziedzinie uczenia się przez wzmacnianie były testowane na dużo prostszych obiektach, takich jak wspomniany wyżej Cart-Pole Swing-Up. Pół-gepard był później stosowany przez innych autorów, np. w pracy [20] z roku 2016.

Uczenie się przez wzmacnianie opiera się na dokonywaniu losowań (decyzji), rejestrowaniu jakości wyniku (nagród i kolejnych stanów) i wyznaczaniu rozkładu, z którego będą dokonywane losowania w przyszłości (polityki decyzyjnej). Jednym z centralnych zagadnień tej dziedziny jest więc wnioskowanie o wartościach oczekiwanych funkcji losowej w danych rozkładach prawdopodobieństwa (optymalizowanej polityki), na podstawie losowań z rozkładów źródłowych (wcześniejszych polityk).

W artykule [2], napisanym wraz z prof. Andrzejem Pacutem, kierownikiem zespołu, w którym pracuję, ogólna postać wyżej opisanego problemu została zaatakowana bezpośrednio. Został zaproponowany tzw. estymator zbalansowany. Wykorzystuje on klasyczny zabieg, a mianowicie mnożenie wartości funkcji przez iloraz gęstości danego rozkładu prawdopodobieństwa i źródłowego. W artykule zaprezentowano nowy sposób ważenia tak preparowanych próbek, a mianowicie waga próbki jest proporcjonalna do pewnej miary odległości między rozkładami danym i źródłowym.[2] Ten sposób estymacji ma bardzo korzystną własność, a mianowicie tak zdefiniowany estymator jest asymptotycznie optymalny w sensie minimalizacji wariancji. Zastosowanie tego sposobu powoduje efektywne wykorzystywanie danych, prowadzi więc do szybkiego uczenia się przez wzmacnianie.

Autonomia algorytmów uczenia się

Szereg algorytmów uczenia się, w tym większość algorytmów uczenia się przez wzmacnianie, ma charakter metod aproksymacji stochastycznej [16]. Metody takie wymagają współczynnika zwanego parametrem kroku. Mimo wielu dekad badań, nie istnieją odporne metody, które wyznaczają ten współczynnik w trakcie działania metody. Może on być wyznaczony eksperymentalnie, ale to polega na kilkukrotnym przeprowadzeniu procesu uczenia się tylko po to, aby zaleźć najwłaściwszy współczynnik. W praktycznych zastosowaniach jest to nie do przyjęcia, gdyż zwykle uczenie dotyczy podatnego na uszkodzenia urządzenia.

Oto formalny opis tego problemu. Poszukujemy minimum ciągłej i różniczkowalnej funkcji $J : R^n \mapsto R$. Nie jest znany jej gradient, natomiast dany jest generator próbek $\xi_t, t = 1, 2, \ldots$ i taka funkcja $g(\theta, x)$, że wartość oczekiwana $Eg(\theta, \xi)$, w której losowe jest ξ, jest równa gradientowi $\nabla J(\theta)$. Klasyczną metodą znalezienia minimum J jest tzw. Procedura Robbinsa-Monro,

[2] Estymator ma postać

$$\widehat{\eta} = \frac{\sum_{i=1}^{N} r(u_i) \frac{\pi(u_i)}{\pi_i(u_i)} \kappa(\pi, \pi_i)}{\sum_{i=1}^{N} \kappa(\pi, \pi_i)}$$

gdzie

$$\kappa(\pi, \pi_i) = \int \left(\frac{\pi(u)}{\pi_i(u)} \right)^2 \pi_i(u) \mathrm{d}u.$$

która poprawia przybliżenie optymalnej wartości θ wg następujących iteracji:

$$\theta_{t+1} = \theta_t - \beta_t g(\theta_t, \xi_t), \quad t = 1, 2, \ldots \tag{4}$$

gdzie β_t to dodatni skalar będący wspomnianym wyżej parametrem kroku. Jakkolwiek istnieją teoretyczne warunki jakie parametry kroku powinny spełniać, to jednak są one w praktyce nieprzydatne. Uniwersalna i odporna metoda, która wyznaczałaby β_t w trakcie działania rekurencji (4) jest wciąż poszukiwana, choć problem jest znany od sześciu dziesięcioleci.

W pracy [5] zaproponowałem mechanizm wyznaczania parametru kroku oparty o następujące zasady:

– proces (4) jest podzielony na okresy,

– w każdym okresie, oprócz tego, że parametr θ jest aktualizowany na podstawie (4), jest także wyznaczana jego alternatywna wartość θ' na podstawie formuły

$$\theta'_{t+1} = \theta'_t - \beta_t g(\theta_{t_0}, \xi_t), \quad t = 1, 2, \ldots \tag{5}$$

przy czym t_0 to początkowa chwila okresu, oraz $\theta'_{t_0} = \theta_{t_0}$.

Powyższe zasady opierają się na następującej własności: zbyt duży parametr kroku powoduje szybko rosnący dystans między θ_t i θ'_t (krzywizna J ma dominujący wpływ na ruch θ_t). Dla zbyt małego parametru kroku, ten dystans pozostaje zbyt mały (krzywizna J ma nieistotny wpływ na ruch θ_t). Wartość β_t zapewniającą szybką zbieżność można osiągnąć stabilizując tempo oddalania się od siebie θ_t i θ'_t.

W artykule [4] zaproponowaliśmy oparty na powyższych zasadach algorytm uczenia się sieci neuronowej on-line, w którym każdy element wektora θ_t posługuje się niezależnym parametrem kroku. Algorytm ten został także wzbogacony o szereg wniosków z własności procesu (4). W rezultacie, okazał się on wyraźnie szybszy niż Rozszerzony Filtr Kalmana [17] zastosowany do tych samych problemów.

W artykule [3] zaproponowaliśmy wersję metody wyznaczającej parametr kroku dopasowana do działania z algorytmami typu Aktor-Krytyk wprowadzonymi wcześniej w [1]. Działanie takiego algorytm polega na jednoczesnym dopasowaniu do siebie dwóch struktur: polityki decyzyjnej oraz tzw. Krytyka, czyli aproksymatora funkcji wartości. Tworzy to sytuację, w której parametry polityki decyzyjnej poruszają się w stronę maksimum funkcji zmieniającej się wraz z Krytykiem, zaś parametry Krytyka poruszają się w stronę minimum funkcji zmieniającej się wraz z polityką decyzyjną. W tej sytuacji, zgodnie z własnościami przedstawionymi w [14], zbieżność Krytyka powinna być w granicy nieskończenie szybsza niż zbieżność polityki decyzyjnej. Z zasady tej czyni użytek algorytm zaproponowany w [3].

Algorytmy uczenia się przez wzmacnianie w warunkach gęstej dyskretyzacji czasu

Algorytm uczenia się przez wzmacnianie dokonuje wyboru decyzji posługując się stochastyczną polityką decyzyjną, czyli rozkładem prawdopodobieństwa decyzji parametryzowanym przez stan. Decyzja musi być losowana, aby algorytm uczenia się mógł rozpoznać jakość różnych decyzji i przypisać do wszystkich stanów te najlepsze.

Na ogół polityka decyzyjna ma następującą strukturę. Jej podstawą jest aproksymator (np. sieć neuronowa), na którego wejściu jest stan decydenta, a na wyjściu wartość oczekiwaną rozkładu prawdopodobieństwa, z którego losowana jest decyzja. Kolejne dwie decyzje są zależne jedynie poprzez stan, do którego doprowadziła pierwsza.

Z powyższych okoliczności wynikają następujące trudności w stosowaniu uczenia się przez wzmacnianie w automatyce i robotyce. Po pierwsze, typowy sterownik cyfrowy generuje sterowania z dużą częstotliwością. W rezultacie, każda z nich trwa krótko, a tym samym trudno jest z dużą dokładnością rozpoznać jej konsekwencje. Im gęstsza jest bowiem dyskretyzacja czasu, tym więcej jest pośrednich chwil czasowych między decyzją, a pojawieniem się jej efektów.

Druga z trudności polega na tym, że w realnych zastosowaniach sterowniki faktycznie wyznaczają pewne fizyczne wartości, takie jak napięcia, które nie mogą być diametralnie zmieniane między kolejnymi chwilami czasowymi. Tymczasem realizacja wyżej opisanej stochastycznej polityki decyzyjnej oznacza w większości przypadków dodanie do gładkiego przebiegu sterowań białego szumu. Im gęstsza dyskretyzacja czasu, tym bardziej jest to kłopotliwe techniczne i na większe uszkodzenia wystawia sprzęt sterowany w taki sposób.

Do czasu mojego artykułu [6] zagadnienie gęstej dyskretyzacji czasu sterowania właściwie nie było podejmowane w literaturze uczenia się przez wzmacnianie w postaci dyskretnej, odpowiadającej realiom. Istniała natomiast ścieżka badań nad uczeniem się przez wzmacnianie w czasie ciągłym [18], która jednak była bardzo odległa od zastosowań.

W artykule [6] przedstawiłem koncepcję stochastycznej polityki decyzyjnej, w której czas jest podzielony na okresy. W ich trakcie między decyzjami występuje odpowiednio zaprojektowana zależność stochastyczna (poza zależnością wynikającą z dynamiki środowiska, polegającą na tym, że decyzja wynika ze stanu, a ten jest skutkiem poprzedniej decyzji). W artykule zostało pokazane, że taka polityka decyzyjna może być optymalizowana przez szeroką klasę algorytmów ucznia się przez wzmacnianie opartych na gradiencie logarytmu gęstości decyzji. Ta klasa obejmuje m.in. metody typu Aktor-Krytyk. W takim zastosowaniu decyzje z pojedynczego okresu stanowią łączą się w makro-decyzję, zaś celem algorytmu uczącego staje się manipulowanie gęstością prawdopodobieństw poszczególnych makro-decyzji (zwiększanie gęstości tych „dobrych” i zmniejszanie tych „złych”). Pomysł ten stanowi rozwiązanie problemu znikomego wpływu pojedynczej decyzji na proces sterowania. Dla dowolnie gęstej dyskretyzacji czasu można dobrać wystarczającą długość okresu, aby makro-decyzja wywierała istotny wpływ na ten proces.

W artykule [7] zbadałem własności polityki decyzyjnej skonstruowanej w następujący sposób: Decyzja jest sumą dwóch składników. Pierwszy z nich jest wyznaczany przez aproksymator (np. sieć neuronową), którego wejściem jest stan, zaś parametry są optymalizowane w procesie uczenia się. Drugim składnikiem jest autoskorelowany proces stochastyczny (np. średniej ruchomej lub autoregresyjny). Przy dowolnej dyskretyzacji czasu parametry tego procesu można dobrać w taki sposób, aby taka polityka była technicznie realizowalna, tzn. nie dotykał jej problem zbyt szybko zmieniających się sterowań. Taką politykę można w przybliżeniu optymalizować metodami typu Aktor-Krytyk. Z powodzeniem zastosowali ją autorzy pracy [20], gdzie jest ona optymalizowana przy użyciu wyspecjalizowanego algorytmu. W nieopublikowanym jeszcze materiale proponuję inny, jeszcze sprawniejszy algorytm optymalizujący taką politykę.

Sprawdzian: uczący się chodzić humanoid

Opracowane przeze mnie metody działały zadowalająco w symulacjach z nietrywialnymi problemami sterowania. Naturalnym etapem ich badań stała się konfrontacja ze złożonym robotem. Dlatego jeszcze pod koniec 2009 roku rozpocząłem projekt badawczy, którego celem było powstanie robota humanoidalnego ze sterownikiem zdolnym do prowadzenia obliczeń na dużych sieciach neuronowych; tym sterownikiem stał się mały, ale w pełni sprawny komputer osobisty

z systemem Linux. Kolejnymi celami tego projektu była budowa oprogramowania sterującego, implementacja odpowiednich metod uczenia się przez wzmacnianie oraz przeprowadzenie testowego procesu nauki chodzenia. Skala tego przedsięwzięcia była znaczna. O ile nie był to z pewnością pierwszy humanoidalny robot kroczący, to z pewnością był to jeden z pierwszych robotów uczących się chodzić, więc trudno było o jakiekolwiek wzorcowe badania tego rodzaju. W rezultacie liczba rozmaitych trudności koncepcyjnych i technicznych, które musiałem pokonać realizując ten projekt, była ogromna.

Pierwsze rezultaty projektu zostały opublikowane w artykule [9]. Początkowy, dosyć niezdarny, mechanizm sterowania robotem określa pozycje docelowe dla serwomechanizmów. Te pozycje są modyfikowane przez politykę decyzyjną optymalizowaną algorytmem uczenia sie przez wzmacnianie opisanym w [3]. Polityka decyzyjna ma postać sieci neuronowej, której wyjście jest modyfikowane przez szum normalny.

Artykuł [8] opisuje dojrzały rezultat projektu. Została tam zaprezentowana koncepcja polityki sterowania opartej o trzy elementy. Pierwszym z nich jest cykl w przestrzeni konfiguracji robota, którego pokonywanie składa się na początkowy, dosyć niezdarny chód. To pokonywanie polega na rzutowaniu w każdej chwili pozycji robota w przestrzeni konfiguracji na cykl i wyznaczaniu docelowej pozycji dla silników jako tej, która jest na cyklu po pewnym kwancie czasu za rzutem. Drugim elementem polityki jest sieć neuronowa, która wprowadza poprawki do docelowych pozycji silników. W końcu trzecim elementem jest proces stochastyczny, którego wartości również modyfikują docelowe pozycje silników. Uczenie sieci neuronowej jest celem działania algorytmu uczenia się przez wzmacnianie.

Nowy materiał, który się tam pojawił obejmuje m.in. nowe elementy w algorytmie uczącym oraz zredefiniowany opis stanu robota. Opis ten zawiera m.in. prędkość i pochylenie robota (wektor grawitacji w układzie współrzędnych związanym z tułowiem robota). Oba te wektory okazały się na tyle trudne do rekurencyjnej estymacji, że zagadnieniu temu poświęciłem osobny artykuł [19].

Film z robotem opisanym w artykule [8] jest w Internecie pod adresem [24], a także na płycie CD zawierającej niniejszy Autoreferat, w elektronicznym załączniku do tego artykułu.

6 Opis pozostałej działalności naukowo-badawczej

Głównym tematem mojej pracy naukowo-badawczej jest uczenie się przez wzmacnianie, zaś kluczowe osiągnięcia na tym polu zostały przedstawione w poprzednim rozdziale autoreferatu. Inne obszary mojej działalności naukowo-badawczej stanowią wsparcie dla głównego, albo zupełnie niezależnie uważam je za interesujące. Poniżej przedstawiam moje wybrane prace należące do obszarów innych niż główny. Te obszary to robotyka, impulsowe sieci neuronowe i sterowanie predykcyjne.

Robotyka

Próbując zastosować uczenie się przez wzmacnianie w robocie kroczącym, niespodziewanie natrafiłem na trudny problem: skąd zbudowany z tanich elementów robot ma pozyskiwać dokładną informację o swojej prędkości i pochyleniu? Silniki takiego robota mają niedokładne i działające z niewielką częstotliwością enkodery pozycji. Zatem estymacja prędkości robota kroczącego na podstawie ich odczytów i modelu kinematyki byłaby obarczona dużym błędem. Co więcej, nawet gdyby enkodery były bardzo dokładne, to pomiar taki nie uwzględniałby ślizgania się

robota na jego stopie. Na szczęście robot, którym posługiwałem się w badaniach był wyposażony był także w czujnik inercyjny, na który składa się żyroskop mierzący prędkość kątową i akcelerometr mierzący przyspieszenie.

Wraz ze współpracownikami z Wydziału Mechatroniki Politechniki Warszawskiej opracowałem mechanizm rekurencyjnej estymacji prędkości i pochylenia robota kroczącego, opisany w artykule [19]. Mechanizm ten oparty jest o Rozszerzony Filtr Kalmana, w którym:

- wejściem są odczyty żyroskopu i akcelerometru,
- obserwacją jest odczyt prędkości z modelu kinematyki uwzględniający rotację stopy robota, estymowaną na podstawie odczytu z żyroskopu (w jego korpusie),
- na estymowany stan robota składa się prędkość robota, wektor grawitacji w układzie współrzędnych związanym z czujnikiem inercyjnym (mówi o pochyleniu robota), oraz obciążenia żyroskopu i akcelerometru.

Estymaty stanu robota uzyskane opisywaną metodą są bardzo dokładne, nawet jeśli zastosowany sprzęt jest niskiej jakości. Jest to trudna do przecenienia zaleta tej metody, ponieważ można ją z powodzeniem stosować w niskonakładowych robotach.

Impulsowe sieci neuronowe

Człowiek zawdzięcza takie swoje predyspozycje jak percepcja, pamięć czy rozwiązywanie problemów sieciom neuronowym, z których jest zbudowany jego mózg. Badania nad sztucznymi sieciami neuronowymi były inspirowane nadzieją, że podobnymi predyspozycjami uda się dzięki nim obdarzyć programy komputerowe. W tej chwili wiadomo, że struktury takie jak perceptron wielowarstwowy nie oferują takich możliwości, natomiast nowe nadzieje są wiązane z najdokładniejszymi znanymi modelami naturalnych sieci neuronowych, czyli impulsowymi sieciami neuronowymi (ang. spiking neural networks). Prowadzone są intensywne prace badawcze nad mechanizmami uczenia się takich struktur.

W pracy [22] napisanej wraz z moim magistrantem zaprezentowałem koncepcję uczenia się impulsowych sieci neuronowych opartą o nakładanie autoskorelowanego szumu na wagi neuronów w populacji (zbiorze) sieci wykonujących to samo zadanie. Rejestrowana jest jakość działania różnych sieci i na podstawie ich porównania estymowany jest kierunek poprawy w przestrzeni wag. Rozwiązanie okazuje się efektywne, tzn. pozwala ono nauczyć sieć aproksymacji wieloargumentowej i wielowartościowej funkcji.

Sterowanie predykcyjne

Efektowną demonstracją naturalnej inteligencji jest planowanie działań. Opiera się ono na przeglądzie scenariuszy przyszłych zdarzeń i wyborze tego najlepszego. Taka sama metodyka w zastosowaniu do automatycznego sterowania obiektami nosi nazwę „sterowanie predykcyjne”.

W pracy [23] napisanej wraz z kolegami z Zakładu Sztucznej Inteligencji Instytutu Systemów Elektronicznych Politechniki Warszawskiej, zaproponowałem algorytm sterowania predykcyjnego, którego przeznaczeniem było sterowanie poczynaniami bota w grze komputerowej. Został on zaimplementowany w grze *Half-Life*. Algorytm ten oparty jest o losowy przegląd scenariuszy zdarzeń i obejmuje zbieranie statystyk umożliwiających lepsze sterowanie w przyszłości, zawiera więc element uczenia się.

7 Dodatkowa bibliografia

[10] R.S.Sutton, A.G.Barto. Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

[11] C.Watkins, P.Dayan. „Q-learning," Machine Learning, 8, 279–292, 1992.

[12] A.G.Barto, R.S.Sutton, C.W.Anderson, „Neuronlike adaptive elements that can learn difficult learning control problems," IEEE Transactions on Systems, Man, and Cybernetics, 13, 834–846, 1983.

[13] H.Kimura, S.Kobayashi. „An analysis of actor/critic algorithm using eligibility traces: Reinforcement learning with imperfect value functions," In Proceedings of the 15th international conference on machine learning, pp.278–286, 1998.

[14] V.Konda, J.Tsitsiklis. „Actor–critic algorithms," SIAM Journal on Control and Optimization, 42(4), 1143–1166, 2003.

[15] S.Bhatnagar, R.S.Sutton, M.Ghavamzadeh, M.Lee. „Natural actor–critic algorithms," Automatica, Vol.45, Issue 11, pp. 2471–2482, 2009.

[16] H.J.Kushner, G.Yin, Stochastic Approximation Algorithm and Applications, Springer-Verlag, New York, 1997.

[17] Y.Iiguni, H.Sakai, H.Tokumaru, „A real time learning lagorithm for a multilayered neural network based on extended Kalman filter," IEEE Transactions on Signal Processing, 45(6), pp. 959–966, 1992.

[18] R.Munos. „Policy Gradient in Continuous Time," Journal of Machine Learning Research 7, pp. 771–791, 2006.

[19] **P.Wawrzyński**, J.Mozaryn, J.Klimaszewski, „Robust Estimation of Walking Robots Velocity and Tilt Using Proprioceptive Sensors Data Fusion," Robotics and Autonomous Systems, Vol. 66, pp. 44–54, Elsevier, 2015.

[20] T.P.Lillicrap, J.J.Hunt, A.Pritzel, N.Heess, T.Erez, Y.Tassa, D.Silver, D.Wierstra, „Continuous control with deep reinforcement learning", International Conference on Learning Representations, 2016.

[21] M.P.Deisenroth, G.Neumann, J.Peters, "A Survey on Policy Search for Robotics," Foundations and Trends in Robotics 2 (1-2), 1–142, 2013.

[22] P.Suszynski, **P.Wawrzyński**, "Learning population of spiking neural networks with perturbation of conductances," Proceedings of Int. Joint Conference on Neural Networks, August 4-9, 2013, Dallas TX, USA, pp. 332–337, IEEE, 2013.

[23] **P.Wawrzyński**, J. Arabas, P. Cichosz, "Predictive Control for Artificial Intelligence in Computer Games," Lecture Notes in Artificial Intelligence 5097, pp. 1137–1148, Springer-Verlag, 2008.

[24] www.youtube.com/watch?v=O2rx4Bdwn24