WYBRANE ZAGADNIENIA SZTUCZNEJ INTELIGENCJI

Uczenie ze wzmocnieniem i nadzorowane w grach

SPRAWOZDANIE

Marta Maślankowska, 228003

Daniel Popek, 228030

SZCZEGÓŁY IMPLEMENTACJI

Implementacja Q-learningu

Gracz Q-learningowy uczy się (trenuje) na podstawie gier rozegranych przez dwóch graczy losowych (wybierających dowolne pole spośród możliwych). Informacja o wygranej lub przegranej propagowana jest wstecz po wszystkich stanach gry osobno dla gracza zaczynającego, którym w niniejszej implementacji zawsze jest gracz X, a osobno dla O. Tym sposobem tablica Q-learningowa aktualizowana jest dla każdego stanu, ucząc się tak jakby 2x szybciej.

Plansza składa się z 0, 1 oraz -1. 0 oznacza puste pole, możliwe do zajęcia, 1 jest X, natomiast -1 kółko.

Tablica Q-learningowa zawiera wszystkie permutacje -1, 0 oraz 1 dla każdego pola. Łatwo zauważyć, że jest ich zbyt dużo - około 4x za dużo; gdyż niektóre stany gry nigdy nie będą miały prawa zaistnieć (np. plansza z samymi krzyżykami). Nie jest to jednak powód do zmartwienia, gdyż podczas rozgrywki na wyuczonej tablicy nigdy też w tym momencie nie zdarzy się odwiedzenie takiego niemożliwego pola. Tablica zatem nie jest odfiltrowywana z niepotrzebnych potencjalnych plansz.

Implementacja zakłada odpowiednią kolejność argumentów - zawsze najpierw podawany jest gracz X, zaczynający, a następnie gracz O. Dlatego też osobno przeprowadzane są testy ze względu na gracza, który zaczyna.

Implementacja sieci neuronowej

Sieć neuronowa została zaimplementowana przy użyciu biblioteki keras (w Pythonie). Wykorzystuje ona domyślnie jedną warstwę ukrytą i podane poniżej w sprawozdaniu parametry.

Uczona jest na podstawie wytrenowanych wartości w tablicy Q-learningowej, zatem najpierw, przed samym wytrenowaniem sieci (która powinna doprowadzić do dużo lepszego uogólnienia rozgrywek), są puszczane losowe rozgrywki, które uaktualniają wartości w tablicy. Następnie dzielone są one na odpowiednie wejście i wyjście i sieć neuronowa uczy się w sposób nadzorowany odpowiednich etykiet wyjściowych.

SPOSOBY REPREZENTACJI DANYCH

Jednym z kluczowych założeń/wyborów implementacji rozwiązania opartego na tablicy Q-learningowej był rodzaj reprezentacji danych. Po wyuczeniu wartości w tablicy, opierając się na przebiegach zadanej liczby gier graczy losowych (domyślnie przyjętą wartością rozegranych gier jest 5000), należało zadecydować o tym jakie sieć neuronowa będzie przyjmować wejście i, w zależności od tego wyboru, co zwróci na wyjściu.

9in-9out

Pierwszym, najbardziej narzucającym się i intuicyjnym sposobem reprezentacji danych, jest dostarczenie sieci neuronowej na wejściu aktualnego stanu planszy, a na wyjściu otrzymanie wartości z tablicy Q-learningowej dla tej planszy.

Ten sposób reprezentacji będzie w dalszych badaniach przyjmowany jako domyślny.

27in-9out

Innym z testowanych rozwiązań jest zmiana wartości wejściowej - inny sposób reprezentacji planszy. Zamiast dostarczania sieci neuronowej planszy, która składa się z -1 dla kółka, 0 dla pustych pól oraz 1 dla krzyżyka, dostarczana jest rozbita reprezentacja tej samej treści.

Najpierw brane są pod uwagę pola, na których zaznaczony jest krzyżyk - wtedy w 9-elementowej tablicy wypełnionej 0, zamieniane są na 1 te pola, na których jest x. Podobnie dla kółek i pól pustych. Tym sposobem każdą z tych trzech składowych zapisujemy jedynie za pomocą zer i jedynek.

Przykładowo mając planszę postaci:

x | x | o

  | | o

  | o | x

Czyli [1, 1, -1, 0, 0, -1, 0, -1, 1], zostaje ona zamieniona na:
[**1, 1, 0, 0, 0, 0, 0, 0, 1**, **0, 0, 1, 0, 0, 1, 0, 1, 0**, **0, 0, 0, 1, 1, 0, 1, 0, 0**]
krzyżyk **kółko** **puste pole**

18in-1out

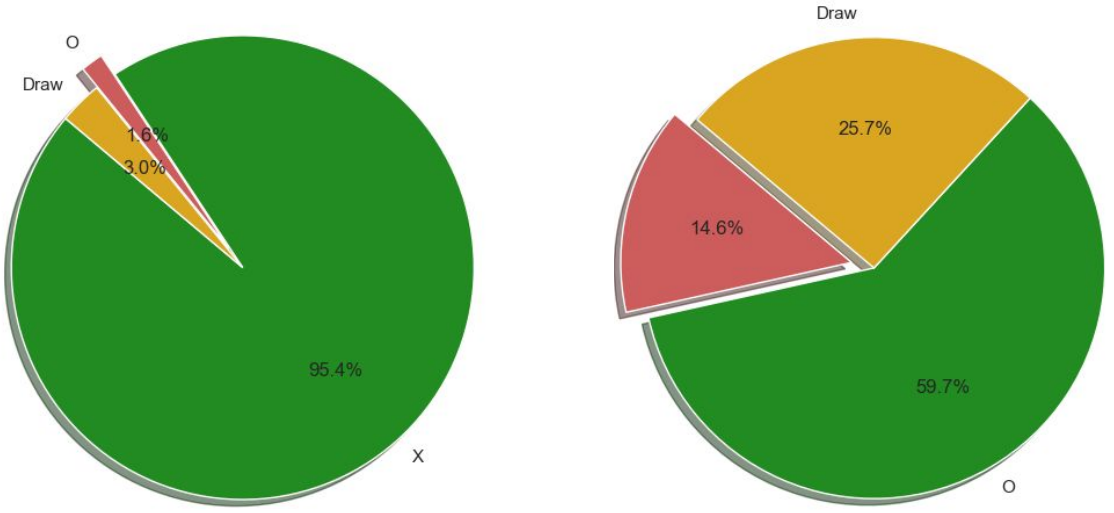
Ostatnim z testowanych sposób reprezentacji danych jest przyjęcie 18-elementowego wektora na wejście i uzyskanie liczby na wyjściu.

Wektor wejściowy utworzony jest z połączenia 9-elementowej planszy oraz 9-elementowego wektora *one-hot* odwzorowującego akcję, która jako następna ma zostać wykonana. Wtedy na wyjściu zwracana jest odpowiednia liczba z Q-learningowej tablicy, odpowiadająca wartości podanej na wejściu akcji.

BADANIA

Do badań związanych ze skutecznością sieci neuronowej zostały przyjęte pewne parametry domyślne, tak aby ułatwić testowanie i prezentację wyników; mianowicie:

- wartości domyślne dla Q-learningu - alpha = 0.7, gamma = 0.95 oraz wartość początkowa w tablicy init = 0.2
- liczba gier treningowych (Q-learning) - 5000
- liczba epok uczenia - 10
- funkcja straty - błąd średniokwadratowy
- optymalizacja - Adam
- liczba powtórzeń (dla ilu nauczonych sieci uśrednione były wyniki) - 10



Powyższe wykresy przedstawiają procent wygranych dla parametrów domyślnych sieci neuronowej. Wpierw ukazane są wyniki, gdy sieć neuronowa jest graczem zaczynającym, a następnie, gdy jest drugim z kolei graczem - na zielono zwizualizowane są wyniki sieci, na czerwono wyniki gracza losowego, a na żółto remisy.

Można stwierdzić, że to, co interesuje nas najbardziej, to liczba nieprzegranych, a nie wygranych, rozgrywek. Dla domyślnych parametrów osiągana jest zatem ponad 98,5% skuteczność, gdy sieć kontroluje gracza X (zaczynającego), natomiast około 85% skuteczność w przeciwnym razie.

Najlepsze wyniki

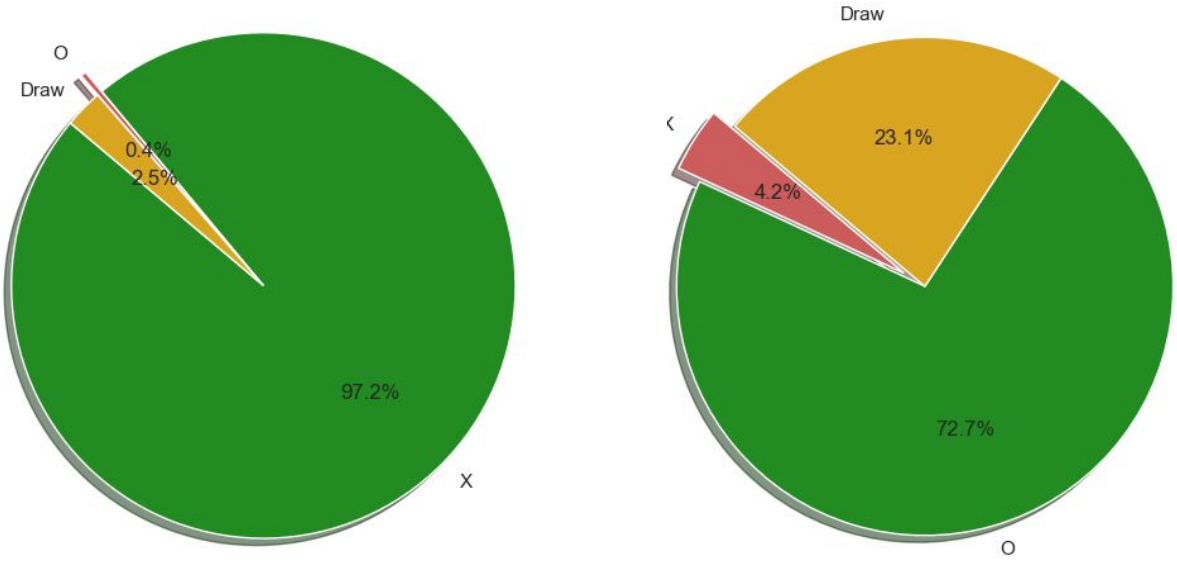
Po przeprowadzeniu wszystkich badań, które zostaną poniżej zaprezentowane, wyłonione zostały najlepsze parametry (a nie domyślne).

Są to:

- wartości domyślne dla Q-learningu
- liczba gier treningowych (Q-learning) - 7500
- liczba epok uczenia - 10
- funkcja straty - błąd średniokwadratowy (zlogarytmowany)
- optymalizacja - Adadelta

liczba powtórzeń (dla ilu nauczonych sieci uśrednione były wyniki) - 10

Z takimi parametrami następnie sprawdziliśmy czy faktycznie ich kombinacja daje lepsze wyniki i uzyskaliśmy następujące rezultaty:



Dla gracza X (zaczynającego) wzrosła zarówno liczba wygranych gier - z 95,4% do 97,2%, tym samym zmalała liczba przegranych do zaledwie 0,4%.

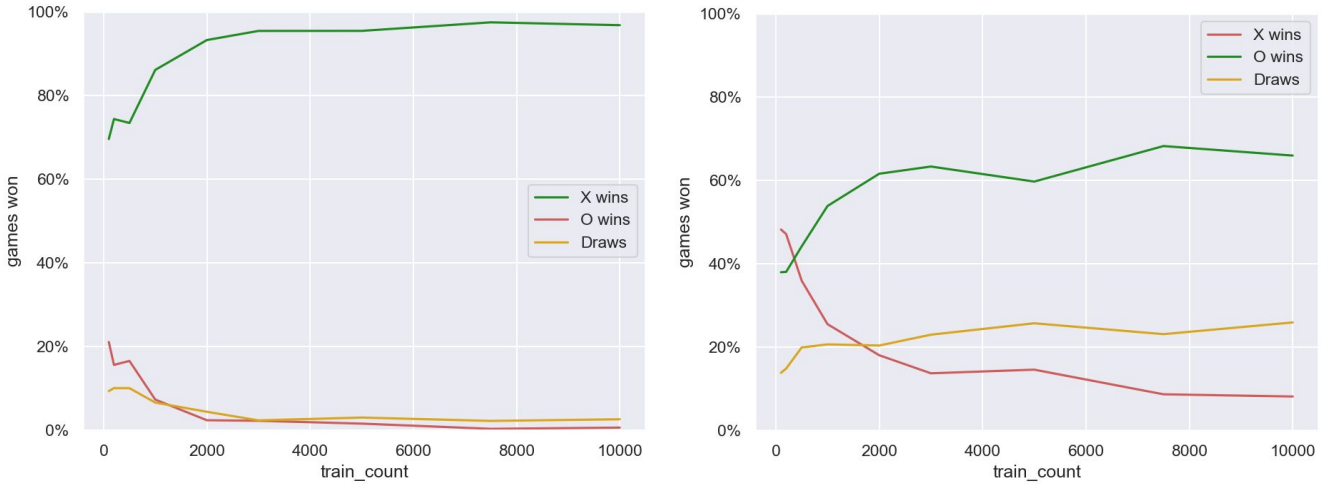
Natomiast dla gracza O również została uzyskana znacząca poprawa. Podczas gdy liczba zremisowanych meczów się praktycznie nie zmieniła, to znacząco wzrosła liczba wygranych, bo aż o 13%, tym samym zmniejszając liczbę przegranych do zaledwie 4,2%.

Można zatem uznać rozgrywki kontrolowane przez sieć neuronową jako bardzo dobre i zdecydowanie lepsze niż zwykła tablica Q-learningowa.

Eksperyment 1.Skuteczność sieci neuronowej w zależności od liczby gier treningowych w Q-learningu

Badana była przede wszystkim skuteczność (liczba wygranych) w zależności od liczby gier treningowych rozegranych między dwoma losowymi graczami na etapie uczenia wartości w tablicy Q-learningowej.

Badana była następująca liczba gier: 100, 200, 500, 1000, 2000, 3000, 5000, 7500 oraz 10000.



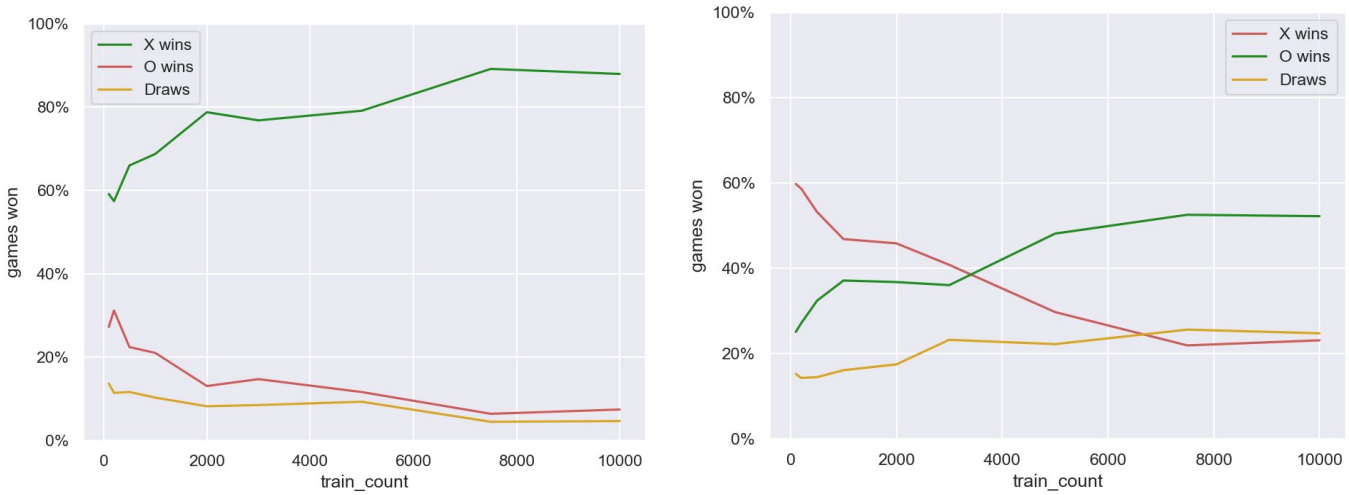
Po prawej stronie widzimy jak modeluje się stosunek wygranych do przegranych i zremisowanych gier, gdy sieć neuronowa jest graczem zaczynającym, a po prawej gdy jest druga z kolei. Znaczenie kolorów jest takie samo - zielonym oznaczona jest sieć, czerwonym przeciwnik (gracz losowy), a żółtym remisy.

Widać, że o ile dla sytuacji, w której sieć neuronowa zaczyna, już przy około 2000 rozgrywek treningowych osiągane są bardzo zadowalające wyniki (około 90-95% wygranych i zaledwie kilka procent przegranych), o tyle dla odwrotnej sytuacji można by przebadać jeszcze więcej danych - liczba wygranych przeciwnika bowiem cały czas widocznie maleje.

Inne sposoby reprezentacji danych

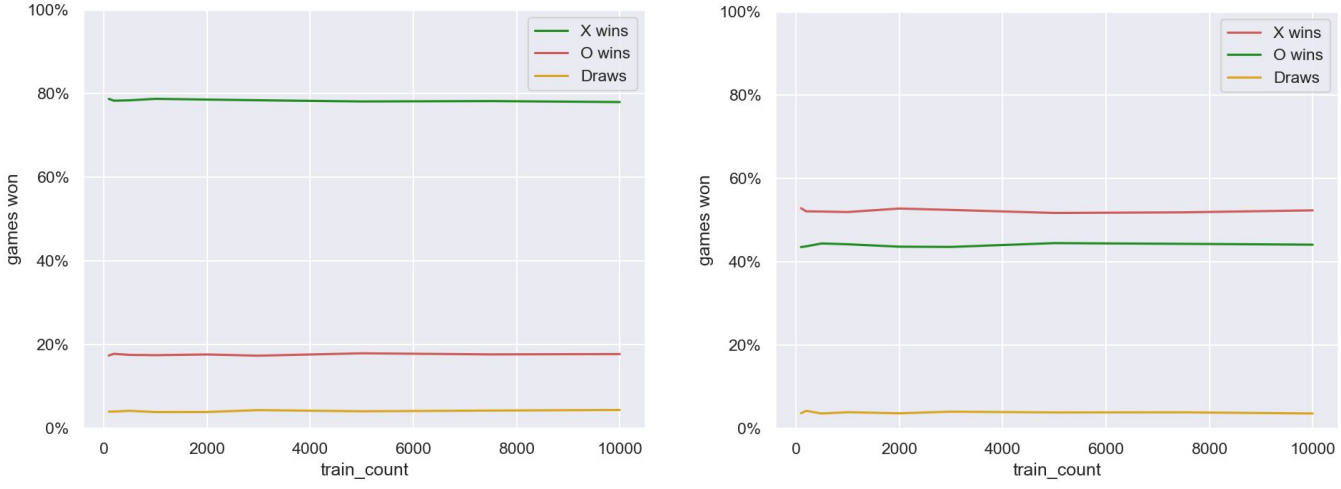
Przedstawione powyżej zestawienia dotyczą parametrów domyślnych - reprezentacji danych oznaczanej jako 9in-9out. Warto jednak sprawdzić jak radzą sobie pozostałe z rozważanych reprezentacji tj. 27in-9out oraz 18in-1out.

27in-9out



Od razu widać, że dla tej reprezentacji danych sieć uczy się wolniej. Cały czas, podczas wzrastania liczby rozgrywek treningowych, widać tendencję wzrostową dla liczby wygranych, jednakże dla ostatniej z badanych wartości - 10000 gier - osiągane są zdecydowanie gorsze wyniki. Również na samym początku, co wyraźnie rzuca się w oczy do około 3000 gier, gdy sieć neuronowa steruje nie zaczynającym graczem, widać, że liczba wygranych przeciwnika zdecydowanie dominuje.

18in-1out



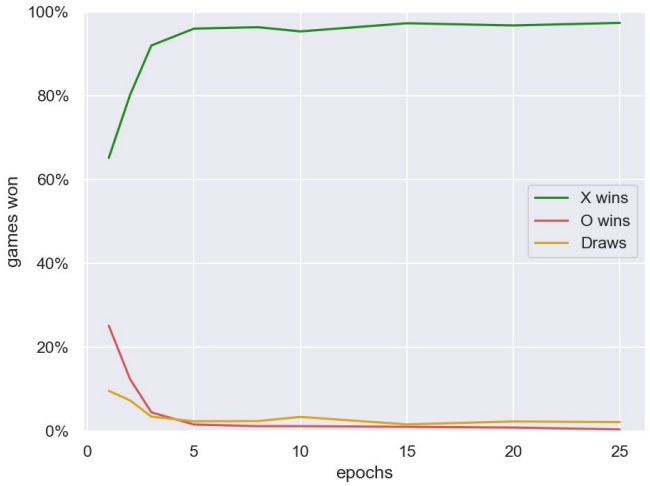
Dla tego sposobu reprezentacji danych dzieje się rzecz niezwykła, bowiem jakość modelu jest niemalże niezależna od liczby rozgrywek uczących.

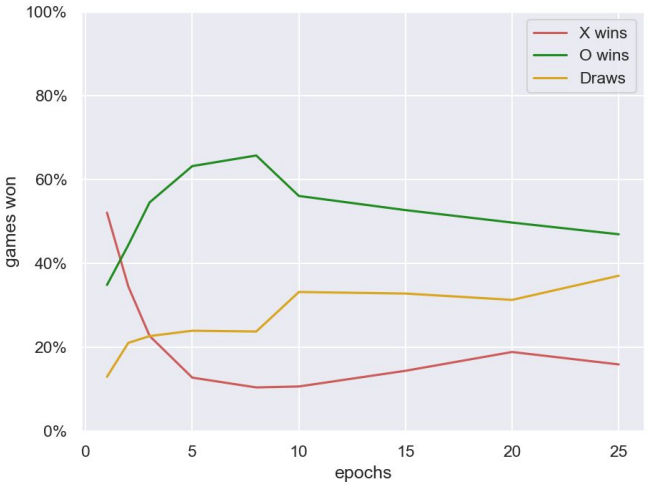
Wynika to z tego, że sieć neuronowa uczy się jedynie odwzorowywać wartości tablicy Q-learningowej na 1 (w przypadku jakiejkolwiek wartości) lub -1 w przypadku, gdy w tablicy Q-learningowej znajdowała się wartość zabroniona.

Eksperyment 2. Skuteczność sieci neuronowej w zależności od liczby epok uczenia

Założenia:
Badania, jak zostało to opisane powyżej, mierzono dla wartości domyślnych, zmieniając liczbę epok. Każda kombinacja parametrów została wytrenowana 10 razy, a następnie uśredniona. Gier testowych zostało przeprowadzonych 1000 i na tej podstawie powstały wykresy porównawcze.

Badana liczba epok: 1, 2, 3, 5, 8, 10, 15, 20 oraz 25





Jak widać sieć neuronowa przy zaczynaniu (reprezentowaniu gracza X) bardzo szybko zaczyna osiągać dobre wyniki i wystarczy niewiele epok (mniej niż 5).

W przeciwnym wypadku natomiast sprawa ma się zupełnie inaczej i najwyraźniej dochodzi do przeuczenia. Po 8 epoce następuje załamanie i pojawia się tendencja wzrastająca. Słusznym zatem (mniej więcej) okazał się wybór domyślnego parametru liczby epok wynoszącej 10, gdyż w przypadku większej liczby epok moglibyśmy otrzymywać dużo gorsze wyniki.

Warto jednak spostrzec, że choć po 8 epokach liczba wygranych gier wyraźnie spada, to liczba zremisowanych gier przez wszystkie epoki utrzymuje tendencję wzrostową.

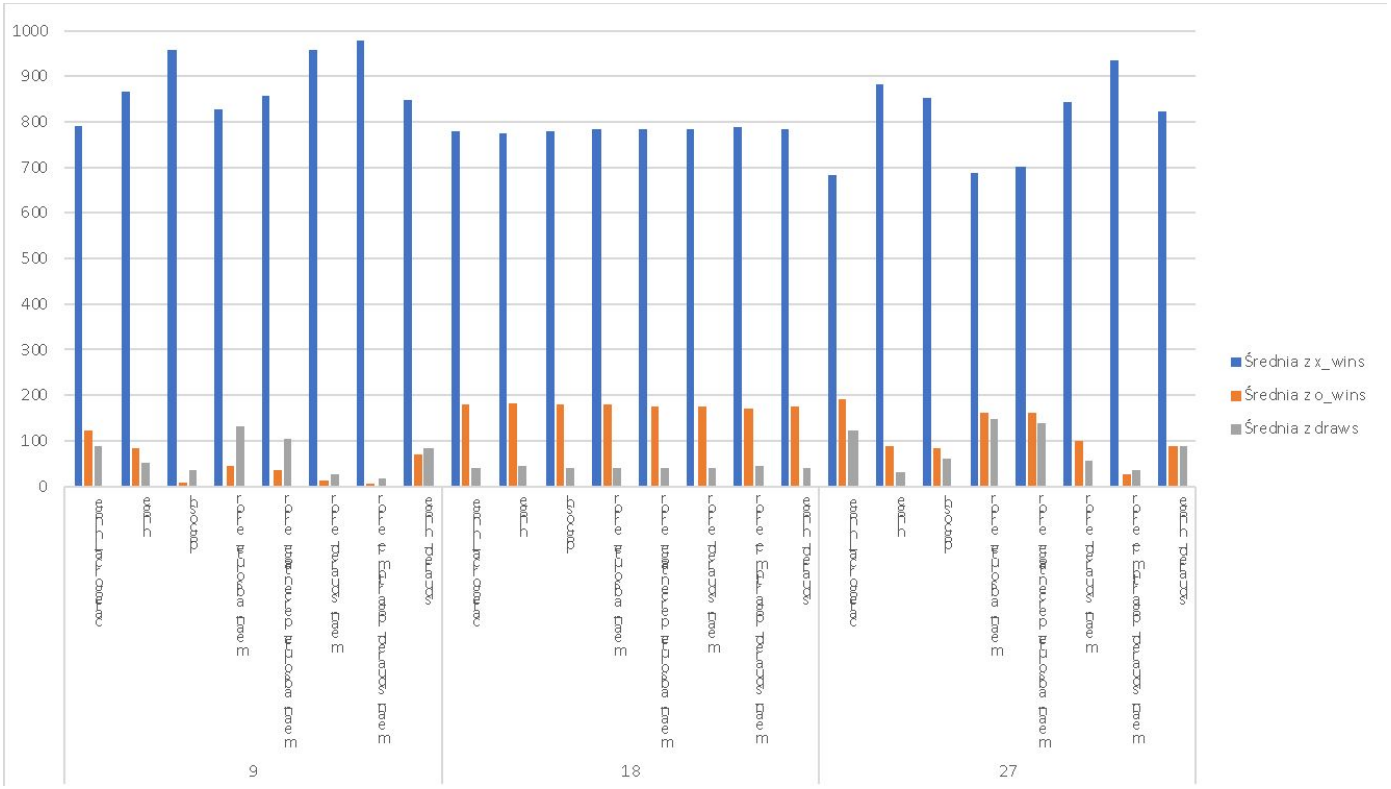
Eksperyment 3. Badanie wpływu funkcji straty na wyniki

Założenia:
Mierzono wyniki gracza Q-Learning w 1000 grach względem gracza losowego w zależności od rodzaju błędu, służącego do wyuczenia sieci neuronowej. Eksperyment powtarzano 10 razy a wyniki uśredniano. Uwzględniono sytuację, w której rozpoczyna gracz Q-Learning

oraz sytuację, w której zagrywa jako drugi. Badania prowadzono dla 3 różnych reprezentacji danych wejściowych.

Dla rozpoczynającego gracza Q-Learning:

Etykiety wierszy	Średnia z x_wins	Średnia z o_wins	Średnia z draws
9	**884,1875**	**47,65**	**68,1625**
categorical_hinge	790,6	123,2	86,2
hinge	863,3	84,8	51,9
logcosh	954,8	9,5	35,7
mean_absolute_error	825,4	43,5	131,1
mean_absolute_percentage_error	856,6	36,3	107,1
mean_squared_error	957,4	13,8	28,8
mean_squared_logarithmic_error	978	1,5	20,5
squared_hinge	847,4	68,6	84
18	**780,9875**	**177,4375**	**41,575**
categorical_hinge	778,9	179	42,1
hinge	772,6	184	43,4
logcosh	779,4	179,2	41,4
mean_absolute_error	780,9	179,7	39,4
mean_absolute_percentage_error	780,9	176,9	42,2
mean_squared_error	784,3	176,5	39,2
mean_squared_logarithmic_error	787,6	168,8	43,6
squared_hinge	783,3	175,4	41,3
27	**800,3125**	**113,75**	**85,9375**
categorical_hinge	682,7	192,9	124,4
hinge	881	89,4	29,6
logcosh	854,4	83	62,6
mean_absolute_error	687,9	161,1	151
mean_absolute_percentage_error	698,9	162,8	138,3
mean_squared_error	841,7	101,9	56,4
mean_squared_logarithmic_error	934,2	29	36,8
squared_hinge	821,7	89,9	88,4
Suma końcowa	**821,8291667**	**112,9458333**	**65,225**

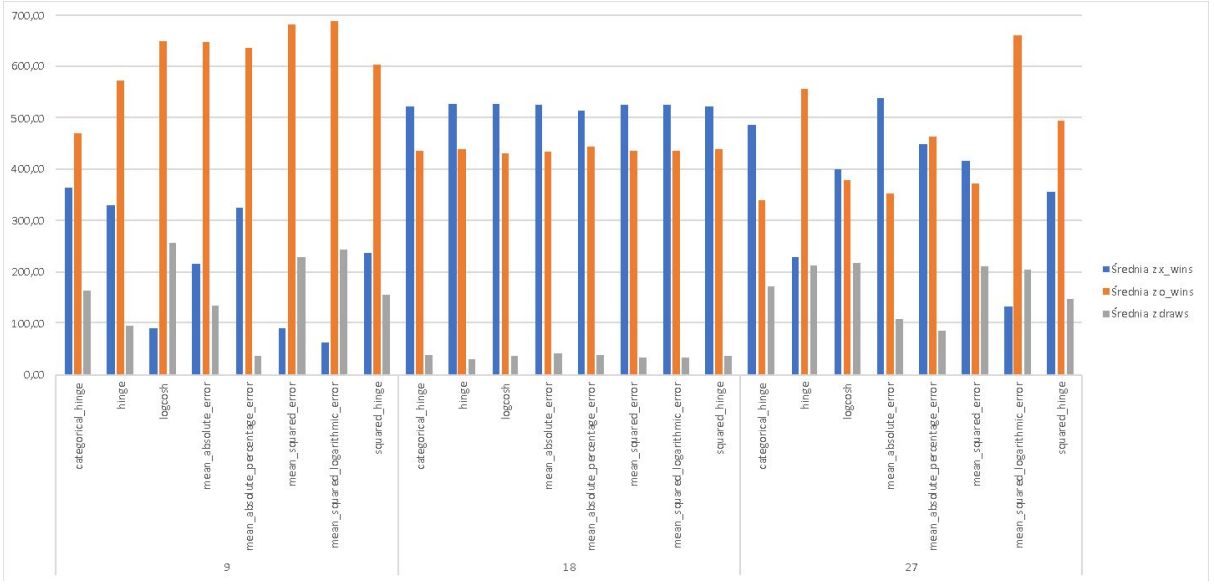


Wniosek:
W przypadku reprezentacji 9-in i 28-in najlepiej radzi sobie błąd mean_squared_logarithmic_error. Dobrze sprawdzają się również błędy MSE, logcosh i hinge.

Dla gracza Q-Learning zagrywającego w drugiej kolejności (jako O)

Etykiety wierszy	Średnia z x_wins	Średnia z o_wins	Średnia z draws
Reprezentacja -9	**214,88**	**619,40**	**165,73**
categorical_hinge	363,40	472,20	164,40
hinge	330,00	572,80	97,20
logcosh	91,00	650,20	258,80
mean_absolute_error	217,00	647,40	135,60
mean_absolute_percentage_error	325,40	636,60	38,00
mean_squared_error	90,60	680,60	228,80
mean_squared_logarithmic_error	63,80	690,40	245,80
squared_hinge	237,80	605,00	157,20
Reprezentacja -18	**524,60**	**438,45**	**36,95**
categorical_hinge	522,80	438,00	39,20
hinge	528,60	440,20	31,20
logcosh	528,40	433,20	38,40
mean_absolute_error	525,20	433,60	41,20
mean_absolute_percentage_error	514,80	446,20	39,00
mean_squared_error	526,60	437,80	35,60
mean_squared_logarithmic_error	526,80	438,60	34,60

squared_hinge	523,60	440,00	36,40
Reprezentacja -27	**376,28**	**452,50**	**171,23**
categorical_hinge	486,80	340,00	173,20
hinge	229,20	556,40	214,40
logcosh	401,60	379,40	219,00
mean_absolute_error	538,00	353,20	108,80
mean_absolute_percentage_error	450,00	462,60	87,40
mean_squared_error	416,00	372,40	211,60
mean_squared_logarithmic_error	132,40	661,20	206,40
squared_hinge	356,20	494,80	149,00
Suma końcowa	**371,92**	**503,45**	**124,63**



Eksperyment 4. Badanie wpływu optymalizacji

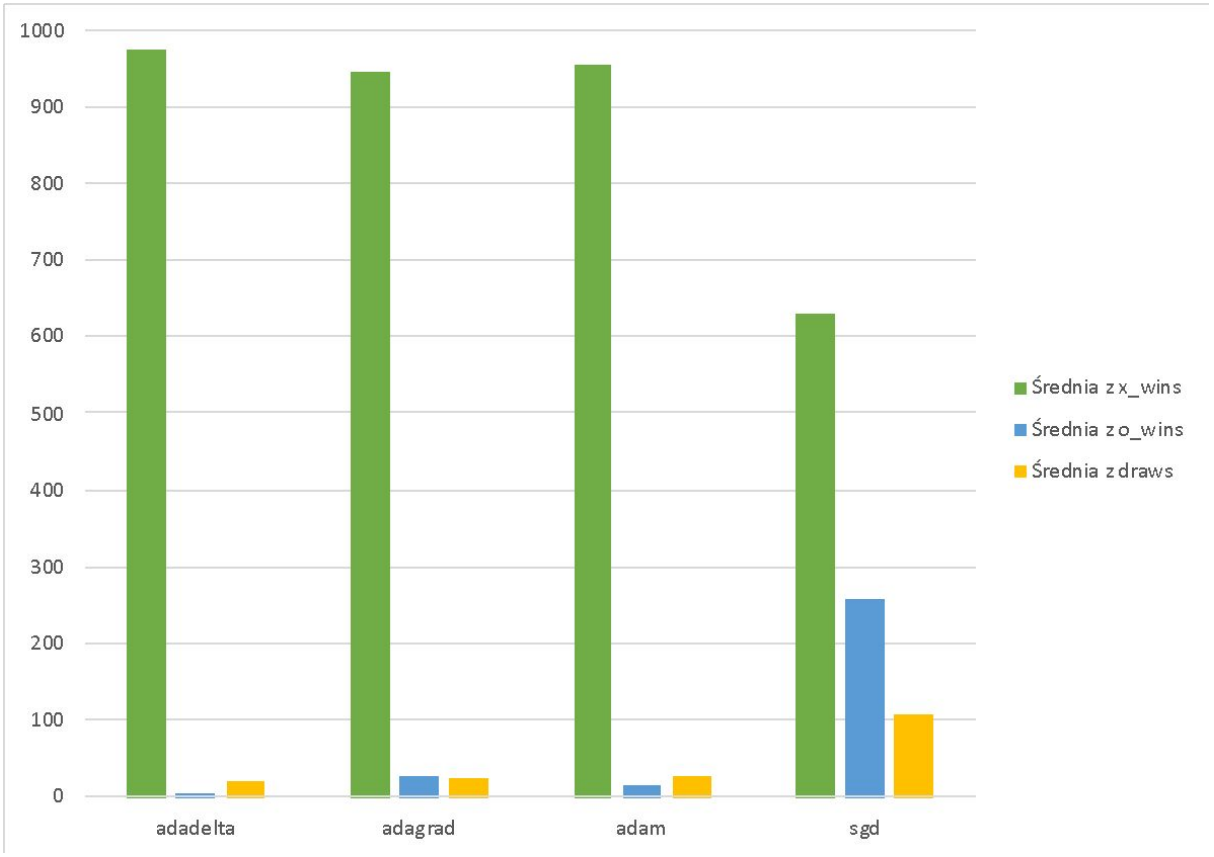
Założenia:

Mierzono wyniki gracza Q-Learning w 1000 grach względem gracza losowego w zależności od rodzaju optymalizacji błędu przy uczeniu sieci neuronowej. Brano pod uwagę optymalizatory: adam, adagrad, adadelta i SGD. Eksperyment powtarzano 10 razy a wyniki uśredniano. Uwzględniono sytuację, w której rozpoczyna gracz Q-Learning oraz sytuację, w której zagrywa jako drugi. Badania prowadzono dla reprezentacji danych wejściowych 9-in.

Dla rozpoczynającego gracza Q-Learning

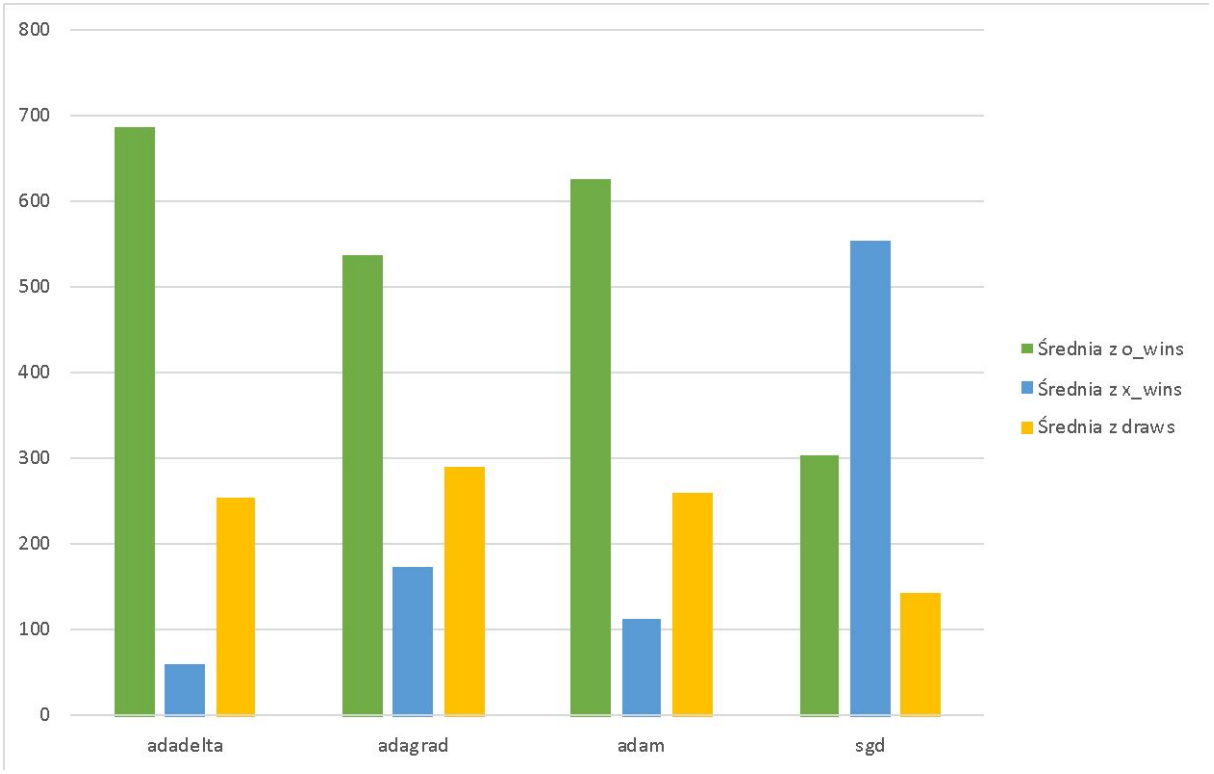
Etykiety wierszy	Średnia z x_wins	Średnia z o_wins	Średnia z draws
adadelta	973,8	5	21,2
adagrad	945,6	29,2	25,2

adam	956,5	14,2	29,3
sgd	630,3	260,2	109,5
Suma końcowa	**876,55**	**77,15**	**46,3**



Dla gracza Q-Learning rozpoczynającego jako drugi:

Etykiety wierszy	Średnia z o_wins	Średnia z x_wins	Średnia z draws
adadelta	686,1	59,1	254,8
adagrad	536	174,4	289,6
adam	626,1	114,1	259,8
sgd	304,1	552,3	143,6
Suma końcowa	**538,075**	**224,975**	**236,95**



Wnioski:
Najlepiej sprawdzają się optymalizatory Adam i Adadelta. SGD radzi sobie zdecydowanie gorzej.

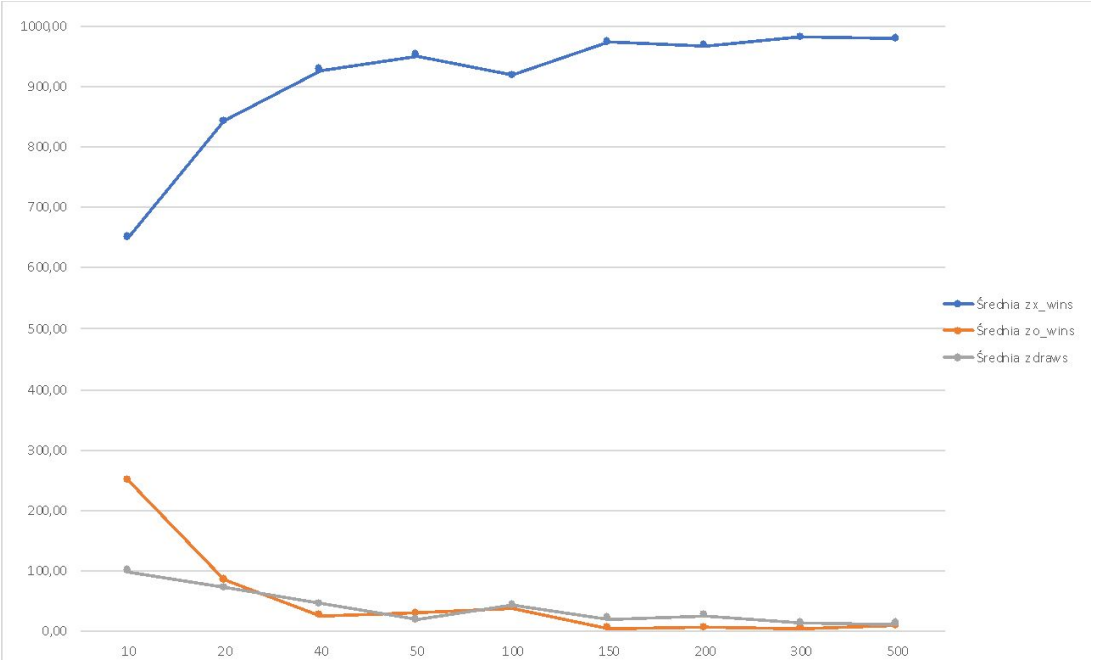
Eksperyment 5. Badanie wpływu liczby neuronów w warstwie ukrytej

Założenia:
Mierzono wyniki gracza Q-Learning w 1000 grach względem gracza losowego w zależności od liczby neuronów w warstwie ukrytej sieci neuronowej. Dla każdej wartości eksperyment powtarzano 5 razy a wyniki uśredniano. Uwzględniono sytuację, w której rozpoczyna gracz Q-Learning oraz sytuację, w której zagrywa jako drugi. Badania prowadzono dla reprezentacji danych wejściowych 9-in.

Przy rozpoczynającym graczu Q-Learningowym

Etykiety wierszy	Średnia z x_wins	Średnia z o_wins	Średnia z draws
10	649,60	250,80	99,60
20	842,40	85,80	71,80
40	927,00	26,80	46,20
50	950,40	30,00	19,60
100	917,80	39,20	43,00
150	973,60	4,80	21,60
200	967,20	6,20	26,60

300	980,80	5,20	14,00
500	978,20	9,00	12,80
Suma końcowa	**909,67**	**50,87**	**39,47**

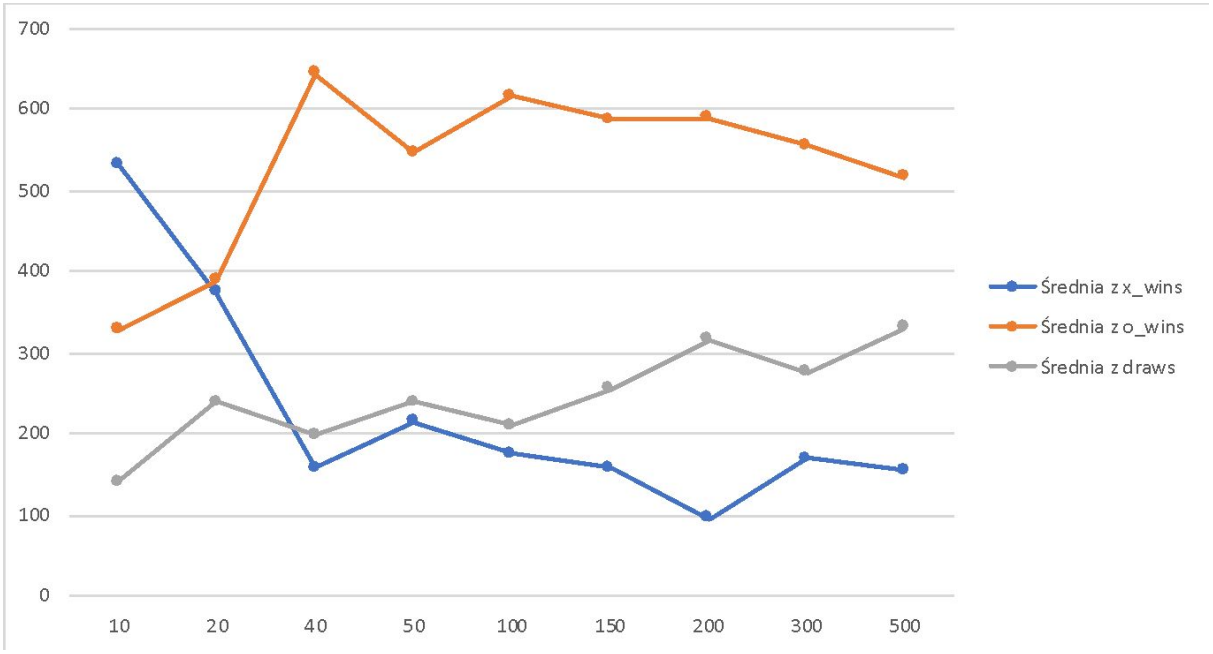


Wniosek:

Wraz ze wzrostem liczby neuronów w warstwie ukrytej rośnie skuteczność gracza Q-Learning.

Dla gracza Q-Learning zagrywającego w drugiej kolejności (jako O):

Etykiety wierszy	Średnia z x_wins	Średnia z o_wins	Średnia z draws
10	532	327	141
20	374	387,4	238,6
40	157,8	643,4	198,8
50	214,6	546,4	239
100	175,2	614,6	210,2
150	158,6	586	255,4
200	96	588,2	315,8
300	170	554,2	275,8
500	154,4	515,8	329,8
Suma końcowa	**225,84**	**529,22**	**244,93**



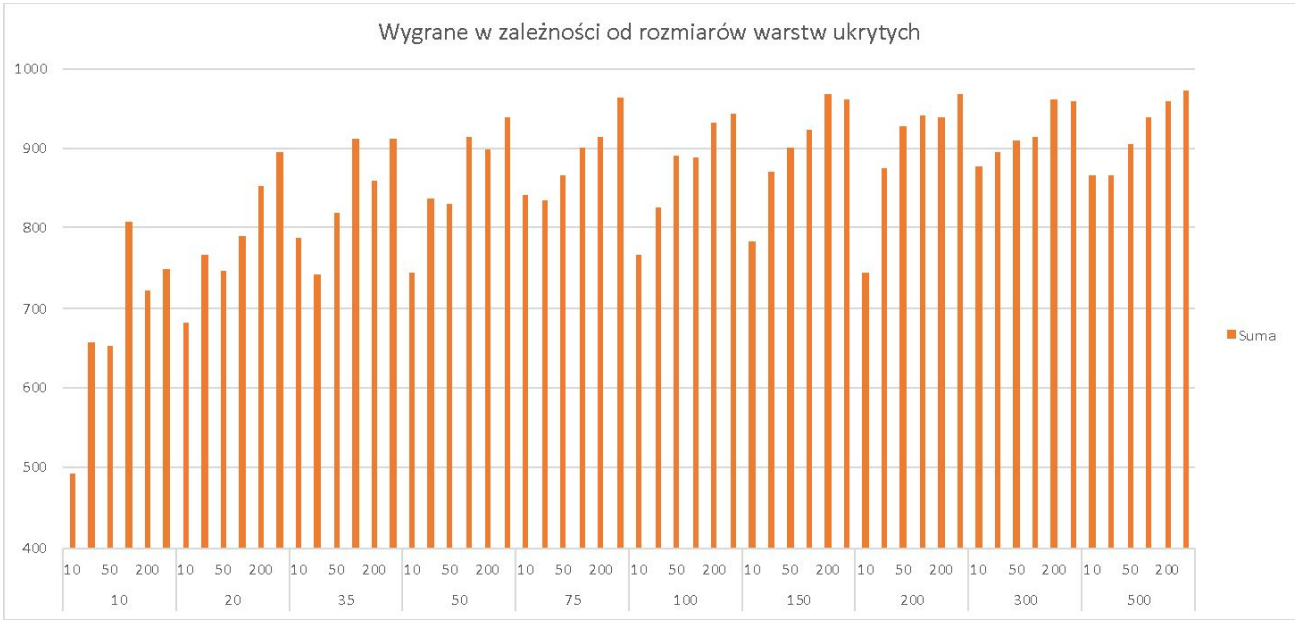
W przypadku, gdy gracz Q-Learning rozpoczyna jako drugi można zaobserwować moment przeuczenia.

Eksperyment 6. Badanie wpływu liczby neuronów w sieci z dwoma warstwami ukrytymi

Założenia:
Podobnie jak we wcześniejszym eksperymencie mierzono wyniki gracza Q-Learning w 1000 grach względem gracza losowego w zależności od liczby neuronów w warstwie ukrytej sieci neuronowej. Zbudowano jednak sieć złożoną z 2 warstw. Badano wiec skuteczność dla par (**hidden_one_size** , **hidden_two_size**). Uwzględniono sytuację, w której rozpoczyna gracz Q-Learning oraz sytuację, w której zagrywa jako drugi. Badania prowadzono dla reprezentacji danych wejściowych 9-in.

hidden_one_size	hidden_two_size	Średnia z x_wins	hidden_one_size	hidden_two_size	Średnia z x_wins
10	10,00	492,60	**100,00**	10,00	768,80
	20,00	657,60		20,00	825,40
	50,00	653,80		50,00	891,60
	100,00	809,40		100,00	889,40
	200,00	723,60		200,00	931,60
	500,00	748,80		500,00	944,80
10 Suma		**680,97**	**100 Suma**		**875,27**
20	10,00	683,00	**150,00**	10,00	783,00
	20,00	767,60		20,00	872,60
	50,00	747,00		50,00	901,20
	100,00	789,40		100,00	923,00

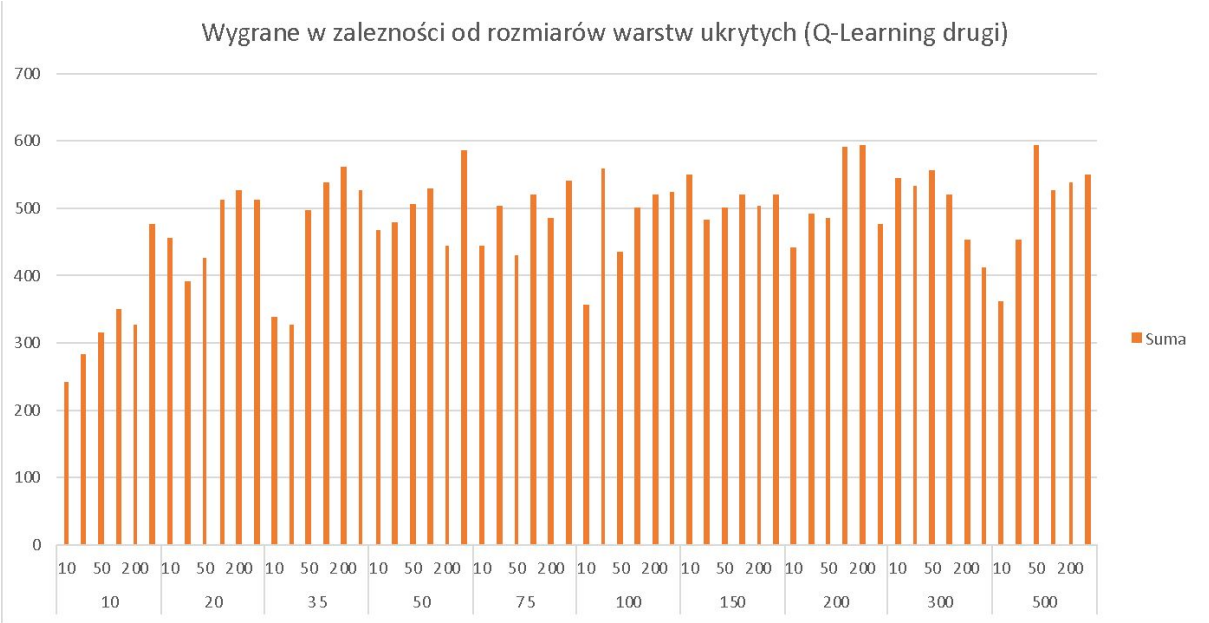
		200,00	852,60		200,00	969,40
		500,00	896,00		500,00	962,80
20 Suma			**789,27**	**150 Suma**		**902,00**
	35	10,00	787,20	**200,00**	10,00	745,40
		20,00	742,40		20,00	875,20
		50,00	819,40		50,00	928,00
		100,00	911,20		100,00	941,40
		200,00	859,60		200,00	939,00
		500,00	912,80		500,00	968,20
35 Suma			**838,77**	**200 Suma**		**899,53**
	50	10,00	745,60	**300,00**	10,00	879,00
		20,00	837,40		20,00	895,80
		50,00	830,00		50,00	910,80
		100,00	914,80		100,00	915,60
		200,00	897,80		200,00	961,20
		500,00	939,20		500,00	959,60
50 Suma			**860,80**	**300 Suma**		**920,33**
	75	10,00	842,40	**500,00**	10,00	866,20
		20,00	834,80		20,00	868,00
		50,00	868,20		50,00	906,40
		100,00	900,80		100,00	938,80
		200,00	915,00		200,00	959,60
		500,00	964,60		500,00	972,20
75 Suma			**887,63**	**500 Suma**		**918,53**



Najlepszy wynik → para 500,500

Przy graczu Q-Learning rozpoczynającym jako drugi

hidden_one_size	hidden_two_size	Średnia z o_wins	hidden_one_size	hidden_two_size	Średnia z o_wins
10	10,00	242,60	**100,00**	10,00	356,20
	20,00	283,60		20,00	558,00
	50,00	314,40		50,00	436,80
	100,00	351,60		100,00	499,20
	200,00	327,00		200,00	521,40
	500,00	477,20		500,00	525,00
10 Suma		**332,73**	**100 Suma**		**482,77**
20	10,00	456,60	**150,00**	10,00	551,40
	20,00	391,20		20,00	481,60
	50,00	427,00		50,00	499,80
	100,00	510,80		100,00	522,00
	200,00	526,00		200,00	504,00
	500,00	511,20		500,00	521,40
20 Suma		**470,47**	**150 Suma**		**513,37**
35	10,00	339,20	**200,00**	10,00	440,20
	20,00	327,80		20,00	491,20
	50,00	497,80		50,00	487,00
	100,00	538,20		100,00	590,40
	200,00	561,40		200,00	594,20
	500,00	526,40		500,00	476,20
35 Suma		**465,13**	**200 Suma**		**513,20**
50	10,00	468,40	**300,00**	10,00	544,80
	20,00	480,20		20,00	532,20
	50,00	505,60		50,00	557,60
	100,00	528,40		100,00	519,40
	200,00	444,80		200,00	453,80
	500,00	584,40		500,00	411,40
50 Suma		**501,97**	**300 Suma**		**503,20**
75	10,00	445,80	**500,00**	10,00	360,80
	20,00	503,00		20,00	452,80
	50,00	429,00		50,00	593,40
	100,00	521,40		100,00	525,80
	200,00	486,00		200,00	539,60
	500,00	541,60		500,00	550,20
75 Suma		**487,80**	**500 Suma**		**503,77**

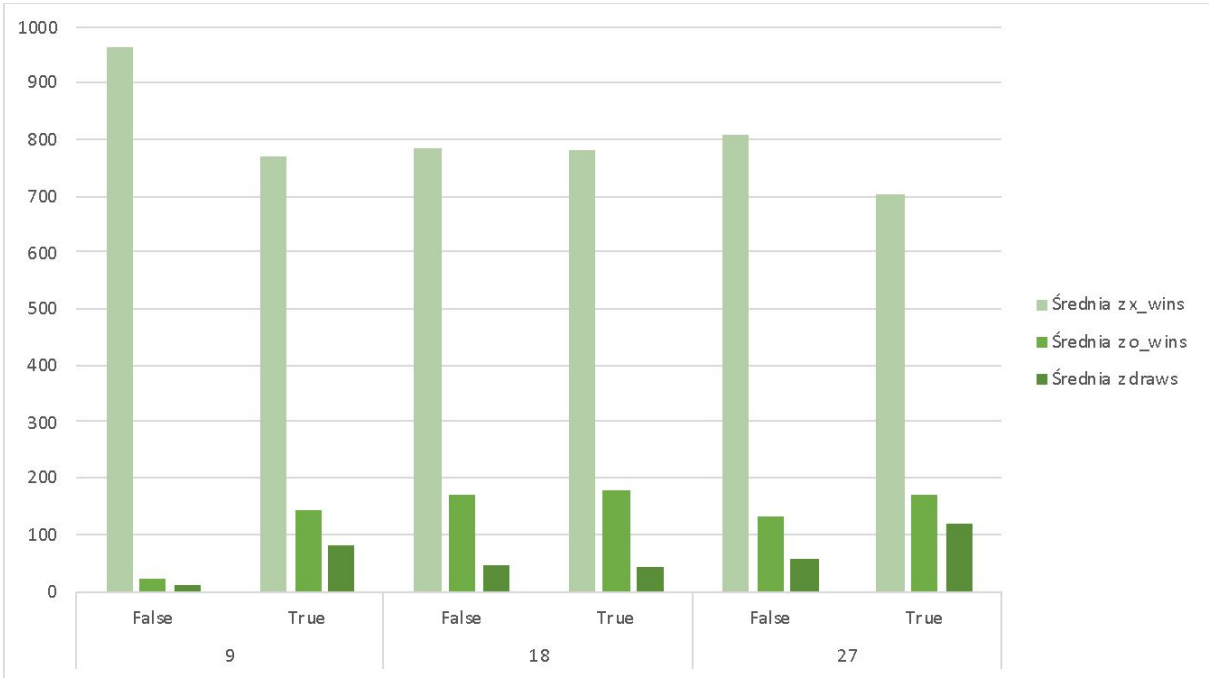


Najlepszy wynik → para (200,200)

Eksperyment 7. Badanie wpływu „filtrowania" wzorców uczących na skuteczność gracza Q-Learning

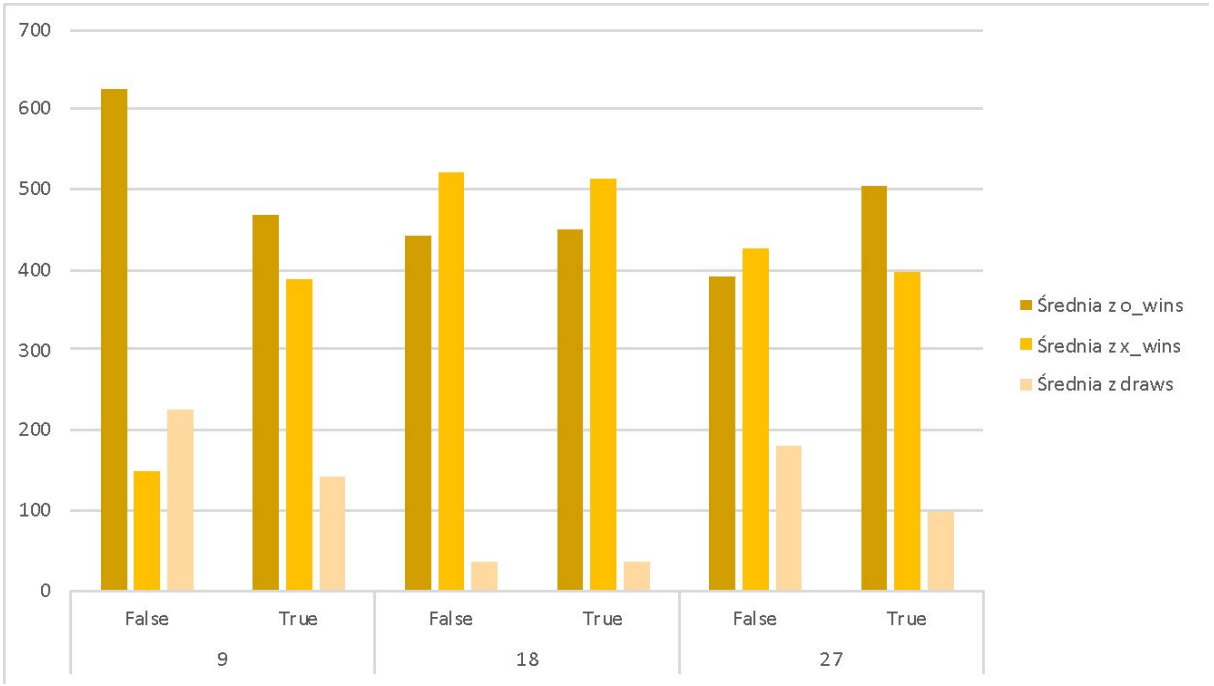
Założenia:
Mierzono wyniki gracza Q-Learning w 1000 grach względem gracza losowego w zależności od zastosowania filtrowania tablicy Q na wejściu sieci neuronowej. Przy uwzględnionym filtrowaniu nie brano pod uwagę podczas uczenia sieci neuronowej wierszy (stan, akcje), w których wartość współczynnika Q nie została zmieniona na skutek uczenia Q-Learningiem. Uwzględniono sytuację, w której rozpoczyna gracz Q-Learning oraz sytuację, w której zagrywa jako drugi. Badania prowadzono dla reprezentacji danych wejściowych 9-in.

Etykiety wierszy	Suma z x_wins	Suma z o_wins	Suma z draws
9	**867,7**	**85**	**47,3**
Bez filtrowania	963,6	24	12,4
Z filtrowaniem	771,8	146	82,2
18	**782,6**	**173,7**	**43,7**
Bez filtrowania	784,2	169,8	46
Z filtrowaniem	781	177,6	41,4
27	**758**	**152,4**	**89,6**
Bez filtrowania	810	132,4	57,6
Z filtrowaniem	706	172,4	121,6
Suma końcowa	**802,76**	**137,03**	**60,2**



Dla gracza Q-Learning rozpoczynającego jako drugi:

Etykiety wierszy	Średnia z o_wins	Średnia z x_wins	Średnia z draws
9	**547,2**	**269**	**183,8**
Bez filtrowania	625	149,8	225,2
Z filtrowaniem	469,4	388,2	142,4
18	**447**	**517,4**	**35,6**
Bez filtrowania	443,4	522	34,6
Z filtrowaniem	450,6	512,8	36,6
27	**448,1**	**411,9**	**140**
Bez filtrowania	391,6	426,6	181,8
Z filtrowaniem	504,6	397,2	98,2
Suma końcowa	**480,76**	**399,43**	**119,8**



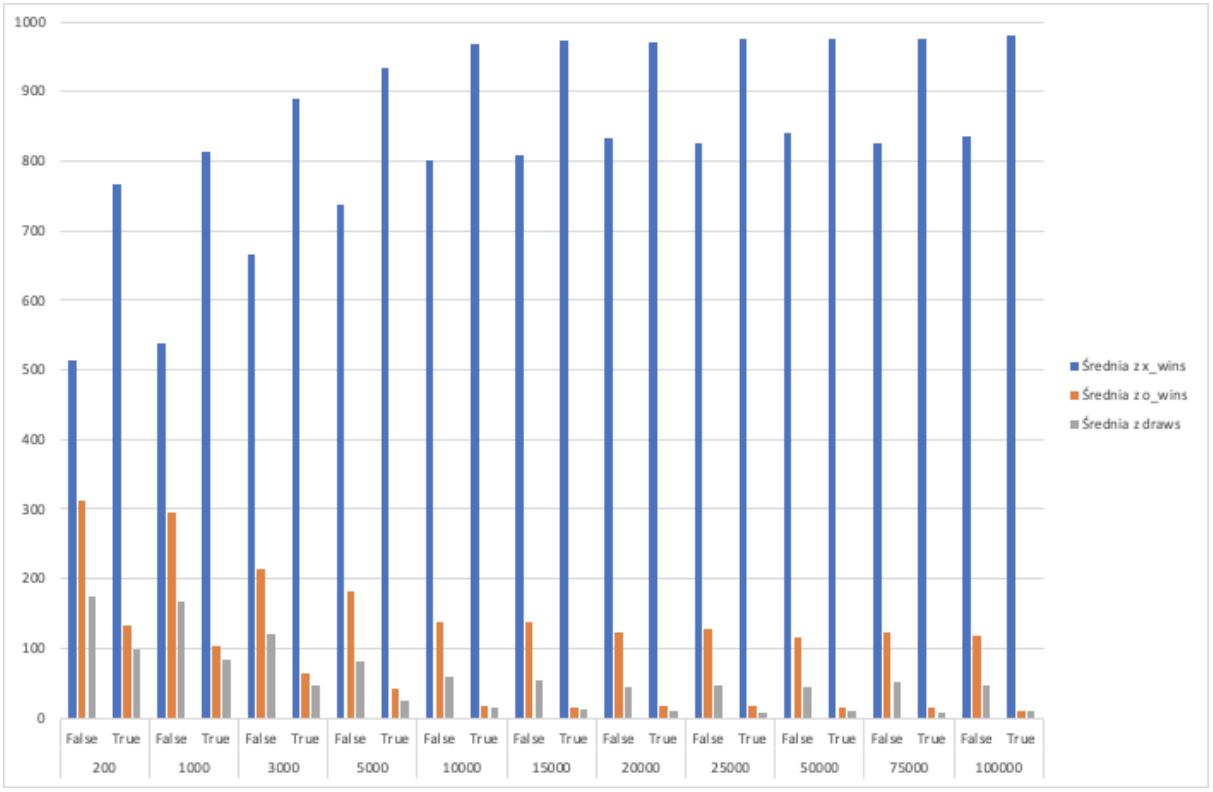
Wniosek:

Wprowadzenie filtrowania do reprezentacji 9-in i 27-in pogarsza skuteczność gracza Q-Learning. W przypadku, gdy Q-Learning rozpoczyna jako drugi w reprezentacji 18-in

Eksperyment 8. Badanie liczby iteracji na efekty wyuczenia Q-Learningu

Etykiety wierszy	Średnia z x_wins	Średnia z o_wins	Średnia z draws
200	**640,9**	**222,4**	**136,7**
False	514,2	311,6	174,2
True	767,6	133,2	99,2
1000	**676,1**	**198,6**	**125,3**
False	538,6	294,2	167,2
True	813,6	103	83,4
3000	**777,9**	**139,1**	**83**
False	666,4	214,2	119,4
True	889,4	64	46,6
5000	**835,6**	**110,8**	**53,6**

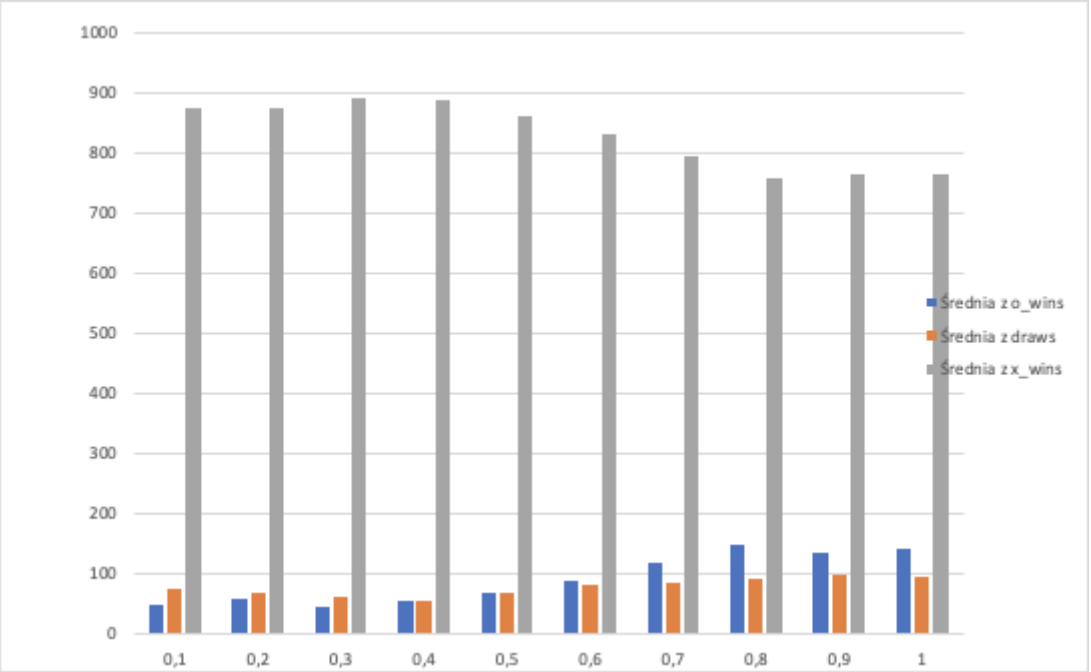
False	738	180,8	81,2
True	933,2	40,8	26
10000	**885**	**78,2**	**36,8**
False	801,8	138,6	59,6
True	968,2	17,8	14
15000	**890**	**76,3**	**33,7**
False	807,6	137	55,4
True	972,4	15,6	12
20000	**901,3**	**70,4**	**28,3**
False	831,8	122,6	45,6
True	970,8	18,2	11
25000	**899,6**	**72,7**	**27,7**
False	824,8	128,4	46,8
True	974,4	17	8,6
50000	**907,7**	**65,1**	**27,2**
False	839,8	115	45,2
True	975,6	15,2	9,2
75000	**900,7**	**69,1**	**30,2**
False	826	122,4	51,6
True	975,4	15,8	8,8
100000	**907,3**	**64**	**28,7**
False	834,6	117,6	47,8
True	980	10,4	9,6
Suma końcowa	**838,4**	**106,1**	**55,6**



WPŁYW PARAMETRU ALPHA

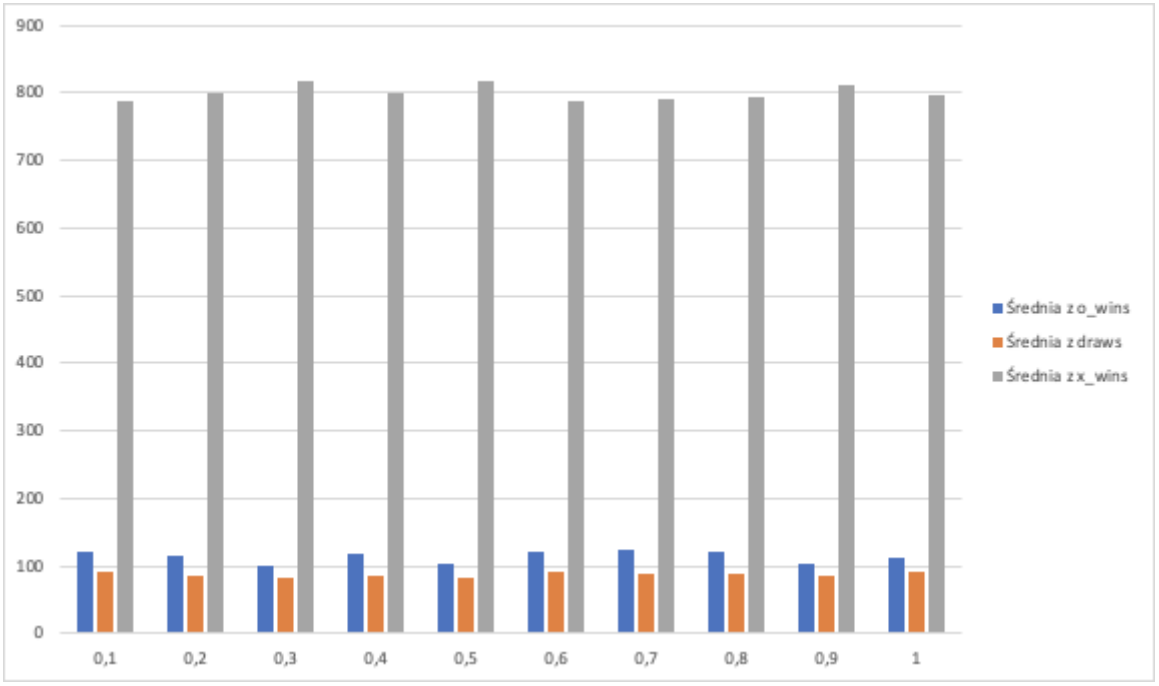
Dla 1000 iteracji algorytmu Q-Learning

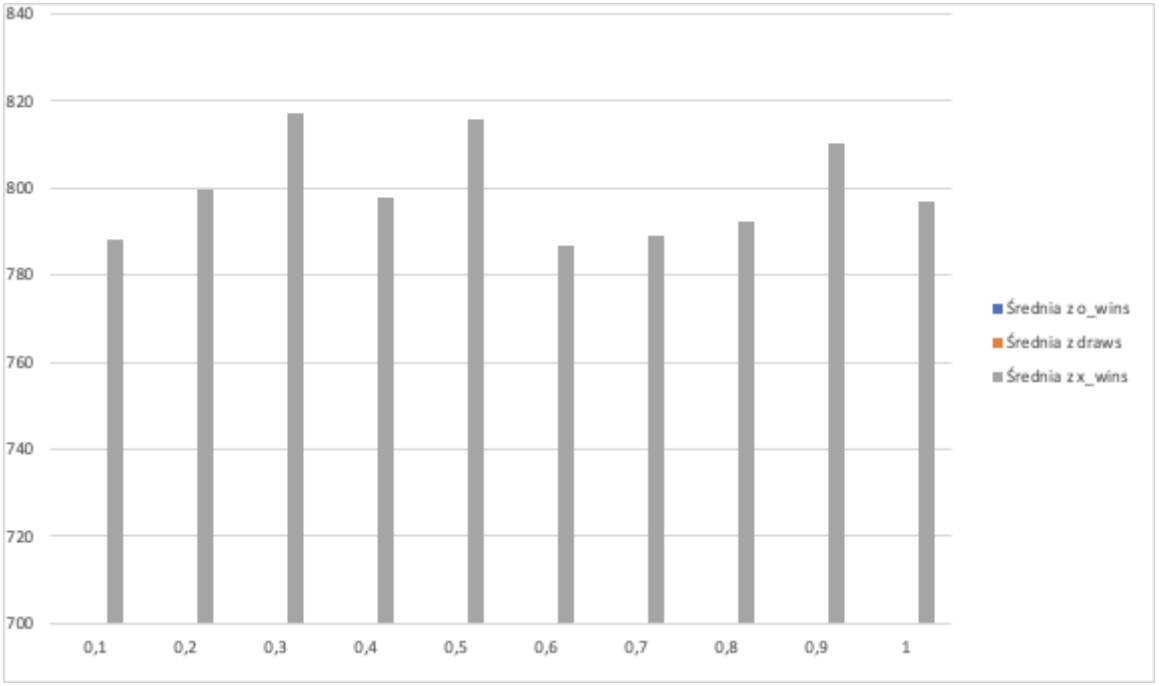
Etykiety wierszy	Średnia z o_wins	Średnia z draws	Średnia z x_wins
0,1	49,8	75,6	874,6
0,2	59,2	68,2	872,6
0,3	47	63,6	889,4
0,4	56,6	57	886,4
0,5	70,8	68,6	860,6
0,6	88,8	81	830,2
0,7	120	86	794
0,8	149	93,4	757,6
0,9	135,4	99,6	765
1	142	95,4	762,6
Suma końcowa	**91,86**	**78,84**	**829,3**



WPŁYW PARAMETRU INIT

Etykiety wierszy	Średnia z o_wins	Średnia z draws	Średnia z x_wins
0,1	121,4	90,4	788,2
0,2	115,2	85	799,8
0,3	100,6	82,2	817,2
0,4	117,8	84,2	798
0,5	102	82,2	815,8
0,6	120,8	92,2	787
0,7	123	87,8	789,2
0,8	120,6	87	792,4
0,9	103	86,8	810,2
1	111,6	91,4	797
Suma końcowa	**113,6**	**86,92**	**799,48**


900
800
700
600
500
400
300
200
100
0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1
Średnia z o_wins
Średnia z draws
Średnia z x_wins


840
820
800
780
760
740
720
700
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1
Średnia z o_wins
Średnia z draws
Średnia z x_wins