Studium Magisterskie

Kierunek Analiza Danych – Big Data

Juliusz Juzaszek
Nr albumu 82285

Wykorzystanie algorytmu głębokiego uczenia przez wzmacnianie dla wyznaczenia optymalnej strategii obrotu akcjami giełdowymi na przykładzie spółek wchodzących w skład indeksu S&P500

Praca magisterska
pod kierunkiem naukowym
dr hab. Ryszarda Szupiluka, prof. SGH
Instytut
Informatyki i Gospodarki Cyfrowej

Warszawa 2024

Spis treści

Wstęp

W ostatniej dekadzie można było zaobserwować znaczący wzrost popularności stosowania algorytmów uczenia maszynowego. Zjawisko to można przypisać kilku kluczowym czynnikom. Po pierwsze, nastąpił znaczący wzrost dostępności dużych zbiorów danych, które są niezbędne do trenowania skomplikowanych modeli i ulepszania algorytmów. Po drugie, nastąpił postęp w technologii obliczeniowej, który umożliwił przetwarzanie dużych zbiorów danych i skomplikowanych modeli w rozsądnym czasie. To przyspieszyło badania i rozwój w dziedzinie uczenia maszynowego. Trzeci czynnik to rozwój i ulepszenia w samych algorytmach uczenia maszynowego, w tym w uczeniu głębokim. Część tych algorytmów istniała już od dłuższego czasu, ale przyrost ilości mocy obliczeniowej i praktycznych zastosowań sprawił, że dopiero stosunkowo niedawno znalazły się one w tak dużym centrum uwagi. Obiecujące rezultaty modeli uczenia maszynowego i związane z tym zwiększone zainteresowanie wokół nich doprowadziło do przyspieszenia rozwoju kolejnych metod. Wreszcie, coraz większa dostępność otwartych zasobów edukacyjnych i platform do nauki umożliwiły szerokiej społeczności badaczy, inżynierów i hobbystów eksperymentowanie, naukę i wdrażanie algorytmów uczenia maszynowego. Ta demokratyzacja dostępu do narzędzi i wiedzy przyczyniła się do szybkiego rozwoju i adopcji technologii w różnych sektorach przemysłu i nauki. Jednym z potencjalnych zastosowań dla algorytmów uczenia maszynowego jest wykorzystanie ich do prognozowania zmienności cen aktywów finansowych takich jak akcje giełdowe. Umożliwia to wyznaczenie optymalnego portfela inwestycyjnego, który potencjalnie wraz z upływem czasu mógłby zacząć przynosić ponadprzeciętne zyski.

Celem niniejszej pracy jest weryfikacja skuteczności metod uczenia maszynowego w wyznaczaniu optymalnej strategii obrotu akcjami giełdowymi spółek wchodzących w skład indeksu S&P500 na podstawie historycznych cen i wskaźników analizy technicznej. Z punktu widzenia teorii rynków efektywnych, która nie dopuszcza możliwości systematycznego osiągania ponadprzeciętnej stopy zwrotu na bazie danych historycznych kierowanie się jakimiś doskonale znanymi zależnościami przy podejmowaniu decyzji inwestycyjnych jest pozbawione większego sensu. Wykorzystanie algorytmu samouczącego się, który potrafi zdobywać całkowicie nową wiedzę obserwowaną przez człowieka jedynie jako ciąg różnych liczb stwarza szansę na odkrycie w latach 2015-2023 w wycenie akcji giełdowych pewnych trudno wykrywalnych niewyeksploatowanych zależności, które mogą pomóc obejść efektywność rynku.

W ramach realizacji celu pracy zostanie zaimplementowany algorytm głębokiego gradientu deterministycznej polityki DDPG, który jest wersją algorytmu głębokiego uczenia przez wzmacnianie przeznaczoną do modelowania decyzji określanych poprzez ciągły zbiór wartości. Został on przedstawiony w pracy naukowej z 2016 roku przez Google DeepMind i jego przeznaczenie pokrywa się z wyzwaniami jakie stawia stworzenie modelu przeznaczonego do tego typu zadania.

Wykorzystanie algorytmu głębokiego uczenia przez wzmacnianie DDPG dla wyznaczenia optymalnego portfela inwestycji w 30 arbitralnie wybranych akcji giełdowych na podstawie danych z lat 2009-2018 było już przedmiotem pracy Xiao-Yang Liu, Zhuoran Xiong, Shan Zhong, Hongyang Yang & Anwar Walid, Practical Deep Reinforcement Learning Approach for Stock Trading opublikowanej w 2018 roku. Niniejsza praca rozszerzy to podejście weryfikując czy sprawdzi się ono na niearbitralnym portfelu akcji giełdowych pochodzących ze zdywersyfikowanego indeksu giełdowego oraz czy rynek w późniejszym okresie czasu po publikacji powyższego badania zgodnie z hipotezą rynku efektywnego zdyskontował w wycenie dostępnych w publicznym obrocie instrumentów finansowych potencjalne przewagi zastosowanego rozwiązania.

Rozdział I przedstawi założenia inwestowania w akcje giełdowe, prognozwania zmienności ich cen, analizy technicznej i wprowadzi odniesienie pracy do adekwatnej wersji hipotezy rynku efektywnego. Rozdział II wprowadzi pojęcie uczenia maszynowego oraz przedstawi krótko potencjalne zastosowania algorytmów uczenia maszynowego w omawianej dziedzinie. Rozdział III przedstawi algorytmy, z których zbudowany jest DDPG oraz uzasadni jego wyjątkowe możliwości, które przesądziły o jego wyborze. Rozdział IV przedstawi specyfikację stworzonego modelu, zmiennych objaśniających i danych na których został on zastosowany. Rozdział V przedstawi rezultaty osiągnięte dla wypróbowanych konfiguracji modelu skupiając się na porównaniu ich z indeksem cen akcji oraz oczekiwaną stopą zwrotu według modelu CAPM.

Rozdział I

Metody i znaczenie analizy technicznej w prognozowaniu zmienności cen akcji

1.1 Wprowadzenie do inwestowania w akcje giełdowe

Inwestowanie w akcje na giełdzie polega na zakupie udziałów w spółkach publicznych, co pozwala inwestorom uczestniczyć w zyskach tych spółek poprzez wypłaty dywidend oraz potencjalne wzrosty wartości akcji. Proces ten odbywa się na zorganizowanych rynkach, takich jak giełdy papierów wartościowych. Fundamentalną różnicą między inwestycją w akcje giełdowe, a takie instrumenty jak lokata bankowa czy obligacje skarbu państwa jest wprowadzenie elementu ryzyka. Zamiast osiągać ustalone dodatnie stopy zwrotu, inwestor może osiągać zyski wyższe niż wynikające ze stopy wolnej od ryzyka, ale w zamian wystawia wszystkie zainwestowane środki na potencjalne straty. Inwestor na rynku akcji giełdowych powinien posiadać podstawową wiedzę na tematy gospodarcze i funkcjonowania rynków finansowych. Inwestowanie wymaga analizy spółek i przebiegu ich wyceny, aby wybrać te o najlepszych perspektywach wzrostu i stabilności. Rozłożenie inwestycji na wiele różnych akcji lub sektorów może zmniejszyć tak zwane ryzyko systematyczne poniesienia strat, ponieważ nie wszystkie segmenty rynku reagują w ten sam sposób na zmiany panujących warunków gospodarczych. Do tego ogranicza to ryzyko specyficzne dla konkretnej spółki, która może na przykład zbankrutować ze względu na popełniane błędy niezależne od warunków otoczenia zewnętrznego. Na podstawie danych historycznych można stwierdzić że inwestycje w akcje giełdowe mogą przynosić zyski przewyższające te wyznaczane przez stopę wolną od ryzyka, ale wymaga to czasu i cierpliwości. Inwestowanie długoterminowe pomaga zminimalizować wpływ krótkoterminowych wahań rynkowych na uzyskany wynik. Inwestowanie może być aktywne w postaci częstego kupna i sprzedaży akcji giełdowych w celu osiągnięcia zysku krótkoterminowego lub pasywne w postaci długoterminowego trzymania akcji z przekonaniem o ich wzroście. Konstruowanie określonych portfeli akcji często jest wynikiem korzystania z określonego doradztwa profesjonalnych instytucji finansowych. Ponadto istnieją indeksy giełdowe wyceniane na podstawie wyników zagregowanych odpowiednio akcji umożliwiając osiąganie ustabilizowanych zysków przy zdywersyfikowanym ryzyku.

1.1.1 Model wyceny aktywów kapitałowych (CAPM)

Modelem służący do ustalenia adekwatnej wyceny danego aktywa kapitałowego na podstawie związanego z nim systematycznego ryzyka rynkowego jest model wyceny aktywów kapitałowych CAPM. Model został opracowany w latach 60 XX wieku niezależnie przez Jacka Treynora, Williama Sharpe'a, Johna Lintnera i Jana Mossina w oparciu o wcześniejsze prace Harry'ego Markowitza dotyczące dywersyfikacji portfela inwestycyjnego.[1,2,3,4,5] Podstawą modelu jest przyjęcie za punkt odniesienia stopy wolnej od ryzyka poniesienia strat, którą może być przykładowo oprocentowanie 10-letnich obligacji skarbu państwa oraz stopy rynkowej, którą stanowi wybrany indeks giełdowy o zdywersyfikowanym ryzyku rynkowym. W ramach modelu oblicza się współczynnik beta, który w odpowiedni sposób określa zaobserwowany stosunek zmienności ocenianego aktywa kapitałowego do wybranego indeksu giełdowego. Uzyskany z modelu wynik pozwala porównać czy faktyczna stopa zwrotu osiągana z danego aktywa kapitałowego jest adekwatna do ponoszonego ryzyka systematycznego. Pokazuje to poniższy wzór:

$$R = R_f + \beta * \left(R_m - R_f\right) \qquad (1)$$

$$\beta = \frac{Cov(R, R_m)}{Var(R_m)} \qquad (2)$$

gdzie:

R – oczekiwana stopa zwrotu

R_f – stopa wolna od ryzyka

R_m – stopa zwrotu z portfela rynkowego

β – współczynnik określający kowariancję stóp zwrotu danego aktywa kapitałowego ze stopami zwrotu portfela rynkowego w stosunku do wariancji stóp zwrotu portfela rynkowego

[1] Capital Asset Pricing Model, Wikipedia, online: https://pl.wikipedia.org/wiki/Capital_Asset_Pricing_Model, dostęp: 13.02.2024 14:02

[2] W. Sharpe, Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk, Journal of Finance, 1964

[3] J. Lintner, The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets, Review of Economics and Statistics, 1965

[4] J. Mossin, Equilibrium in a Capital Asset Market, Econometrica, 1966

[5] H. Markowitz, Portfolio Selection, Journal of Finance, 1952

Zależność pomiędzy beta (β) a oczekiwanym zachowaniem ocenianego aktywa kapitałowego jest następująca:

$\beta = 0$: Brak reakcji na zmiany rynku

$\beta < 1$: Niska wrażliwość na zmiany rynku

$\beta = 1$: Takie same zmiany jak rynek

$\beta > 1$: Wysoka wrażliwość na zmiany rynku

$\beta < 0$: Odwrotna reakcja na zmiany rynku

1.1.2 Prognozowanie zmienności cen akcji giełdowych

Złożoność, różnorodność i wielodziedzinowość wszelkiego rodzaju procesów wpływających na postrzeganie wartości danych spółek przez inwestorów, sposób ich zachowania odnośnie decyzji zakupowo-sprzedażowych, a w konsekwencji na wycenę akcji giełdowych tych spółek sprawia, że prognozowanie przyszłego kształtowania się cen staje się niezwykle skomplikowanym procesem, a pierwszy dylemat pojawia się już na etapie ustalenia sposobu podejścia jakie przyjmuje się do tego celu. Dwie najpowszechniejsze grupy metod dzielą się na analizę techniczną i analizę fundamentalną. Mają one umożliwić wyznaczenie optymalnej inwestycji kierując się zupełnie odrębnymi kryteriami, które sygnalizują przyszłą wycenę akcji giełdowej. Analiza techniczna zajmuje się analizą zmian kursu i wolumenu obrotów danymi instrumentami finansowymi, natomiast analiza fundamentalna opiera się na ocenie wewnętrznej wartości spółki na podstawie takich czynników jak analiza finansowa spółki, jej pozycja na rynku, branża działalności oraz ogólne panujące warunki gospodarcze. Przedmiotem tej pracy będzie wykorzystanie metod analizy techniczej, które dzięki potencjalnym zdolnościom prognozowania zmienności cen pomogą wyznaczyć optymalną strategię budowy portfela akcji giełdowych. Zostaną one omówione w dalszej części tego rozdziału. Metody analizy fundamentalnej nie będą dalej rozwijane ze względu na brak zastosowania dla celu pracy. Przyczyną wyboru takiego a nie innego podejścia jest chęć zachowania jak największej obiektywizacji zmiennych objaśniających, które trafią do modelu uczenia maszynowego służącego do oceny kilkuset spółek. Zastosowanie analizy fundamentalnej wymaga dobrego zrozumienia kontekstu dla sytuacji danej spółki, więc brak należytej kwantyfikacji i standaryzacji płynących z niej wniosków mógłby utrudnić hurtowe zastosowanie narzędzia prognostycznego.

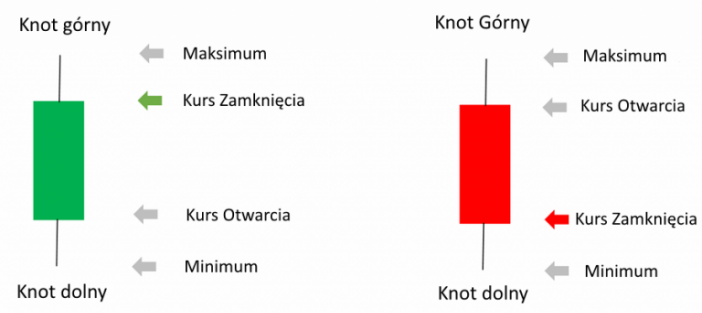
1.2 Wprowadzenie do analizy technicznej

Analiza techniczna jest procesem wykorzystującym specyficzne narzędzia analityczne w celu przewidywania przyszłych kierunków zmian cen na rynku finansowym, opierając się na danych dotyczących historycznych poziomów tych cen. Idea analizy technicznej opiera się na dwóch założeniach. Pierwsze z nich mówi, że zmiany cen w czasie na rynku nie są całkowicie losowe i istnieją dla nich pewne dające się określić trendy, a drugie, że historia się powtarza i zaobserwowanie pewnych zaistniałych zależności w przeszłych wartościach cen pozwala zwiększyć skuteczność przewidywania tych trendów w przyszłości. Akceptacja istnienia trendów, które wykryte we wczesnej fazie zachowują się tak jak przewiduje teoria jest warunkiem koniecznym dla możliwości uzasadnienia zastosowania metod analizy technicznej. Głos przeciwny w tej sprawie zgłasza hipoteza rynku efektywnego, która będzie przedmiotem kolejnego podrozdziału. Zdaniem entuzjastów stosowania analizy technicznej ceny rynkowe wyprzedzają powszechną wiedzę na temat czynników fundamentalnych. Wiele z okresów hossy i bessy zaczynało się od nieznanych lub niedostrzeganych zmian zachodzących w tych czynnikach, a osoby wykorzystujące skutecznie metody analizy technicznej należąc do mniejszości mogą z tej wiedzy korzystać zanim stanie się ona powszechnie znana[6]. Jedną z zalet analizy technicznej jest jej uniwersalność i dostępność. Opracowanie strategii dla każdej z potencjalnych inwestycji przebiega w powtarzalny sposób, a potrzebne dane są zawsze dostępne dla każdego. Analiza techniczna jest stosowana zarówno do podejmowania decyzji inwestycyjnych krótkoterminowych, przeprowadzanych w ciągu jednego dnia, jak i do planowania strategii inwestycyjnych długoterminowych. Najbardziej podstawowym narzędziem wykorzystywanym w ramach analizy technicznej jest analiza wykresu. Stosowane metody mogą nieznacznie różnić się zależnie od typu analizowanego instrumentu finansowego. Przedstawione w dalszej części podrozdziału informacje dotyczą konkretnie metod przeznaczonych dla rynku akcji giełdowych lub przynajmniej rynku kapitałowego. Podstawową informacją wykorzystywaną w analizie technicznej przy zastosowaniu dziennych interwałów czasowych jest rozróżnienie ceny dotyczącej danego dnia na cenę zamknięcia, otwarcia, minimalną i maksymalną. Cena minimalna oznacza najniższą odnotowaną cenę danego dnia, a maksymalna analogicznie najwyższą. Cena otwarcia mówi o początkowej cenie po jakiej były zawierane transakcje kupno-sprzedaż po otwarciu giełdy danego dnia, a cena

[6]J. J. Murphy, Analiza techniczna rynków finansowych, New York Institute of Finance, 1999, wydanie polskie WIG-Press, przekład W. Madej, Warszawa, 1999, s.5

zamknięcia o ostatniej cenie, którą obowiązywała w momencie zamknięcia giełdy. Różnica w tych cenach pozwala badać zmienność i trend cenowy, który miał miejsce danego dnia, a także przewidywać ich kształt w dłuższym okresie. Na rysunku przedstawione są elementy najbardziej charakterystycznego w analizie technicznej wykresu świecowego. Górne i dolne krańce całego wykresu oznaczają cenę minimalną i maksymalną, a górne i dolne krańce tak zwanych „świec" czyli widocznych na rysunku 1 kolorowych słupków pokazują cenę zamknięcia i otwarcia. Jeśli cena otwarcia jest niższa od ceny zamknięcia to oznacza to dodatnią stopę zwrotu między początkiem a końcem dnia i jest przedstawione za pomocą zielonego koloru świecy, w przeciwnym wypadku kolor świecy jest czerwony.

Rysunek 1 *Świece w wykresie świecowym*

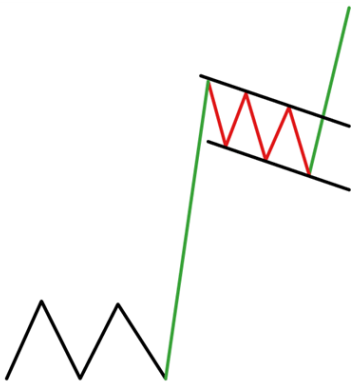


Źródło: Świecie japońskie, opcje24.pl, 2019, online: https://www.opcje24h.pl/swiece-japonskie/

Tak jak opisane powyżej celem przeprowadzenia analizy technicznej jest przewidywanie przyszłego kierunku trendu cenowego na rynku. Sprowadza się to do próby określenia, czy aktualnie zaobserwowany przeszły trend cenowy będzie kontynuowany, czy nastąpi jego zmiana. Jednakże ze względu na dużą zmienność wartości cen akcji giełdowych często obserwuje się różne wzrosty i spadki cen, które nie oznaczają zmiany trendu obserwowanego w dłuższym okresie. Jednym z elementów analizy technicznej jest właściwe przewidzenie momentów wystąpienia punktów tak zwanego wsparcia i oporu, jak również odpowiednie ich wykorzystanie w dalszej analizie. Wsparcie oznacza poziom ceny na którym zatrzymuje się trwający ruch spadkowy. Opór jest przeciwieństwem wsparcia i oznacza poziom ceny na którym zatrzymuje się trwający ruch wzrostowy. Znając te punkty można próbować oszacować przyszły trend. Wyznaczanie przyszłego trendu tą metodą sprowadza się do poszukiwania różnych ustalonych kształtów i schematów na historycznych wykresach cen, a określone punkty wsparcia i oporu mogą właśnie taki kształt tworzyć. Dobrym przykładem jest flaga, która polega na kilkukrotnym naprzemiennym osiągnięciu w krótkim czasie punktów wsparcia i oporu następujących po znaczącym przebiciu wcześniejszego punktu oporu lub wsparcia. Ma

to zwiastować znaczące wybicie ceny w górę lub w dół w zależności od tego czy wspomniane przebicie początkowego punktu dotyczyło oporu czy wsparcia. Przykład flagi zwiastującej wzrost ceny, która zaczyna się od przebicia punktu oporu pokazuje rysunek 2.

Rysunek 2 *Wykres cen akcji giełdowych formujący kształt flagi wzrostowej*



Źródło: W. Venketas, How to Trade Bullish Flag Patterns, DailyFX, 2019, online: https://www.dailyfx.com/education/technical-analysis-chart-patterns/bull-flag.html

1.2.1 Wskaźniki analizy technicznej

Wskaźniki analizy technicznej mimo ich bardziej złożonego podłoża teoretycznego, ułatwiają aplikację metod analizy technicznej. Zmienne te ukazują w skwantyfikowany sposób specyficzne zależności w obserwowanych trendach cenowych, które posiadają zdolność do przewidywania przyszłych zmian w tych trendach. Mają one w założeniu pozwolić w bardziej zmechanizowany i zobiektywizowany sposób przewidzieć punkty wsparcia i oporu oraz kontynuację lub zmianę obecnego trendu. Najprostszym stosowanym wskaźnikiem jest średnia ruchoma. Może być ona przykładowo prosta (SMA) lub wykładnicza (WMA). Pokazuje średnią wartość ceny danego instrumentu finansowego na podstawie ustalonego okna czasowego poprzedzającego dany moment w czasie, dla którego określana jest wartość tego wskaźnika. Pozwala to zbadać trend w jakim w dłuższym okresie porusza się cena przy usunięciu zbędnej dla tego celu krótkoterminowej zmienności.[7] W ramach teorii analizy technicznej uważa się, ze w przebiegu wyceny akcji giełdowych zachodzą różne cykle. Wykorzystanie średnich kroczących od 5 do 40 dni jest uzasadnione chociażby próbą wyznaczenia cyklu miesięcznego. Średnie kroczące wyznaczają trend, więc największą

[7] J. J. Murphy, dz. cyt., s. 173-180

przydatność osiągają w przypadku obecności wyraźnego trendu wzrostowego lub spadkowego. Wystąpienie trendu horyzontalnego może osłabiać zdolność średnich kroczących do predykcji cen.[8] W przypadku gdy w długookresowym przebiegu wykresu cen nie występuje wyraźny trend zastosowanie znajdują oscylatory. Oscylatory przydają się też do wskazania krótkookresowych sytuacji skrajnego odejścia od trendu, gdy należy przewidzieć dobry moment do zakupu lub wykupu akcji giełdowych. Oscylator osiąga skrajne wartości gdy ceny zbyt szybko przemierzyły pewien zakres i należy spodziewać się ich korekty. Powinien on być traktowany jako uzupełnienie podstawowej analizy trendu.[9] Najbardziej podstawowym oscylatorem jest impet, który mierzy różnicę między ostatnią odnotowaną ceną zamknięcia i ceną zamknięcia sprzed ustalonej liczby dni.[10] Wskaźnik ten powoduje dwa problemy. Jego wartość jest bardzo podatna na gwałtowny wzrost lub spadek cen sprzed uwzględnianego do jego wyliczenia okresu nawet jeśli w badanym okresie ceny wykazują niewielkie zmiany. Zakres wartości wskaźnika, wynikający z bezwzględnych zmian wartości cen, nie jest stały, co utrudnia porównywanie różnych wartości tego samego wskaźnika między sobą. Problemy te rozwiązuje wskaźnik siły względnej (RSI), który jest podstawionym do określonego wzoru ilorazem średnich wartości wzrostu i spadku cen zamknięcia z ustalonej liczby ostatnich dni. Skala ustalonych wartości wskaźnika od 0 do 100 informuje o wyprzedaniu lub wykupieniu rynku, to znaczy odpowiednio nadmiernego odejścia obowiązującej ceny od potencjalnego trendu.[11] Innym oscylatorem jest CCI, który normalizuje różnicę między wartością wyliczaną na podstawie bieżącej ceny minimalnej, maksymalnej i zamknięcia, a ustaloną średnią kroczącą. Wartości spoza zakresu wskaźnika wysyłają sygnały kupna lub sprzedaży.[12] Rolę oscylatora może też pełnić porównanie między dwoma średnimi kroczącymi mierzącymi dwa okna czasowe o różnej długości czasu. Dodatnia wartość oscylatora jest osiągana w sytuacji gdy krótkookresowa średnia przewyższa długookresową. Moment, w którym dwie średnie osiągają tę samą wartość w danym punkcie w czasie oznacza sygnał do kupna lub sprzedaży. Przecięcie wykresu przebiegu w czasie długookresowej średniej przez ten sam wykres średniej krótkookresowej od dołu oznacza sygnał kupna, a od góry sygnał sprzedaży. Nadmierne oddalenie się krótkookresowej średniej od długookresowej oznacza krótkoterminowe odchylenie od trendu. Zbyt duże oddalenie się średnich może oznaczać zmianę trendu. W przypadku badania zbieżności i rozbieżności dla przebiegu w czasie dwóch wykładniczych

[8] J. J. Murphy, dz. cyt., s. 187, 188
[9] J. J. Murphy, dz. cyt., s. 197-199
[10] J. J. Murphy, dz. cyt., s. 199
[11] J. J. Murphy, dz. cyt., s. 210, 211
[12] J. J. Murphy, dz. cyt., s. 207

średnich kroczących mówimy o oscylatorze zbieżności-rozbieżności średniej kroczącej (MACD).[13] Innym przydatnym wskaźnikiem analizy technicznej są wstęgi Bollingera. Są to linie powyżej i poniżej przebiegu w czasie średniej wyceny danego aktywa, rozszerzające się lub kurczące w zależności od natężenia jej zmienności. Wyznaczają one obszar, w którym powinien się utrzymywać wykres ceny w zależności od czasu. Wyjście linii kursu poza ten obszar oznacza sygnał krótkotrwałego odwrócenia trendu. Sygnał kupna zostaje wygenerowany, gdy linia kursu spada poniżej dolnej wstęgi, a sprzedaży gdy rośnie powyżej górnej.[14,15] Wartości opisanych powyżej wskaźników zależą albo od trendu w jakim znajduje się dany wykres ceny albo od towarzyszącej mu zmienności. Wskaźnik Aroon pozwala stwierdzić czy analizowana akcja giełdowa rzeczywiście znajduje się aktualnie w trendzie, a jeśli tak to jak silny jest ten trend.[16] Wskaźniki mogą stanowić samodzielną wyczerpującą metodę aplikacji analizy technicznej. Dla analizy w interwałach dziennych ich wartość jest obliczana oddzielnie dla każdego dnia.

1.3 Skuteczność analizy technicznej w kontekście hipotezy rynku efektywnego

Od lat 60 ubiegłego wieku uwaga ekonomistów i inwestorów jest skupiona wokół dominującej w nurcie ekonomii neoklasycznej hipotezy rynku efektywnego. Wspomniana efektywność odnosi się do uwzględniania przez ceny aktywów finansowych wszystkich dostępnych o nich informacji uniemożliwiając tym samym osiągnięcie w długim okresie ponadprzeciętnych zysków z inwestycji względem rynku. Oznacza to możliwość osiągnięcia stóp zwrotu wyższych od obowiązujących indeksów giełdowych wyłącznie w wyniku adekwatnego zwiększania poziomu ryzyka inwestycji w teorii mierzonego w ramach modelu CAPM.[17] W 1970 Eugene Fama opublikował artykuł, w którym zdefiniował trzy najczęściej poruszane w dyskusji naukowej postulowane wersje hipotezy mówiące o tym w jaki dokładnie sposób rynek miałby być efektywny. W hipotezie słabej efektywności rynku dostępne informacje jakie

[13] J. J. Murphy, dz. cyt., s. 189, 198
[14] Wstęgi Bollingera, Edukacja giełdowa, online: https://www.edukacjagieldowa.pl/gieldowe-abc/analiza-techniczna/narzedzia-analizy-technicznej/wstega-bollingera/
[15] Wstęgi Bollingera w tworzeniu strategii inwestycyjnych, admirals, online: https://admiralmarkets.com/pl/education/articles/forex-indicators/wstega-bollingera
[16] Aroon Indicator – wskaźnik analizy technicznej, Trader's Area, online: https://tradersarea.pl/aroon-indicator-wskaznik-analizy-technicznej/
[17] A. G. Titan, The Efficient Market Hypothesis: Review of Specialized Literature and Empirical Research, Procedia Economics and Finance 32, 2015, s. 442-449

miałyby być uwzględniane w cenach dotyczą wszelkich wniosków płynących z historycznych cen, a więc hipoteza ta postuluje brak możliwości osiągnięcia ponadprzeciętnych zysków za pomocą metod analizy technicznej. Hipoteza półsilnej efektywności rynku zakłada, że ceny uwzględniają wszystkie publicznie dostępne informacje, a więc uwzględnia słabą teorię oraz rozszerza ją o wnioski płynące chociażby z analizy fundamentalnej spółki akcyjnej czy wszelkie inne opublikowane informacje, który mogą mieć wpływ na wyniki danego aktywa. Ostatnia silna hipoteza zakłada brak możliwości osiągnięcia ponadprzeciętnych zysków z obrotu aktywami finansowymi zarówno za pomocną publicznie dostępnych informacji jak i w wyniku tzw. „Insider Trading" czyli informacji dostępnych jedynie dla wybranych podmiotów.[18] Chociaż prawdziwość żadnej z hipotez rynku efektywnego nie jest rozstrzygnięta ponad wszelką wątpliwość i zmaga się ze znaczącą ilością krytyki i prób podważenia to jednak nie istnieje żadna konkretna równie poważana hipoteza, która stanowiłaby spójną alternatywę dla hipotezy rynku efektywnego.[19] Nie oznacza to jednak, że należy zaprzestać poszukiwań i nie przyjrzeć się różnorodnym propozycjom i wnioskom na ten temat wysuwanym przez różne prace badawcze. Hipotezy półsilne i silne nie są przedmiotem tej pracy, ponieważ nie dotyczą potencjalnej efektywności konstruowanego w rozdziale IV i V modelu, który bazuje jedynie na danych historycznych. W dalszej części tego podrozdziału zostanie przedstawiona synteza i wnioski płynące z krytyki dotyczącej słabej hipotezy rynku efektywnego oraz powiązanego z nią twierdzenia, że zależność wyceny inwestycji giełdowych w zależności od czasu wykazuje cechy błądzenia losowego. Należy mieć na uwadze, że słaba hipoteza ze względu na jej najbardziej zachowawczą formę cieszy się największym ze wszystkich trzech uznaniem w środowisku ekonomicznym. W wielu pracach naukowych dotyczących innych kwestii hipoteza ta jest przyjmowana a priori jako prawdziwa.[20]

Spośród przyczyn, które mogłyby potencjalnie prowadzić do niesłuszności hipotezy rynku efektywnego należy wymienić czynniki natury psychologicznej, które wpływają na bieżące krótkoterminowe decyzje inwestorów. [20] zauważa, że w literaturze psychologicznej istnieją istotne dowody na to, że przy sporządzaniu prognoz i osądów ludzie mają tendencję do przeceniania najnowszych danych. Jeżeli takie zachowanie jest widoczne na rynkach finansowych, wówczas w długim okresie powinien nastąpić powrót wartości stóp zwrotów z

[18] E. F. Fama, Efficient Capital Markets: A Review of Theory and Empirical Work, The Journal of Finance, 1970
[19] A. G. Titan, dz. cyt.
[20] S. Showalter, J. Gropp, Validating Weak-form Market Efficiency in United States Stock Markets with Trend Deterministic Price Data and Machine Learning, Department of Economics, DePauw University, Greencastle, IN 46135, USA, 2019, s.2

akcji, które odnotowały wyjątkowo dobre lub złe zwroty do średniej rynkowej.[21] Inną przyczyną może być asymetria informacji jakie posiadają poszczególni uczestnicy rynku. Osiągnięcie równowagi rynkowej zakłada osiągnięcie ważonego sumą prawdopodobieństw określonych scenariuszy konsensusu dla prawidłowej wyceny danego aktywa na podstawie dostępnych informacji. Problem pojawia się gdy jeden z uczestników rynku może dysponować przykładowo bardziej skutecznym modelem prognostycznym, co okazując się dopiero a posteriori, sprawia że jego ruchy arbitrażowe będą kontrowane przez uczestników rynku posiadających gorsze informacje. Warto też zwrócić uwagę na różne anomalie, które trwale utrzymują się na rynku mimo powszechnego przeświadczenia o ich istnieniu. Jedną z nich jest chociażby to że małe spółki konsekwentnie osiągają lepsze stopy zwrotu niż duże i nie są one wyjaśnione przez model CAPM[22].

W opracowaniu badawczym [20] dotyczącym walidacji słabej hipotezy rynku efektywnego pojawia się konkluzja, że dostępne odwołania do pozostałych prac w tym zakresie nie pozwalają rozstrzygnąć czy jest ona słuszna czy nie. Część rezultatów jest sprzecznych ze sobą. W związku z tym w ramach przywołanej pracy przeprowadzony został rozszerzony test Dickeya-Fullera z hipotezą zerową, której istotą jest brak informacji jakie może dostarczyć dla szeregu czasowego jego opóźnienie w czasie. Dane dla testu stanowiło 100 losowo wybranych akcji giełdowych pochodzących z indeksu S&P500 z okresu od 1 stycznia 2008 do 1 stycznia 2018. Hipoteza zerowa zostaje odrzucona wskazując na potencjalną stacjonarność badanego szeregu czasowego indeksu cen co jest niezgodne z hipotezą o błądzeniu losowym, które jest niestacjonarne. Wspomniany test nie pozwala wyciągnąć więcej wniosków niż potencjalna stacjonarność, więc nie można stwierdzić co to w praktyce oznacza dla określania przebiegu wartości cen i możliwości osiągnięcia ponadprzeciętnych zysków.[23] W publikacji [21] przywołany zostaje postulat, że iloraz wariancji dla dwóch różnych przedziałów czasowych w procesie błądzenia losowego powinien być w przybliżeniu proporcjonalny do stosunku ich długości w czasie. Test stosunku wariancji wykonany na głównych indeksach giełdowych oraz wybranych portfoliach dużych i małych spółek w latach 1962-1985 dostarcza zróżnicowanych rezultatów. Gdy bazowy okres dla pojedynczej stopy zwrotu wynosi jeden tydzień wszystkie wyniki nie są zgodne z hipotezą sugerującą błądzenie losowe. Gdy okres uwzględnianej stopy zwrotu zostaje podniesiony do 4 tygodni test nie odrzuca hipotezy sugerującej błądzenie losowe dla głównych indeksów i portfoliów złożonych z dużych spółek, ale odrzuca ją dla portfoliów

[21] S. Showalter, J. Gropp, dz. cyt., s.4
[22] W. Kenton, Small Minus Big (SMB): Definition and Role in Fama/French Model, Investopedia, 2020
[23] S. Showalter, J. Gropp, dz. cyt., s.8

złożonych z małych spółek. Rezultaty nie zmieniają się gdy brany pod uwagę dwudziestotrzyletni okres zostaje podzielony na dwa podokresy.[24] Rezultat osiągnięty w pracy [20] również skłania się ku odrzuceniu hipotezy zerowej testu stosunku wariancji. Może to sugerować występujące zjawisko dążenia cen akcji giełdowych do ich długookresowej średniej, co stwarzałoby możliwość jej prognozowania.

Proste modele uczenia maszynowego takie jak regresja logistyczna zastosowane w ramach [20] wytrenowane wyłącznie w oparciu o historyczne dane dotyczące cen nie wykazały mocy predykcyjnej niezbędnej do generowania ponadprzeciętnych stóp zwrotu. Z kolei badanie [22] korzystając z algorytmów sztucznej sieci neuronowej, wektorów nośnych SVM, lasu losowego i Naive Bayes osiągnęło za pomocą 10 wskaźników analizy technicznej dokładność między 74 a 90% w prognozowaniu kierunku ruchu cen w interwałach dziennych 2 wybranych indyjskich indeksów giełdowych oraz akcji giełdowych 2 wybranych spółek na podstawie danych z lat 2003-2012[25]. W ramach badania [23] za pomocą algorytmów sztucznej sieci neuronowej i wektorów nośnych SVM zbudowano dwa modele służące do predykcji kierunku ruchu cen w interwałach dziennych indeksu Istanbul Stock Exchange National 100 Index w latach 1997-2007. Również za pomocą 10 wskaźników analizy technicznej osiągnięto dokładność na poziomie odpowiednio 75% i 71%.[26] Krytyka obydwu badań wskazuje na brak odniesienia uzyskanych rezultatów do potencjalnego ryzyka oraz brak uzasadnienia dla wyboru wskazanych papierów wartościowych.[27]

Istotnym ograniczeniem dla prób osiągnięcia ponadprzeciętnych zysków względem indeksów rynkowych, ale i problemem dla słuszności hipotezy rynku efektywnego pozostaje kwestia opłat transakcyjnych. Konieczność uiszczania opłat przy każdej transakcji stanowi przeszkodę dla skuteczności aktywnych strategii inwestowania. Jak pisze [25] badania empiryczne pokazują że fundusze zarządzane aktywnie w krótkim okresie czasem osiągają stopy zwrotu z portfela wyższe niż wynikające z CAPM, ale w długim okresie są one niższe[28]. Wynika z tego że prób pokonania rynku należy szukać przede wszystkim w strategiach bazujących na buy-and-hold, a to dodatkowo utrudnia zadanie. Z drugiej strony różne opłaty

[24] A. W. Lo and A. C. Mackinlay, A Non-Random Walk Down Wall Street, Princeton University Press, 2001
[25] J. Patel, S. Shah, P. Thakkar and K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Systems with Applications 42, 2015, s. 259 – 268
[26] Y. Kara, M. A. Boyacioglu, Ö. K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange, Expert Systems with Applications 38, 2011, s. 5311-5319
[27] S. Showalter, J. Gropp, dz.cyt., s.4
[28] A. Sławiński, A. Chmielewska, Zrozumieć rynki finansowe, Polskie Wydawnictwo Ekonomiczne, Warszawa 2017

transakcyjne jakie ponoszą poszczególni uczestnicy rynku wprowadzają asymetrię w dostępie do rynku, która stanowi kolejny argument przeciwko hipotezie rynku efektywnego.

Niniejsza praca nie ma stanowić wyczerpującej próby weryfikacji słabej hipotezy rynku efektywnego jako że wydaje się że wymagałoby to szerszego ujęcia dostępnych indeksów giełdowych i kwestii ryzyka oraz dogłębnej analizy obowiązujących opłat transakcyjnych. Jednakże niejednoznaczne wnioski dotyczące obowiązywania tej hipotezy w różnych warunkach stwarzają szanse realizacji założonego celu pracy czyli osiągnięcia ponadprzeciętnych zysków z wykorzystaniem metody głębokiego uczenia przez wzmacnianie na podstawie danych historycznych, w szczególności biorąc pod uwagę potencjalnie zwiększone względem człowieka zdolności metod uczenia maszynowego do wykrywania pewnych wzorców[29], co może usprawnić aplikację analizy technicznej pogłębiając wyciągnięte wnioski i usuwając zbędne przesądy. Szczególną zaletą wykorzystywanego w niniejszej pracy rodzaju algorytmu głębokiego uczenia przez wzmacnianie jest możliwość wykorzystania go do wyznaczania precyzyjnej strategii obrotu akcjami giełdowymi, a nie jedynie przewidywanie prawdopodobieństwa danego kierunku, w którym zmieni się cena.

[29] G. Knowles, AI is better than humans at seeing patterns. Use it in L&D., LinkedIn, 2023

Rozdział II

Wprowadzenie do uczenia maszynowego i jego zastosowań w obrocie akcjami giełdowymi

2.1 Wprowadzenie do uczenia maszynowego

Uczenie maszynowe to dziedzina, która została zdefiniowana w ramach badań poświęconych rozwojowi szeroko pojętej sztucznej inteligencji. Stanowi zbiór algorytmów służących do analizy danych. Aby zrozumieć uczenie maszynowe należy odwołać się do ogólnie ujętego sposobu w jakim możemy modelować decyzje. Dysponując pewnymi informacjami, dla których należy uzyskać jakieś wnioski musimy stworzyć jakiś związek pomiędzy tym co dostarczamy na wejściu i otrzymujemy na wyjściu. Związek ten zazwyczaj przyjmuje postać pewnego zbioru reguł. W podejściu polegającym na wykorzystaniu poznanej w jakiś sposób wiedzy pomagającej podejmować właściwe decyzje do modelu dostarczane są informacje wejściowe oraz związane z nimi reguły postępowania, które poprzez zastosowanie na tych informacjach zapewniają oczekiwany efekt na wyjściu. W przypadku algorytmów uczenia maszynowego dostarczane są informacje, jednak wypracowanie reguł wiążących je ze sobą na wejściu i wyjściu następuje całkowicie w wyniku działania algorytmu, a nie w wyniku ingerencji zewnętrznej. Uczenie maszynowe można, więc zdefiniować jako rodzinę algorytmów budujących na podstawie przykładowych danych model podejmowania decyzji bez ingerencji zewnętrznej w kształtowanie związku między wspomnianymi danymi i decyzjami. Zadaniem modelu uczenia maszynowego może ale nie musi być przewidywanie założonych prawdziwych wartości wyjściowych. W zależności od tego czy zastosowano kryterium oczekiwanych rezultatów wyróżniamy uczenie nadzorowane i nienadzorowane. Poza tym wyróżnia się jeszcze uczenie przez wzmacnianie w którym nie dostarcza się zestawu danych uczących tylko środowisko opierające się na określonym zbiorze reguł. W uczeniu nadzorowanym modele uczą się na podstawie zestawu danych zawierających wejścia oraz odpowiadające im wyjścia w postaci etykiet. Celem jest nauczenie modelu generalizacji przydatnych do prognozowania wartości wyjść na nowych, niewidzianych danych. Przykłady zastosowań to klasyfikacja (np. rozpoznawanie spamu) i regresja (np. przewidywanie cen domów). Podstawowe elementy nadzorowanego uczenia maszynowego stanowią dane, wybrany algorytm uczący, funkcja straty oceniająca błąd między przewidywaniami a rzeczywistymi wynikami oraz metoda minimalizacji funkcji straty.

W uczeniu nienadzorowanym algorytmy analizują i grupują dane wejściowe bez etykiet, próbując odkryć ukryte wzorce lub struktury danych. Zastosowania obejmują grupowanie danych, redukcję wymiarowości oraz wykrywanie anomalii.[30] W uczeniu przez wzmacnianie model uczony jest poprzez interakcje ze środowiskiem, w którym dąży do maksymalizacji sumy nagród. Jest ono często stosowane w problemach decyzyjnych, takich jak gry czy robotyka, gdzie algorytm ma się nauczyć strategii osiągania celu.[31] Można jeszcze wyróżnić uczenie półnadzorowane, które łączy elementy uczenia nadzorowanego i nienadzorowanego, wykorzystując ograniczoną ilość etykietowanych danych wraz z dużą ilością danych nieetykietowanych, które wymagają od modelu przypisania im jakiejś etykiety.[32] Podzbiór algorytmów uczenia maszynowego stanowią algorytmy uczenia głębokiego. Są one przeznaczone dla wyjątkowo skomplikowanych zagadnień, w których zależność między danymi na wyjściu i wejściu jest modelowana za pomocą kolejno modelujących się pośrednich wartości, których część wynika z danych wejściowych, część wynika z pozostałych wartości pośrednich, a część dodatkowo określa dane wyjściowe. Przykładem algorytmu nadzorowanego uczenia głębokiego jest głęboka sieć neuronowa DNN, która zostanie omówiona w kolejnym rozdziale.

2.1.1 Podstawowe pojęcia uczenia maszynowego

Aby móc klarownie przyswoić opis procesu uczenia maszynowego i głębokiego należy być zaznajomionym z pewnym słowniczkiem pojęć, które stanowią normę w nomenklaturze dotyczącej omawianych algorytmów. Dane, które są używane do modelowania są dzielone w sposób uznany za optymalny na zbiór treningowy i testowy. W uczeniu nadzorowanym zbiór treningowy służy do wyznaczenia parametrów modelu, a zbiór testowy sprawdza jak model radzi sobie z estymacją wartości wyjściowych na podstawie danych wejściowych, które nie były użyte do jego trenowania. Ma to ogromne znaczenie dla prawidłowości estymacji dokonywanych przez model, ponieważ wyuczone reguły muszą być uogólnione w taki sposób, by sprawdzać się na całej populacji określonego rodzaju danych, a nie wyłącznie na związkach specyficznych dla zbioru treningowego. Wspomniane parametry odnoszą się do wartości

[30] Opracowanie własne na podstawie D. Soni, Supervised vs. Unsupervised Learning, Towards Data Science, 2018
[31] Opracowanie własne na podstawie 9 Real-Life Examples of Reinforcement Learning, Santa Clara University, online: https://onlinedegrees.scu.edu/media/blog/9-examples-of-reinforcement-learning, dostęp: 18.02.24 23:22
[32] A. Bewtra, The Ultimate Guide to Semi-Supervised Learning, v7labs, 2022,
online: https://www.v7labs.com/blog/semi-supervised-learning-guide dostęp: 19.02.2024 00:03

określanych przez sam algorytm w trakcie uczenia, z kolei hiperparametry odnoszą się do ustalonych ręcznie wartości, które wpływają na sposób funkcjonowania algorytmu. Sam proces uczenia oznacza określanie przez dany algorytm parametrów modelu w taki sposób, by wartości wyjściowe zwracane przez model jak najbardziej pokrywały się z założonymi prawdziwymi wartościami. Nauczenie się przez model zbyt szczegółowych reguł, które pozwalają na bardzo skuteczne wiązanie ze sobą danych w zbiorze treningowym i jednocześnie bardzo słabe wiązanie ze sobą danych w zbiorze testowym jest określane jako przeuczenie. Ze względu na konieczność doboru wartości hiperparametrów, które pozwolą na osiągnięcie optymalnej wartości funkcji straty, ale jednocześnie nie doprowadzą do przeuczenia się modelu często wyróżnia się też zbiór walidacyjny, który pozwala określić taką konfigurację hiperparametrów, która nie doprowadzi do przeuczenia się modelu na zbiorze danych treningowych i jednocześnie pozwala uniknąć dobrania wartości hiperparametrów, które doprowadzą do przeuczenia się modelu na zbiorze testowym.[33,34] Zbiór testowy pozwala na finalną ocenę wyuczonych na zbiorze treningowym uogólnionych reguł i dobranych hiperparametrów. W praktyce decyzja o wyodrębnieniu zbioru walidacyjnego będzie zależała od ilości dostępnych danych i typu rozwiązywanego problemu. Istnieją techniki umożliwiające stworzenie takiego zbioru na podstawie wybranych danych należących do zbioru treningowego.

Proces tworzenia modelu uczenia maszynowego będzie sprowadzał się do następujących kroków:

- zebranie i przygotowanie odpowiedniego zestawu danych
- wyodrębnienie kategorii danych takich jak poszczególne zmienne
- podział danych na zbiór treningowy, testowy i opcjonalnie walidacyjny
- analiza danych w celu oszacowania potencjalnych związków
- dobór typu algorytmu uczenia maszynowego adekwatnego dla zadanego problemu
- trening różnych konfiguracji modelu z różnymi wartościami hiperparametrów i opcjonalnie różnymi typami algorytmu na podstawie posiadanych danych
- ocena uzyskanych wyników i wybór najlepszego modelu
- ciągłe monitorowanie skuteczności modelu w trakcie dedykowanego zastosowania w celu wykrycia potrzeby aktualizacji modelu i użytych danych treningowych

[33] Opracowanie własne na podstawie J. Bergstra, Y. Bengio, Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research, 13, 2012, s. 282

[34] Opracowanie własne na podstawie I. Goodfellow , Y. Bengio, & A. Courville, Deep Learning, MIT Press, 2016, s. 118, 119

2.2 Zastosowania uczenia maszynowego w obrocie akcjami giełdowymi

Zastosowania modeli uczenia maszynowego w dziedzinie inwestycji kapitałowych i giełdowych są bardzo szerokie i większość z nich wychodzi poza temat niniejszej pracy.
Adekwatnym dla tematyki tej pracy przykładem jest prognozowanie cen jako szeregu czasowego. Do tego celu służą przede wszystkim modele takie jak ARIMA, GARCH czy rekurencyjne sieci neuronowe LSTM. Algorytmy uczenia maszynowego mogą służyć do zawierania transakcji z dużą szybkością w oparciu o wcześniej zdefiniowane kryteria, takie jak określone wzorce w danych. Określa się to mianem "Algorithmic Trading". Podobna reguła obowiązuje przy zastosowaniu tych algorytmów na dłuższych interwałach czasowych. Różnica polega tylko na tym, że w przypadku bardzo krótkiego interwału czasowego decyzje wyznaczane przez algorytmy muszą być aplikowane w ciągu ułamków sekund, więc ich kontrola staje się przez to ograniczona. Wprowadza się wtedy często takie mechanizmy jak "Stop Loss", które mają automatycznie zamknąć pozycję i wycofać środki gdy straty przekroczą określony poziom. Algorytmy mogą też służyć do wybiórczego dostrzegania pewnych wzorców w danych finansowych, umożliwiając wykrywanie określonych typów ryzyka dla zainwestowanych portfelów czy na przykład anomalie sugerujące próbę oszustwa. Z kolei algorytmy głębokiego uczenia przez wzmacnianie, które są przedmiotem tej pracy mogą również przewidywać wzorce cenowe w czasie i wykorzystać to by pomóc zoptymalizować alokację aktywów finansowych poprzez identyfikację właściwej strategii ich dywersyfikacji w celu maksymalizacji zysków i minimalizacji ryzyka. Algorytmy uczenia maszynowego mogą też wykraczać całkowicie poza sferę analizy danych cenowych rozszerzając znacząco swoją przydatność dla celów inwestycyjnych. Chociażby algorytmy przetwarzania naturalnego języka mogą dokonywać oceny nastrojów na rynku poprzez analizę wiadomości finansowych, mediów społecznościowych i raportów finansowych. Niniejsza praca skupi się na aplikacji metod głębokiego uczenia przez wzmacnianie w celu wyznaczenia optymalnej strategii alokacji środków pieniężnych w akcje giełdowe.

Rozdział III
Przegląd wiedzy o zastosowanych algorytmach

3.1 Głębokie sieci neuronowe (DNN)

3.1.1 Inspiracja dla DNN

Pomysł na algorytm głębokiej sieci neuronowej (DNN) wziął się z prób naśladowania procesów zachodzących w ludzkim mózgu. Inspiracją były biologiczne neurony i ich sposoby przetwarzania informacji. Neurony komunikują się ze sobą poprzez synapsy, czyli punkty połączeń, gdzie sygnały są przekazywane od jednego neuronu do drugiego, a siła synaptycznych połączeń może się zmieniać, co jest podstawą uczenia się i pamięci u człowieka. Choć temat ten jest często poruszany w opracowaniach tłumaczących podstawy działania algorytmu DNN należy traktować go jako anegdotę, gdyż opisywany model nie ma w praktyce nic wspólnego z neurobiologią i jest całkowicie matematyczną koncepcją. Jak pisze Francois Chollet w książce „Deep Learning with Python" cała mistyczna otoczka tego, że technologia ta przypomina działanie naszego mózgu, jest zbędna.[35]

3.1.2 Konstrukcja sieci DNN

Głęboka sieć neuronowa DNN składa się z trzech głównych rodzajów warstw: jednej warstwy wejściowej, co najmniej dwóch warstw ukrytych oraz jednej warstwy wyjściowej.[36] Matematycznie warstwy są reprezentowane przez tensory, które służą uogólnieniu takich obiektów jak skalar, wektor czy macierz. Uogólnienie polega na tym ze tensory o poszczególnej liczbie wymiarów mogą być reprezentowane przez wspomniane obiekty, a pozostałe tensory o innej liczbie wymiarów odzwierciedlają analogiczne obiekty dla danej liczby wymiarów. Jako że konstrukcje takie jak wektor czy macierz reprezentujące tensory odpowiednio pierwszego i drugiego rzędu można intuicyjnie postrzegać jako zbiory liczb o określonej strukturze, będą one pozwalały lepiej zrozumieć jaką strukturę liczb stanowią takie obiekty jak tensory trzeciego czy czwartego rzędu. Układ liczb stanowiących te obiekty można zwizualizować jako odpowiednio wektor macierzy oraz macierz macierzy. Należy jednak pamiętać, ze takie

[35] F. Chollet, Deep Learning with Python, Manning Publications Co., 2018, s.8
[36] Najbardziej uproszczony model perceptronu składa się z jednej warstwy ukrytej, ale formalnie nie stanowi to głębokiego uczenia

sformułowania stanowią obrazową intuicję, a nie definicję matematyczną, która pozwalałaby w ten sposób ekstrapolować właściwości jednych obiektów na drugie. Rzeczywista matematyczna natura tensorów wykracza poza tę intuicję, chociażby w kontekście wykonywanych na nich operacji.[37,38]

Warstwy są ze sobą połączone za pomocą parametrów, których proces obliczania dzieli się na propagację przednią i wsteczną. Zostały one dokładnie opisane poniżej.

3.1.3 Propagacja przednia

Propagacja przednia jest wykorzystywana do obliczenia wartości dla warstwy wyjściowej na podstawie wartości wprowadzonych do warstwy wejściowej poprzez zastosowanie obowiązujących wartości parametrów wag. Jest ona przeprowadzana zarówno w procesie uczenia jak i w trakcie wykorzystywania gotowego modelu.

Warstwa wejściowa w sieci neuronowej DNN służy do odbierania zbioru zmiennych objaśniających budowanego modelu i składa się z n neuronów. Neuron otrzymuje sygnały wejściowe i zwraca jakiś wynik. Każdy neuron warstwy wejściowej przyjmuje na wejściu skalar odpowiadający wartości danej zmiennej objaśniającej. Dane na wejściu dla wszystkich neuronów tej warstwy jednocześnie można przedstawić jako wektor $x \in \mathrm{R}^n$ gdzie każdy element wektora x_i odpowiada danej zmiennej objaśniającej:

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_n] \qquad (3)$$

gdzie:
n – liczba cech wejściowych

W przypadku wektora wejściowego zmiennych objaśniających dla więcej niż jednej próbki danych będzie on reprezentowany za pomocną macierzy o wymiarach $m \times n$, gdzie m jest wymiarem informującym, z której próbki danych pochodzą wartości dla danych zmiennych objaśniających[39].

[37] Opracowanie własne na podstawie S. Steinke, What's the difference between a matrix and a tensor?, Medium, 2017
[38] Opracowanie własne na podstawie M.A. Akivis i V.V. Goldberg, An Introduction to Linear Algebra and Tensors, Dover Publications, 2012, s. 50-54
[39] Dopuszczalne są również warstwy wejściowe o większej liczbie wymiarów, gdzie dodatkowe podlegające uczeniu warstwy mają za zadanie przekształcić je do tensora dwuwymiarowego przeznaczonego dla warstwy wejściowej DNN. Przykładem są konwolucyjne sieci neuronowe, które pozostają poza tematyką tej pracy.

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \qquad (4)$$

Dane na wyjściu neuronów warstwy wejściowej z reguły będą takie same jak na wejściu, opcjonalnie można przeprowadzić ich normalizację.

Warstwy ukryte i wyjściowa również będą składać się z pewnej liczby n neuronów, gdzie n może przyjmować różną wartość dla każdej warstwy. Aby na podstawie warstwy wejściowej obliczyć wartości w neuronach kolejnych warstwach sieci DNN należy użyć tak zwanych wag. Wagi są tensorami parametrów wyliczanych przez algorytm w celu wyznaczenia odpowiednich związków między neuronami sąsiednich warstw sieci neuronowej DNN. W przypadku obliczeń dotyczących pojedynczej próbki danych pojedynczy neuron pierwszej warstwy ukrytej będzie otrzymywał na wejściu iloczyn skalarny wektora warstwy wejściowej składającego się z n neuronów oraz odpowiadającemu mu wektora n parametrów wag dedykowanych specjalnie do przeprowadzenia obliczeń dla konkretnego neuronu w danej konkretnej warstwie. Każdy z neuronów warstw ukrytych oraz warstwy wyjściowej będzie posiadał swój własny dedykowany wektor wag służących do obliczenia jego wartości na wyjściu na podstawie wyjść z wszystkich neuronów poprzedniej warstwy.

Wektor wag prezentuje się następująco:

$$\mathbf{w}_i^{(j)} = \left[w_1^{(j)}, w_2^{(j)}, w_3^{(j)}, \ldots, w_n^{(j)}\right] \qquad (5)$$

gdzie:

i – numer obliczanego neuronu w kolejnej warstwie ukrytej (indeks dolny)

j – numer kolejnej warstwy ukrytej, do której należy obliczany neuron (indeks górny)

n – ilość neuronów w poprzedniej warstwie

Macierz wag służąca do obliczenia wejść do wszystkich neuronów w kolejnej warstwie prezentuje się następująco:

$$\mathbf{W}^{(j)} = \begin{pmatrix} w_{11}^{(j)} & \cdots & w_{1n}^{(j)} \\ \vdots & \ddots & \vdots \\ w_{m1}^{(j)} & \cdots & w_{mn}^{(j)} \end{pmatrix} \qquad (6)$$

gdzie:

m – ilość neuronów w kolejnej warstwie

n – ilość neuronów w poprzedniej warstwie

j – numer kolejnej warstwy wyjściowej

Skalar stanowiący wyjście z danego neuronu warstwy ukrytej lub wyjściowej będzie rezultatem tak zwanej funkcji aktywacji, która jako argument przyjmuje rezultat wspomnianego iloczynu skalarnego wyjść neuronów poprzedniej warstwy i odpowiedniego wektora wag. Funkcja aktywacji służy wprowadzeniu nieliniowej zależności między neuronami warstw i jest oznaczana symbolem sigma σ.

Wyróżniamy następujące najczęściej stosowane funkcje aktywacji:

-ReLU: $f(x) = \max(0, x)$ (7)

Zwraca tylko wartości nieujemne.

- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$ (8)

Zwraca tylko wartości w przedziale (0,1)

- Tanh: $f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$ (9)

Zwraca tylko wartości w przedziale (-1,1)

- Softmax: $f(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{K}(e^{x_j})}$ (10)

gdzie i = 1, 2, …, K oznacza numer neuronu w warstwie

Przydatna dla sieci z warstwą wyjściową składającą się z więcej niż jednego neuronu. Przekształca logity na rozkład prawdopodobieństwa.[40]

Funkcje Sigmoid i Softmax są przydatne w problemach klasyfikacji pozwalając przekształcić wartości warstwy wyjściowej na rozkłady prawdopodobieństwa przynależności do określonej klasy.

Wzór na wyjście i-tego neuronu pierwszej warstwy ukrytej dla pojedynczej próbki danych będzie się prezentować następująco:

$$h_i^{(1)} = \sigma\left(\mathbf{x} \cdot \mathbf{w}_i^{(1)}\right) \quad (11)$$

[40] Opracowanie własne na podstawie S. Sharma, Activation Functions in Neural Network, Towards Data Science, 2017, online: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, dostęp: 31.01.2024 22:05

gdzie:

$\mathbf{x}$ – wektor warstwy wejściowej

$\mathbf{w}_i^{(1)}$ – wektor wag przeznaczony dla obliczenia i-tego neuronu (indeks dolny) warstwy ukrytej numer 1 (indeks górny)

$h_i^{(1)}$ – skalar na wyjściu i-tego neuronu (indeks dolny) warstwy ukrytej numer 1 (indeks górny)

$\cdot$ – symbol iloczynu skalarnego wektorów

σ – funkcja aktywacji

Wektor wyjść neuronów danej j-tej warstwy ukrytej dla pojedynczej próbki danych będzie się prezentować następująco:

$$\mathbf{h}^{(j)} = \left[h_1^{(j)}, h_2^{(j)}, h_3^{(j)}, \dots, h_n^{(j)}\right] \qquad (12)$$

gdzie:

j – numer warstwy ukrytej

n – ilość neuronów w warstwie ukrytej

Wzór na wyjście i-tego neuronu j-tej warstwy ukrytej dla pojedynczej próbki danych:

$$h_i^{(j)} = \sigma\left(\mathbf{h}^{(j-1)} \cdot \mathbf{w}_i^{(j)}\right) \qquad (13)$$

gdzie:

$\mathbf{h}^{(j-1)}$ – wektor wyjść neuronów poprzedniej warstwy ukrytej numer j - 1 (indeks górny)

$\mathbf{w}_i^{(j)}$ – wektor wag przeznaczony dla obliczenia i-tego neuronu (indeks dolny) warstwy ukrytej numer j (indeks górny)

$h_i^{(j)}$ – skalar wyjścia i-tego neuronu (indeks dolny) kolejnej warstwy ukrytej numer j (indeks górny)

$\cdot$ – symbol iloczynu skalarnego wektorów

σ – funkcja aktywacji

Dla pojedynczej próbki danych wektor wyjść wszystkich neuronów określonej warstwy powstaje jako iloczyn wektora wyjść wszystkich neuronów poprzedniej warstwy oraz dedykowanej dla kolejnej warstwy macierzy wag.

Wzór na wektor wyjść wszystkich neuronów pierwszej warstwy ukrytej:

$$\mathbf{h}^{(1)^{\mathrm{T}}} = \sigma\left(\mathbf{W}^{(1)}\mathbf{x}^{\mathrm{T}}\right) \qquad (14)$$

gdzie:

$\mathbf{x}^{\mathrm{T}}$ – transponowany wektor warstwy wejściowej

$\mathbf{W}^{(1)}$ – macierz wag przeznaczona do obliczenia warstwy ukrytej numer 1 (indeks górny)

$\mathbf{h}^{(1)^{\mathrm{T}}}$ – transponowany wektor wyjść neuronów warstwy ukrytej numer 1 (indeks górny)

Wzór na wektor wyjść wszystkich neuronów j-tej warstwy ukrytej:

$$\mathbf{h}^{(j)^{\mathrm{T}}} = \sigma\left(\mathbf{W}^{(j)}\mathbf{h}^{(j-1)^{\mathrm{T}}}\right) \qquad (15)$$

gdzie:

$\mathbf{h}^{(j-1)^{\mathrm{T}}}$ – transponowany wektor wyjść neuronów poprzedniej warstwy ukrytej numer j – 1 (indeks górny)

$\mathbf{W}^{(j)}$ – macierz wag przeznaczona do obliczenia warstwy ukrytej numer j (indeks górny)

$\mathbf{h}^{(j)^{\mathrm{T}}}$ – transponowany wektor wyjść neuronów warstwy ukrytej numer j (indeks górny)

Wzór na wektor wyjść wszystkich neuronów warstwy wyjściowej:

$$\mathbf{y} = \sigma\left(\mathbf{W}^{(j+1)}\mathbf{h}^{(j)^{\mathrm{T}}}\right) \qquad (16)$$

gdzie:

$\mathbf{h}^{(j)^{\mathrm{T}}}$ – transponowany wektor wyjść neuronów poprzedniej ostatniej warstwy ukrytej numer j (indeks górny)

$\mathbf{W}^{(j+1)}$ – macierz wag przeznaczona do obliczenia warstwy wyjściowej, macierz jest oznaczona w indeksie górnym jako j+1, ponieważ numer ostatniej warstwy ukrytej został oznaczony jako j, więc potrzebna jest j+1 macierz wag

$\mathbf{y}$ – wektor wyjść neuronów warstwy wyjściowej

3.1.4. Propagacja wsteczna

Celem propagacji wstecznej jest określenie wartości wag, które pozwalają na podstawie danych wejściowych osiągnąć wartości danych wyjściowych możliwie zbliżone do ich prawdziwych wartości. Mechanizmem określającym stopień niedopasowania danych wyjściowych

zwróconych przez sieć do prawdziwych danych wyjściowych jest funkcja straty. Znalezienie wartości wag, które minimalizują wartość tej funkcji jest głównym zadaniem propagacji wstecznej osiąganym za pomocą algorytmu spadku gradientu.[41,42] Postać funkcji straty zależy od typu problemu jaki ma zostać rozwiązany przez DNN.

Wyróżniamy następujące najczęściej stosowane funkcje straty:

- Błąd Średniokwadratowy (MSE)

$$MSE(y, y') = \frac{1}{N} * \sum_{i=0}^{N} (y_i - y'_i)^2 \qquad (17)$$

gdzie N – liczba próbek danych

Stosowany w regresji do minimalizowania różnicy między przewidywanymi a rzeczywistymi wartościami.

- Entropia Krzyżowa (CE)

Wzór dla dwóch klas:

$$CE(y, y') = -\frac{1}{N} * \sum_{i=0}^{N} (y_i * \log(y'_i) + (1 - y_i) * \log(1 - y'_i)) \qquad (18)$$

Wzór dla wielu klas:

$$CE(y, y') = -\frac{1}{N} * \sum_{i=0}^{N} \sum_{j=0}^{C} y_{ij} * \log(y'_{ij}) \qquad (19)$$

gdzie C – liczba klas/neuronów w warstwie wyjściowej

Stosowana w problemach klasyfikacyjnych, mierzy różnicę między rozkładami prawdopodobieństwa.

Wartość y' w podanych wzorach na funkcję straty oznacza wartości wyjścia neuronów w warstwie wyjściowej sieci neuronowej DNN. Oznacza to że wartość y' określa się za pomocą wzoru 16. Pozwala to przyjąć, że funkcja straty L o jakiejś ustalonej postaci jest funkcją wszystkich wag L(W) w modelu DNN co z kolei pozwala na wyznaczenie gradientu funkcji straty dla wszystkich wag ∇L(W). Jest to potrzebne dla przeprowadzenia procedury algorytmu spadku gradientu. Aktualizacja każdej z wag w modelu DNN w ramach algorytmu spadku gradientu będzie dokonywana wykorzystując zależność przedstawioną poniższym wzorem:

$$w_{mnt}^{(j)} = w_{mn;t-1}^{(j)} - \alpha * \frac{dL}{dw_{mn;t-1}^{(j)}} \qquad (20)$$

[41] ang. Gradient Descent

[42] Opracowanie własne na podstawie F. Chollet, dz. cyt., s. 49, 60

gdzie:

α – współczynnik uczący o wartości $(0, 1]$ jako hiperparametr

$w_{mnt}^{(j)}$ – waga dla m neuronu kolejnej warstwy j, n neuronu poprzedniej warstwy j – 1 w t-tej iteracji propagacji wstecznej

Powyższy wzór oznacza, że gdy wzrost wartości danej wagi w określonym punkcie funkcji straty powoduje spadek jej wartości to jest on pożądany, ponieważ różnica między przewidywanymi i prawdziwymi wartościami wyjściowymi zmniejszyła się. W przeciwnym razie należy zmniejszyć wartość wagi. Prędkość zmian wartości wagi wynikająca z wartości $\frac{dL}{dw_{mn;t-1}^{(j)}}$ jest regulowana hiperparametrem w postaci współczynnika uczącego α.[43]

W faktycznym procesie uczenia przedstawiona powyżej postać wzoru spadku gradientu nie jest zbyt użyteczna. Łatwo wpada ona w lokalne minima funkcji straty co nie pozwala dotrzeć do jej globalnego minimum. Do tego współczynnik uczący o stałej wartości dla każdej wagi nie jest zbyt praktyczny. Istnieją różne algorytmy używane do aktualizacji parametrów wag. Jednym z najpopularniejszych algorytmów adresujących opisane problemy jest „Adam". Wprowadza on pojęcie momentum, które zastępuje $\frac{dL}{dw_{mn;t-1}^{(j)}}$ z wzoru 18 ważoną hiperparametrem β_1 sumą $\frac{dL}{dw_{mn;t-1}^{(j)}}$ i poprzedniego momentum. Standardowo zbliżanie się wartości wagi do jakiegoś optimum funkcji straty powoduje spadek wartości $\frac{dL}{dw_{mn;t-1}^{(j)}}$ i spowolnienie zmian wagi w kolejnych iteracjach algorytmu. Wprowadzenie momentum pozwala uśrednić zmiany wartości wagi w kolejnych iteracjach algorytmu propagacji wstecznej, co umożliwia wydostanie się z lokalnego minimum i eksplorację pozostałych możliwości. Innym usprawnieniem jest wprowadzenie parametru v_t, który pozwala modyfikować współczynnik uczący dla różnych wag. Poniżej znajduje się przedstawienie tego algorytmu.[44,45,46]

Wzór na momentum:

[43] Opracowanie własne na podstawie F. Chollet, dz. cyt., s. 49-52

[44] Opracowanie własne na podstawie Francois Chollet, dz. cyt., s. 51

[45] Opracowanie własne na podstawie Akash Ajagekar, Adam, Cornell University, 2021, online: https://optimization.cbe.cornell.edu/index.php?title=Adam, dostęp: 31.01.2024 22:19

[46] Opracowanie własne na podstawie Vitaly Bushaev, Adam – latest trends in deep learning optimization, Towards Data Science, 2018, online: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c, dostęp: 31.01.2024 22:27

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * \frac{dL}{dw_{mn;t-1}^{(j)}} \qquad (21)$$

gdzie β_1 jest hiperparametrem

$m'_t = \frac{m_t}{1-\beta_1{}^t}$ <- Uzgodnienie wartości zapobiegające dążeniu parametru do zera

Wzór na v_t:

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * \left(\frac{dL}{dw_{mn;t-1}^{(j)}}\right)^2 \qquad (22)$$

gdzie β_2 jest hiperparametrem

$v'_t = \frac{v_t}{1-\beta_2{}^t}$ <- Uzgodnienie wartości zapobiegające dążeniu parametru do zera

Finalny wzór „Adam":

$$w_{mnt}^{(j)} = w_{mn;t-1}^{(j)} - \alpha * \left(\frac{m'_t}{\sqrt{v'_t}} + e\right) \qquad (23)$$

gdzie e jest hiperparametrem

Dodatkowym usprawnieniem procesu uczenia jest regularyzacja L2, która dodaje kwadrat wartości wagi jako składnik do funkcji straty, karząc w ten sposób algorytm za zbyt duże wartości wag, które mogą prowadzić do powstania reguł dla danych treningowych, które są zbyt szczegółowe dla danych testowych i oznaczają przeuczenie.[47]

Mając wiedzę o procesach propagacji przedniej i wstecznej można przedstawić pełny przebieg uczenia się algorytmu. Faza treningu podzielona jest na epizody. W każdym epizodzie każda próbka danych treningowych musi jednokrotnie przejść przez propagację przednią w celu obliczenia prognozowanych wartości wyjściowych, a następnie przez propagację wsteczną w celu porównania ich z wartościami prawdziwymi i wyliczenia na tej podstawie nowych wag. Istnieją trzy podejścia do obliczania gradientu funkcji straty w ramach jednego epizodu zależnie od sposobu użycia dostępnych próbek danych:

- probabilistyczny gradient

Nowe wartości wag są obliczane odrębnie kolejno dla każdej próbki danych.

- gradient pełnego wsadu

[47] Opracowanie własne na podstawie S. Yıldırım, L1 and L2 Regularization — Explained, Towards Data Science, 2020, online: https://towardsdatascience.com/l1-and-l2-regularization-explained-874c3b03f668, dostęp: 31.01.2024 23:17

Nowe wartości wag są obliczane dla wszystkich próbek danych treningowych jednocześnie. Funkcja straty jest uśredniana.

- gradient mini-wsadu

Nowe wartości wag są obliczane odrębnie kolejno dla próbek danych treningowych pogrupowanych w określoną liczbę mniejszych wsadów.[48]

Optymalna liczba epizodów treningu pozwala na osiągnięcie docelowych wartości odrębnych od funkcji straty statystyk oceniających dopasowanie danych wyjściowych zwracanych przez siec neuronową DNN do danych prawdziwych. Gotowy algorytm DNN za pomocą propagacji przedniej pozwala nam obliczać przewidywane wartości wyjściowe dla danych dla których prawdziwe wartości nie są znane.

3.2 Uczenie przez wzmacnianie (RL)

3.2.1 Wprowadzenie

Uczenie przez wzmacnianie (RL) jest algorytmem uczenia maszynowego, który stanowi odrębną kategorię wobec uczenia nadzorowanego i nienadzorowanego. W przeciwieństwie do nich optymalizacja algorytmu nie opiera się na dopasowaniu parametrów modelu do danych treningowych, lecz ustalenia optymalnego sposobu postępowania na podstawie odgórnie określonego zbioru reguł. Algorytm początkowo próbuje znaleźć rozwiązanie metodą prób i błędów, by finalnie kierować się zdobytą wiedzą o korzyściach płynących z określonych decyzji[49].

3.2.2 Podstawowe pojęcia stosowane w nomenklaturze algorytmu

Znajomość poniższych pojęć wprowadzonych z ich unikalnym dla omawianego zagadnienia znaczeniem jest wymagana aby wytłumaczyć sposób funkcjonowania algorytmu[50]:

[48] Opracowanie własne na podstawie R. Odegua, Stochastic vs Mini-batch training in Machine learning using Tensorflow and python, Coinmonks, 2018, online: https://medium.com/coinmonks/stochastic-vs-mini-batch-training-in-machine-learning-using-tensorflow-and-python-7f9709143ee2, dostęp: 31.01.2024 22:45
[49] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 2018, s. 1, 2
[50] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 6, 7, 48-50, 55

Agent – obiekt uczący się podejmować decyzje
Stan – zbiór informacji o *Agencie* określony dla niego w danym czasie
Akcja – zbiór możliwych do podjęcia decyzji przez agenta w danym *Stanie* prowadzących do określenia nowego *Stanu*
Nagroda – informacja zwrotna dla *Agenta* będą wynikiem podjęcia danej *Akcji* w danym *Stanie*
Środowisko – zbiór wszystkich możliwych *Stanów*, *Akcji* i *Nagród* jakie mogą dotyczyć danego *Agenta*
Stopa dyskontowa – hiperparametr określający stopień zależności między wartością nagrody a odległością w czasie potrzebnym do jej uzyskania
Polityka – funkcja *Akcji* jakie podejmuje *Agent* w zależności od *Stanu*

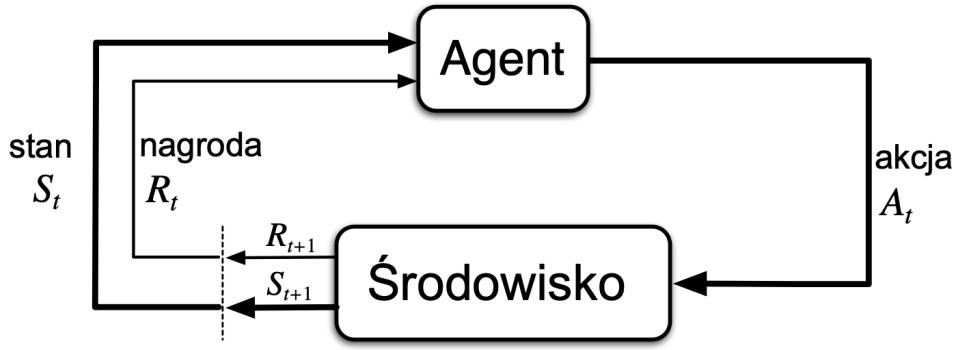
3.2.3 Proces decyzyjny Markowa (MDP)

Proces decyzyjny Markowa (MDP) to matematyczna struktura stanowiąca sekwencję podejmowanych decyzji, których rezultat jest częściowo losowy, a częściowo zależy od *Agenta* podejmującego decyzję[51]. Jest on traktowany jako matematycznie wyidealizowana forma problemu uczenia się przez wzmacnianie (RL) dla dyskretnej skończonej przestrzeni możliwych *Akcji* i *Stanów*[52]. Z tego powodu jest on użyty w niniejszej pracy jako przykład wyjaśniający podstawy teoretyczne funkcjonowania algorytmu RL.

W ramach MDP określamy *Środowisko* składające się ze zbioru możliwych *Stanów, Akcji i Nagród.* Dla każdego kroku w czasie określanego jako t, gdzie t = 0,1,2,3…, *Agent* otrzymuje określony *Stan* i w odpowiedzi musi wybrać odpowiednią *Akcję*. W wyniku wybrania danej *Akcji* krok później *Agent* otrzymuje *Nagrodę* i nowy *Stan*, które są zdefiniowane przez *Środowisko* jako odpowiedź na konkretną *Akcję* podjętą w konkretnym *Stanie*.
Celem *Agenta* jest maksymalizacja sumy *Nagród* osiągniętych we wszystkich krokach w czasie[53].

[51] Markov Decision Process, Wikipedia, online: https://en.wikipedia.org/wiki/Markov_decision_process, dostęp: 28.01.2024 23:32
[52] R.S. Sutton, A.G. Barto, dz. cyt., s. 47
[53] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 47-50

Rysunek 3 *Poglądowa ilustracja interakcji między Agentem i Środowiskiem*



Źródło: R.S. Sutton, A.G. Barto, dz. cyt., s. 48

Zbiór wybranych przez algorytm *Akcji* i uzyskanych Stanów i *Nagród* dla skończonej liczby kroków można przedstawić za pomocą poniższego ciągu:

$$S_0, A_0, R_1, S_1, A_1, \dots, R_T, S_T \; dla \; T \in \mathbb{C}_+ \qquad (24)$$

gdzie S_t, A_t, R_t to odpowiednio *Stan*, *Akcja* i *Nagroda* w danym kroku w czasie t, a t należy do przedziału $[0, \; T]$.
Skumulowaną nagrodę względem czasu t można przedstawić w następujący sposób:

$$G_t = R_{t+1} + R_{t+2} + \dots + R_T \qquad (25)$$

Uwzględniając stopę dyskontową γ taką że $0 \leq \gamma \leq 1$ można wprowadzić do powyższego wzoru zależność między wpływem wartości *Nagrody* na wartość skumulowanej nagrody a odległością w czasie potrzebnym do jej uzyskania. Uzyskany w ten sposób wzór prezentuje się następująco:

$$G_t = R_{t+1} + \gamma * R_{t+2} + \gamma^2 * R_{t+3} + \dots + \gamma^{T-1} * R_T \qquad (26)$$

Istotą MDP jest fakt, że *Stany* i *Nagrody* w kroku w czasie t+1, które są określone na podstawie *Stanów* i *Akcji* w kroku w czasie t nie są z nimi związane za pomocą funkcji deterministycznej lecz probabilistycznej, która określa z jakim prawdopodobieństwem *Środowisko* dla danego

Stanu i *Akcji* może zwrócić daną kombinację *Nagrody* i *Stanu*[54]. Opisana zależność prezentuje się następująco:

$$p(s', r|s, a) = P(S_{t+1} = s',\ R_{t+1} = r \mid S_t = s,\ A_t = a\ ) \qquad (27)$$

gdzie s', s, r, a są wszystkimi możliwymi *Stanami, Nagrodami i Akcjami*

Ostatecznym problemem jaki wynika z opisu MDP jest rozstrzygnięcie na jaką *Akcję* powinien zdecydować się *Agent* znajdujący się w danym *Stanie*. Zachowanie to będzie wynikało z *Polityki*, która jest niczym innym jak funkcją mapującą *Stan Agenta* na *Akcję* jaką on podejmie. Celem *Polityki* jest wybieranie takiej sekwencji *Akcji* w zależności od uzyskanych *Stanów*, by zmaksymalizować skumulowaną nagrodę. Wartość *Akcji* w danym *Stanie* będziemy postrzegać jako skumulowaną nagrodę uzyskaną we wszystkich kolejnych krokach w czasie przy założeniu nieustannego przestrzegania *Polityki* i określać to jako *Oczekiwana Skumulowana Nagroda*. Polityki dzielimy na:

- deterministyczne – jednoznacznie określają jaka *Akcja* zostanie podjęta w danym *Stanie*

$$\pi: S_t \rightarrow\ A_t \qquad (28)$$

- probabilistyczne – wybór *Akcji* jest funkcją prawdopodobieństwa *Stanu*

$$\pi(a|s) =\ P(A_t = a \mid S_t = s) \qquad (29)$$

Wybór *Akcji od Stanu* będzie zależeć od funkcji π lub $\pi(a|s)$ zależnie od tego czy chcemy wybrać konkretną *Akcję* czy przypisać określone prawdopodobieństwa dla wyboru poszczególnych *Akcji*[55]. Podejście probabilistyczne będzie dawać *Agentowi* możliwość skutecznej eksploatacji *Środowiska*, w którym kilka różnych wariantów *Stanów* i *Nagród* w odpowiedzi na daną *Akcję* w danym *Stanie* może być zwróconych z prawdopodobieństwem większym niż 0, gdyż wprowadza to niepewność odnośnie tego, które wybory okażą się optymalne z punktu widzenia celu *Polityki*. Odrębny problem stanowi to, że dopiero uprzednia możliwość eksploracji różnych wariantów daje *Agentowi* możliwość nauczenia się jakie

[54] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 48-49
[55] Od tego momentu stwierdzenie „wybór *Akcji*" w późniejszej części pracy oznacza nie tyle wybór konkretnej *Akcji* co zastosowanie funkcji *Polityki* w celu wyznaczenia kolejnej *Akcji*

Nagrody są związane z określonymi *Akcjami* w określonych *Stanach*. Prowadzi to do dylematu na jakiej podstawie *Polityka* powinna rozstrzygać o wyborze określonych *Akcji*[56] na etapie uczenia, gdy prawidłowe wartości *Oczekiwanych Skumulowanych Nagród* nie są jeszcze znane[57].

3.2.4 Funkcja wartości *Akcji* Q(s, a) i wyznaczanie optymalnej *Polityki*

Funkcja wartości *Akcji* służy do określania w jaki sposób *Polityka* powinna się zmieniać pod wpływem eksploracji nowych wariantów sekwencji *Stanów* i *Akcji*. Poniższy wzór określa *Oczekiwaną Skumulowaną Nagrodę* dla każdej możliwej *Akcji* jaką można wykonać w każdym możliwym *Stanie* w określonym kroku w czasie t[58]:

$$Q_\pi(s,a) = \mathrm{E}_\pi[G_t \mid S_t = s,\ A_t = a] = \mathrm{E}_\pi[\sum_{k=0}^{\infty} \gamma^k * R_{t+k+1} \mid S_t = s,\ A_t = a] \qquad (30)$$

Następujące równanie będzie prawdziwe:

$$Q_\pi(S_t, A_t) = R_{t+1} + \gamma * Q_\pi(S_{t+1}, A_{t+1}) \qquad (31)$$

Funkcja wartości *Akcji* na bieżąco stanowi informację o korzyściach płynących z określonych decyzji i stanowi podstawę kształtowania związku między konkretnymi *Stanami* i *Akcjami* przez *Politykę*. Przykładowo *Polityka* może zakładać, że zawsze wybiera w danym *Stanie Akcję* z najwyższą wartością Q(s,a). Jest to logiczne ponieważ Q(s,a) oznacza *Oczekiwaną Skumulowaną Nagrodę*, którą *Polityka* ma się kierować.
Dana polityka π' jest bezsprzecznie lepsza od polityki π wtedy i tylko wtedy, gdy dla wszystkich możliwych *Stanów* s i *Akcji* a prawdziwe jest równanie $Q\pi'(s,a) > Q\pi(s,a)$. Optymalna polityka jest lepsza lub równoważna wszystkim pozostałym i jest pewne, że istnieje, ale nie musi być unikalna[59].

Aby wyznaczyć optymalną *Politykę* należy oszacować wartości Q(s,a) dla wszystkich możliwych kombinacji *Stanów* i *Akcji*. Jest to problematyczne ze względu na fakt, że tak jak zostało opisane wyżej Q(s,a) dla danego *Stanu* i *Akcji* zależy od tego które kolejne *Akcje* zostaną

[56] Patrz przypis 51
[57] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 58
[58] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 58
[59] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 63

wybrane[60] i które kolejne *Stany* zostaną osiągnięte w wyniku stosowania wyznaczanej *Polityki*. *Agent* może wyznaczyć Q(s,a) analitycznie lub wielokrotnie powtórzyć sekwencję wyboru kolejnych *Akcji*[61] w określonej skończonej liczbie kroków w czasie w ramach metody Monte Carlo. Wykonanie jednego pełnego powtórzenia sekwencji zwanego epizodem pozwala dla każdego *Stanu* i *Akcji* uzyskać wiedzę, które kolejne *Akcje* zostały wybrane przez *Politykę* i które kolejne *Stany zostały osiągnięte* po wybraniu danej *Akcji* w danym *Stanie*, co umożliwia obliczenie Q(s,a) dla każdej ich kombinacji uzyskanej przez *Agenta* w danym epizodzie. W praktyce metoda Monte Carlo będzie preferowana jako że *Środowiska* przeznaczone dla algorytmu RL zazwyczaj są zbyt skomplikowane, by uzyskać jednoznaczne optymalne rozwiązanie[62]. Takie podejście rodzi pewne problemy, ponieważ *Agent* może wielokrotnie przechodzić przez daną kombinację *Stanu* i *Akcji* uzyskując różne skumulowane nagrody będące następstwem różnic w decyzjach podjętych w późniejszych krokach lub nawet różnic w wynikach wynikających z nieznanego probabilistycznego związku *Nagród* i *Stanów* w kroku t+1 z *Stanami* i *Akcjami* w kroku t. Istnieją różne sposoby podejścia do tych problemów. Sposobem radzenia sobie z różnicami w wynikach wynikającymi z nieznanego probabilistycznego związku *Nagród* i *Stanów* w kroku t+1 z *Stanami* i *Akcjami* w kroku może być zastosowanie hiperparametru uczącego Epsilon-Greedy pozwalające algorytmowi w fazie uczenia dokonać wyboru *Akcji*[63] postrzeganej w danym momencie jako optymalnej z prawdopodobieństwem 1 – epsilon oraz całkowicie losowej z prawdopodobieństwem epsilon. Pozwala to na poszukiwanie optymalnych rozwiązań, które cechują się niskim prawdopodobieństwem wystąpienia. Zazwyczaj optymalna jest zmienna wartość tego hiperparametru, która maleje do zera z każdym epizodem uczenia tak aby na początku promować swobodną eksplorację *Środowiska* przez *Agenta*, a następnie skupić się na eksploatacji *Środowiska* za pomocą oszacowanej optymalnej *Polityki*[64]. Z kolei do przykładowych sposobów radzenia sobie z następstwem różnic w decyzjach podjętych w późniejszych krokach należą strategia krocząca lub algorytmy tymczasowej różnicy.

Strategia kroczącego określania wartości danego Q(s,a) po każdym epizodzie metody Monte Carlo polega na wyznaczaniu jego aktualnej wartości jako dotychczasowa wartość Q(s,a) powiększona o różnicę wyznaczonej w danym epizodzie skumulowanej nagrody danej

[60] Patrz przypis 51
[61] Patrz przypis 51
[62] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 73
[63] Patrz przypis 51
[64] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 100-103

Akcji i *Stanu* i dotychczasowego Q(s,a) podzieloną przez numer epizodu[65]. Opisany wzór prezentuje się następująco:

$$Q(S_t, A_t) = Q(S_t, A_t) + \frac{1}{N} * \left(G_t - Q(S_t, A_t)\right) \qquad (32)$$

gdzie N – numer epizodu czyli jednokrotnego powtórzenia wszystkich kroków w czasie
W powyższym wzorze alternatywnie zamiast parametru 1/N można użyć stałego współczynnika uczącego alfa o wartościach z przedziału (0,1)[66]. Oprócz uzgadniania wartości Q(s,a) po każdym epizodzie pozostaje jeszcze kwestia powtórzenia kombinacji danej *Akcji* i *Stanu* w ramach tego samego epizodu. W tym przypadku można na przykład użyć wartości wynikającej z pierwszych odwiedzin lub średniej ze wszystkich odwiedzin, obydwie metody będą zapewniać konwergencję do optymalnego rozwiązania[67].

Algorytmy tymczasowej różnicy rozszerzają metodę Monte Carlo o możliwość wyliczania parametrów w oparciu o dotychczasowe estymacje parametrów. Różnią się one od strategii kroczącej brakiem konieczności wyliczania skumulowanej nagrody w celu oszacowania nowej wartości Q(s,a). Oznacza to że *Agent* nie musi czekać do końca epizodu, by poznać wszystkie kolejne decyzje i nagrody i móc zaktualizować aktualne oszacowanie Q(s,a), zamiast tego dokonuje się to każdorazowo podczas przechodzenia przez daną kombinację *Stanu* i *Akcji*. Algorytmy tymczasowej różnicy opierają się na wyliczaniu wartości Q(s,a) w oparciu o użycie dotychczas oszacowanych wartości Q(s, a) dla *Akcji* możliwych do podjęcia w *Stanach*, których osiągnięcie będzie skutkiem wyboru danej *Akcji*[68], dla której szacujemy nową wartość Q(s,a). Oznacza to, że poszczególne wartości Q(s,a) służą do wzajemnego wyliczania się wedle określone klucza. Pozwala to skorzystać podczas uczenia z dotychczasowej wiedzy o *Oczekiwanej Skumulowanej Nagrodzie* dla danych *Akcji* i *Stanów*, zamiast każdorazowo na nowo wyliczać efekty zastosowanej *Polityki*. Dla algorytmu tymczasowej różnicy w wariancie Sarsamax[69] wartość wyznaczonej w danym epizodzie skumulowanej nagrody z wzoru 30 na metodę kroczącą zostaje zamieniona na sumę nagrody

[65] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 30-32, 109, 110
[66] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 33
[67] R.S. Sutton, A.G. Barto, dz. cyt., s. 92
[68] Patrz przypis 51
[69] Istnieją różne warianty metody Sarsa, które różnią się sposobem wyboru wartości Q(s',a') w kroku t+1 służącej do obliczenia wartości Q(s,a) w kroku t

wynikającej z danej *Akcji* i maksymalnej wartość $Q(S_{t+1},A_{t+1})$[70] jaką posiada jedna z *Akcji* w *Stanie* osiągniętym w wyniku *Akcji* dla której szacujemy nowe $Q(S_t,A_t)$[71]. Uzyskany wzór prezentuje się następująco:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha * \left(R_{t+1} + \gamma * \max_{A_{t+1}} A(s)\, Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right) \qquad (33)$$

gdzie A(S) – zbiór wszystkich możliwych *Akcji*

Alternatywnie możemy użyć metody Expected Sarsa, która zamienia maksymalną wartość $Q(S_{t+1},A_{t+1})$ na wartość oczekiwaną w wyniku optymalnego wyboru *Akcji*[72] w kroku w czasie t+1[73]. To znaczy zamiast maksymalnej wartości *Akcji* przyjmujemy oczekiwaną wartość *Akcji* wynikającej z wyboru *Akcji* jaki dyktuje zastosowana *Polityka.* Uzyskany wzór prezentuje się następująco:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha * \left(R_{t+1} + \gamma * \sum_{A_{t+1}}^{A(S)} \pi(A_{t+1}|S_{t+1}) Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right) \qquad (34)$$

Tabela 1 *Porównanie metody Monte Carlo i tymczasowej różnicy*

Monte Carlo	Tymczasowa różnica
Wymagane poznanie skumulowanej nagrody z całego epizodu, aby zaktualizować parametry modelu	Aktualizacja parametrów modelu następuje po każdym kroku w czasie na podstawie wykorzystania przez model własnych estymacji
Znając prawdziwe skumulowane nagrody dobrze konwerguje do optymalnego rezultatu	Opierając się na niepewnych estymacjach model może być stronniczy i mieć problem z konwergencją do optymalnego rezultatu
Wysoka wariancja rezultatów ze względu na dużą ilość alternatywnych wariantów sekwencji *Agenta*	Dzięki estymacjom dobrze zachowuje wiedzę z pozostałych prób redukując wariancję rezultatów

[70] Parametr ten stanowi *Oczekiwaną Skumulowaną Nagrodę* w kroku w czasie o jeden dalej niż *Oczekiwana Skumulowana Nagroda* jaką staramy się oszacować, dlatego należy uwzględnić ustalony hiperparametr stopy dyskontowej
[71] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 129-132
[72] Patrz przypis 51
[73] Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 133, 134

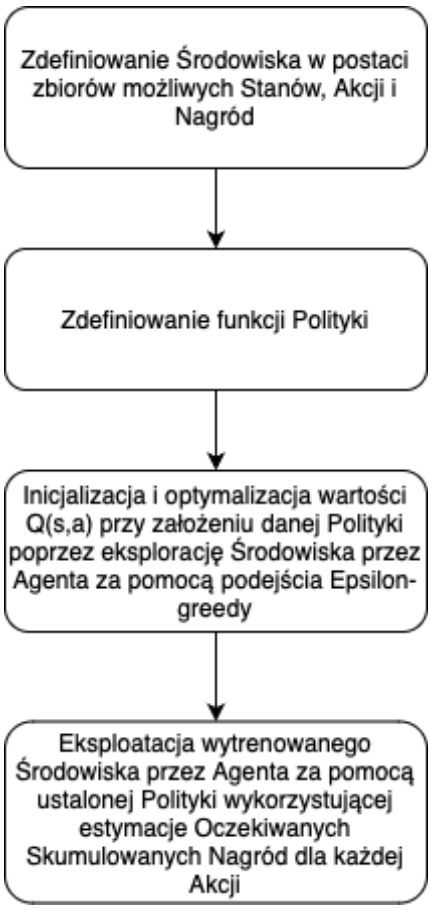
Źródło: Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 33, 92, 129-134

Poniższy wzór stanowi konstatację dotychczasowych wzorów przedstawiając *Oczekiwaną skumulowaną nagrodę* w danym *Stanie* przy założeniu przestrzegania ustalonej *Polityki*. Ustalona *Polityka* z określonym prawdopodobieństwem zwraca na podstawie *Stanu* podjętą *Akcję*, która z określonym prawdopodobieństwem skutkuje zwróceniem przez *Środowisko* określonego nowego *Stanu* i *Nagrody* wraz z pozostałą *Oczekiwaną skumulowaną nagrodą* wynikającą z dalszego przestrzegania *Polityki*:

$$\mathrm{E}_{\pi}[G_t \mid S_t = s] = \sum_{a}^{A(S)} \pi(a|s) \sum_{s'}^{S} \sum_{r}^{R} p(s', r|s, a)[R_{t+1} + \gamma * Q_{\pi}(S_{t+1}, A_{t+1})] \quad (35)$$

gdzie S – zbiór wszystkich możliwych *Stanów* R – zbiór wszystkich możliwych *Nagród*

Rysunek 4 *Poglądowy przebieg algorytmu RL dla MDP*



Źródło: Opracowanie własne na podstawie R.S. Sutton, A.G. Barto, dz. cyt., s. 1, 2, 6, 7, 30-33, 48-50, 55, 58, 63, 73, 92, 100-103, 109, 110, 129-132, 133, 134

Tak jak opisane na początku tego podrozdziału *Środowisko* w ramach MDP składa się ze skończonego dyskretnego zbioru możliwych *Stanów*. W przypadku praktycznych zastosowań, dla których zbiór możliwych *Stanów* jest ciągły istnieją dwie możliwości: dyskretyzacja ciągłej przestrzeni oraz aproksymacja funkcji mapującej ciągły *Stan* na *Akcję.* W praktyce obydwie metody przysparzają sporych problemów. Omówiona w kolejnym podrozdziale metoda głębokiego uczenia przez wzmacnianie DRL stanowi właśnie aproksymację takiej funkcji i umożliwia najbardziej efektywne uczenie się *Agenta* na bazie *Środowiska* z ciągłym zbiorem wartości *Stanów.*[74]

3.3 Głębokie uczenie przez wzmacnianie (DRL)

Algorytm głębokiego uczenia przez wzmacnianie (DRL) jest połączeniem algorytmów głębokiego uczenia (DL) i uczenia przez wzmacnianie (RL). Dotychczasowe podejście w algorytmie RL polegające na eksplorowaniu *Środowiska* w celu estymowania parametrów Q(s,a) będzie się dokonywać poprzez skojarzenie ze sobą *Stanów* i Akcji za pomocą głębokiej sieci neuronowej DNN. Zbiór algorytmów głębokiego uczenia przez wzmacnianie dzieli się na metody stosujące sieć przewidują wartości Q(s,a) tak zwaną sieć krytyka, metody stosujące sieć wyznaczającą gotową *Politykę* tak zwaną sieć aktora oraz metody łączące zastosowanie obydwu tych sieci.

3.3.1 Krytyk

W sieci neuronowej krytyka warstwą wejściową do sieci jest *Stan*, którego wartość musi być reprezentowana poprzez jednowymiarowy tensor. Wyjściem są Q(s,a) dla każdej z możliwych do podjęcia *Akcji*. Funkcja straty jest zbudowana w oparciu o różnicę między estymowanymi przez sieć wartościami Q'(s,a,W), a prawdziwymi wartościami Q_π(s,a) przy założeniu przestrzegania danej *Polityki*:

$$L(W) = \mathrm{E}_\pi\left[\left(Q_\pi(S_t, A_t) - Q'(S_t, A_t, W)\right)^2\right] \qquad (36)$$

[74] Opracowanie własne na podstawie Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver & Daan Wierstra, Continous Control with Deep Reinforcement Learning, Google Deepmind, London, UK, 2016, online: https://arxiv.org/abs/1509.02971

gdzie W - wagi w modelu sieci neuronowej DL

Prawdziwe wartości $Q_\pi(S_t, A_t)$ nie są znane, do ich obliczenia używany jest wzór Sarsamax, który wymaga znajomości wartości $Q_\pi(S_{t+1}, A_{t+1})$. Wartości te potencjalnie mogą być już znane, ponieważ obliczyła je sieć neuronowa, która została użyta do estymowania prawdziwych wartości każdego $Q_\pi(s, a)$. Wynika z tego że sieć sama musi sobie wyliczyć założone prawdziwe wartości wyjściowe, do których aspirować mają jej estymowane wartości wyjściowe[75]:

$$Q_\pi(S_t, A_t) = R_{t+1} + \gamma * \max_{A_{t+1}} A(S)\, Q'(S_{t+1}, A_{t+1}, W) \qquad (37)$$

Doprowadza to do problemu, w którym założone prawdziwe wartości $Q_\pi(s,a)$ w funkcji straty zmieniają się wraz z aktualizacją parametrów sieci DL. By tego uniknąć wartości $Q_\pi(s, a)$ są wyznaczane do gradientu funkcji straty w danej iteracji propagacji wstecznej algorytmu sieci DL na podstawie odrębnej sieci DL tzw. „sieci celu", która posiada nieaktualne parametry domyślnego modelu sieci DL aktualizowane co kilku iteracji algorytmu. Mechanizm nosi nazwę „stałych Q-celów".[76] Rezultatem jest poniższy wzór funkcji straty:

$$L(W) = \mathrm{E}_\pi \left[\left(R_{t+1} + \gamma * \max_{A_{t+1}} A(s)\, Q'(S_{t+1}, A_{t+1}, W^-) - Q'(S_t, A_t, W) \right)^2 \right] \qquad (38)$$

gdzie W^- - wagi w modelu DL „sieci celu"

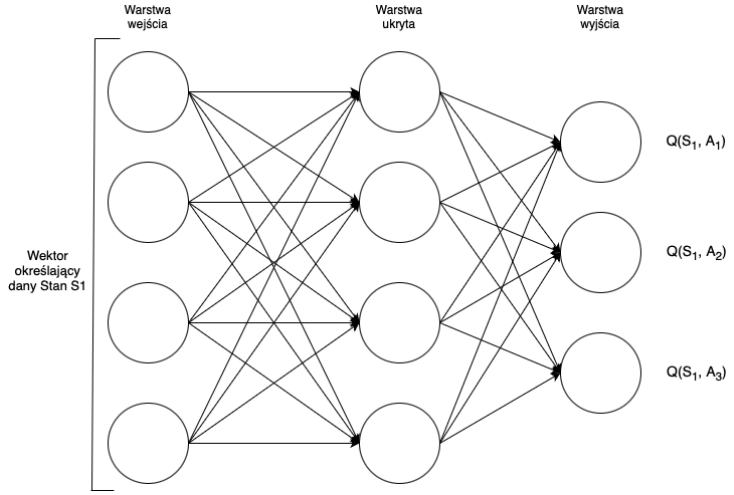
Istotną cechą uczenia się algorytmu DRL jest fakt, że mimo procedury aktualizacji wag sieci DNN przeprowadzanej w każdym kolejnym kroku w czasie, zastosowaną metodą tej aktualizacji nie jest stochastyczny gradient lecz gradient mini-wsadu. To znaczy, że w każdym kolejnym kroku w czasie *Agent* poza aktualnym doświadczaniem *Nagrody* w wyniku podjęcia *Akcji* w danym *Stanie* potrzebuje użyć przynajmniej kilkudziesięciu innych kombinacji *Stanów*, *Akcji* i *Nagród*, by skonstruować wsad do sieci. Do tego celu służy „bufor powtórzeń", który zapamiętuje wszystkie kombinacje $(S_t, A_t, R_{t+1}, S_{t+1})$ zwane doświadczeniami, które w danym epizodzie odwiedził *Agent*. Bufor ma określony limit, jeśli jest on przekroczony najwcześniej dodane doświadczenia są usuwane zgodnie z algorytmem kolejki FIFO. Bufor służy do tego,

[75] Przy założeniu, że prawdziwe wartości $Q_\pi(s,a)$ wynikają z metody Sarsamax

[76] Opracowanie własne na podstawie Miguel Morales, Grokking Deep Reinforcement Learning, Manning Publications Co., 2020, s. 276-280

by za każdym razem losować z niego mini-wsad. Wielkość bufora i mini-wsadu są hiperparametrami modelu.[77]

Rysunek 5 *Sieć neuronowa DNN[78] krytyka w głębokim uczeniu przez wzmacnianie DRL dla Stanu z trzema możliwymi Akcjami do podjęcia*



Źródło: Opracowanie własne

3.3.2 Aktor

Alternatywnie dla przewidywania wartości funkcji Q(s,a) można zastosować na wyjściu sieci DNN funkcję aktywacji softmax tak by optymalizować algorytm pod wyznaczanie gotowej *Polityki* w postaci prawdopodobieństwa wyboru określonej *Akcji*. Przykładem takiego algorytmu jest algorytm „REINFORCE", którego celem jest wyznaczyć taką *Politykę*, która maksymalizuje zdobytą nagrodę. Sumę nagród oznaczamy w następujący sposób:

$$R(\tau) = \sum_{t=0}^{T-1} R(S_t, A_t) \qquad (39)$$

gdzie $\tau = (S_0, A_0, \ldots, S_{T-1}, A_{T-1})$ oznacza zbiór *Stanów* i *Akcji* ze wszystkich kroków czasowych, dla których została wyznaczona nagroda

[77] Ryan Sander, Introduction to Experience Replay for Off-Policy Deep Reinforcement Learning, Towards Data Science, 2022, online: https://towardsdatascience.com/a-technical-introduction-to-experience-replay-for-off-policy-deep-reinforcement-learning-9812bc920a96, dostęp: 02.02.2024 17:01
[78] Uproszczona na rysunku do jednej warstwy ukrytej

Celem jest maksymalizacja funkcji celu $J(\pi_W)$ w postaci wartości oczekiwanej zbioru nagród $R(\tau)$ uzyskanych dla *Polityki* wyznaczonej przez siec z parametrami W:

$$J(\pi_W) = \mathrm{E}_{\pi_W}[R(\tau)] \qquad (40)$$

Jako że *Polityka* zwracana przez sieć określa prawdopodobieństwa określonych decyzji skutkującymi różnymi możliwymi sekwencjami Stanów i Akcji τ, wartość oczekiwaną nagród będziemy rozumieli jako sumę nagród dla danych sekwencji τ przemnożoną przez prawdopodobieństwo osiągnięcia tej sekwencji przy danych wagach modelu:

$$J(\pi_W) = \sum_{\tau} P(\tau|W)R(\tau) \qquad (41)$$

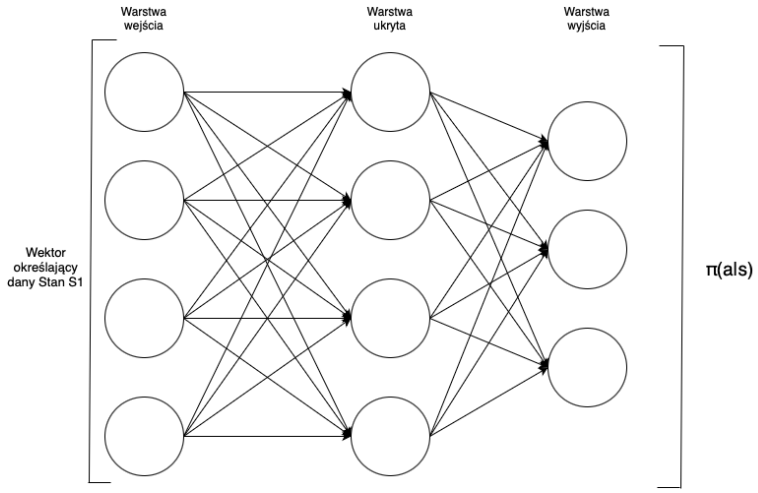
Aby znaleźć globalne optimum funkcji należy wyznaczyć jej gradient względem wag sieci. Algorytm nosi nazwę wzniesienia gradientu i jest analogiczny do spadku gradientu, po prostu zamiast globalnego minimum wyznacza się globalne maksimum funkcji. Algorytm „REINFORCE" stanowi przykład metody Monte Carlo, a nie tymczasowej różnicy, ponieważ aby poznać wszystkie *Nagrody* służące do aktualizacji parametrów modelu, musimy najpierw poznać całą sekwencję wybranych Akcji oraz uzyskanych Stanów i Nagród. Ostateczny wzór gradientu funkcji $J(\pi_W)$ będzie uwzględniał sumę gradientów dla wszystkich kroków w czasie poprzez rozbicie czynnika $P(\tau|W)$ na prawdopodobieństwa wyboru określonych *Akcji* przez *Politykę* zależnych od prawdopodobieństwa *Stanów* zwróconych przez *Środowisko* w odpowiedzi na te *Akcje.* Takie podejście będzie wymagało eksploracji różnych możliwych sekwencji zależnych od realizacji określonych prawdopodobieństw, a więc wprowadzenia N prób uzyskania różnych sekwencji *Stanów* i *Akcji.* Finalny wzór $J(\pi_W)$ po różnych przekształceniach będzie wyglądał następująco[79] [80]:

$$J(\pi_W) = \frac{1}{N} * \sum_{\tau}^{N} \sum_{t=0}^{T-1} (\log \pi_W(A_t, S_t) + \log P(S_{t+1}|S_t, A_t))R(\tau) \qquad (42)$$

[79] Opracowanie własne na podstawie Dhanoop Karunakaran, „REINFORCE — a policy-gradient based reinforcement Learning algorithm", Intro to Artificial Intelligence, 2020, online: https://medium.com/intro-to-artificial-intelligence/reinforce-a-policy-gradient-based-reinforcement-learning-algorithm-84bde440c816, dostęp: 31.01.2024 21:15

[80] Opracowanie własne na podstawie Miguel Morales, dz. cyt., s. 340-349

Rysunek 6 *Sieć neuronowa DNN[81] aktora w głębokim uczeniu przez wzmacnianie DRL dla Stanu z trzema możliwymi Akcjami do podjęcia i Polityką określoną przez warstwę wyjściową*



Źródło: Opracowanie własne

3.3.3 Aktor-Krytyk

W niniejszym podrozdziale zostały omówione metody DRL służące do przewidywania *Oczekiwanej Skumulowanej Nagrody* lub optymalnej *Polityki*. Zbiór metod Aktor-Krytyk stanowi połączenie tych dwóch metod, a jego istotą jest skonstruowanie dwóch odrębnych sieci neuronowych DNN[82], które wspierają się wzajemnie w procesie uczenia, lecz jedna z nich przewiduje *Oczekiwanej Skumulowaną Nagrodę*, a druga optymalną *Politykę*. Tytułowy Aktor to opisany powyżej algorytm do wyznaczania gotowej *Polityki* w postaci prawdopodobieństwa wyboru określonej *Akcji*, z kolei Krytyk to opisany na początku podrozdziału algorytm estymujący wartości funkcji *Akcji* Q(s,a). Wartości wyjściowe obydwu sieci służą dla siebie nawzajem w celu wyliczenia funkcji straty każdej z nich. Estymacje optymalnej *Polityki* wynikają z oszacowań Q(s,a) drugiej sieci, a tak wyznaczona *Polityka* pozwala poruszać się *Agentowi* w celu dokonywania kolejnych oszacowań Q(s,a) zamiast poruszać się na podstawie właśnie tych oszacowań. Żadne z tych podejść nie odchodzi całkowicie od schematu częściowego utwierdzania się algorytmów w błędnych przekonaniach, ale mechanizm zdobywania z każdym krokiem kolejnej *Nagrody* pozwala konsekwentnie dążyć do poprawnego rozwiązania. Powstaje pytanie, że skoro zarówno metody Krytyka jak i Aktora-

[81] Uproszczona na Rysunku do jednej warstwy ukrytej
[82] Omówiona wcześniej metoda „stałych Q-celów" nie jest metodą Aktor-Krytyk, ponieważ przyczyna skonstruowania dwóch sieci DNN jest w tym przypadku zupełnie inna. Jest to o tyle istotne, że obydwie metody będą stosowane razem.

Krytyka kierują się początkowo błędnymi estymacjami to jak się to ma do metody Aktora, która jest przecież metodą Monte Carlo. Podstawową opisaną w podrozdziale 2.2 różnicą między metodami Monte Carlo, a tymczasowej różnicy była ilość kroków w czasie jakie musiał wykonać *Agent*, by algorytm mógł zaktualizować swoje parametry. W przypadku metod Aktor-Krytyk istnieje możliwość całkowitej modulacji tego aspektu. Algorytm może kierować się wyłącznie swoimi estymacjami w następnym kroku lub przechodzić wszystkie kroki w czasie, by poznać faktyczną skumulowaną nagrodę, ale może także na przykład przejść t kroków w czasie zamiast jednego, a następnie posłużyć się swoją estymacją, którą zastanie t kroków dalej. Może też wykorzystać wszystkie możliwe kombinacje to znaczy wykonać jeden krok i posłużyć się gotową estymacją, wykonać dwa kroki i posłużyć się gotową estymacją, …, wykonać T kroków i poznać faktyczną skumulowaną nagrodę, a następnie uśrednić wyniki z każdego z tych podejść. Dokładna konfiguracja będzie zależała od wybranej wersji algorytmu.[83]

3.3.4 Głęboki gradient deterministycznej *Polityki* (DDPG)

Głęboki gradient deterministycznej *Polityki* jest przykładem algorytmu DRL w wersji Aktor-Krytyk. Składa się z dwóch sieci DNN, z których jedna wyznacza optymalną *Politykę* nakazującą wybór jednej konkretnej *Akcji*, a więc *Politykę* deterministyczną π, a nie *Politykę* probabilistyczną $\pi(a|s)$, a druga wyznacza wartość Q(s, a) dla tej konkretnej jednej *Akcji*, którą wskazała deterministyczna *Polityka*. Celem takiego podejścia jest możliwość wykorzystania neuronów warstwy wyjściowej sieci wyznaczającej optymalną *Politykę*, by zamiast wskazywać prawdopodobieństwa danych *Akcji* wskazywały ciągłą wartość każdego z elementów wektora jednej konkretnej *Akcji.* Pozwala to by podjęcie *Akcji* zamiast być zdarzeniem zerojedynkowym oznaczało podjęcie *Akcji*, ale „w jakimś stopniu”. Sprawia to, że zbiór wartości *Akcji* może stać się zbiorem ciągłym co znacząco zwiększa liczbę potencjalnych praktycznych zastosowań takiego algorytmu. Z tego też powodu algorytm DDPG oprócz należenia do zbioru algorytmów DRL typu Aktor-Krytyk należy też do zbioru algorytmów ciągłej kontroli DRL (DRL-CC).[84]

Tak jak zostało opisane w metodzie Aktor-Krytyk wartości wyjściowe obydwu sieci służą dla siebie nawzajem w celu wyliczenia funkcji straty każdej z nich. Sieć neuronowa DNN Aktora na podstawie danego *Stanu* t zwraca deterministycznie określoną *Akcję* do podjęcia. Następnie *Środowisko* na podstawie tej *Akcji* zwraca *Stan* t+1 i *Nagrodę*. Z kolei sieć

[83] Opracowanie własne na podstawie Miguel Morales, dz. cyt., s. 350, 351, 362, 363
[84] Opracowanie własne na podstawie

neuronowa DNN Krytyka na podstawie danego Stanu t i określonej przez sieć Aktora *Akcji* zwraca wartość Q(s,a) tej *Akcji* w tym *Stanie.* Jako, że celem całego algorytmu RL jest wybór *Akcji* o jak najwyższym Q(s,a) to sieć Krytyka staje się tym samym funkcją celu $J(W_A)$ sieci Aktora. Aby wyliczyć gradient Q(s,a) sieci Krytyka względem wag sieci Aktora wprowadza się gradient łańcuchowy, w którym gradient Q(s,a) zależy od wektora ciągłych wartości reprezentujących daną *Akcję*, a wektor ten jest zwracany przez sieć Aktora, więc zależy od wag sieci Aktora.[85,86] Przedstawia to poniższy wzór:

$$\frac{dJ(W_A)}{dW_A} = \frac{dQ(s,a,W_K)}{d\pi(s,W_A)} * \frac{d\pi(s,W_A)}{dW_A} \qquad (43)$$

gdzie:

W_A – wagi sieci Aktora

W_K – wagi sieci Krytyka

$a = \pi(s,W_A)$

Finalna postać wzoru będzie jeszcze doprecyzowana o fakt, że stanowi wartość oczekiwaną powyższego równania dla wielu *Stanów* z mini-wsadu doświadczeń zwracanego z bufora powtórzeń, który jest wykorzystywany również w ramach algorytmu DDPG.

Funkcja straty sieci Krytyka będzie analogiczna do wzorów 32 i 33. Jedyna różnica polega na tym, że zamiast wyznaczania A_{t+1} z wzoru 33 za pomocą Sarsamax należy wyliczyć tę wartość z sieci Aktora posiłkując się *Stanem* t+1 zwróconym przez *Środowisko* w odpowiedzi na *Akcję*, której wartość Q(s,a) stanowi estymowaną wartość wyjściową sieci Krytyka uwzględnianą w jego funkcji straty. Fakt, że funkcja straty sieci Krytyka jest analogiczna do wzorów 34 i 35 powoduje, że potrzebna jest również funkcja sieci celu Krytyka. Ponadto wspomniane wyznaczenie A_{t+1}, które wymaga użycia sieci Aktora nie jest wykonywane w metodzie DDGP przez faktyczną sieć Aktora, ale przez jeszcze jedną sieć, która analogicznie do sieci celu Krytyka jest historyczną kopią sieci Aktora. Poniżej znajduje się podsumowanie funkcji spełnianej przez każdą z sieci DNN konstruowanych w ramach DDGP:

- Sieć Aktora: Wyznacza deterministyczną *Politykę* informując *Agenta* jaką konkretną *Akcję* ma podjąć
- Sieć Krytyka: Stanowi funkcję celu dla sieci Aktora

[85] Opracowanie własne na podstawie Timothy P. Lillicrap, Jonathan J. Hunt, ..., Daan Wierstra, dz. cyt. oraz Miguel Morales, dz. cyt., s. 377-384

- Sieć celu Aktora: Dostarcza elementy potrzebne do wyliczenia funkcji straty sieci Krytyka
- Sieć celu Krytyka: Również dostarcza elementy potrzebne do wyliczenia funkcji straty sieci Krytyka[87]

Wzór na funkcję straty sieci Krytyka:

$$L(W_K) = (R_{t+1} + \gamma * Q'(S_{t+1}, \pi(S_{t+1}, W_{AC}), W_{KC}) - Q'(S_t, A_t, W_K))^2 \qquad (44)$$

gdzie:

W_K – wagi sieci Krytyka

W_{AC} – wagi sieci celu Aktora

W_{KC} – wagi sieci celu Krytyka

Finalna postać wzoru będzie jeszcze doprecyzowana o fakt, że stanowi wartość oczekiwaną powyższego równania dla wielu *Stanów* z mini-wsadu doświadczeń zwracanego z bufora powtórzeń, analogicznie jak w funkcji celu sieci Aktora.

Aktualizacja wag sieci celu zachodzi inaczej niż w opisanej na początku podrozdziału metodzie „stałych Q-celów". Po każdej iteracji treningu wartości wag sieci celu zbliżają się do aktualnych wartości wag sieci swoich odpowiedników w stopniu równym ustalonemu współczynnikowi tau τ. Przyjmuje on bardzo małe wartości rzędu jednej tysięcznej, aby zapewnić stabilność efektów uczenia w czasie Proces ten nosi nazwę „miękkiej aktualizacji" wag. [88]

$$W_{KC} = \tau * W_K + (1 - \tau) * W_{KC} \qquad (45)$$

$$W_{AC} = \tau * W_A + (1 - \tau) * W_{AC} \qquad (46)$$

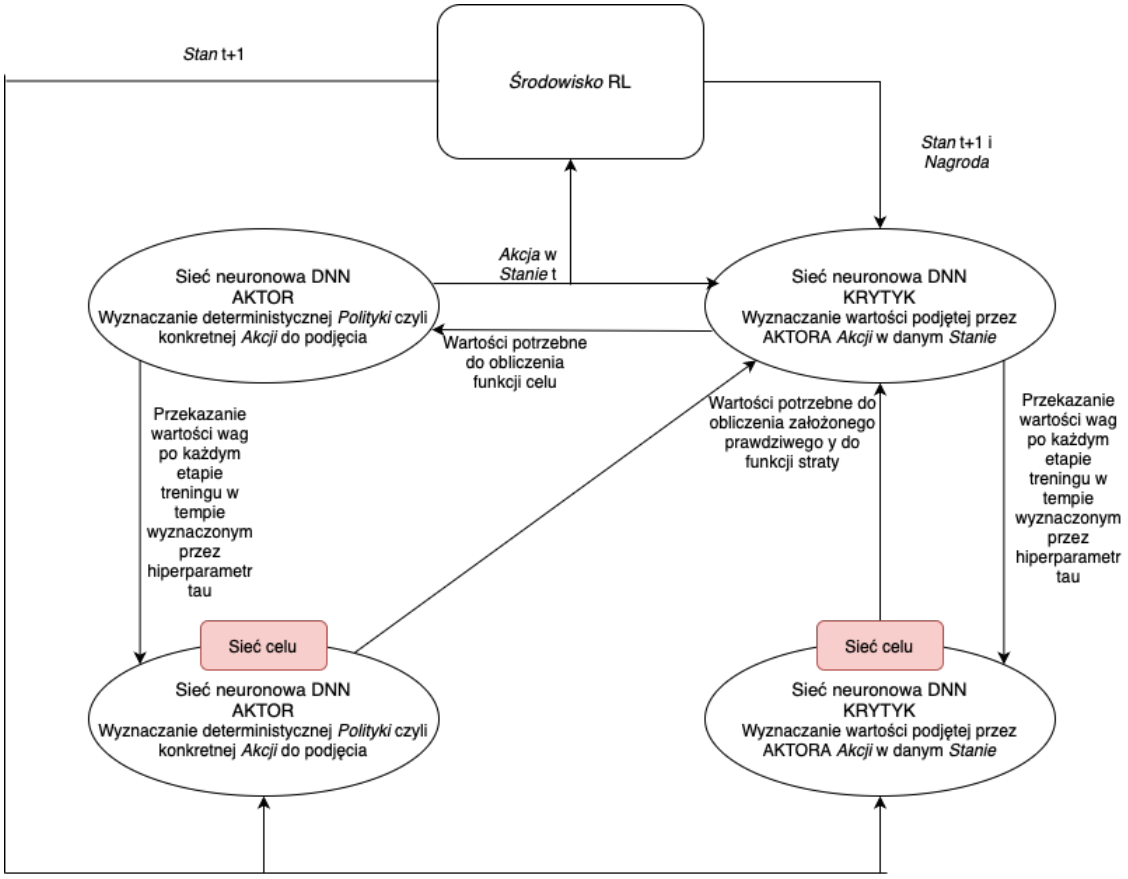
Ze względu na to, że deterministyczna *Polityka* zwraca jedną konkretną *Akcję*, która jest wektorem ciągłych wartości, nie stosuje się hiperparametru epsilon-greedy. Zamiast tego w celu umożliwienia sporadycznej eksploracji *Środowiska* niezgodnej z optymalną *Polityką* dodaje się do tego wektora losowe wartości w postaci hałasu wyznaczanego przez proces Ornsteina–Uhlenbecka[89]

[87] Opracowanie własne na podstawie Timothy P. Lillicrap, Jonathan J. Hunt, ..., Daan Wierstra, dz. cyt. oraz Miguel Morales, dz. cyt., s. 377-384

[88] Opracowanie własne na podstawie Timothy P. Lillicrap, Jonathan J. Hunt, ..., Daan Wierstra, dz. cyt. oraz Miguel Morales, dz. cyt., s. 377-384

[89] Ornstein–Uhlenbeck process, Wikipedia, online: https://en.wikipedia.org/wiki/Ornstein–Uhlenbeck_process

Rysunek 7 *Struktura algorytmu głębokiego gradientu deterministycznej Polityki DDPG*



Źródło: Opracowanie własne na podstawie Timothy P. Lillicrap, Jonathan J. Hunt, ..., Daan Wierstra, dz. cyt. oraz Miguel Morales, dz. cyt., s. 377-384

Całkowity przebieg algorytmu DDGP prezentuje się następująco[90]:

1. Losowa inicjalizacja wag sieci Aktora i Krytyka
2. Inicjalizacja wartości wag sieci celu na podstawie wag sieci podstawowych
3. Inicjalizacja bufora powtórzeń

Dla każdego z epizodów:

4. Inicjalizacja początkowego *Stanu*

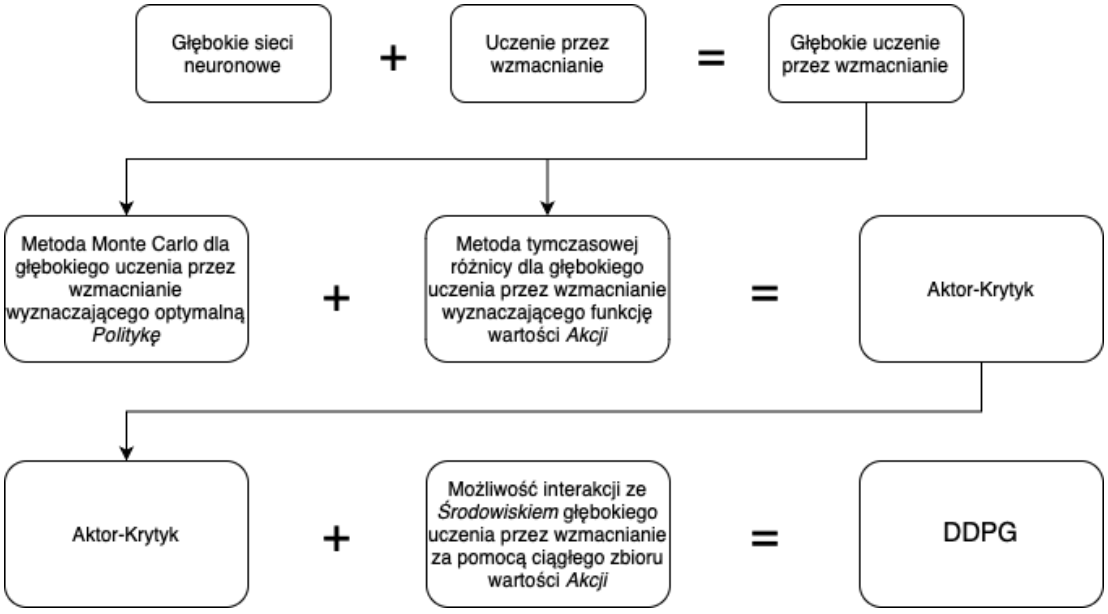
Dla każdego z kroków w czasie:

5. Sieć Aktora wyznacza konkretną *Akcję* + hałas
6. *Środowisko* zwraca *Stan* i *Nagrodę* dla *Akcji*
7. Zapis doświadczenia $(S_t, A_t, R_{t+1}, S_{t+1})$ do bufora powtórzeń
8. Losowanie mini-wsadu N doświadczeń z bufora powtórzeń
9. Obliczenie funkcji straty i optymalizacja Krytyka
10. Obliczenie funkcji celu i optymalizacja Aktora
11. Aktualizacja wag w sieciach celu

DDPG stanowi finalne połączenie wszystkich omawianych dotychczas metod. Głębokie sieci neuronowe DNN pozwalają na efektywne uczenie się *Środowisk* uczenia przez wzmacnianie RL. Uzyskane w ten sposób metody DRL umożliwiają *Stan* o ciągłym zbiorze wartości i dzielą się na te wykorzystujące metodę tymczasowej różnicy umożliwiającą aproksymację Q(s,a) oraz metodę Monte Carlo szukającą od razu aproksymacji optymalnej *Polityki*. Połączenie tych dwóch stanowi zbiór metod Aktor-Krytyk, gdzie *Agent* zamiast uczyć się na podstawie swoich własnych do pewnego momentu błędnych estymacji Q(s,a) uczy się ich na podstawie wyborów *Akcji*[91] wynikających z estymacji optymalnej *Polityki* wyznaczanej przez drugą sieć. Tak jak zostało opisane wyżej DDPG to wersja metody Aktor-Krytyk dla *Akcji* o ciągłym zbiorze wartości[92]. Implementacja i wykorzystanie tego algorytmu dla wyznaczenia optymalnej strategii obrotu akcjami giełdowymi będzie przedmiotem rozdziału IV i V.

[90] Timothy P. Lillicrap, Jonathan J. Hunt, ..., Daan Wierstra, dz. cyt.
[91] Patrz przypis 51
[92] DDPG jako metoda Aktor-Krytyk znajduje się całkowicie po stronie metody tymczasowej różnicy i nie jest metodą Monte Carlo

Rysunek 8 *Poglądowa ilustracja zależności między omawianymi metodami uczenia maszynowego*



Źródło: Opracowanie własne

Rozdział IV

Metodologia zastosowana w modelowaniu decyzji kupno-sprzedaż w obrocie akcjami giełdowymi za pomocą algorytmu głębokiego uczenia przez wzmacnianie

4.1 Opis zastosowanego modelu

W niniejszym rozdziale zostanie przedstawiona implementacja algorytmu głębokiego uczenia przez wzmacnianie (DRL) w wersji głębokiej deterministycznej polityki gradientu (DDPG) przedstawionej w podrozdziale 3.3 w celu wyznaczenia optymalnej strategii obrotu akcjami giełdowymi[93]. Istota zadania wykonywanego w ramach modelu będzie polegała na rozdysponowaniu ograniczonych środków między akcje giełdowe spółek należących do indeksu S&P500. Startowa kwota pieniężna, którą dysponuje *Agent* będzie mogła być przez niego być rozdysponowana w takim stopniu jaki w danym kroku w czasie *Agent* uzna za słuszny. Przykładowo gdyby na rynku panowała głęboka bessa, *Agent* nie jest zmuszony, by posiadać jakiekolwiek akcje giełdowe, z kolei w przypadku znaczącej hossy ma on możliwość zainwestować wszystkie posiadanie środki. *Agent* może zakupić dowolną ilość akcji giełdowych każdej spółki, ale ilość akcji giełdowych jednej spółki możliwa do zakupu lub sprzedaży w jednym kroku w czasie jest ograniczona przez ustalony w ramach modelu limit. Limit ten ma zapobiec wyuczeniu się strategii inwestycyjnej opierającej się na pojedynczych akcjach giełdowych, które akurat w okresie treningowym radzą sobie nadzwyczajnie dobrze. Decyzje odnośnie transakcji odbywają się w interwałach dziennych. *Agent* każdego dnia może podjąć decyzję o dokupieniu określonej ilości akcji giełdowych każdej ze spółek o ile posiadane środki mu na to pozwalają, a także może sprzedać określoną ilość akcji giełdowych spółek, które w danym momencie posiada. Może także zdecydować się nie modyfikować żadnej z pozycji i czekać na kolejne kroki w czasie. Epizod treningu kończy się wraz z upływem wszystkich kroków w czasie stanowiących zadany okres treningowy. Celem *Agenta* jest maksymalizacja wartości posiadanych aktywów finansowych rozumianych jako suma aktualnego salda środków pieniężnych oraz wartości wszystkich posiadanych akcji giełdowych po cenie zamknięcia z ostatniego dnia treningu. Wszystkie decyzje jakie podejmuje *Agent*

[93] Warto mieć na uwadze, że w tym rozdziale sformułowanie *Akcja* oznacza tak jak dotychczas abstrakcyjną interakcję w ramach *Środowiska* DRL, która może oznaczać na przykład zakup wielu akcji giełdowych w różnej ilości, z kolei sformułowanie akcja giełdowa oznacza faktyczną akcję giełdową.

wynikają z opisu *Stanu* w jakim znajduje się w danym kroku w czasie. Składa się on z posiadanych środków pieniężnych, struktury portfela akcji giełdowych wraz z ich cenami zamknięcia z poprzedniego kroku w czasie oraz 9 wskaźników analizy technicznej. Te elementy zostaną omówione dokładniej w kolejnym podrozdziale. *Stany Środowiska* można opisać jako ciągłe. Każdy z jego elementów przyjmuje postać dowolnej liczby rzeczywistej. Tak jak to zostało opisane w podrozdziale 3.2 taka cecha uniemożliwiłaby skorzystanie z najprostszej formy algorytmu uczenia przez wzmacnianie dla skończonego procesu MDP ze względu na brak możliwości jednoznacznego powiązania określonego *Stanu* z daną *Akcją* i wymagałaby zastosowania aproksymacji funkcji *Akcji* w zależności od *Stanu.* Tak jak to zostało opisane w podrozdziale 3.3 taką aproksymację stanowi głęboka sieć neuronowa DNN. Z kolei zbiór możliwych *Akcji* pozornie przyjmuje postać dyskretną, ponieważ liczba spółek, dla których możemy kupić skończoną ilość akcji giełdowych również jest skończona. Jeżeli jednak weźmiemy pod uwagę liczby dotyczące modelu budowanego w tej pracy czyli 333 spółki, dla których możemy kupić do 200 akcji giełdowych, okazuje się że zbiór możliwych *Akcji* składa się z 200^{333} możliwych do uzyskania konfiguracji. To zdecydowanie zbyt wiele, by przyjmowanie zbioru *Akcji* jako dyskretnego miało w tym przypadku jakiekolwiek praktyczne uzasadnienie. Ze względu na potraktowanie zbioru *Akcji* jako ciągłego algorytm DDPG został wybrany jako adekwatna wersja algorytmu DRL dla tego problemu, ponieważ stanowi on właśnie przykład algorytmu ciągłej kontroli w głębokim uczeniu przez wzmacnianie. Tak jak opisane w podrozdziale 3.3 wartości zwracane przez warstwę wyjściową sieci neuronowej Aktora w DDPG stanowią nie tyle prawdopodobieństwo wyboru określonej *Akcji* co deterministyczną decyzję odnośnie wyboru konkretnej *Akcji* oznaczającej stopień natężenia wyboru każdej z akcji giełdowych co następnie na podstawie maksymalnej możliwej ilości akcji giełdowych do zakupu w jednym kroku w czasie może być przeliczone na oszacowaną dyskretną ilość akcji giełdowych do zakupu lub sprzedaży, zostało to omówione dokładniej w kolejnym podrozdziale. Model składa się z 4 sieci neuronowych stanowiących sieci Aktora i Krytyka oraz sieci celu dla Aktora i Krytyka.

Zarówno w sieci Aktora jak i Krytyka liczba warstw ukrytych oraz liczba ich neuronów zależy od wersji modelu. Zastosowano algorytm optymalizacji „Adam", regularyzację L2 oraz funkcję aktywacji ReLU dla wszystkich warstw ukrytych. W modelu sieci Krytyka tensor *Akcji* jest dołączany do pierwszej warstwy ukrytej wyliczonej z tensora *Stanu.*

Tabela 2 *Ustalone hiperparametry dla wszystkich konfiguracji modelu*[94]

Rozmiar bufora powtórzeń	10000
Rozmiar mini-wsadu do treningu sieci	128
Stopa dyskontowa gamma	0.99
Stopa miękkiej aktualizacji parametrów sieci celu Aktora i Krytyka	0.001
Współczynnik uczący sieci Aktora	0.0001
Współczynnik uczący sieci Krytyka	0.001

4.2 Opis *Środowiska*, danych i zmiennych użytych w ramach modelu

4.2.1 Dane

Dla bardziej klarownego opisu *Środowiska* ten podrozdział rozpocznie się od opisania danych, które zostały użyte w ramach zaimplementowanego modelu.

Bazę stanowią dane z Yahoo Finance odnośnie cen maksymalnych, minimalnych, otwarcia i zamknięcia oraz wielkości obrotu z każdego dnia między końcem 2014 roku a grudniem 2023 roku dla 333 spółek, które obecnie wchodzą w skład indeksu S&P500, a które znajdowały się w nim już przed okresem, z którego pochodzą dane. Następnie na podstawie tych danych dla każdego dnia zostały obliczone wartości 9 różnych wskaźników analizy technicznej, które cechowały każdą z akcji spółek. Okres treningowy dla algorytmu przypada na lata 2015 – 2018, a następnie jest rozszerzany aż do 2023 roku. Z kolei pierwsze dziesięć miesięcy 2019 roku stanowi pierwotny okres testowy, który w późniejszym etapie również podlega modyfikacji. Wspomniana końcówka 2014 roku została uwzględniona w pozyskanych danych aby możliwe było obliczenie poprawnych wartości wskaźników technicznych od początku pierwotnego okresu treningowego.

[94] Rekomendacja dla wybranych wartości hiperparametrów pochodzi z T. P. Lillicrap, J. J. Hunt, ..., D. Wierstra, dz. cyt.

4.2.2 Wskaźniki analizy technicznej

Poniższe wskaźniki analizy technicznej zostały uwzględnione jako zmienne objaśniające dla *Agenta*:

- 30 dniowa prosta średnia krocząca (SMA)
- 60 dniowa prosta średnia krocząca (SMA)
- Zbieżność-rozbieżność średniej ruchomej (MACD)
- 30 dniowy wskaźnik siły względnej (RSI)
- 30 dniowy Commodity Channel Index (CCI)
- Górna i dolna wstęga Bollingera
- Indeks kierunkowy (DX)
- Aroon

4.2.3 *Środowisko*

Tak jak opisane w podrozdziale 3.2 elementy *Środowiska* stanowią *Stan*, *Akcja* i *Nagroda*. *Stan* składa się ze skalara określającego saldo środków pieniężnych, 333-elementowego wektora określającego ilość akcji giełdowych każdej ze spółek w portfelu *Agenta*, 333-elementowego wektora określającego cenę zamknięcia akcji giełdowych każdej ze spółek w poprzednim kroku w czasie oraz macierzy 333x9 określającej wartość każdego ze wskaźników technicznych dla każdej z akcji giełdowych.

Po spłaszczeniu do jednowymiarowego tensora dla sieci DNN *Stan* składa się z 3664 skalarów. *Akcja* składa z 200-elementowego wektora określającego ilość akcji giełdowych każdej ze spółek jaką należy dokupić lub sprzedać. Jej wartości są wyznaczone z warstwy wyjściowej sieci Aktora za pomocą funkcji aktywacji tanh. Przedział (-1,1) pozwala oszacować jaką dyskretną ilość w postaci części maksymalnego limitu akcji giełdowych jakie można kupić w jednym kroku w czasie należy kupić lub sprzedać. *Nagrodę* po każdym kroku stanowi różnica między wartością posiadanych aktywów finansowych między krokiem w czasie t+1 i t, gdzie wartość aktywów finansowych dla kroku t+1 jest rozumiana jako suma salda środków pieniężnych posiadanych w okresie t+1 oraz wartości wszystkich posiadanych okresie t+1 akcji giełdowych po cenie zamknięcia z okresu t. Wartość ta jest przemnożona przez pewną liczbę ułamkową dla efektywności obliczeniowej. Skumulowana nagroda jest określona jako suma *Nagród* z każdego kroku w czasie w danym epizodzie treningu. Będzie ona służyć do porównywania różnych wersji modeli między sobą. Dodatnia wartość skumulowanej nagrody

oznacza, że model częściej i mocniej osiąga dzienne wartości stopy zwrotu o wartości dodatniej, a nie ujemnej, a jego końcowa wartość posiadanych aktywów finansowych powinna być większa od startowego salda środków pieniężnych.

Początkowe saldo środków pieniężnych zostało ustalone na 5 milionów dolarów, a maksymalna ilość akcji giełdowych pojedynczej spółki, które można kupić lub sprzedać w jednym kroku w czasie na 200. Opłata transakcyjna została ustalona na 0,1% wartości transakcji.

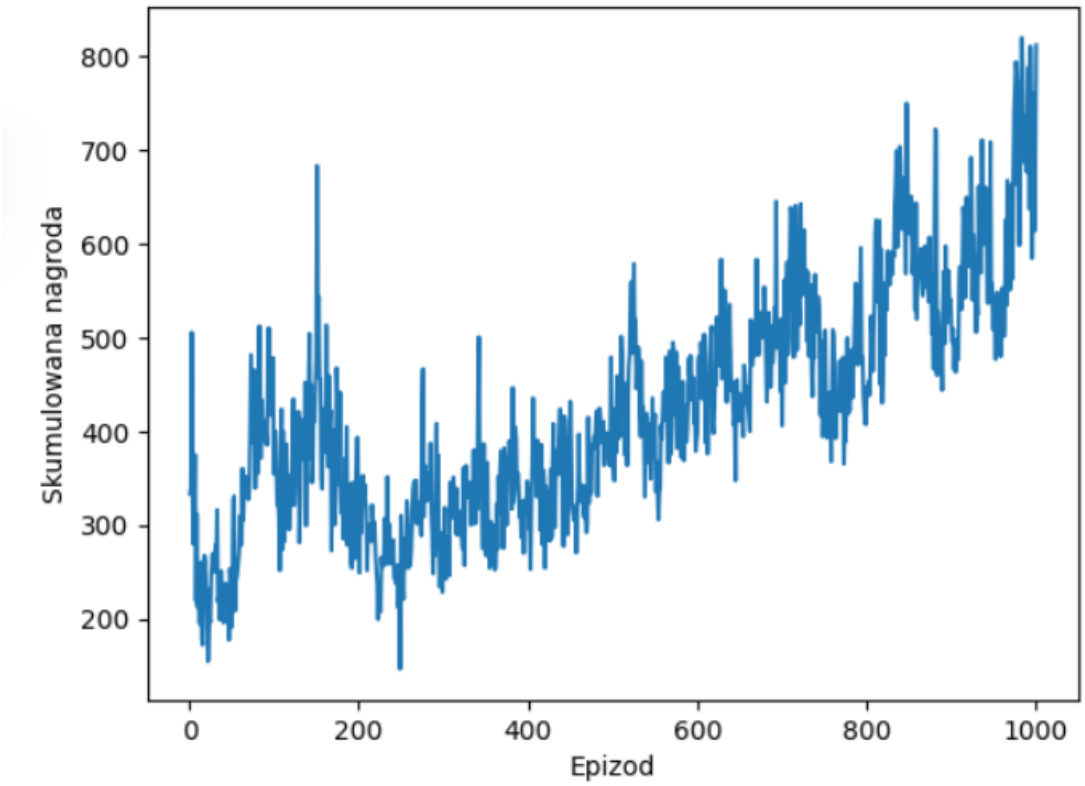
Rozdział V

Podsumowanie rezultatów modelowania decyzji kupno-sprzedaż w obrocie akcjami giełdowymi za pomocą algorytmu głębokiego uczenia przez wzmacnianie

5.1 Przebieg treningu dla różnych konfiguracji modelu

Pierwszy model (niebieski) składający się w sieciach Aktora i Krytyka z dwóch warstw ukrytych o liczbie neuronów odpowiednio 2000 i 1000 został wytrenowany w 1000 epizodach. Jak widać na rysunku 9 wraz z przemijaniem kolejnych epizodów model jest w stanie konwergować do coraz lepszych rezultatów na zbiorze treningowym. Można wręcz powiedzieć, że tysiąc epizodów to za mało, by wyczerpać pełne możliwości uczenia się przez *Agenta,* aczkolwiek należy pamiętać o ryzyku przeuczenia na zbiorze treningowym.

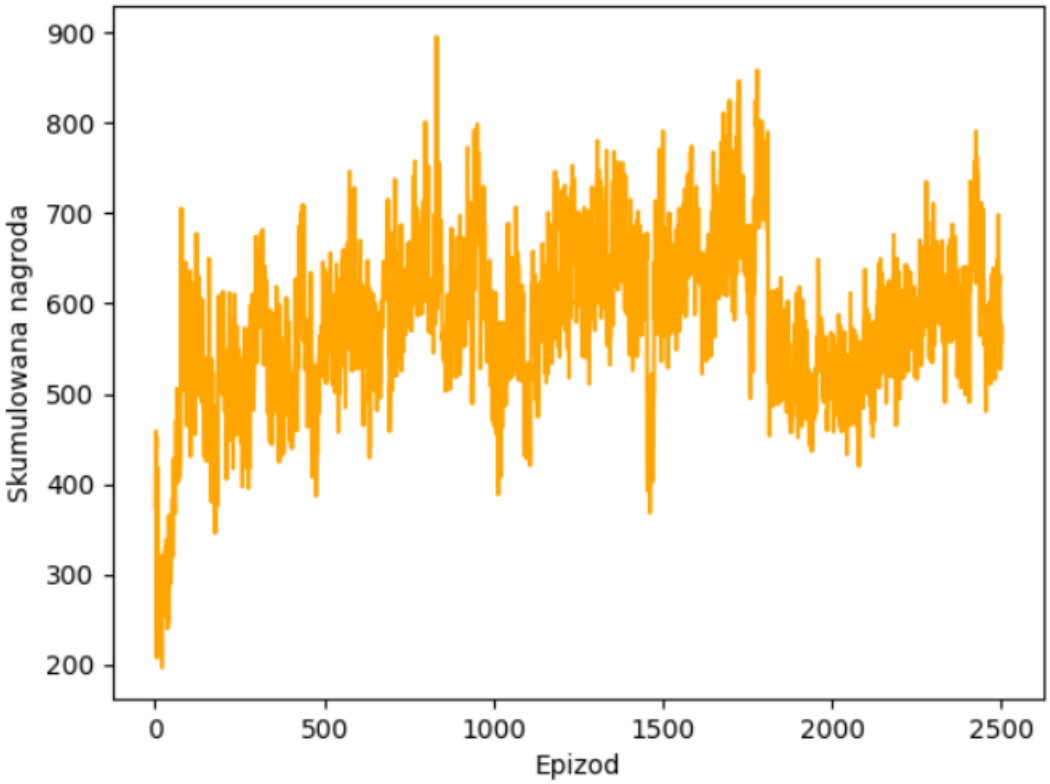
Rysunek 9 *Skumulowana nagroda osiągnięta przez Agenta w danym epizodzie treningu modelu niebieskiego*



Źródło: Opracowanie własne

Drugi model (pomarańczowy) ma bliźniaczą konfigurację do pierwszego, ale został wytrenowany w 2500 epizodach. Jak można zobaczyć na rysunku 1000 epizodów okazało się dobrym momentem na zakończenie treningu, a kolejne epizody nie przyniosły poprawy skumulowanej nagrody.

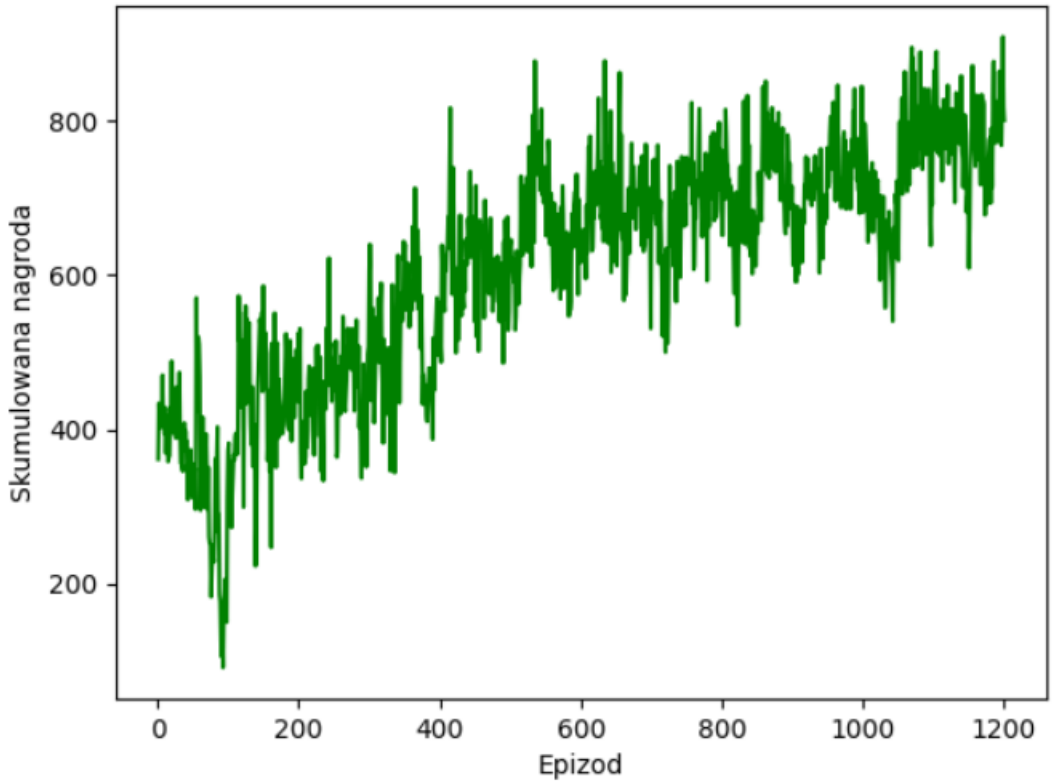
Rysunek 10 *Skumulowana nagroda osiągnięta przez Agenta w danym epizodzie treningu modelu pomarańczowego*



Źródło: Opracowanie własne

Trzeci mniejszy model (zielony) składający się w sieciach Aktora i Krytyka z dwóch warstw ukrytych o liczbie neuronów odpowiednio 400 i 300 został wytrenowany w 1200 epizodach. Na podstawie wartości skumulowanej nagrody można stwierdzić, że model wykazuje większe zdolności uczenia od modelu niebieskiego.

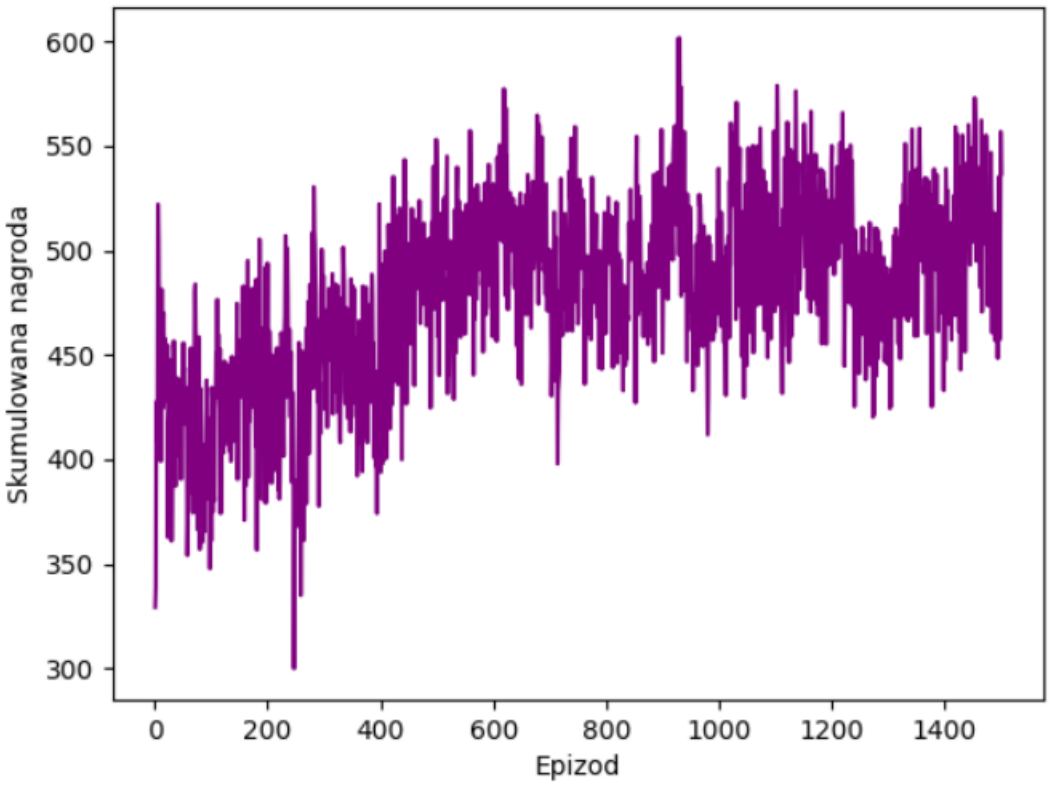
Rysunek 11 *Skumulowana nagroda osiągnięta przez Agenta w danym epizodzie treningu modelu zielonego*



Źródło: Opracowanie własne

Czwarty model (fioletowy) składający się w sieciach Aktora i Krytyka z trzech warstw ukrytych o liczbie neuronów odpowiednio 400, 300 i 200 został wytrenowany w 1500 epizodach. Na podstawie wartości skumulowanej nagrody można stwierdzić, że model wykazuje najgorsze zdolności uczenia ze wszystkich wymienionych konfiguracji modelu. Już w okolicach 600 epizodu model przestał zwiększać skumulowaną nagrodę i do końca treningu pozostał w stagnacji.

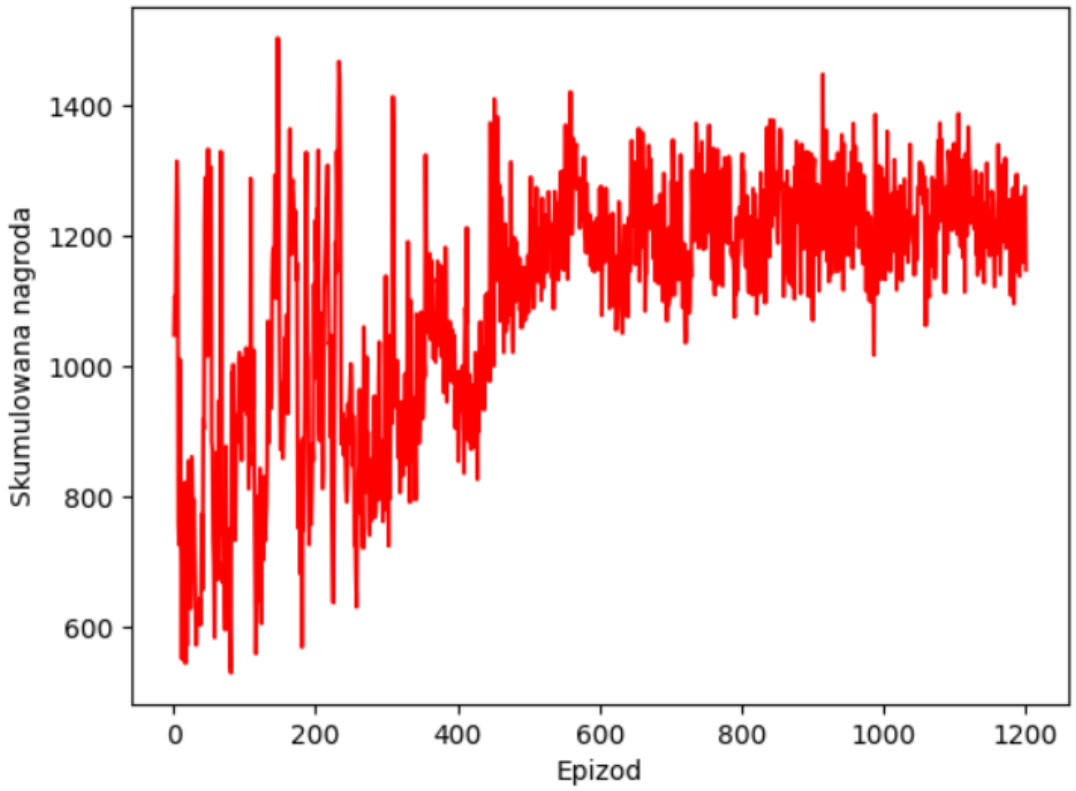
Rysunek 12 *Skumulowana nagroda osiągnięta przez Agenta w danym epizodzie treningu modelu fioletowym*



Źródło: Opracowanie własne

Piąty model (czerwony) ma taką samą konfigurację jak model zielony, ale zmienione zostało założenie o maksymalnej liczbie akcji giełdowych pojedynczej spółki jakie może zakupić lub sprzedać *Agent* w okresie treningowym w jednym kroku w czasie. Ta liczba została zwiększona z 200 do 2000. Jak widać na rysunku 13, ta konfiguracja modelu umożliwia osiągnięcie najwyższej skumulowanej nagrody w okresie treningowym w porównaniu ze wszystkimi innymi wypróbowanymi konfiguracjami.

Rysunek 13 *Skumulowana nagroda osiągnięta przez Agenta w danym epizodzie treningu modelu czerwonym*



Źródło: Opracowanie własne

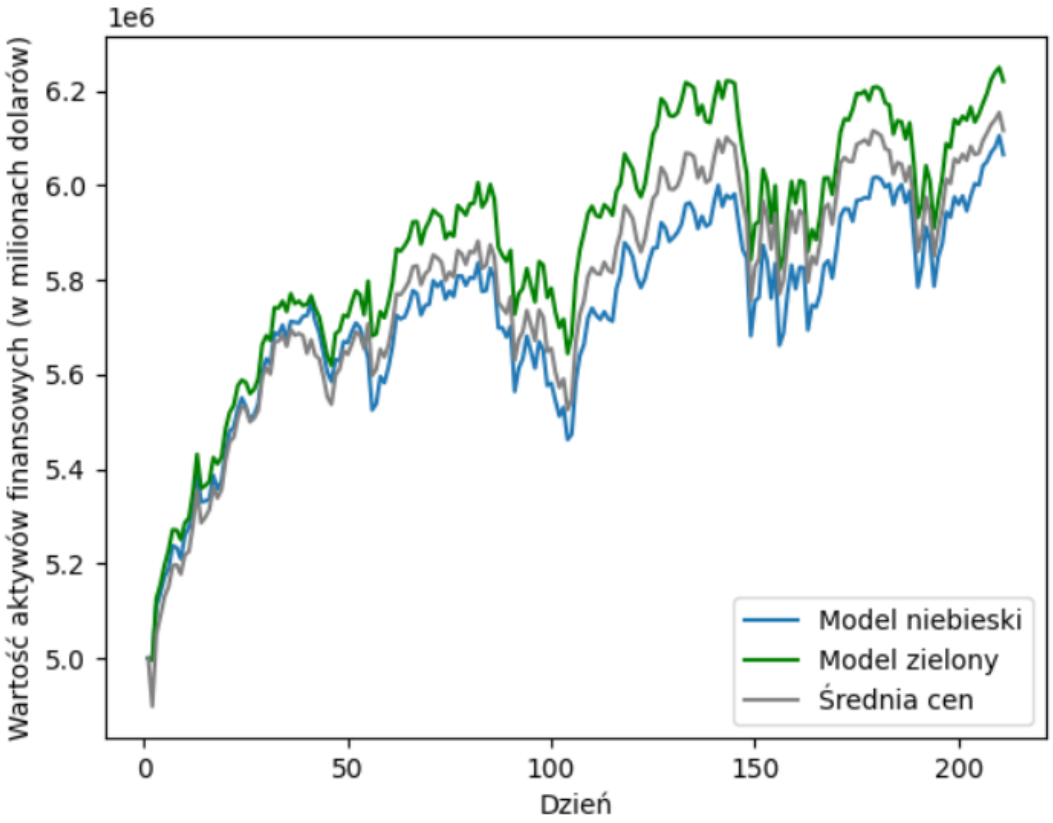
5.2 Testowanie modelu na potreningowym okresie czasu

Testowanie modeli na potreningowym okresie czasu będzie odbywać się w dwóch wariantach. Pierwszy wariant czyli strategia zachowawcza zakłada utrzymanie takiego samego limitu zakupu lub wykupu akcji giełdowych jednej spółki w jednym kroku w czasie w okresie testowym jak w okresie treningowym. Drugi wariant czyli strategia agresywna będzie modyfikować to założenie. Indeksem porównawczym dla rezultatów modelu będzie indeks cen akcji giełdowych wszystkich 333 spółek uwzględnionych w modelu ważony ceną akcji giełdowej każdej ze spółek czyli po prostu średnia ich cen.

5.2.1 Strategia zachowawcza

Wśród modeli z maksymalnym limitem zakupu lub wykupu 200 akcji giełdowych jednej spółki w jednym kroku w czasie w trakcie treningu najlepszy pod względem osiągniętej skumulowanej nagrody okazał się mniejszy model zielony o warstwach ukrytych sieci Aktora i Krytyka o wielkości 400 i 300 neuronów, a zaraz zanim nieznacznie gorzej radził sobie model niebieski o warstwach ukrytych o wielkości 2000 i 1000 neuronów. Jak pokazuje rysunek 14 i tabela 3 model zielony był w stanie w trakcie okresu testowego styczeń – październik 2019 wygenerować wyższą stopę zwrotu z początkowego salda środków pieniężnych niż indeks średnich cen akcji giełdowych wszystkich 333 spółek uwzględnionych w modelu. Różnica między stopami zwrotu wyniosła 1,9 punktu procentowego. Największa zmiana w różnicy w stopach zwrotu miała miejsce w okresie styczeń-kwiecień 2019, a więc w ciągu pierwszych 4 miesięcy po okresie treningowym. W kolejnym miesiącach wykres modelu zielonego podążał wraz z indeksem cen utrzymując swoją przewagę. Model niebieski początkowo podążał wraz z indeksem, a następnie zaczął do niego wyraźnie tracić i należy go odrzucić jako wyraźnie słabszy model.. Modele pomarańczowy i fioletowy zgodnie z oczekiwaniami po rezultatach treningu również były słabsze od modelu zielonego. Finalnie model zielony okazał się na zbiorze testowym najlepszą wersją ze wszystkich próbowanych konfiguracji modelu z limitem zakupu lub wykupu akcji giełdowych jednej spółki w jednym kroku w czasie wynoszącym 200. Będzie on użyty do kolejnych analiz w dalszej części rozdziału.

Rysunek 14 *Porównanie wartości aktywów finansowych posiadanych w danym dniu przez Agenta DDPG-DRL do średniej cen rozważanych akcji giełdowych w okresie styczeń-październik 2019 w modelu zielonym i niebieskim*



Źródło: Opracowanie własne

Tabela 3 *Porównanie stopy zwrotu między strategią zachowawczą Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz indeksem S&P500*[95] *przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL*

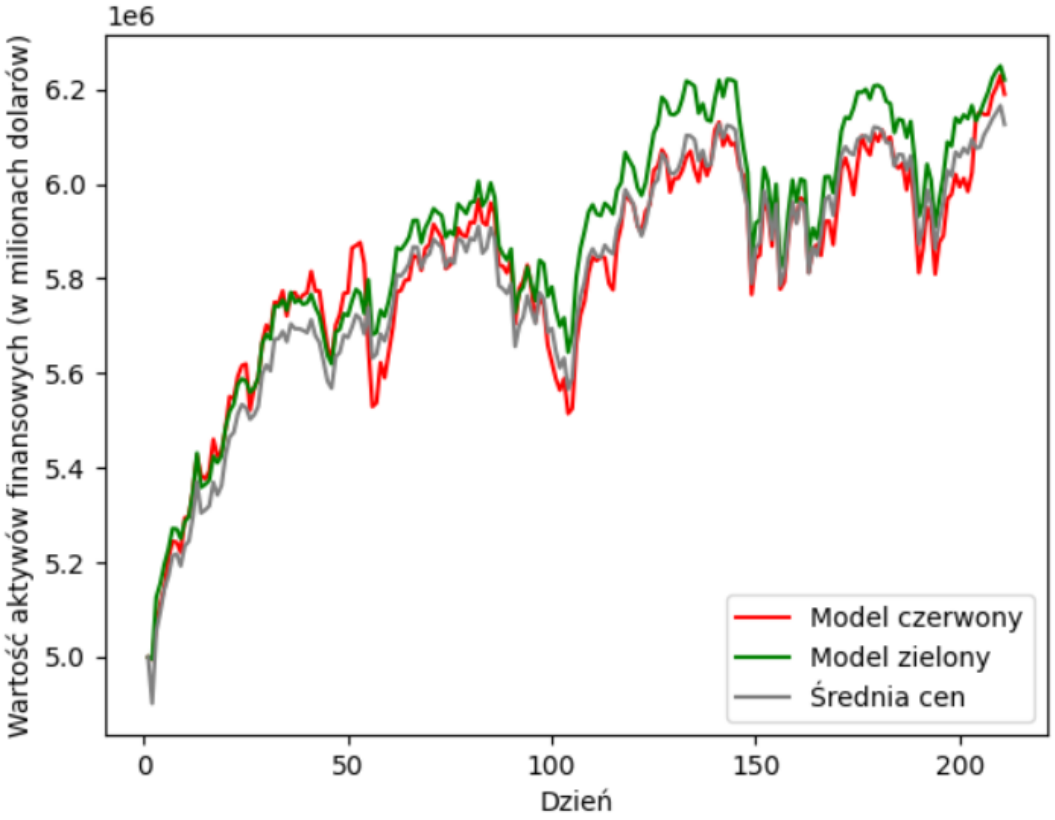
Stopa zwrotu	Model zielony DDPG-DRL	Średnia cen rozważanych spółek	S&P500
Styczeń 2019	10,4%	9,3%	7,7%
Styczeń – Kwiecień 2019	20,1%	18,2%	17,3%
Styczeń – czerwiec 2019	21,1%	19,2%	20,0%
Styczeń – październik 2019	24,4%	22,5%	21,4%

Jednym z powodów, dla których próbowane modele nie są w stanie wyraźnie lub w ogóle przekroczyć indeksu cen może być zbyt niski limit akcji giełdowych jednej spółki jakie *Agent* może zakupić do swojego portfela w danym kroku w czasie. Ceny niektórych akcji giełdowych są zbyt niskie, by mieć znaczący udział w portfelu i *Agent* może nie być w stanie w pełni skorzystać ze swojej wiedzy. Teza ta została zweryfikowana pozytywnie. Model czerwony, który ma zwiększony limit akcji giełdowych jednej spółki osiąga w trakcie okresu treningowego znacznie wyższe wartości skumulowanej nagrody co oznacza znacznie większą przewagę wartości dodatnich stóp zwrotu nad ujemnymi. Niestety jak pokazuje rysunek 15 model czerwony nie potrafi za pomocą swoją wiedzy przekroczyć stóp zwrotu modelu zielonego w okresie testowym, a przez większość czasu nawet wyraźnie do niego traci. Oznacza to że zwiększenie limitu zakupu akcji giełdowych w jednym kroku w czasie mimo znacznego zwiększenia zwrotów w okresie treningowym nie przełożyło się na zdolność generowania większych dodatnich zwrotów na zbiorze testowym, a doprowadziło jedynie do znacznego przeuczenia się modelu na zbiorze treningowym i niepotrzebnego zwiększenia ryzyka specyficznego konkretnych spółek wynikającego ze zwiększenia możliwej koncentracji wartości pojedynczej spółki w portfelu. Należy podkreślić, ze dziesięciokrotnie większa liczba akcji giełdowych jednej spółki jakie *Agent* modelu czerwonego może kupić w jednym kroku w

[95] Ze względu na wykluczenie z modelu spółek, które w późniejszym okresie wypadły z indeksu S&P500 porównanie między modelem, a tym indeksem jest nieuprawnione, ale wartości te zostały przywołane, by uzasadnić, że porównanie do średniej cen akcji giełdowych rozważanych spółek mimo innej metodologii stanowi dobre uogólnienie porównania do S&P500.

czasie znacząco zwiększa ryzyko strat w wyniku negatywnych szoków egzogenicznych dotyczących poszczególnych spółek, na które *Agent* nie jest przygotowany, dlatego wybór tego modelu przy podobnych rezultatach byłby niewłaściwy.

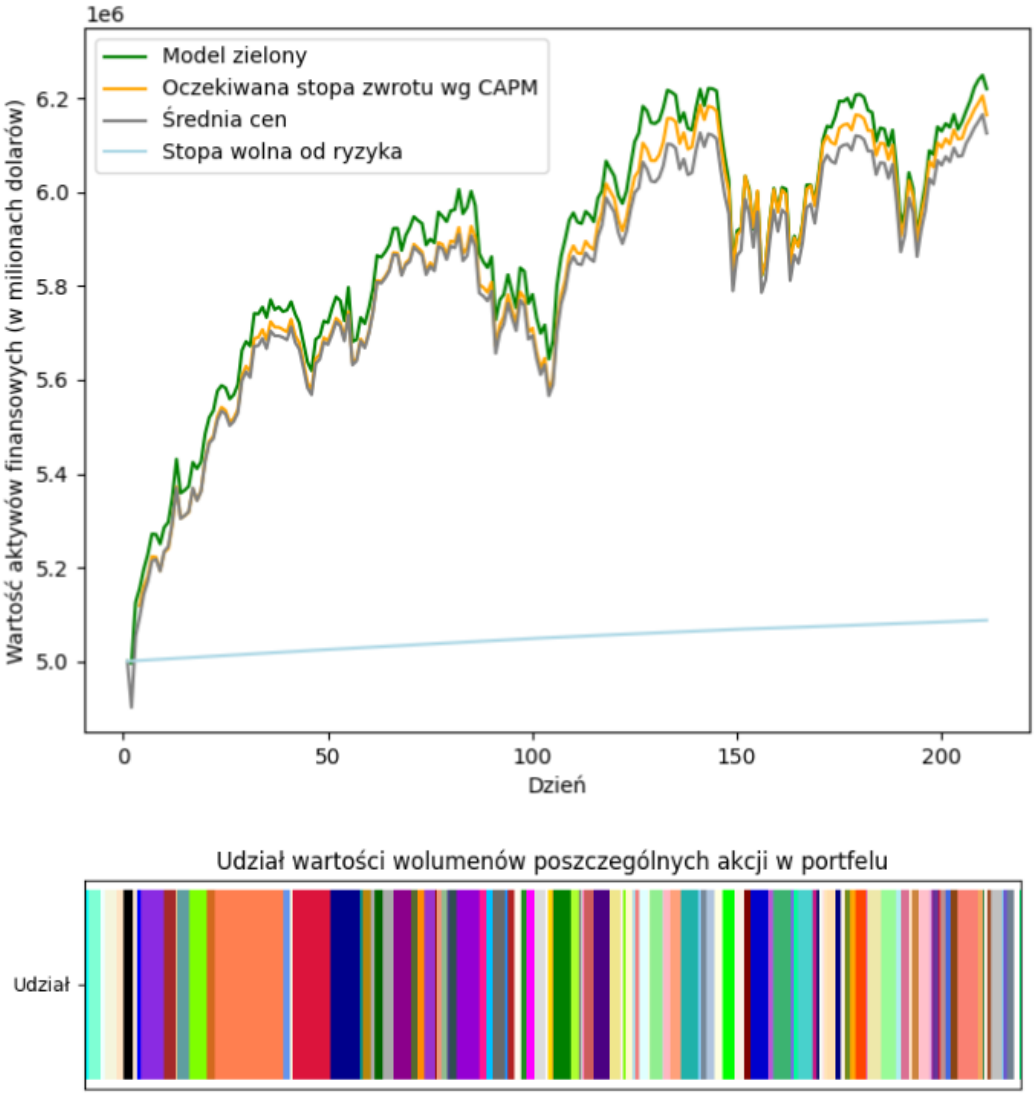
Rysunek 15 *Porównanie wartości aktywów finansowych posiadanych w danym dniu przez Agenta DDPG-DRL do średniej cen rozważanych akcji giełdowych w okresie styczeń-październiku 2019 w modelu zielonym i czerwonym*



Źródło: Opracowanie własne

Rysunek 16 i tabela 4 pokazują porównanie między modelem zielonym, a oczekiwaną stopą zwrotu uwzględniającą ryzyko systematyczne wyliczoną według wzoru CAPM przy przyjęciu oprocentowania 10-letnich amerykańskich obligacji skarbowych jako stopy wolnej od ryzyka oraz średniej cen akcji giełdowych rozważanych w modelu spółek jako stopy rynkowej. Współczynnik beta został wyliczony metodą kroczącą uwzględniając wszystkie stopy zwrotu do danego momentu w czasie.

Rysunek 16 *Porównanie wartości aktywów finansowych posiadanych w danym dniu przez Agenta DDPG-DRL do średniej cen rozważanych akcji giełdowych, oczekiwanej stopy zwrotu według CAPM oraz oprocentowania 10-letnich amerykańskich obligacji skarbowych w okresie testowym styczeń-październik 2019 wraz z udziałem wartości wolumenów poszczególnych akcji giełdowych w portfelu w ostatnim dniu okresu testowego dla strategii zachowawczej modelu zielonego*



Źródło: Opracowanie własne

Tabela 4 *Porównanie stopy zwrotu między strategią zachowawczą Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz oczekiwaną stopą zwrotu według CAPM przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL*

Stopa zwrotu	Model zielony DDPG-DRL	Oczekiwana stopa zwrotu wg CAPM	Średnia cen rozważanych spółek
Styczeń 2019	10,4%	9,4%	9,3%
Styczeń – Kwiecień 2019	20,1%	18,5%	18,2%
Styczeń – czerwiec 2019	21,1%	19,9%	19,2%
Styczeń – październik 2019	24,4%	23,3%	22,5%

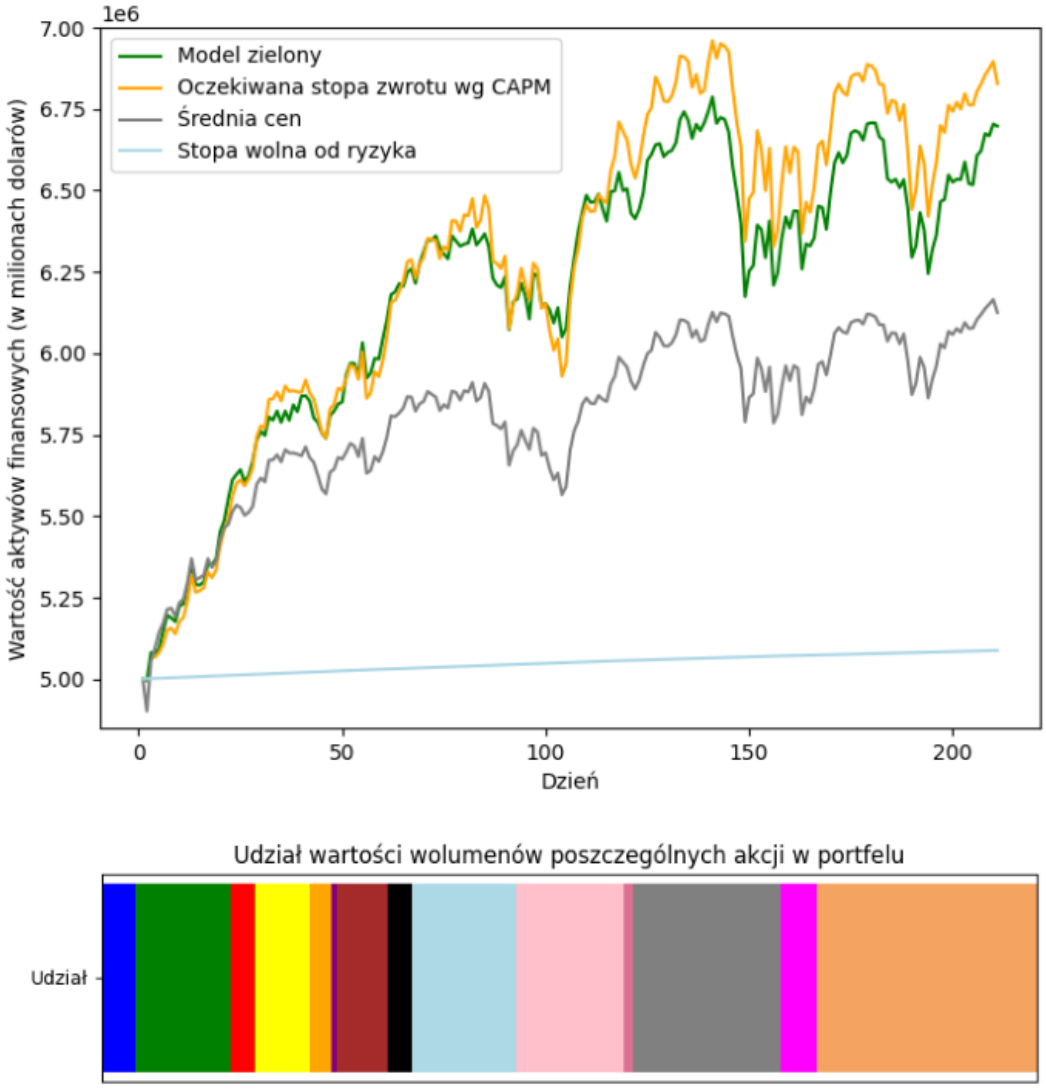
Strategia zachowawcza modelu zielonego umożliwiła osiągnięcie ponadprzeciętnych zysków w okresie testowym względem rynku zarówno na podstawie porównania z zastosowanym indeksem cen jak i z wyliczoną na jego podstawie oczekiwaną stopą zwrotu CAPM.

5.2.2 Strategia agresywna

Jak zostało opisane wyżej na podstawie treningu modelu czerwonego udało się ustalić, że zwiększanie limitu zakupu akcji giełdowych w jednym kroku znacząco zwiększa przewagę wartości dodatnich stóp zwrotu nad wartościami ujemnych stóp zwrotu. Ze względu na przeuczenie nie ma to jednak przełożenia z okresu treningowego na okres testowy. Strategią umożliwiającą *Agentowi* zwiększenie potencjalnych zysków jest korzystanie z wiedzy zdobytej z okresu treningowego z niskim limitem zakupu lub wykupu akcji giełdowych w jednym kroku w czasie w celu uniknięcia przeuczenia, przy jednoczesnym zwiększeniu tego limitu na zbiorze testowym umożliwiającym koncentrację środków w wybranych akcjach giełdowych, które model uznaje za najlepsze. W przypadku utrzymania limitu 200 na zbiorze treningowym i zwiększenia go do 2000 na zbiorze testowym model zielony w znaczącym stopniu przewyższa zwrot indeksu średniej cen rozważanych spółek. Jednakże po obliczeniu oczekiwanej stopy zwrotu według modelu CAPM uwzględniającej ryzyko systematyczne wynikające ze zwiększonej zmienności portfela względem indeksu cen wynika, że osiągnięte zyski w rzeczywistości mogą nie być ponadprzeciętne, ponieważ

strategia ta obarczona jest ryzykiem poniesienia strat w przyszłości. Rezultat strategii agresywnej modelu zielonego w okresie testowym pokazuje rysunek 17 i tabela 5. Zamiast dywersyfikacji nieznacznie odbiegającej od indeksu cen model inwestuje większość środków w kilkanaście najlepszych jego zdaniem akcji giełdowych.

Rysunek 17 *Porównanie wartości aktywów finansowych posiadanych w danym dniu przez Agenta DDPG-DRL do średniej cen rozważanych akcji giełdowych, oczekiwanej stopy zwrotu według CAPM oraz oprocentowania 10-letnich amerykańskich obligacji skarbowych w okresie testowym styczeń-październik 2019 wraz z udziałem wartości wolumenów poszczególnych akcji giełdowych w portfelu w ostatnim dniu okresu testowego dla strategii agresywnej modelu zielonego*



Źródło: Opracowanie własne

Tabela 5 *Porównanie stopy zwrotu między strategią agresywną Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz oczekiwaną stopą zwrotu według CAPM przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL*

Stopa zwrotu	Model zielony DDPG-DRL	Oczekiwana stopa zwrotu wg CAPM	Średnia cen rozważanych spółek
Styczeń 2019	9,7%	9,2%	9,3%
Styczeń – Kwiecień 2019	27,6%	29,5%	18,2%
Styczeń – czerwiec 2019	29,8%	33.3%	19,2%
Styczeń – październik 2019	33,9%	36,6%	22,5%

5.2.3 Strategia mieszana w długim okresie

Aby rozstrzygnąć czy ryzyko wskazywane przez CAPM rzeczywiście się zrealizuje przy dalszym przestrzeganiu predykcji modelu DDPG-DRL, zostanie on wytrenowany na 9 dodatkowych kilkuletnich okresach treningowych. Każdy z nich będzie się kończyć przed przeznaczonym dla niego kroczącym półrocznym okresem testowym składającym się z danego półrocza w latach 2019-2023. Analiza poprzednich okresów testowych, a w szczególności ostatniego minionego będzie pozwalała oszacować czy w kolejnym półroczu należy zastosować strategię agresywną czy zachowawczą lub czy istnieje inna konfiguracja modelu, która potencjalnie może sprawdzić się lepiej. Strategia zachowawcza zostanie zastosowana, gdy strategia agresywna z poprzedniego półrocza nie pokona indeksu cen. Zbiorczy rezultat przedstawia rysunek 18 oraz tabele 6 i 7. Tabela 6 przedstawia porównanie stóp zwrotu osiągniętych wyłącznie w ramach danego półrocznego okresu testowego, tabela 7 przedstawia skumulowany efekt 5-letniego przestrzegania predykcji modelu dla 10 kolejnych okresów testowych. Z kolei rysunek 19 oraz tabele 8 i 9 przedstawiają wariant, w którym w całym 5-letnim okresie testowym zastosowana została wyłącznie strategia agresywna zielonego modelu. Tabela 8 i 9 przedstawiają analogicznie to co 6 i 7, ale dla przypadku wyłącznie strategii agresywnej.

Tabela 6 *Porównanie stopy zwrotu między wybraną strategią mieszaną (A – agresywna, Z - zachowawcza) Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz oczekiwaną stopą zwrotu według CAPM przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL oddzielnie dla każdego półrocza w okresie styczeń 2019 – grudzień 2023*

Stopa zwrotu	Model zielony DDPG-DRL	Oczekiwana stopa zwrotu wg CAPM	Średnia cen rozważanych spółek
Czerwiec 2019 A	+29,8%	+33,3%	+19,2%
Grudzień 2019 A	+11,8%	+9,8%	+7,5%
Czerwiec 2020 A	+5,6%	-7,0%	-6,9%
Grudzień 2020 A	+18,0%	+19,8%	+19,2%
Czerwiec 2021 Z	+16,5%	+17,2%	+17,8%
Grudzień 2021 Z	+8,6%	+9,3%	+10,2%
Czerwiec 2022 Z	-15,6%	-14,7%	-17,0%
Grudzień 2022 A	+1,2%	+6,7%	+6,6%
Czerwiec 2023 Z	+10,7%	+10,3%	+8,8%
Grudzień 2023 Z	+7,1%	+8,0%	+7,5%

Konfiguracja modelu zielonego była stosowana niezmiennie przez pełne 5 lat, ponieważ żadna inna próbowana konfiguracja modelu na minionych okresach testowych nie dawała pewności osiągnięcia lepszych rezultatów. Na podstawie znajdującej się niżej tabeli 8 można stwierdzić, że w II półroczu 2020, I 2021, II 2021, II 2022 i I 2023 strategia agresywna okazała się być słabsza od indeksu cen. Był to powód, dla którego w strategii mieszanej w następujących po nich okresach została zastosowana strategia zachowawcza co pokazuje powyższa tabela 6. Nie zawsze okazało się to opłacalne, przykładowo w okresach II 2021, I 2022, II 2023 strategia agresywna wygenerowała ostatecznie wyższe stopy zwrotu mimo wyboru strategii zachowawczej. Należy również zauważyć, że według stóp CAPM wybór strategii zachowawczej zamiast agresywnej nie oznaczał niższego ryzyka w każdym okresie co stawia pod znakiem zapytania sens stosowania długookresowo strategii mieszanej zamiast po prostu agresywnej. W dłuższym okresie strategia zachowawcza modelu zielonego wykazała też jedynie zdolność do naśladowania, a nie prześcignięcia indeksu cen, a prawie cały wzrost powyżej indeksu cen był zasługą okresów z zastosowaną strategią agresywną.

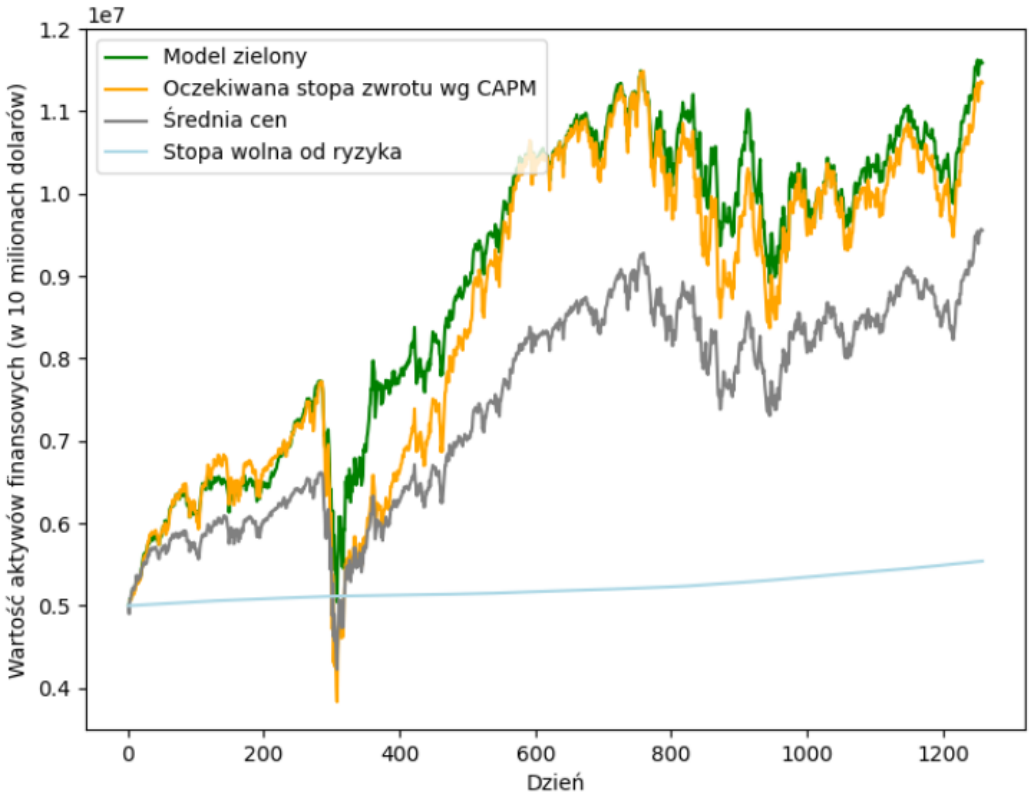
Tabela 7 *Porównanie stopy zwrotu między długookresową strategią mieszaną (A – agresywna, Z - zachowawcza) Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz oczekiwaną stopą zwrotu według CAPM przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL kumulatywnie w kroczącym okresie testowym styczeń 2019 – grudzień 2023*

Stopa zwrotu	Model zielony DDPG-DRL	Oczekiwana stopa zwrotu wg CAPM	Średnia cen rozważanych spółek
Czerwiec 2019 A	29,8%	33,3%	19,2%
Grudzień 2019 A	45,1%	44,0%	28,1%
Czerwiec 2020 A	53,3%	23,3%	19,3%
Grudzień 2020 A	80,9%	69,1%	42,3%
Czerwiec 2021 Z	107,6%	109,3%	67,6%
Grudzień 2021 Z	129,0%	128,5%	84,7%
Czerwiec 2022 Z	93,2%	78,4%	53,3%
Grudzień 2022 A	95,4%	92,6%	63,3%
Czerwiec 2023 Z	116,4%	111,0%	77,7%
Grudzień 2023 Z	131,7%	126,8%	91,1%

Jak widać w tabelach 7 i 9 ostatecznie w całym 5-letnim okresie strategia mieszana osiągnęła nieznacznie wyższą stopę zwrotu niż strategia agresywna, ale paradoksalnie uzyskując nieproporcjonalnie wyższy rezultat oczekiwanej stopy zwrotu według modelu CAPM, co może sugerować wyższe ryzyko powodowane zmiennością posiadanego portfela akcji giełdowych. Warto jednak zwrócić uwagę na fakt, że celem analizy długookresowej było sprawdzenie czy strategia agresywna osiąga ponadprzeciętne rezultaty przypadkiem i czy problem z pokonaniem oczekiwanej stopy zwrotu CAPM w pierwszym okresie testowym przewiduje osiągnięcie ponadprzeciętnych strat w przyszłości, które wyrównają uzyskaną stopę zwrotu w okolice indeksu cen. Choć w niektórych okresach w określonych warunkach model niezależnie od strategii miał problem by pokonać indeks cen, to jednak nic takiego się nie wydarzyło. Można stwierdzić że oszacowane ryzyko wynikające z modelu CAPM nie jest zagrożeniem dla *Agenta* DDPG-DRL, ponieważ jak pokazują rysunki 18 i 19 model ma zdolność, by dzięki swoim predykcjom konsekwentnie unikać jego realizacji. W szczególności warto zwrócić uwagę,

że największą stopę zwrotu strategia agresywna osiągnęła w I półroczu 2020 roku w trakcie odbicia po krachu pandemicznym.

Rysunek 18 *Porównanie wartości aktywów finansowych posiadanych w danym dniu przez Agenta DDPG-DRL do średniej cen rozważanych akcji giełdowych, oczekiwanej stopy zwrotu według CAPM oraz oprocentowania 10-letnich amerykańskich obligacji skarbowych w kroczącym okresie testowym styczeń 2019 – grudzień 2023 dla strategii mieszanej modelu zielonego*



Źródło: Opracowanie własne

Tabela 8 *Porównanie stopy zwrotu między strategią agresywną Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz oczekiwaną stopą zwrotu według CAPM przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL oddzielnie dla każdego półrocza w okresie styczeń 2019 – grudzień 2023*

Stopa zwrotu	Model zielony DDPG-DRL	Oczekiwana stopa zwrotu wg CAPM	Średnia cen rozważanych spółek
Czerwiec 2019 A	29,8%	33,3%	19,2%
Grudzień 2019 A	11,8%	9,8%	7,5%
Czerwiec 2020 A	+5,6%	-7,0%	-6,9%
Grudzień 2020 A	+18,0%	+19,8%	+19,2%
Czerwiec 2021 A	+11,9%	+12,3%	+17,8%
Grudzień 2021 A	+9,8%	+11,2%	+10,2%
Czerwiec 2022 A	-14,9%	-13,4%	-17,0%
Grudzień 2022 A	+1,2%	+6,7%	+6,6%
Czerwiec 2023 A	+7,6%	+11,7%	+8,8%
Grudzień 2023 A	+10,3%	+7,7%	+7,5%

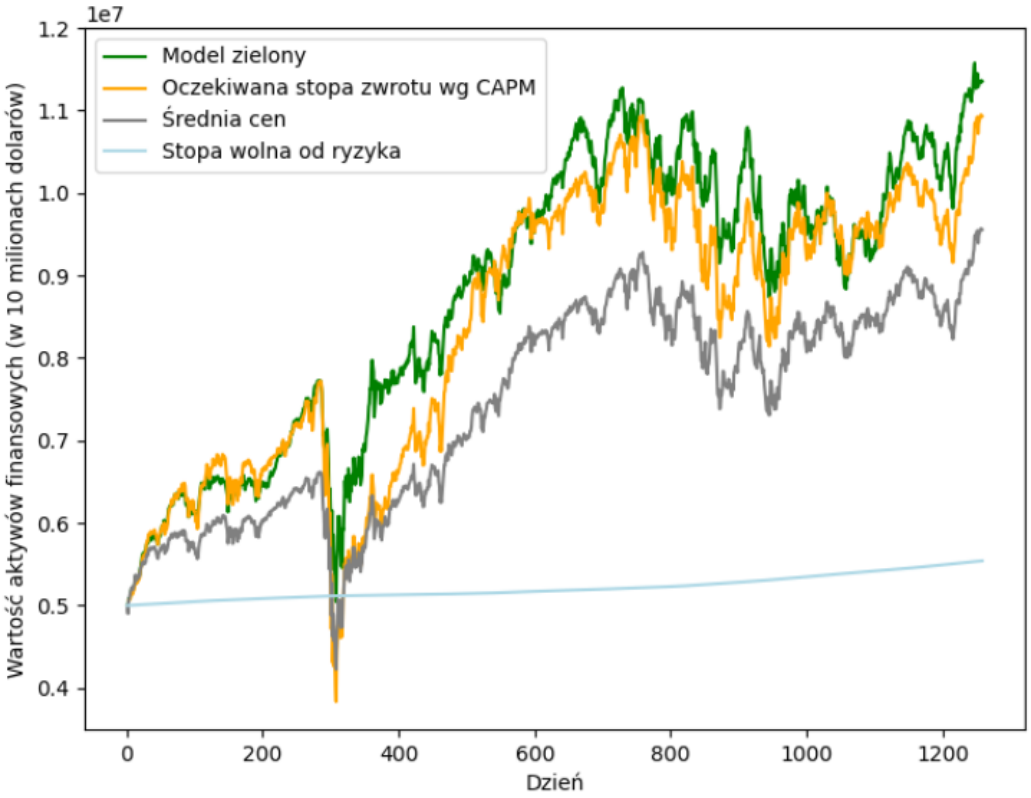
Jak widać w tabeli 8 strategia agresywna w niektórych okresach miała problem, by pokonać indeks cen, a wręcz przynosiłaby straty względem niego. Problem ten został całkowicie zdiagnozowany, ponieważ model w tych właśnie okresach utracił całkowicie swoją zdolność do rozróżniania wartości akcji giełdowych między sobą i zaczął rozkładać swoje środki losowo i z równomiernym wolumenem bez faworyzowania żadnej konkretnej wybranej spółki względem innych. Wynikało to z tego, że olbrzymia hossa, która zaczęła się w II półroczu 2020 niedługo po pandemii sprawiała, że większość akcji giełdowych znajdowało się w fazie wzrostu, a model wykazujący się w ocenie inwestycji dużą czułością na zmieniające się kierunki cen nie potrafił wypracować w okresie treningowym narzędzi, które pozwoliły by mu tak skutecznie rozróżniać możliwe alternatywy wzrostowe między sobą. O ile strategia zachowawcza biorąca pod uwagę wszystkie akcje giełdowe była w stanie znaleźć w tym okresie odpowiednie strategie inwestycyjne, pozwalające jej zachować swoją oszacowaną skuteczność predykcyjną, to strategia agresywna nie była w stanie znaleźć kilkunastu najlepszych alternatyw, gdy potencjalnie optymalny wybór był naprawdę szeroki. To znaczy strategia agresywna nadal potrafiła znaleźć spółki, które rosną, ale nie potrafiła znaleźć tych, które rosły najbardziej, co stwarzało problem z pokonaniem szybko rosnącego

indeksu cen. Próby innych konfiguracji modelu nie rozwiązały tego problemu. Strategia agresywna zaczęła odzyskiwać swoje zdolności rozróżniające w I półroczu 2022, by w II półroczu 2023 ponownie zacząć generować istotną przewagę stopy zwrotu nad indeksem cen. Rysunek 20 pokazuje, że zdolność strategii agresywnej do generowania ponadprzeciętnych zysków w II półroczu 2023 roku. Warto zauważyć, że faktyczny problem ze złożeniem nielosowego portfela akcji giełdowych w okresie olbrzymiej hossy mógł być przyczyną obniżonej zmienności przebiegu wykresu strategii agresywnej i osiągnięcia niższej oczekiwanej stopy zwrotu CAPM od strategii mieszanej.

Tabela 9 *Porównanie stopy zwrotu między strategią agresywną Agenta DDPG-DRL modelu zielonego, średnią ceną wszystkich rozważanych spółek oraz oczekiwaną stopą zwrotu według CAPM przy uwzględnieniu założonych opłat transakcyjnych Agenta DDPG-DRL kumulatywnie w kroczącym okresie testowym styczeń 2019 – grudzień 2023*

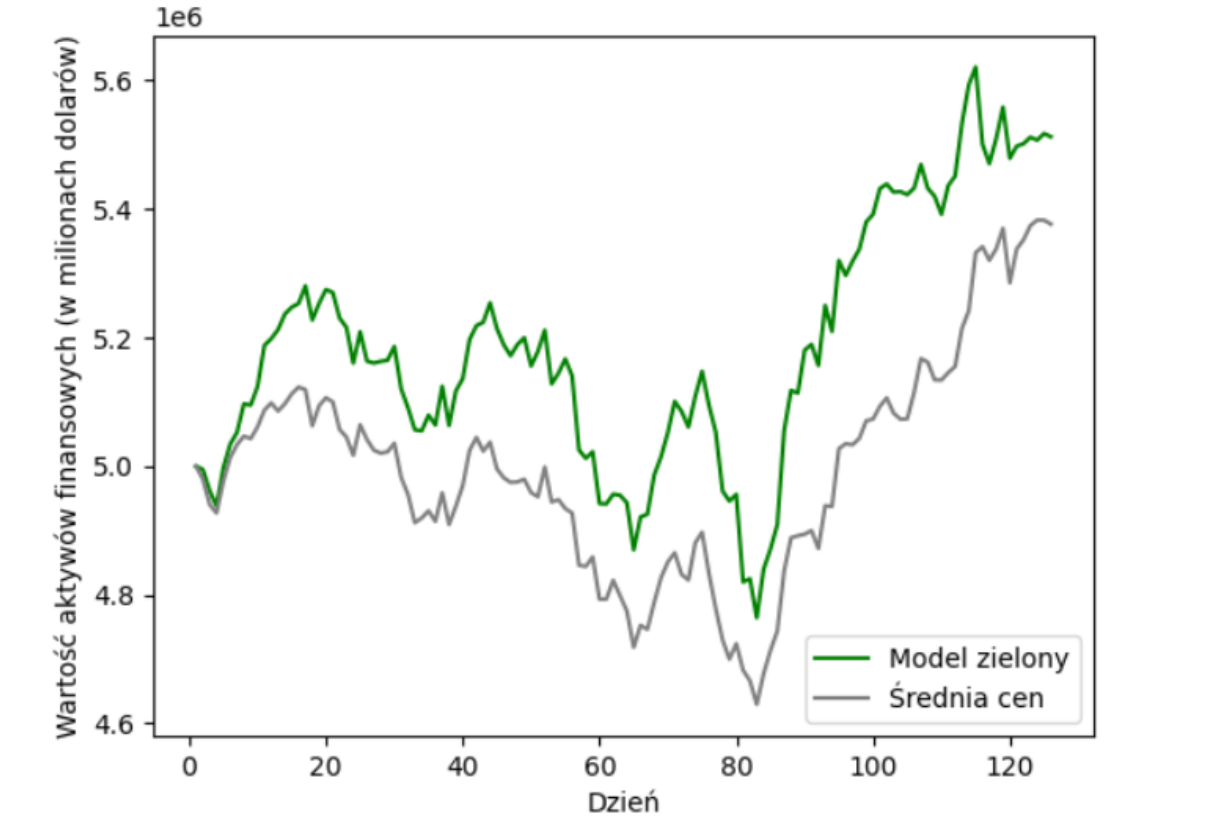
Stopa zwrotu	Model zielony DDPG-DRL	Oczekiwana stopa zwrotu wg CAPM	Średnia cen rozważanych spółek
Czerwiec 2019 A	29,8%	33,3%	19,2%
Grudzień 2019 A	45,1%	44,0%	28,1%
Czerwiec 2020 A	53,3%	23,3%	19,3%
Grudzień 2020 A	80,9%	69,1%	42,3%
Czerwiec 2021 A	102,4%	94,4%	67,6%
Grudzień 2021 A	122,2%	117,7%	84,7%
Czerwiec 2022 A	89,2%	72,8%	53,3%
Grudzień 2022 A	91,4%	86,3%	63,3%
Czerwiec 2023 A	105,9%	101,8%	77,7%
Grudzień 2023 A	127,0%	118,6%	91,1%

Rysunek 19 *Porównanie wartości aktywów finansowych posiadanych w danym dniu przez Agenta DDPG-DRL do średniej cen rozważanych akcji giełdowych, oczekiwanej stopy zwrotu według CAPM oraz oprocentowania 10-letnich amerykańskich obligacji skarbowych w kroczącym okresie testowym styczeń 2019 – grudzień 2023 dla strategii agresywnej modelu zielonego*



Źródło: Opracowanie własne

Rysunek 20 *Strategia agresywna modelu zielonego w okresie testowym II półrocza 2023 roku w porównaniu ze średnią cen rozważanych akcji giełdowych*



Źródło: Opracowanie własne

Zakończenie

Algorytmy uczenia maszynowego znajdują się nadal w fazie dynamicznego rozwoju, a przedstawiony w tej pracy algorytm głębokiego deterministycznego gradientu polityki DDPG jest efektem ostatnich lat badań nad metodami uczenia przez wzmacnianie. Eksploracja możliwych zastosowań ledwo nadąża za rozwojem teorii, a wiele z omawianych w poprzednich rozdziałach koncepcji całkiem niedawno dopiero pojawiało się w formie artykułów naukowych jako zupełnie nowe oryginalne rozwiązania. Celem pracy było sprawdzenie czy algorytm głębokiego uczenia przez wzmacnianie jest w stanie całkowicie samemu nauczyć się wniosków płynących z historycznych cen akcji giełdowych i analizy technicznej i wykorzystać je w ramach strategii obrotu tymi akcjami przynosząc ponadprzeciętne zyski względem indeksu cen. Z założenia nie jest to łatwe zadanie jako że teoria efektywnego rynku sugeruje, że ponadprzeciętny zysk osiągany w długim terminie na rynku jest mało prawdopodobny. Model zezwalający na opieranie strategii inwestycyjnej na sporej ilości akcji giełdowych pojedynczych spółek okazał się zauważalnie lepszy od indeksu cen, ale trzeba mieć świadomość, że jest to obarczone zwiększonym ryzykiem. W trakcie badanego 5-letniego okresu szacowane według modelu CAPM ryzyko strat się nie zrealizowało. Wyniki dla modelu ograniczającego ryzyko poprzez zmuszenie go do bardzo silnej dywersyfikacji środków były niekonkluzywne. Należy jednak podkreślić, że większość z wypróbowanych konfiguracji modelu na danych wykorzystanych w tej pracy na prawie żadnym etapie nie osiągnęła stopy zwrotu istotnie niższej od indeksu cen. Sprawia to że wykorzystanie tego mechanizmu do skonstruowania swojego portfela akcji staje się dosyć atrakcyjne. Na koniec trzeba jeszcze wspomnieć o tym, że mimo znaczącej ewolucji w świecie metod uczenia maszynowego wiele z nich jak chociażby sieci neuronowe czy klasyczne uczenie przez wzmacnianie bez wykorzystania głębokiego uczenia istnieją już od bardzo dawna, a najbardziej kluczową przyczyną ogromnego zainteresowania rozwojem i nowymi zastosowaniami tych algorytmów jest postęp w dostępnej mocy obliczeniowej, która jest niezbędna dla ich efektywnego wykorzystania. Pozornie nie jest to temat związany z tematyką tej pracy jednak warto mieć świadomość jak istotne praktyczne reperkusje to wywołuje. Korzystając z dowolnych najmocniejszych procesorów CPU dostępnych w sprzedaży detalicznej wykonanie obliczeń potrzebnych do uzyskania wszystkich wyników przedstawionych w tej pracy zajęłoby wiele tygodni, a uzyskany w ten sposób model mimo niewątpliwej teoretycznej słuszności zdążyłby już najprawdopodobniej stracić swój termin ważności. Dzięki kilku tysiącom rdzeni procesorów graficznych GPU obsługujących technologię NVIDIA CUDA wykonanie tych

obliczeń jest możliwe w obrębie jednego dnia i daje to szerokie możliwości tworzenia skutecznych modeli potrzebnych na bieżąco.

Bibliografia

[1] Chollet F., Deep Learning with Python, Manning Publications Co., 2018
[2] Sutton R.S., Barto A.G., Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 2018
[3] Morales M., Grokking Deep Reinforcement Learning, Manning Publications Co., 2020
[4] Lillicrap Timothy P., Hunt J., Pritzel A., Heess N., Erez T., Tassa Y., Silver D. & Wierstra D., Continous Control with Deep Reinforcement Learning, Google Deepmind, London, UK, 2016, online: https://arxiv.org/abs/1509.02971
[5] Sharma S., Activation Functions in Neural Network, Towards Data Science, 2017, online: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
[6] Ajagekar A., Adam, Cornell University, 2021, online: https://optimization.cbe.cornell.edu/index.php?title=Adam
[7] Bushaev V., Adam – latest trends in deep learning optimization, Towards Data Science, 2018, online: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c
[8] Yıldırım S., L1 and L2 Regularization — Explained, Towards Data Science, 2020, online: https://towardsdatascience.com/l1-and-l2-regularization-explained-874c3b03f668
[9] Odegua R., Stochastic vs Mini-batch training in Machine learning using Tensorflow and python, Coinmonks, 2018, online: https://medium.com/coinmonks/stochastic-vs-mini-batch-training-in-machine-learning-using-tensorflow-and-python-7f9709143ee2
[10] Sander R., Introduction to Experience Replay for Off-Policy Deep Reinforcement Learning, Towards Data Science, 2022, online: https://towardsdatascience.com/a-technical-introduction-to-experience-replay-for-off-policy-deep-reinforcement-learning-9812bc920a96
[11] Karunakaran D., „REINFORCE — a policy-gradient based reinforcement Learning algorithm", Intro to Artificial Intelligence, 2020, online: https://medium.com/intro-to-artificial-intelligence/reinforce-a-policy-gradient-based-reinforcement-learning-algorithm-84bde440c816
[12] Markov Decision Process, Wikipedia, online: https://en.wikipedia.org/wiki/Markov_decision_process
[13] Xiao-Yang Liu, Zhuoran Xiong, Shan Zhong, Hongyang Yang & Anwar Walid, Practical Deep Reinforcement Learning Approach for Stock Trading, arXiv.org, 2018

[14] Liu, Xiao-Yang and Li, Zechu and Zhu, Ming and Wang, Zhaoran and Zheng, Jiahao, ElegantRL: Massively Parallel Framework for Cloud-native Deep Reinforcement Learning, Github, 2021, online: https://github.com/AI4Finance-Foundation/ElegantRL
[15] Liu, Xiao-Yang and Li, Zechu and Yang, Zhuoran and Zheng, Jiahao and Wang, Zhaoran and Walid, Anwar and Guo, Jian and Jordan, Michael I, ElegantRL-Podracer: Scalable and elastic library for cloud-native deep reinforcement learning, NeurIPS, Workshop on Deep Reinforcement Learning, 2021
[16] Udacity, Deep Reinforcement Learning, Github, 2020, online: https://github.com/udacity/deep-reinforcement-learning
[17] Murphy J. J., Analiza techniczna rynków finansowych, New York Institute of Finance, 1999, wydanie polskie WIG-Press, przekład Madej W., Warszawa, 1999
[18] E. F. Fama, Efficient Capital Markets: A Review of Theory and Empirical Work, The Journal of Finance, 1970
[19] Titan A. G., The Efficient Market Hypothesis: Review of Specialized Literature and Empirical Research, Procedia Economics and Finance 32, 2015
[20] Showalter S., Gropp J., Validating Weak-form Market Efficiency in United States Stock Markets with Trend Deterministic Price Data and Machine Learning, Department of Economics, DePauw University, Greencastle, IN 46135, USA, 2019
[21] Lo A. W. and Mackinlay A. C, A Non-Random Walk Down Wall Street, Princeton University Press, 2001
[22] Patel J., Shah S., Thakkar P. and Kotecha K., Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Systems with Applications 42, 2015
[23] Kara Y., Boyacioglu M. A., Baykan Ö. K., Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange, Expert Systems with Applications 38, 2011
[24] Kenton W., Small Minus Big (SMB): Definition and Role in Fama/French Model, Investopedia, 2020
[25] Sławiński A., Chmielewska A., Zrozumieć rynki finansowe, Polskie Wydawnictwo Ekonomiczne, Warszawa 2017
[26] Knowles G., AI is better than humans at seeing patterns. Use it in L&D., LinkedIn, 2023
[27] Wstęgi Bollingera, Edukacja giełdowa, online: https://www.edukacjagieldowa.pl/gieldowe-abc/analiza-techniczna/narzedzia-analizy-technicznej/wstega-bollingera/

[28] Wstęgi Bollingera w tworzeniu strategii inwestycyjnych, admirals, online: https://admiralmarkets.com/pl/education/articles/forex-indicators/wstega-bollingera
[29] Aroon Indicator – wskaźnik analizy technicznej, Trader's Area, online: https://tradersarea.pl/aroon-indicator-wskaznik-analizy-technicznej/
[30] Świecie japońskie, opcje24.pl, 2019, online: https://www.opcje24h.pl/swiece-japonskie/
[31] Venketas W., How to Trade Bullish Flag Patterns, DailyFX, 2019, online: https://www.dailyfx.com/education/technical-analysis-chart-patterns/bull-flag.html
[32] Soni D., Supervised vs. Unsupervised Learning, Towards Data Science, 2018
[33] 9 Real-Life Examples of Reinforcement Learning, Santa Clara University, online: https://onlinedegrees.scu.edu/media/blog/9-examples-of-reinforcement-learning
[34] Bewtra A., The Ultimate Guide to Semi-Supervised Learning, v7labs, 2022, online: https://www.v7labs.com/blog/semi-supervised-learning-guide
[35] Steinke S., What's the difference between a matrix and a tensor?, Medium, 2017
[36] Akivis M.A. i Goldberg V.V., An Introduction to Linear Algebra and Tensors, Dover Publications, 2012, s. 50-54
[37] Bergstra J., Bengio Y., Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research, 13, 2012, s. 282
[38] Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning. MIT Press, 2016, s. 118, 119
[39] Sharpe W., Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk, Journal of Finance, 1964
[40] Lintner J., The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets, Review of Economics and Statistics, 1965
[41] Mossin J., Equilibrium in a Capital Asset Market, Econometrica, 1966
[42] Markowitz H., Portfolio Selection, Journal of Finance, 1952
[43] PyTorch Documentation, online: https://pytorch.org/docs/stable/index.html

Spis rysunków

Spis tabel

Streszczenie

Celem niniejszej pracy jest implementacja algorytmu głębokiego gradientu deterministycznej polityki DDPG stanowiącego algorytm ciągłej kontroli głębokiego uczenia przez wzmacnianie DRL w celu wyznaczenia optymalnej strategii obrotu akcjami giełdowymi na przykładzie spółek wchodzących w skład indeksu S&P500 z użyciem historycznych cen akcji oraz wskaźników analizy technicznej. Zakres pracy obejmuje kompleksowe przedstawienie składowych algorytmów metody DDPG-DRL w postaci głębokiej sieci neuronowej DNN, metod uczenia przez wzmacnianie RL oraz metod głębokiego uczenia przez wzmacnianie DRL, charakterystykę wybranych wskaźników technicznych i zastosowanej metodologii modelowania wraz z opisem konkretnych konfiguracji modelu oraz podsumowanie rezultatów dla zastosowanej implementacji, danych i okresów treningowych oraz testowych.

Algorytmy głębokiego uczenia w odpowiedzi na dostarczane informacje wypracowują reguły wiążące je ze sobą w wyniku działania algorytmu, a nie w wyniku ingerencji zewnętrznej. Wyróżniamy uczenie nadzorowane takie jak głębokie sieci neuronowe, uczenie nienadzorowane oraz uczenie przez wzmacnianie. Algorytm uczenia przez wzmacnianie RL wprowadza abstrakcyjne pojęcia takie jak stan, akcja i nagroda. Wartość akcji w algorytmie RL jest rozumiana jako oczekiwana skumulowana nagroda wynikająca z wyboru danej akcji pod warunkiem dalszego przestrzegania optymalnej polityki wyboru akcji. Algorytm głębokiego uczenia przez wzmacnianie jest połączeniem głębokich sieci neuronowych i uczenia przez wzmacnianie w celu aproksymacji wartości akcji, które musi podjąć agent na podstawie ciągłego zbioru wartości stanów algorytmu RL. DDPG jest odmianą algorytmu głębokiego uczenia przez wzmacnianie służącą deterministycznego wyznaczaniu akcji o ciągłym zbiorze wartości na podstawie stanu.

Wykorzystanie algorytmu DDPG pozwala na stworzenie modelu estymującego kształt optymalnego portfela akcji giełdowych w danym dniu. Agent modelu ma możliwość dowolnie kupować, sprzedawać i trzymać określoną ilość akcji giełdowych spółek wchodzących w skład indeksu S&P500, a celem jest maksymalizacja stopy zwrotu z początkowego salda posiadanych środków pieniężnych. Wybór algorytmu DDPG jest uzasadniony charakterystyką danych wejściowych i wyjściowych oraz złożonością całego problemu.

Uzyskany model w przypadku ryzykownej strategii zezwalającej agentowi na inwestowanie znaczącej ilości środków w akcje giełdowe pojedynczych spółek zapewnia zauważalnie wyższą stopę zwrotu od indeksu cen wszystkich rozważanych spółek. W przypadku strategii zmuszającej agenta do dywersyfikacji środków osiągnięte rezultaty są niekonkluzywne. Model nie ma problemu by odwzorować indeks cen, ale ponadprzeciętny zysk nie jest zagwarantowany.