

UNIWERSYTET GDAŃSKI

WYDZIAŁ ZARZĄDZANIA

Kierunek: Informatyka i ekonometria

Specjalność: Informatyka ekonomiczna

Patryk Jacek Laskowski

Nr albumu 240939

**Wykorzystanie sztucznych sieci neuronowych do predykcji
wartości kursów giełdowych w języku Python**

**Praca licencjacka napisana
w Katedrze Informatyki Ekonomicznej
pod kierunkiem dr Anny Lenart**

Sopot 2019

Spis treści

Wstęp.....	3
1. Podstawy sztucznej inteligencji	5
1.1. Od początków sztucznej inteligencji do uczenia głębokiego	5
1.2. Metody uczenia maszynowego	10
1.3. Rodzaje i zasady działania sztucznych sieci neuronowej	15
2. Wprowadzenie do rynków finansowych	26
2.1. Charakterystyka rynków finansowych	26
2.2. Funkcje i podział giełdy papierów wartościowych.....	31
2.3. Inwestycje na giełdzie papierów wartościowych	34
3. Narzędzia wykorzystywane w trakcie budowy modelu predykcyjnego.....	39
3.1. Język programowania Python	39
3.2. Podstawowe biblioteki wykorzystywane do uczenia maszynowego	42
3.3. Miary wykorzystywane do oceny wyników i ich porównywania.....	46
4. Wykorzystanie SSN do szacowania przyszłej wartości akcji firmy Apple	51
4.1. Etapy budowy modelu uczenia maszynowego.....	51
4.2. Wybór konstrukcji modelu sztucznych sieci neuronowych	57
4.3. Utworzenie oraz trenowanie modelu predykcyjnego SSN	61
4.4. Porównanie i interpretacja wyników uczenia maszynowego	68
Zakończenie	82
Bibliografia	84
Netografia	86
Spis rysunków.....	88

Wstęp

Stale rozwijający się poziom zaawansowania technologii powoduje, że wykorzystywanie rozwiniętych programów komputerowych, kiedyś dostępnych jedynie dla nielicznych, dzisiaj staje się coraz bardziej powszechne dla wszystkich. Już od bardzo dawna, naukowcy wykorzystując nauki matematyczne, statystyczne czy informatyczne starali się udoskonalać podejmowanie decyzji, by przyspieszyć, a nawet automatyzować pewne czynności, jednocześnie minimalizując poziom błędów spowodowanych czynnikiem ludzkim. Próby te pozwoliły znacznie zwiększyć skuteczność działań w wielu dziedzinach. Dzisiejsza technologia jest tak powszechnie dostępna i wysoce rozwinięta, że bezbłędne przeprowadzanie ogromnej ilości obliczeń w ciągu kilku chwil jest dostępne już praktycznie dla każdego. Można stwierdzić, że aktualnie nie istnieje żaden obszar, który nie korzystałby przynajmniej w minimalnym stopniu z tych udogodnień.

Wraz z nieustannie rozwijającymi się naukami o komputerach, psychologią czy biologią pojawia się realna szansa na stworzenie maszyn, potrafiących się uczyć. Ich celem, jest służyć ludziom, korzystając z nabytych w trakcie trenowania umiejętności, powodując jedynie pozytywne skutki w obszarach ich wdrożeń. Pożądana automatyzacja i wizje naukowców oparte o kategorię literacką *science-fiction* popychają naprzód badania nad rozwojem sztucznej inteligencji. Potencjalnych jej pożądanych zastosowań można wyliczać bez końca, od wprowadzenia usprawnień w życiu codziennym podczas eliminowania spamu w skrzynkach mailowych, aż po tak skomplikowane działania jak rozpoznawanie i leczenie chorób.

Chęć osiągnięcia pewnego poziomu niezależności w świecie, gdzie podstawą większości transakcji jest pieniądź, implikuje potrzebę jego zarabiania. Jednym z ciekawych sposobów pozyskiwania waluty jest gra na giełdzie papierów wartościowych, gdzie zyski bezpośrednio powiązane są z szybkością reakcji oraz umiejętnością skutecznej analizy i generowaniem właściwych decyzji inwestycyjnych. Najważniejszym komponentem sukcesu jest trafne prognozowanie przyszłości będące wypadkową wszystkich składowych wymienionych wcześniej. Obszar finansów również podatny na wykorzystanie technologii, posiada liczne wdrożenia prototypów maszyn, wspomagających decyzje inwestycyjne, a czasem nawet bezpośrednio je podejmujące, generując nierzadko pożądany przychód.

Celem niniejszej pracy jest weryfikacja hipotezy o możliwości satysfakcjonującego wdrożenia prostej sztucznej sieci neuronowej, będącej elementem sztucznej inteligencji, do prognozowania przyszłych zmian na rynku kapitałowym akcji. Sprawdzona zostanie jej skuteczność w podejmowaniu decyzji kupna-sprzedaży walorów elektronicznego rynku „*National Association of Securities Dealers Automated Quotations*” (NASDAQ), jednej z największej giełd na świecie.

Praca składa się z czterech rozdziałów. Pierwszy z nich poświęcono zagadnieniom związanym ze sztuczną inteligencją oraz uczeniem maszynowym. W pierwszej kolejności opisana jest historia jej powstawania, uwzględniająca zdarzenia mające kluczowy wpływ na jej dzisiejszą formę. Następnie przedstawiona zostaje propozycja podstawowego podziału uczenia maszynowego, by w końcu zrozumieć czym jest i jak działa szczególnie komponent sztucznej inteligencji, czyli sztuczna sieć neuronowa.

W rozdziale drugim zawarto charakterystykę rynków finansowych, przede wszystkim skupiając się na rynku kapitałowym, będącym podstawowym elementem badań poniższej pracy. Zadaniem rozdziału trzeciego jest zaznajomienie z narzędziami wykorzystywanymi w ramach omawianego badania, przedstawiając wykorzystany język programowania, jego biblioteki oraz miary, którymi posługiwać się będzie analityk do stwierdzenia skuteczności modelu. Ostatni rozdział zawiera praktyczną część pracy, w której opisane są również zasady budowy sieci. Następnie zgodnie z wytycznymi utworzone są pożądane modele. Stosując powstałe algorytmy szacowane są prognozy, by następnie oceniona została ich jakość wyłaniając najskuteczniejszy rodzaj sieci.

Do realizacji założonego w pracy celu wykorzystane zostały następujące metody badawcze: analiza literatury tradycyjnej i elektronicznej, analiza opisowa, sposoby analizy statystycznej i ekonometrycznej wraz z przeprowadzeniem prognoz w oparciu o systemy informatyczne. Wszystkie wymienione działania zostały połączone przy użyciu języka programowania Python wraz z dedykowanymi do tego dodatkowymi bibliotekami.

Praca została napisana z wykorzystaniem aktualnej literatury krajowej i zagranicznej oraz materiałów elektronicznych z zakresu: informatyki ekonomicznej, statystyki, ekonomii, ekonometrii, sztucznej inteligencji, technologii informatycznych, uczenia głębokiego oraz płytkiego, uczenia maszynowego i prognozowania.

1.Podstawy sztucznej inteligencji

1.1.Od początków sztucznej inteligencji do uczenia głębokiego

Chcąc zrozumieć, czym jest uczenie maszynowe, a w szczególności sieci neuronowe, należy najpierw zdefiniować trzy najważniejsze pojęcia jakimi są sztuczna inteligencja (ang. *artificial intelligence*, *AI*), uczenie maszynowe (ang. *machine learning*, *ML*) oraz głębokie uczenie (ang. *deep learning*, *DL*).

Już w starożytnej Grecji odnaleźć można znaki o początkach rozważań będących ziarnem, z którego po wielu stuleciach wyrosła znana nam dzisiaj dziedzina badań sztucznej inteligencji. Jednym z pierwszych śladów uznawanych przez niektórych za pierwsze kroki w tym kierunku jest mit o Pigmalionie, który w wolnym tłumaczeniu można zinterpretować jako powstanie pierwszej sztucznej istoty inteligentnej. Później na przestrzeni lat wraz z rozwojem nauk i świadomości człowieka powstawało wiele nowych pomysłów, idei a nawet prototypów zbliżających ludzkość do przełomowego momentu jakim było w roku 1946 powstanie komputera. „Elektroniczny, Numeryczny Integrator i Komputer” (ang. *Electronic Numerical Integrator And Computer*, *ENIAC*), uznawany jest za pierwszy na świecie komputer. Finansowany był przede wszystkim przez wojsko amerykańskie, z racji jego głównego celu jakim było wspomaganie obliczeń balistycznych. Kolejnym kamieniem milowym jest publikacja ukazana na łamach magazynu filozoficznego w 1950 roku przez Alana Turinga zatytułowana „*Computing Machinery and Intelligence*”. Wtedy A. Turing jako pierwszy oficjalnie postawił pytanie o możliwość myślenia przez maszyny. Artykuł opisywał Test Turinga, czyli swojego rodzaju grę w naśladowanie uznawaną przez znaczną część naukowców jako test maszyn na obecność inteligencji. Interakcja polegająca na próbie odgadnięcia przez sędziego czy rozmawia z programem, czy z człowiekiem. Komunikacja odbywa się jedynie poprzez tekst z wyłączeniem jakichkolwiek wizualnych i dźwiękowych informacji¹ o rozmówcy. Maszyna w tym wypadku powinna posiadać wiedzę z szerokiego spektrum zagadnień, gdyż nie ma żadnych nałożonych ograniczeń dotyczących tematyki rozmowy².

Narodziny „dzisiejszej” sztucznej inteligencji - kategorii naukowej, która została wykreowana czerpiąc inspiracje z takich dziedzin jak logika, informatyka, psychologia oraz również tak specyficznych dyscyplin badawczych jak nauka o mózgu i umyśle³ datuje się na lato roku 1956. Zorganizowana została wtedy pierwsza konferencja, celem której było zgromadzenie badaczy takich dziedzin jak automatyka, sieci neuronowe czy konstrukcja

¹ Informacja – „treść komunikatu przekazywanego za pomocą danych”. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 70.

² Kisiielewicz A., Sztuczna inteligencja i logika, Wydawnictwo WNT, Warszawa 2015, s. 33-37.

³ Kasperski M., Sztuczna Inteligencja, Helion, Gliwice 2003, s 13.

systemów⁴ pretendujących do miana inteligentnych. Podczas jej trwania organizator John McCarthy zaproponował nazwę nowej dziedziny – „*artificial intelligence*”, która została przyjęta przez wszystkich z entuzjazmem. Jednym z 10 uczestników był Marvin Minsky, amerykański naukowiec, którego prace miały ogromny wpływ na jej rozwój. Definiuje on pojęcie sztucznej inteligencji jako „dziedzina wiedzy, która postawiła sobie za cel i przedmiot badań maszyny, które potrafiłyby rozwiązywać zadania, przy rozwiązywaniu których człowiek korzysta ze swojej inteligencji”⁵. W trakcie dalszych rozważań i prowadzonych eksperymentów, kategoria rozwijała się coraz bardziej tworząc kolejne specjalności w jej zakresie.

Uczenie maszynowe (ang. *Machine Learning*, ML) wykreowane zostało właśnie podczas badań nad sztuczną inteligencją podczas drugiej połowy XX wieku. W 1959 roku Arthur Lee Samuel wprowadził ten termin po raz pierwszy. Jest to kategoria interdyscyplinarna uwzględniająca przede wszystkim informatykę, matematykę i statystykę. Autor definiuje pojęcie jako "*Field of study that gives computers the ability to learn without being explicitly programmed*"⁶ co tłumaczy się jako dziedzinę umożliwiającą komputerom uczenie się, bez jawnego ich programowania. A. Samuel był pionierem gier komputerowych⁷ i sztucznej inteligencji, jak również ojcem pierwszego programu samouczącego się, który odniósł wielki sukces⁸. Dziedzina rozwijała się za sprawą prób stworzenia programów, będących w stanie pozyskiwać wiedzę z danych⁹ i wykorzystywania ich do budowania prognoz¹⁰. Pozwoliło to na znaczne zoptymalizowanie mozolnej pracy ludzkiej, gdzie dotychczas ręcznie trzeba było wykonywać mnóstwo obliczeń analizując ogromne stopy informacji. Dzięki stale rozwijającej

⁴ System – „wyodrębniona część otaczającej rzeczywistości mająca pewną wewnętrzną strukturę, a więc składająca się z części uporządkowanych według ustalonych reguł określających ich wzajemne relacje”. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 90.

⁵ Kasperski M., Sztuczna Inteligencja, Helion, Gliwice 2003, s. 17.

⁶ Samuel A. L., Some studies in machine learning using the game of checker, IBM Journal of Research and Development, New York 1959, s. 535.

⁷ Grę komputerową należy rozumieć jako „program komputerowy, dzięki któremu użytkownik może rywalizować z komputerem bądź innymi graczami (na tym samym komputerze, bądź za pośrednictwem sieci), którego użytkowanie polega na rozwiązywaniu zadania logicznego lub zręcznościowego, co może służyć celom rozrywkowym lub edukacyjnym”. Kaczmarek-Templin B., Web 2.0 – treści cyfrowe a wirtualne światy gier. Wybrane problemy, WSS 2014, s. 89.

⁸ W roku 1956 zaprezentowano w telewizji dzieło A. Samuela – program grający w warcaby na poziomie turniejowym. Obalił on teorię o możliwości wykonywania poleceń jedynie poprzez ich jawne zaprogramowanie, gdyż algorytm niejednokrotnie pokonywał samego kreatora. Kisielewicz A., Sztuczna inteligencja i logika, Wydawnictwo WNT, Warszawa 2015, s. 42.

⁹ Dane - „fakty, zdarzenia, transakcje, które zostały zapisane. Stanowią one surowy materiał wejściowy, z którego produkowane są informacje”. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 69.

¹⁰ Prognozowanie jest racjonalnym i naukowym przewidywaniem przyszłych zdarzeń na podstawie prawdopodobnych zdarzeń dokonuje się wyboru w ramach określonego systemu, w oparciu historię przebiegu prognozowanego zjawiska. Ampuła D., Proces predykcji zdarzeń i zjawisk, Wojskowy Instytut Techniczny Uzbrojenia, Warszawa 2012, s. 37-38.

się technologii¹¹, idącej w parze z coraz większą skutecznością algorytmów oraz ich dostępnością, uczenie maszynowe częściej obecne jest w życiu codziennym. Od prostych filtrów antyspamowych w skrzynkach mailowych, aż po tak zaawansowane projekty jak samochody autonomiczne. Stały rozkwit dziedziny jaką jest ML skutkuje coraz większą liczbą wdrożeń zakończonych sukcesem.¹²

Kolejnym omawianym pojęciem jest uczenie głębokie¹³, czyli podzbiór będący częścią uczenia maszynowego, wykorzystujący algorytmy, których budowa i działanie inspirowane są biologicznymi sieciami neuronowymi. Zatem mówiąc o głębokim uczeniu wciąż na uwadze należy mieć ideę uczenia maszynowego, gdzie algorytmy są trenowane by rozwiązywać problemy w oparciu o dane, a następnie dokonywać podobnych wyborów na nowych obserwacjach. *Deep learning* to stosunkowo młoda dziedzina, bezpośrednio związana ze sztucznymi sieciami neuronowymi - SSN (ang. *Artificial Neural Network*, *ANN*). Początki badań nad SSN datuje się na rok 1943, gdy dwaj neurobiolodzy Warren McCulloch i Walter Pitts stworzyli teoretyczny model¹⁴ komórki mózgowej (model MCP¹⁵). Według ich koncepcji ludzki mózg przetwarza informację za pomocą sieci złożonej z ogromnej ilości¹⁶ pojedynczych komórek nerwowych zwanych neuronami. Jak widać na rysunku 1.1. komórka składa się z ciała, w którym zawarte jest jądro komórkowe, do której podłączone są rozgałęzienia zwane dendrytami oraz długiej wychodzącej z niej nici nazywanej aksonem. Zakończenia aksonu nazywa się synapsami. W sieci neurony łączą się ze sobą za pomocą synaps i dendrytów.

¹¹ Technologia informatyczna – IT (ang. *Information Technology*), to wszystko co dotyczy sprzętu informatycznego i telekomunikacyjnego wraz z ich oprogramowaniem oraz zasadami posługiwania się nimi. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 41.

¹² Rashka S., Python. Uczenie maszynowe, Gliwice, Helion 2018, s. 26.

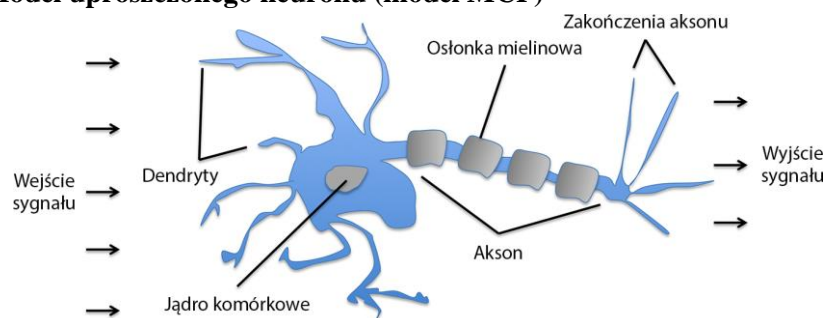
¹³ Wyróżnia się zarówno uczenie głębokie jak i uczenie płytkie. Obie grupy odnoszą się one do tej samej dziedziny, a różnica między nimi wynika z poziomu skomplikowania budowy modelu. Mówi się o uczeniu płytkim, gdy liczba warstw sieci, o których będę pisał dalej w tym rozdziale, nie przekracza dwóch. Uczenie głębokie dotyczy wszystkich tych sieci, których ilość warstw ukrytych jest większa niż dwa.

¹⁴ Model – „reprezentacja pewnego systemu oryginalnego na potrzeby konkretnego podmiotu. Model jest rezultatem procesu konstrukcji i jest sformułowany w konkretnym języku modelowania”. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 52.

¹⁵ MCP – nazwa pochodząca od pierwszych liter imion badaczy (ang. *McCulloch-Pitts neuron*).

¹⁶ Szacuje się, że ludzki mózg złożony jest z około 100 miliardów pojedynczych neuronów.

Rysunek 1.1. Model uproszczonego neuronu (model MCP)



Źródło: Raschka S., Python. Uczenie maszynowe, Helion, Gliwice 2018, s. 42.

Działanie neuronu przedstawiono na zasadzie bramki logicznej¹⁷, posiadającej wiele wejść (dendrytów), które dostarczają impulsy, czyli informacje przetwarzane następnie w jądrze. Wygenerowana w jego środku wartość, wydalana jest jako sygnał wyjściowy biegnący przez akson, aż do synaps. Impuls może być na tyle silny, że może powodować uwolnienie się związków chemicznych na ich końcach. Synapsy połączone są z dendrytami innych komórek otrzymujących wspomniane produkty reakcji w razie ich uwolnienia. W ten sposób neurony komunikują się ze sobą.¹⁸ Bodźcem do rozpoczęcia prac nad omawianym modelem było zainteresowanie działaniem ludzkiego mózgu oraz jego anatomia. Nie przeszkodziło to jednak by ich dzieło zaimplementowane zostało do takich dziedzin naukowych jak matematyka oraz informatyka¹⁹. Około 14 lat od publikacji modelu MCP, w roku 1957 Frank Rosenblatt i Charlie Wightman bazując na nim stworzyli działającą matematyczną reprezentację biologicznej komórki neuronowej – perceptron²⁰. Był on stworzony jako liniowy model zdolny do klasyfikacji binarnej, czyli potrafił przydzielić obiekt na podstawie jego parametrów do jednej z dwóch kategorii. Na rysunku 1.2. widnieje reprezentacja perceptronu. Porównując z poprzednim rysunkiem 1.1. można zobaczyć jak niewiele różnią się od siebie teoretycznie model biologiczny i matematyczny.

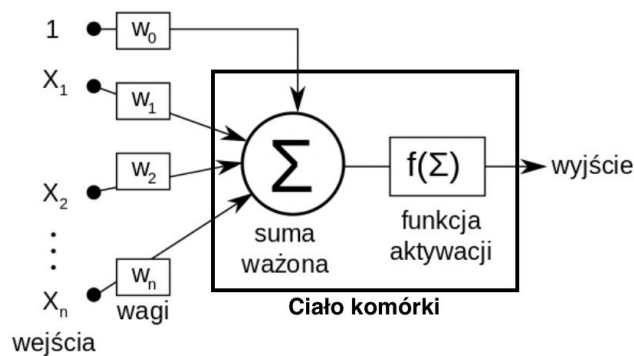
¹⁷ Bramka logiczna – element konstrukcyjny maszyn i mechanizmów, realizujący fizycznie pewną prostą funkcję logiczną, której argumenty oraz sama funkcja mogą przybierać jedną z dwóch wartości, np. 0 lub 1. Przykładowo, bramka logiczna LUB (ang. OR), pobierająca dwie wartości wejściowe, które równe są 1 lub 0 każda, oznaczające odpowiednio prawdę i fałsz zwraca wartość 1 w przypadku, gdy przynajmniej jedna z wartości wejściowych jest równa 1. Jeżeli obie wartości wejściowe równe są 0, wtedy bramka logiczna zwróci wartość 0. Technical Site, Bramki logiczne, <http://technical.elektronowolt.net/?page=bramki>, dostęp dnia 01.07.19.

¹⁸ Kisielewicz A., Sztuczna inteligencja i logika, Wydawnictwo WNT, Warszawa 2015, s. 287.

¹⁹ Szeliga M., Data Science i uczenie maszynowe, Wydawnictwo Naukowe PWN, Warszawa 2017, s. 193.

²⁰ W 1958 roku „New York Times” opublikował artykuł, w którym opisał perceptron jako „Zarodek elektronicznego komputera, który będzie mógł chodzić, mówić pisać, reprodukować się i być świadomym własnego istnienia”. Patterson J., Gibson A., Deep learning. Praktyczne wprowadzenie, Helion, Gliwice 2017, s. 57

Rysunek 1.2. Schemat neuronu matematycznego



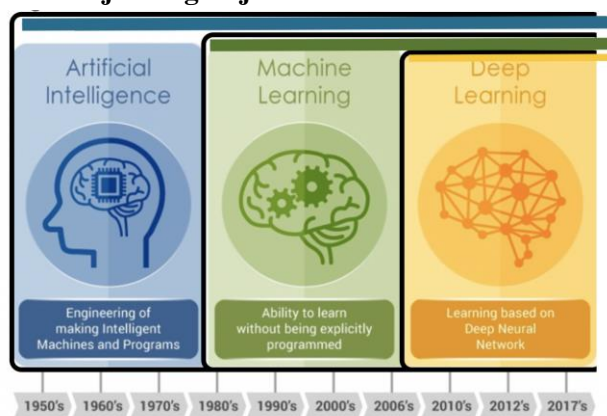
Źródło: Opracowanie własne na podstawie: Kawaguchi K., The McCulloch-Pitts Model of Neuron, <http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node12.html>, dostęp dnia 01.07.19.

Adekwatnie do idei komórek biologicznych ich matematyczne odpowiedniki również mogą być łączone ze sobą tworząc sieci. W sztucznych sieciach neuronowych pojedyncze neurony zazwyczaj uporządkowane są w warstwy (ang. *layers*), gdzie pierwsza i ostatnia z nich nazywane są odpowiednio warstwą wejścia (ang. *input layer*) oraz warstwą wyjścia (ang. *output layer*). Wszystkie inne zawierające się między nimi nazywane są ukrytymi (ang. *hidden layer*). Schemat perceptronu na rysunku 1.2. posiada tylko jedną warstwę. Istotną różnicą w matematycznym modelu jest pojawienie się funkcji aktywacji (ang. *activation function*), symulującej pracę pobudzenia neuronu. Jest to bardzo ważny element każdej komórki, odpowiadający za regulację siły sygnału przez nią przechodzącą. Opisuując neuron biologiczny, wspomniana została zależność uwolnienia związków chemicznych przez aksony od mocy wartości wyprodukowanej przez jądro komórkowe. Funkcja aktywacji pełni rolę regulującą siłę tego sygnału by odpowiednio akson stymulować. Ciało komórki, zaznaczone na rysunku 1.2, przetwarza wartości płynące z dendrytów, przedstawionych jako „wejście”, by później odpowiednio wygenerować informację wyjściową. Wielkość wytworzona przez komórkę płynie dalej, w tym przypadku niosąc informację o pobudzeniu neuronu. Zgodnie z akapitem wyżej, perceptron jednowarstwowy na rysunku 1.2. jest klasyfikatorem binarnym. Jego wyjście stanowi o tym czy neuron został pobudzony czy nie, w ten sposób reprezentując przynależność do jednej z dwóch klas.

Koncepcja wywołała ogromne poruszenie i rozkwit badań w kierunku sztucznej inteligencji. Niestety lata 70 XX w. okazały się momentem próby dla owych odkryć. Krytyka wynikająca z rezultatów badań Marviniego Minskiego oraz Seymoura Paperta spowolniła dynamiczny rozwój badań nad sieciami neuronowymi. Podkreślała ona głównie ograniczoność możliwości wykorzystania jednowarstwowych sztucznych sieci neuronowych. Przez kolejne lata zmniejszony entuzjazm sprowadził badania na boczny tor. Wielu badaczy nie dając za wygraną utrzymywało swoją fascynację i chęć rozwoju nad wykorzystaniem matematycznych

neuronów. Wraz z rozwojem dziedzin takich jak matematyka i informatyka, możliwości komputerów były coraz większe. Sytuacja diametralnie się zmieniła, gdy w drugiej połowie lat 80, XX w. wynaleziony został nowy sposób uczenia bardziej złożonych perceptronów. David Rumelhart, Ronald Williams oraz Geoffrey Hinton zaprezentowali metodę wstecznej propagacji błędów (ang. *backpropagation*²¹) do trenowania algorytmów wielowarstwowych perceptronów. Jest to pomysł w jaki istotność poszczególnych połączeń między neuronami wyrażana jako „waga” (ang. *weight*) są aktualizowane w taki sposób by pewna wybrana miara błędu była coraz to mniejsza osiągając w końcu poprzez kolejne iteracje²² poziom satysfakcjonujący.²³ To wielkie odkrycie dało początek kolejnej fali entuzjazmu sieci neuronowych, która z pewnymi wahaniami trwa do dziś.

Rysunek 1.3. Ewolucja sztucznej inteligencji



Źródło: Genc O., Notes on Artificial Intelligence, Machine Learning and Deep Learning for curious people, <https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2>, dostęp dnia 03.04.19.

Rysunek 1.3 wizualizuje podstawową zależność trzech omówionych w tym rozdziale terminów, jakimi są sztuczna inteligencja, uczenie maszynowe oraz głębokie uczenie. Kategorie zostały przedstawione na osi czasu, odpowiednio zagnieżdżone w sobie odwzorowując rzeczywiste relacje.

1.2. Metody uczenia maszynowego

Nauka wymaga również dodania pojęcia autonomiczności, gdyż można zarówno uczyć się samemu jak również można zyskiwać nowe umiejętności wspomagając się dodatkowymi

²¹ Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, Cambridge 2006, s. 241, 242.

²² Iteracja (łac. iteratio – powtarzanie) – czynność powtarzania wielokrotnie tego samego procesu w pętli, aż do spełnienia wyznaczonego warunku.

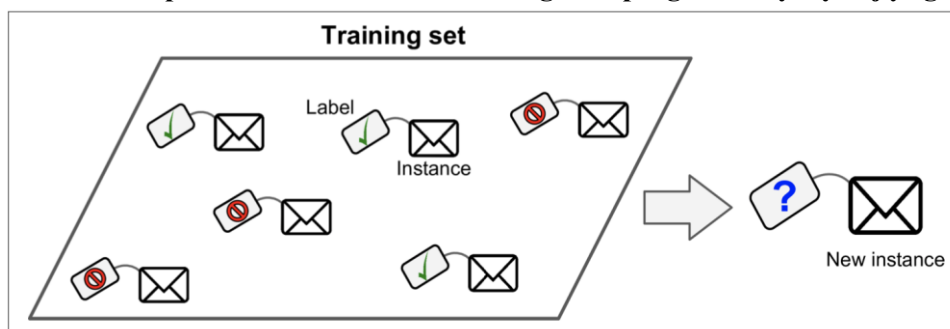
²³ Szeliga M., Data Science i uczenie maszynowe, Wydawnictwo Naukowe PWN, Warszawa 2017, s. 194.

informacjami płynącymi z zewnątrz. Z perspektywy rodzaju nadzorowania procesu wyróżnia się trzy podstawowe kategorie uczenia:

- uczenie nadzorowane,
- uczenie przez wzmocnienie,
- uczenie nienadzorowane²⁴.

Pierwsze z nich - uczenie nadzorowane (ang. *supervised learning*) - jest klasyczną i najczęściej stosowaną metodą strojenia algorytmu uczenia maszynowego. Słowo „nadzór” odnosi się do informacji, która jest oczekiwanym wyjściem dostarczany w parze z informacją wejściową. Można powiedzieć, że ten dodatkowy element pełni funkcję nauczyciela. Na podstawie oczekiwanej wartości wyjściowej maszyna porównuje wynik swojego szacunku z tym znajdującym się na „etykiecie”, uzyskując natychmiast informację zwrotną o jakości swoich przypuszczeń. W ten sposób metodą prób i błędów, algorytm „uczy się” poszukując optymalnego ustawienia parametrów, dzięki którym różnica pomiędzy wartością wyjściową a podaną wartością oczekiwaną będzie minimalna.

Rysunek 1.4. Schemat procesu uczenia nadzorowanego dla programu wykrywającego spam



Źródło: Geron A., Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly, Sebastopol 2019, s. 8.

Na rysunku 1.4 po lewej stronie widoczne są dane zaopatrzone w etykiety, czyli wartości wejściowe. Po prawej zaś stronie znajduje się nowy obiekt bez przyporządkowanej kategorii, którą to właśnie komputer powinien rozpoznać kierując się podobieństwem nowego obiektu do obiektów z puli posiadających swój typ. Jest to przykład jednej z podkategorii uczenia nadzorowanego nazywany klasyfikacją. Na przedstawionej ilustracji zaprezentowany jest konkretnie klasyfikator binarny²⁵, odnoszący się do jednej z dwóch etykiet reprezentujących

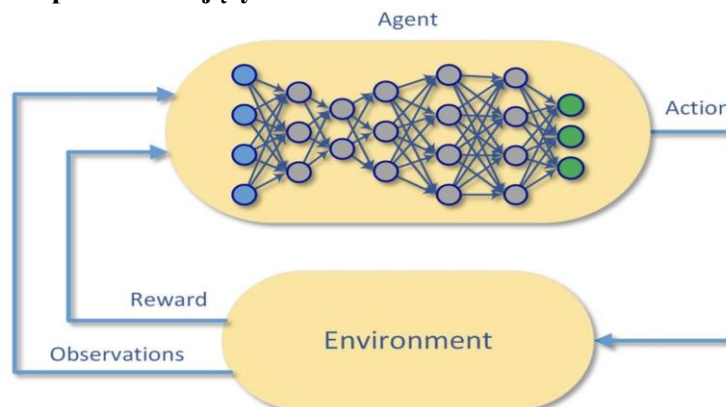
²⁴ Cichosz P., Systemy uczące się, Wydawnictwo Naukowo-Techniczne, Warszawa 2000, s. 33-35.

²⁵ Nazwa klasyfikacji binarnej odnosi się do systemu liczbowego dwójkowego, który posiada tylko dwie cyfry 0 i 1. Wartości można rozumieć jako dwojaki stan rzeczy: coś jest lub czegoś nie ma, prawda lub fałsz, tak lub nie. Komputery zbudowane są w oparciu o ten system ze względu na prostotę jego odwzorowania przez przepływ prądu, rozumiany jako 0 – brak prądu, 1 – prąd płynie. Innym typem jest klasyfikacja wieloklasowa. Wrycza S.,

przynależność do klasy spam lub nie. Zadaniem algorytmu jest ustalenie, do którego z dwóch przypadków należy nowa instancja.

Kolejnym rodzajem sposobu trenowania maszyn jest uczenie przez wzmocnienie (ang. *reinforcement learning*) inaczej nazywane uczeniem z krytykiem. Ten proces jest nieco podobny do poprzedniego. W tym przypadku komputer nazywany „agentem”²⁶, zamiast znanych wartości opisujących dane wejściowe, opiera się na funkcji, która jest wskaźnikiem jakości uczenia się modelu. To podejście można sobie łatwo wyobrazić przytaczając klasyczną metodę „kija i marchewki”²⁷. Maszyna otrzymuje nagrody lub kary opierając się na informacjach zwrotnych jakie zapewnia jej środowisko. Popołniając kolejne kroki system stara się maksymalizować wzmocnienie pozytywne, unikając jednocześnie kar, czyli negatywnych wzmocnień, co jednoznacznie jest ze wzrostem skuteczności. Poniższy przykład (rysunek 1.5) obrazuje prosty schemat działania uczenia z krytykiem.

Rysunek 1.5. Schemat przedstawiający uczenie ze wzmocnieniem



Źródło: Mohhamadi M., A conceptual structure of a deep reinforcement learning system, https://www.researchgate.net/figure/A-conceptual-structure-of-a-deep-reinforcement-learning-system_fig3_319121340, dostęp dnia 23.04.19.

Omówiony rodzaj uczenia można zaimplementować na przykład w silniku aplikacji grającej w szachy, gdzie agent dokonuje kolejne wybory ruchów figur w oparciu o aktualne rozłożenie pionków, czyli środowiska. Jego informacją zwrotną będzie zwycięstwo, sytuacja nagradzana bądź przegrana, czyli kara²⁸.

W trakcie trenowania modelu uczonego w sposób nadzorowany, wykorzystywane są z góry znane wartości wyjściowe poszczególnych wystąpień w bazie danych. Używając uczenia

Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 65-67.

²⁶ Agent to „program (jednostka) postrzegająca środowisko, w którym się znajduje, za pomocą sensorów i oddziałuje na to środowisko wykonując akcje”. Śnieżyński B., Uczenie się pojęć przez agentów jako metoda generowania strategii, Wydawnictwa AGH, Kraków 2013, s. 1.

²⁷ Metoda kija i marchewki wywodzi się z teorii motywacji. Jest to metoda polegająca na dawaniu nagród (marchewki) za zachowania pozytywne i używania kar (kija) za zachowania negatywne.

²⁸ Raschka S., Python. Uczenie maszynowe, Helion, Gliwice 2018, s. 29-30.

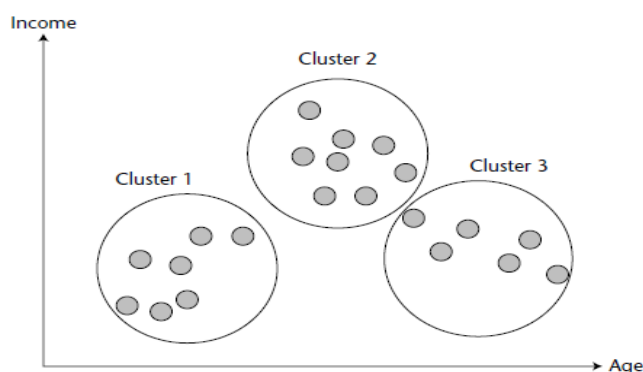
z „nauczycielem”, to jest, systemem motywacyjnym, relatywnie łatwe jest zdefiniowanie prawidłowych odpowiedzi, gdyż znany jest cel, do którego model dąży.

W przypadku trzeciej wymienionej kategorii jaką jest nauka nienadzorowana (ang. *unsupervised learning*) analitik posługuje się danymi bez oznaczeń lub takimi, których struktura jest bliżej nieokreślona. To o tyle szczególna metoda, gdyż algorytm trenuje się sam, co oznacza, że w trakcie ustalania parametrów nie bierze udziału żadna funkcja nadzorująca, określając jasno cel działania, jak i dane nie posiadają dodatkowych informacji, o ich bezpośrednim przyporządkowaniu do kategorii. Celem działania może być wizualizacja czy kompresja danych. Są to operacje pozwalające lepiej zrozumieć korelację²⁹ pomiędzy wartościami.

By łatwiej było wyobrazić sobie ideę uczenia nienadzorowanego, wykorzystane zostaną przykłady najpopularniejszych technik, czyli klasteryzacja i redukcja wymiarowości.

Pierwsze z nich nazywane również analizą skupień (grupowanie, ang. *clustering*) jest techniką szeroko stosowaną do badania rozkładu danych, która na podstawie dostępnych informacji dobiera elementy w podzbiory. W ten sposób możliwe jest odnajdywanie nieznanych dotychczas zależności. Zasadą jej działania jest dobranie takich grup, której elementy są do siebie maksymalnie podobne w jej obrębie, jednocześnie będąc różne od wystąpień spoza jej kręgu.

Rysunek 1.6. Przykład działania algorytmu metody analizy skupień



Źródło: Data mining tasks – data mining, <https://www.wisdomjobs.com/e-university/data-mining-tutorial-199/data-mining-tasks-1871.html>, dostęp dnia 11.05.19.

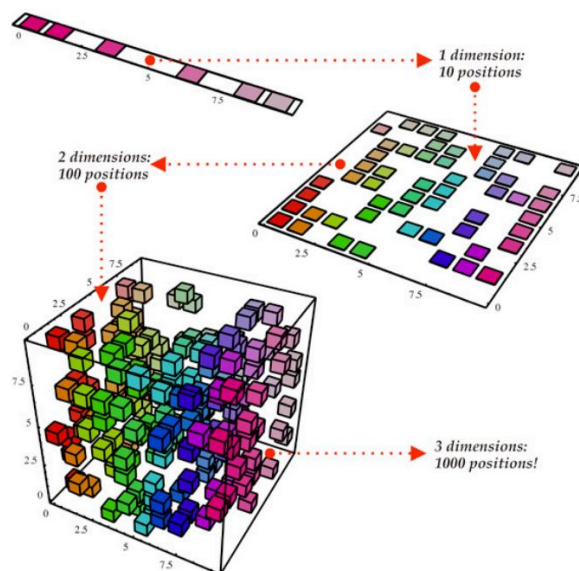
Przedstawiony rysunek 1.6 wizualizuje koncept działania algorytmu klastrującego. W tym przypadku model poszukiwał takich powiązań między obserwowanymi zmiennymi by wyznaczyć trzy podgrupy opierając się na wielkości przychodu (oś pionowa) i liczby lat (oś pozioma). Powyższą technikę wykorzystuje się między innymi do wykrywania anomalii w

²⁹ Korelacja – współzależność, wzajemne powiązanie. Bojańczyk M., Regresja i korelacja na światowych rynkach – w pałapce metod ilościowych, Oficyna Wydawnicza SGH, Warszawa 2013, s. 77.

zbiorach. Zjawiska typowe, znajdują się w jednej grupie, a każde wyjątkowe wystąpienie, które trafia poza jej obręb interpretowane jako odchylenie od normy. Taki schemat używany jest na przykład do wykrywania transakcji przeprowadzanych skradzionymi kartami przez złodziei.

Drugim wspomnianym i bardzo popularnym elementem uczenia nienadzorowanego jest redukcja liczby wymiarów (ang. *dimensionality reduction*). Metoda ta ma na celu ułatwienie pracy nad danymi a w szczególności ich analizy. W rzeczywistym świecie często dane są wielowymiarowe, przekładając się na znaczną grupę wartości pomiarowych co stanowi jeden z podstawowych wyzwań dla nośników danych czy skuteczności algorytmów uczenia maszynowego. Zadaniem modeli redukcyjnych jest kompresja danych bez utraty kluczowych informacji jakie ze sobą niosą, zachowując większość istotnych parametrów. Często zmiana liczby wymiarów wykorzystywana jest również w celach wizualizacji. Rysunek 1.7 przedstawia różnicę między danymi o różnej liczbie wymiarów.

Rysunek 1.7. Porównanie danych w różnych wymiarach



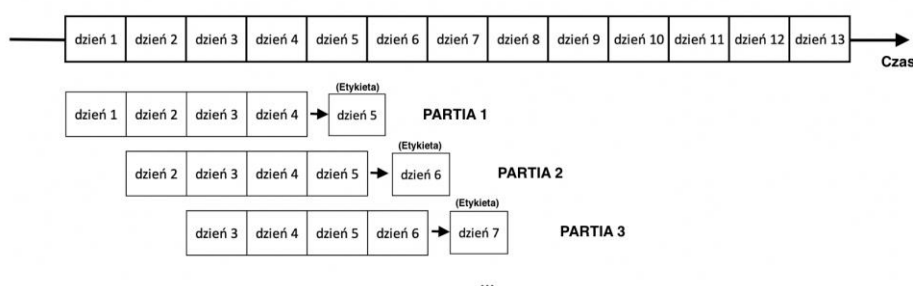
Źródło: Dude S., Curse of dimensionality, <https://bigsnarf.wordpress.com/2013/06/14/curse-of-dimensionality/>, dostęp dnia 28.06.19.

Na samym dole widoczny jest zestaw danych opisanych w trójwymiarowej kostce. Można odnaleźć w nich wiele wartości je opisujących jak wysokość, szerokość, długość (reprezentujące trzy wymiary), ale także inne jak kolor wraz z jego nasyceniem. Zmieniając odpowiednio punkt widzenia te same obiekty wyświetlone będą w formie dwuwymiarowej płaszczyzny. Dzięki modyfikacji perspektywy zredukowany zostaje znacząco rozmiar obiektu bez utraty pewnej ilości danych. Idąc krok dalej, uzyskaną z trójwymiarowego obiektu płaszczyznę można przedstawić jako jednowymiarową prostą. Powyższy sposób ma na celu ukazanie pewnej idei stojącej za redukcją liczby wymiarów przez algorytmy nienadzorowanego

uczenia. Zazwyczaj jednak osiągnięcie wybranych zamierzeń jest nieco bardziej skomplikowane.

W czwartym rozdziale pracy, wykorzystany zostaje nadzorowany typ uczenia maszynowego. Oznacza to, że każda obserwacja posiadać będzie swoją wartość, która ją opisuje. To właśnie ją model ma za zadanie jak najlepiej odwzorować poprzez „odnalezienie” i „nauczenie się” schematu działania danych. Wartościami wejściowymi, do których algorytm będzie oszacowywał etykietę, będą ceny zamknięcia akcji na giełdzie dla firmy Apple z pewnego o okresu czasu w przeszłości. Przykładowy sposób działania pokazany został na poniższym rysunku.

Rysunek 1.8. Idea wykorzystania uczenia nadzorowanego wykorzystana w projekcie niniejszej pracy



Źródło: Opracowanie własne.

W tym przypadku etykieta definiowana jest jako pierwsza wartość ceny zamknięcia, występująca zaraz po wprowadzonych danych historycznych. Ideę zastosowania przedstawiono na rysunku 1.8.

1.3. Rodzaje i zasady działania sztucznych sieci neuronowej

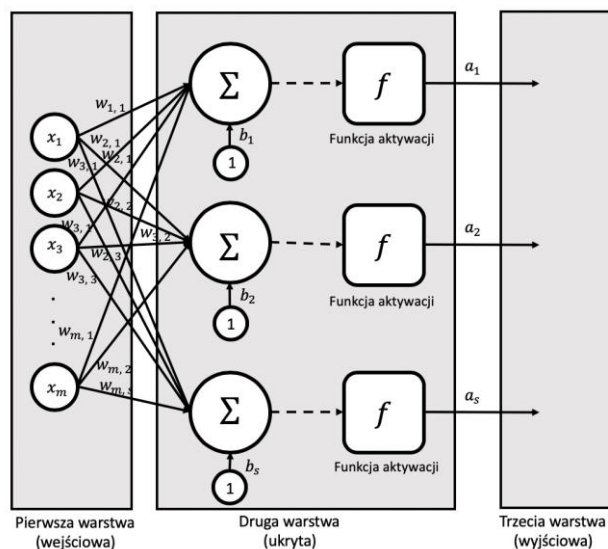
Pojedynczy neuron, będący podstawowym elementem sieci neuronowych jest względnie prostą jednostką, której „umiejętności” są dosyć mocno ograniczone. Takie mało skomplikowane podmioty, można jednak łączyć z łatwością nadając im tym samym coraz to większy poziom skomplikowania, a co za tym idzie coraz to większe możliwości. Jak już opisano wcześniej, struktury, składające się z wielu pojedynczych neuronów połączonych ze sobą nazywane są sieciami neuronowymi. Obiekty te łączy się najczęściej w kolumny nazywane warstwami.

Sposób działania całej sieci neuronowej zależy w dużej mierze od jej architektury, która determinowana jest poprzez właściwości takie jak liczba neuronów, ilość warstw oraz typy połączeń pomiędzy nimi. Gdy mowa jest o matematycznych imitacjach sieci neuronowych, jeden z kluczowych elementów, to również funkcja aktywacji występująca w

każdej warstwie. Dotychczas wspomniano tylko o jednej z nich, reprezentującą wartość pobudzenia neuronu interpretowaną jako przynależność do grupy 0 lub 1 (brak pobudzenia lub pobudzenie). Jest ich jednak znacznie więcej i wykorzystuje się je w zależności od obranego celu. Najważniejsze kategorie SSN oraz wybrane funkcje aktywacji omówione zostaną w tym podrozdziale.

Jeden z podstawowych podziałów sieci neuronowych uwzględnia ilość warstw w nich zawartych, gdzie najprostsza, posiada strukturę jednowarstwową (ang. *single-layer neural network*). Przykładowy schemat takiej sieci zaprezentowany jest na poniższym rysunku 1.9.

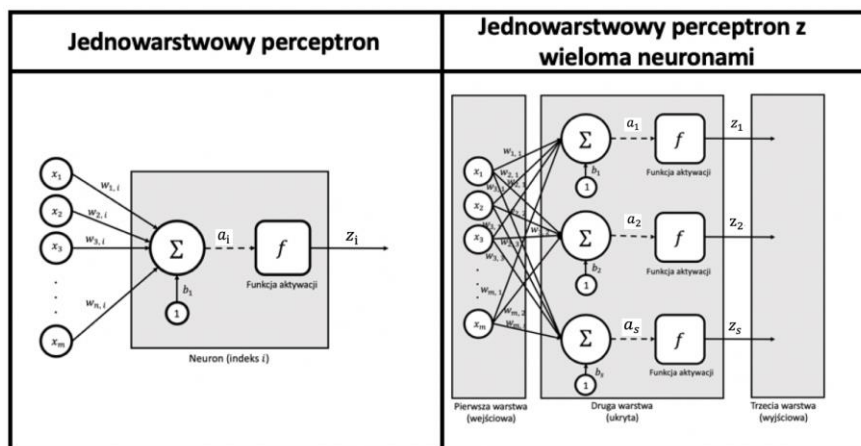
Rysunek 1.9. Jednowarstwowa sieć neuronowa



Źródło: Opracowanie własne na podstawie: Patterson J., Gibson A., Deep learning. Praktyczne wprowadzenie, Helion, Gliwice 2017, s. 58.

Należy zwrócić uwagę na pewien niuans związany z użyciem słowa „warstwa”. Pierwszy model neuronu opisany w pracy nazywany jest perceptronem, który składa się z pojedynczego neuronu i jednej warstwy. Gdy przywołuje się termin „warstwa” na myśli ma się blok złożony najczęściej z wielu neuronów. Następny rysunek 1.10 wskazuje na różnicę, którą warto zanotować, by lepiej zrozumieć działanie sieci neuronowych oraz ich podział.

Rysunek 1.10. Porównanie perceptronów o różnej strukturze



Źródło: Opracowanie własne na podstawie: Hagan M. T. [i. in.], *Neural Network Design*, Martin Hagan, Stwillwater 2014, s. 2-9.

Obie przedstawione podstawowe sieci posiadają jedynie jedną warstwę ukrytą natomiast ta znajdująca się po prawej stronie rysunku dodatkowo zbudowana jest z większej liczby neuronów. Jego poziom skomplikowania jest wyższy z racji różnicy w ilości użytych parametrów natomiast wciąż określa się go mianem jednowarstwowego.

Przepływ informacji na przedstawionym rysunku zilustrowany jest od strony lewej, gdzie wprowadzane są dane, do strony prawej, po której algorytm zwraca użytkownikowi wartości szacunku. Jak widać na załączonym na rysunku 1.9, sieć nazwana jednowarstwową, de facto posiada trzy warstwy, lecz z uwagi na fakt, że każda SSN posiada pierwszą i ostatnią warstwę, odpowiednio wejściową (ang. *input layer*) i wyjściową (ang. *output layer*), w trakcie nazywania pod uwagę bierze się jedynie warstwy pomiędzy nimi, określane ukrytymi (ang. *hidden layer*)³⁰. W modelach widocznych na rysunku 1.10. przedstawiono model sieci o m wejściach. Wartości wprowadzane zapisane są jako \mathbf{x} . Reprezentują one dane, na których podstawie model szacuje wartość wyjściową \mathbf{z} , czyli pożądaną wartość. Neurony łączą się z pierwszą ukrytą warstwą, zazwyczaj na zasadzie każdy z każdym. Do wszystkich połączeń przypisana jest wielkość wagi \mathbf{w} będąca reprezentacją istotności jaką posiada dane połączenie. Warstwa ukryta sumuje wszystkie ilorazy wartości \mathbf{x} oraz \mathbf{w} im odpowiadające dodając jeszcze sygnał progowy \mathbf{b} (ang. *bias*) zwracając wynik \mathbf{a} . Omówione działanie można przedstawić następująco:

$$\mathbf{a} = \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 + \dots + \mathbf{w}_m\mathbf{x}_m$$

Suma \mathbf{a} przekazywana jest dalej do funkcji aktywacji (ang. *activation function*), która po przetworzeniu informacji wsadowej zwraca wartość \mathbf{z} , będącą w tym wypadku wartością

³⁰ Flasiński M., *Wstęp do sztucznej inteligencji*, Polskie Wydawnictwo Naukowe PWN, Warszawa 2011, s. 159-177.

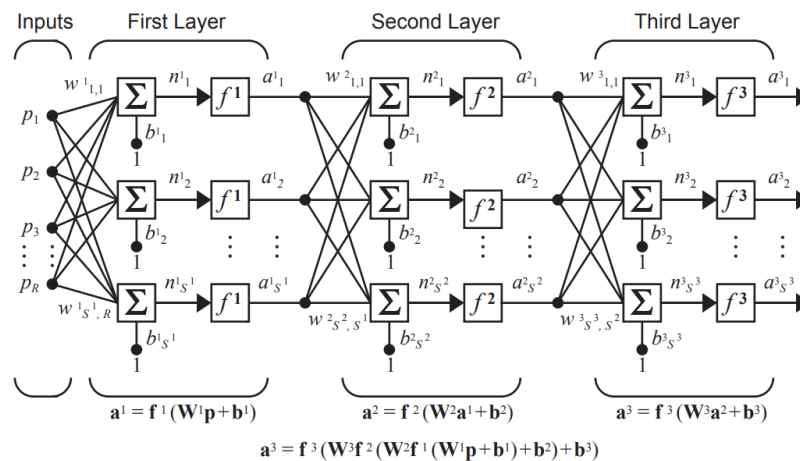
ostateczną zwracaną przez jednowarstwową sieć neuronową. To działanie w formie równania wygląda jak niżej:

$$\mathbf{z} = \mathbf{f}(\mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 + \dots + \mathbf{w}_m\mathbf{x}_m) = \mathbf{f}(\mathbf{a})$$

Wektor parametrów \mathbf{z} jest w opisanej sieci jednowarstwowej, blokiem wartości wyjściowych modelu³¹.

Przy wymagających problemach jedna warstwa, może okazać się niewystarczająca. Chcąc więc zwiększyć potencjał można posłużyć się jedną z metod jaką jest dodawanie kolejnych warstw. W ten sposób powstała konstrukcja o liczbie ukrytych bloków neuronów większej od jeden, określana jest mianem wielowarstwowej sieci neuronowej (ang. *multi-layer neural network*³²) pokazana na rysunku 1.11.

Rysunek 1.11. Trójwarstwowa sztuczna sieć neuronowa



Źródło: Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-9.

W strukturze takiego modelu neurony jednej warstwy przekazują sygnały do warstwy po niej następującej. Znajdując analogię do opisanej wyżej prostej sieci, wartości wyjściowe \mathbf{a} jednej ukrytej warstwy, stają się wartościami wejściowymi kolejnej ukrytej warstwy. Informacja w ten sposób przepływa przez sieć, aż do jej końca, gdzie zwraca ostateczny wektor wyników szacunku \mathbf{a} w warstwie wyjściowej.

Nie ma informacji na temat górnej granicy wolumenu ukrytych sieci co oznacza, że liczebność bloków w pełni zależy od upodobań i celów twórcy. Należy jednak zdawać sobie sprawę, że każda kolejna warstwa ukryta znacząco wydłuża proces uczenia całej sieci.

³¹ Najman K., Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie, Uniwersytet Gdański, Sopot 2003, s. 162-167. (praca doktorska)

³² Hearty J., Zaawansowane uczenie maszynowe z językiem Python, Helion, Gliwice 2017, s. 42, 43.

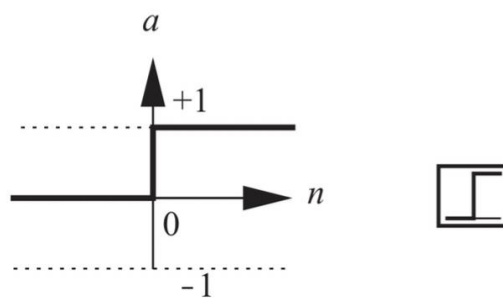
Inne kryterium podziału sieci opiera się o typ funkcji aktywacji w nich zastosowanej. Idea stojąca za wykorzystaniem funkcji aktywacji w neuronach matematycznych jest próbą odwzorowania sposobu komunikacji pomiędzy sobą ich biologicznych odpowiedników. Każdy neuron będący częścią prawdziwego mózgu aktywowany jest poprzez wykorzystanie jego potencjału. Jeśli wartość znajduje się w określonym przedziale neuron zostaje aktywowany. Zatem wykorzystanie funkcji aktywującej obiekt symulować ma pracę jego rzeczywistych pierwowzorów. Zazwyczaj dzieli się je ze względu na liniowość równania wyróżniając przy tym dwie jego postaci: postać liniową oraz postać nieliniową³³.

Obie z powyższych kategorii posiadają wiele przykładów zawierających się w nich funkcji. Niektóre programy pozwalają nawet na tworzenie własnych w zależności od potrzeb jakie stwarza rozwiązywane zagadnienie. Wybór konkretnej z nich determinowany jest od zadania jakiego analityk ma sprostować, konwencje ich użycia zostaną opisane w końcowej części tego podrozdziału.

Jako pierwszy przykład wybrana została funkcja skoku jednostkowego (ang. *hardlim*, *Hard Limit Transfer Function*). Prezentowana jako pierwsza, z uwagi na fakt wykorzystania jej do budowy modelu perceptronu w 1957 roku. Funkcja jak widać na rysunku 1.12. zwraca tylko jedną z dwóch wartości: 1 lub 0. Pierwotnie te wartości interpretowane były jako pobudzenie lub jego brak. Wartości progowe warunkujące wartość wyjściową funkcji skokowej wyglądają następująco:

$$f(a) = \begin{cases} 1 & \text{jeśli } a \geq 0 \\ 0 & \text{jeśli } a < 0 \end{cases}$$

Rysunek 1.12. Skokowa funkcja liniowa



Źródło: Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-4.

Gdy wartość sumy **a** jest większa bądź równa zero, funkcja generuje wartość 1 pobudzając kolejne komórki. Inaczej zwracana jest wartość 0.

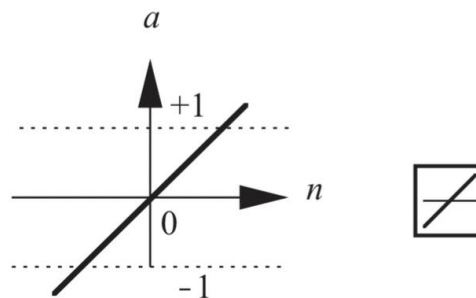
³³ Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-3, 2-4.

Innym przykładem z tej samej kategorii jest prosta liniowa funkcja aktywacji (ang. *linear activation function*). Nazywana również funkcją tożsamościową, z uwagi na jej działanie. Zwracana przez nią wartość jest dokładnie taka sama jak wartość do niej wprowadzona³⁴. Jej równanie wygląda następująco:

$$f(a) = a$$

Rysunek 1.13 jest graficzną reprezentacją zakresu w jakim działa omawiany rodzaj funkcji.

Rysunek 1.13. Liniowa funkcja aktywacji



Źródło: Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-5.

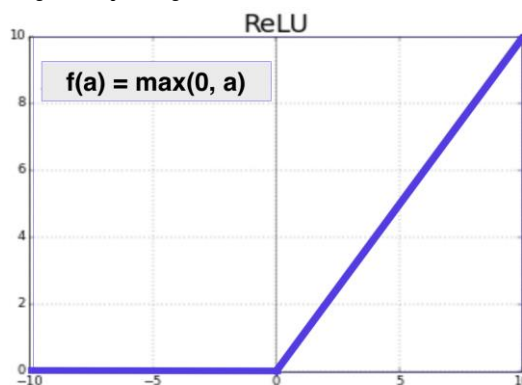
Ostatnim przykładem funkcji liniowej, opisanym w tej pracy jest obcięta funkcja liniowa – ReLU (ang. *rectified linear unit*). Nazywana również pozytywną funkcją liniową. Jest ona o tyle ciekawa, że jej przebieg niewiele różni się od prostej funkcji liniowej z wyjątkiem, że nie zwraca ona wartości ujemnych. Gdy wprowadzona wielkość jest większa od 0, zachowuje się identycznie do liniowej, natomiast gdy wartość wejściowa spada poniżej 0 wtedy funkcja zwraca 0. Matematycznie zapisuje się to w następujący sposób:

$$f(a) = \max(0, a)$$

Funkcja $f(a)$ zwraca maksymalną wartość z podanego zbioru $(0, a)$, z czego wynika brak możliwości zwrócenia liczby mniejszej od 0.

³⁴ Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-1-2-5.

Rysunek 1.14. Liniowa funkcja aktywacji ReLU



Źródło: Opracowanie własne

Obecnie funkcja przedstawiona na rysunku 1.14. bardzo zyskuje na popularności ze względu na praktycznie szybszy proces strojenia parametrów z wykorzystaniem ReLU w stosunku do pierwotnie bardzo popularnej sigmoidalnej funkcji, która opisana będzie w kolejnym przykładzie. Co więcej, dzięki odciętej funkcji liniowej w stosunku do zwykłej liniowej można poradzić sobie z problemem zanikającego gradientu. To zagadnienie, nie zostanie omówione w poniższej pracy ze względu na stopień zaawansowania problemu³⁵. W rozdziale 4. funkcja liniowa ReLU jest jedną z najczęściej wykorzystywanych przy budowie modeli³⁶.

Kolejną omawianą jest kategoria nieliniowych funkcji aktywacji. Jej podstawowy i najbardziej znany przykład opiera się o funkcję logistyczną (ang. *logistic sigmoid activation function*) nazywaną także funkcją sigmoidalną. Wykorzystuje ona przekształcenie logistyczne dzięki czemu jej ekstremalne wartości ulegają jedynie zmniejszeniu wciąż pozostając w zbiorze. Funkcję tę wykorzystuje się do przekształcenia dowolnej wartości w wielkość z przedziału między 0 a 1. Zwrócony wynik interpretowany jest jako poziom prawdopodobieństwa, przy czym 1 reprezentuje zdarzenie pewne. Wraz ze zmniejszaniem się wartości wyjściowej prawdopodobieństwo również maleje. Równanie funkcji wygląda następująco:

$$f(a) = \frac{1}{1 + e^{-a}}^{37}$$

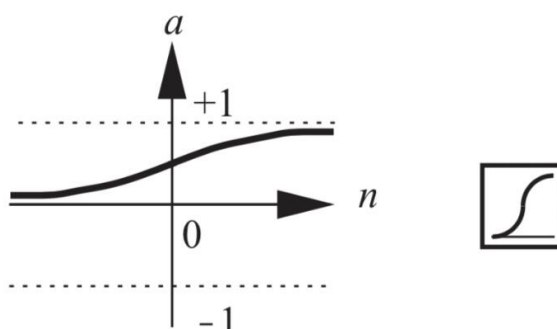
³⁵ Problem zanikającego gradientu jest spotykany w trakcie uczenia maszyn, gdy parametry optymalizowane są za pomocą metody gradientu prostego. Jest to jedna z popularnych metod matematycznych poszukiwania satysfakcjonujących wartości parametrów w sposób znacznie lepszy niż ich losowa aktualizacja. Chcąc zrozumieć proces „uczenia” modelu w stopniu zaawansowanym, zapoznanie się z tą metodą jest niezbędne.

³⁶ Zocca V., Spacagna G., Slater D., Roelants P., Deep learning. Uczenie głębokie z językiem Python, Helion, Gliwice 2017, s. 56–59.

³⁷ Kelleher J. D., Namee B. M., D’Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 357.

Obszar działania powyższego równania przedstawiony został na rysunku 1.15.

Rysunek 1.15. Logistyczna funkcja aktywacji



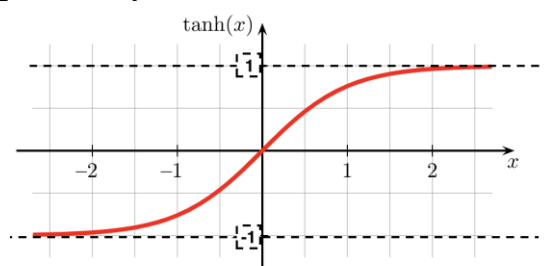
Źródło: Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-6.

Sigmoida nie uzyskuje wartości ujemnych. Jej zakres jest ograniczony w przedziale $< 0, 1 >$ w przeciwieństwie do funkcji tangensa hiperbolicznego (ang. *tanh*, *hyperbolic tangent sigmoid*), który jest kolejnym przykładem z kategorii nieliniowych.

Nakładając ograniczenia dla argumentów przyjmuje ona wartości z zakresu od -1 do 1. Dzięki temu wartość każdej wagi bardziej znormalizowane ze względu na symetryczne rozmieszczenie wokół zera. Gdy wartość a jest mniejsza od -1 lub większa od 1, przekształcana jest odpowiednio do wartości zbliżonej do -1 lub 1 mieszającej się we wspomnianym zakresie. Matematyczna reprezentacja funkcji oraz jej graficzna postać wyglądają następująco:

$$f(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

Rysunek 1.16. Tangens hiperboliczny



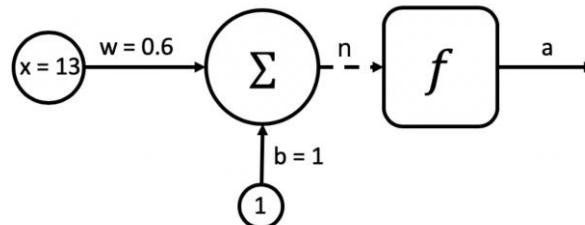
Źródło: Opracowanie własne na podstawie: Patterson J., Gibson A., Deep learning. Praktyczne wprowadzenie, Helion, Gliwice 2017, s. 75.

Jak widać na rysunku 1.16, funkcja tangensa hiperbolicznego jest symetrycznie rozmieszczona wokół zera, a jej granie wyznaczają punkty 1 oraz -1.

Dla porównania wszystkich wymienionych funkcji, przedstawię symulację jednej iteracji dla prostego perceptronu z jednym neuronem, posiadające jedno wejście oznaczone kolejno $x = 13$, odpowiadającą mu wagę $w = 0.6$ oraz wartość sygnału progowego $b = 1$.

$$n = wx + b = 0.6 * 13 + 1 = 8.8$$

Rysunek 1.17. Przykład prostego perceptronu z jednym neuronem



Źródło: Opracowanie własne.

Poniżej podane są wartości wyjściowe a neuronu, którego budowa przedstawiono na rysunku 1.17., wykorzystując każdą z wymienionych w tym rozdziale funkcji aktywacji:

- liniowa funkcja skokowa: $a = \text{hardlim}(8.8) = 1$
- funkcja prosta liniowa: $a = \text{linear}(8.8) = 8.8$
- funkcja pozytywna liniowa: $a = \text{relu}(8.8) = 8.8$
- funkcja logistyczna: $a = \text{sigmoid}(8.8) = 0.99985$
- funkcja tangensa hiperbolicznego: $a = \tanh(n) = 0.99999$

Wybór powyższych zależy w dużej mierze od celu, który analityk³⁸ chce osiągnąć. Przykładowo funkcję sigmoidalną wykorzystuje się w procesie klasyfikacji binarnej jak i wieloklasowej. W pierwszym przypadku warstwa wyjściowa może mieć jeden neuron, którego to wartość pomiędzy 0 a 1 będzie interpretowana jako przynależność do jednej z dwóch klas. Przy czym najlepiej, gdy wielkości będą jak najbardziej skrajne zapewniając tym samym wysoki poziom prawdopodobieństwa zaklasyfikowania do jednej z kategorii. Dla przypadku wieloklasowego, gdy celem projektu jest budowa sieci decydującej jaką wartość od 0 do 9 przedstawia wprowadzona, używając sigmoidalnej funkcji aktywacji, jednym z pomysłów jest umieszczenie dziesięciu neuronów w warstwie wyjściowej. Każdy z neuronów odpowiadać będzie jednej z dziesięciu wartości od 0 do 9. Funkcja zwracając wynik odpowiednio z przedziału od 0 do 1 dla każdego neuronu ostatniej warstwy, będzie informowała o poziomie pewności przynależności argumentu wejściowego do cyfry odpowiadającej konkretnemu

³⁸ Analiza danych to dziedzina zajmująca się zbieraniem i grupowaniem faktów naukowych będących faktycznym ilościowym lub jakościowym materiałem źródłowym, wyjaśnianiem ich by w końcu przedstawić efekty w formie opisu, Apanowicz J., Metodologia ogólna, Wydawnictwo Diecezji IV Pelplińskiej, Gdynia 2002, s. 7, 8.

neuronowi wyjściowemu. W najlepszym przypadku jeden z neuronów warstwy wyjściowej zwróci wartość 1 a cała reszta 0, co oznaczać będzie stuprocentową pewność przynależności do tylko jednej klasy.

Funkcje liniowe, takie jak liniowa prosta i liniowa pozytywna bardzo dobrze nadają się do szacowania linii regresji. Będą one jednak w stanie dopasowywać się jedynie do danych, których ułożenie wskazuje na liniowość. W przypadku problemów nieliniowych dobrym pomysłem może być użycie funkcji tangensa hiperbolicznego lub innych nieliniowych. Dla każdej warstwy funkcje wybiera się oddzielnie co daje możliwość mieszania ich ze sobą. Brak jest jednoznacznych wytycznych mówiących o doborze funkcji, finalnie, to jakie będą zastosowane w modelu zależy od jego twórcy. Często testuje się różne rozwiązania, dochodząc do satysfakcjonującego efektu metodą prób i błędów³⁹.

Dotychczas omówiony został podział sieci ze względu na ilość warstw wykorzystanych oraz typów funkcji aktywacji w nich zawartych. Istnieje jeszcze jeden podział sieci, którego klasyfikator opiera się na sposobie połączeń neuronów wpływając tym samym na kierunek przepływu informacji. Jest wiele różnych rodzajów połączeń, natomiast wśród omawianych w ramach pracy (warstwowych sieciach neuronowych), wyróżnia się dwie podstawowe kategorie⁴⁰: jednokierunkowe i rekurencyjne.

Pierwszą z nich, poruszana już wielokrotnie w tej pracy, jest jednokierunkowa sieć neuronowa (ang. *feedforward*). Jej działanie jest intuicyjne, ponieważ sygnał przebiega tylko w jedną stronę od wejścia (wprowadzanych danych) do wyjścia (rozwiązania proponowanego przez model). Wszystkie dotychczasowe przykłady były reprezentacją zawierającą się w tej kategorii. Zazwyczaj mimo wielu sposobów realizacji, stosuje się połączenie między neuronami na zasadzie każdy z każdym. Jednokierunkowa sieć jest najstarszym i najlepiej przebadanym rodzajem sieci, co wpływa na poziom zaufania i dużą popularność wykorzystania w praktyce. Atutem tego rodzaju jest z pewnością fakt, że w przypadku zbyt dużej wykorzystanej liczbie neuronów, model w takcie dostosowania wag jest w stanie pominąć mało istotne, nadając im wagi bliskie zeru.

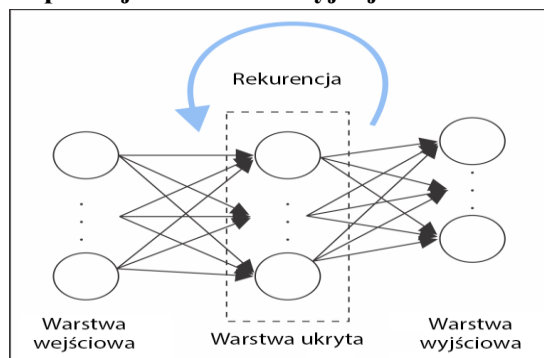
Gdy sygnał wyjściowy jednej warstwy przekazywany jest z powrotem łącząc się z wejściem tej samej lub innej, znajdującej się wcześniej warstwy to nazywa się ją rekurencyjną siecią neuronową (ang. *recurrent*). Praktyce taki stan rzeczy oznacza przepływ danych w dwóch kierunkach prowadząc przekaz zarówno do przodu z jednoczesnym wysyłaniem

³⁹ Najman K., Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie, Uniwersytet Gdański, Sopot 2003, s. 131-138. (praca doktorska)

⁴⁰ Najman K. M., Najman K., Samouczące się sztuczne sieci neuronowe w grupowaniu klasyfikacji danych, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2013, s. 126.

wartości zwrotnych. Jak widać na rysunku 1.18. jest to struktura bardzo podobna do jednokierunkowej sieci, z wyjątkiem zastosowania dodatkowej pętli sprzężenia zwrotnego⁴¹.

Rysunek 1.18. Ogólny model prostej sieci rekurencyjnej



Źródło: Raschka S., Python. Uczenie maszynowe, Helion, Gliwice 2018, s. 372.

Rekurencyjny model pobudza neurony przez określony czas, a następnie tymczasowo je dezaktywuje, przekazując sygnał do kolejnych komórek, które same w przyszłości zostaną stymulowane. Ich problemem jest poziom trudności w trenowaniu w porównaniu do sieci jednokierunkowych. Wynika to z braku dopasowanej metody optymalizacji wag dla tak specyficznej budowy oraz obecności dodatkowego parametru jakim jest przesunięcie czasowe funkcji. Ten typ sieci znajduje swoje zastosowanie między innymi w przypadku pracy z szeregami czasowymi oraz osiąga bardzo dobre rezultaty w takich dziedzinach jak rozpoznawanie mowy czy tłumaczeniu języków⁴².

⁴¹ Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014, s. 2-13-2-15.

⁴² Raschka S., Python. Uczenie maszynowe, Helion, Gliwice 2018, s. 371, 372.

2.Wprowadzenie do rynków finansowych

2.1.Charakterystyka rynków finansowych

W gospodarce, dzięki dokonywanym transakcjom kapitał nieustannie się przemieszcza. Podstawowe podmioty, jakimi są gospodarstwa domowe, przedsiębiorstwa i jednostki administracji wciąż realizują szereg umów, na których podstawie dochodzi do powstawania wzajemnych powiązań. Ten nieprzerwany obrót dóbr i usług odbywa się poprzez rynek finansowy, definiowany jako ogół wszystkich transakcji kupna-sprzedaży wykorzystujący wiele form kapitału pieniężnego, w różnym horyzoncie czasowym. Jego podstawowym elementem jest instrument finansowy, czyli kontrakt, na mocy którego jedna ze stron zyskuje zobowiązania finansowe wobec drugiej.

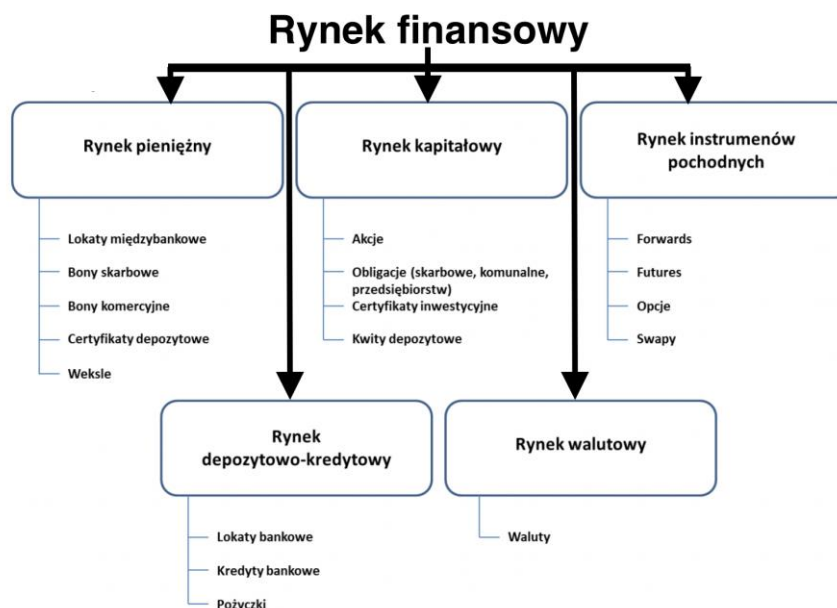
Z punktu widzenia rodzaju instrumentu, rynek kapitałowy, w ramach której przeprowadzona jest analiza w niniejszej pracy, jest jedną z pięciu kategorii rynku finansowego. Biorąc pod uwagę ten sposób podziału wyróżnia się następujące pojęcia⁴³:

- rynek pieniężny,
- rynek kapitałowy,
- rynek instrumentów pochodnych,
- rynek depozytowo-kredytowy,
- rynek walutowy.

W tym rozdziale zwięźle zostaną opisane pierwsza, trzecia, czwarta i piąta kategoria, by na końcu szczegółowo scharakteryzować drugą z nich, czyli rynek kapitałowy. Rysunek 2.1. jest graficzną reprezentacją zaproponowanego kryterium podziału.

⁴³ Górski M., Rynkowy system finansowy, Polskie Wydawnictwo Ekonomiczne, Warszawa 2009, s. 29.

Rysunek 2.1. Kategorie rynku finansowego z punktu widzenia rodzaju instrumentu finansowego



Źródło: Opracowanie własne na podstawie: Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo Naukowe PWN, Warszawa 2010, s. 21.

Na początku omówiony zostanie rynek pieniężny charakteryzujący się najwyższą płynnością aktywów⁴⁴ finansowych wykorzystywanych przez podmioty w celu zaspokojenia zapotrzebowania na kapitał obrotowy⁴⁵ zwiększając tym samym stopień ich płynności. Jest to cecha składników określająca poziom trudności zawierania transakcji dotyczących dóbr wszelkiego rodzaju w dowolnym momencie. Pieniądz wyróżnia się absolutną płynnością, jednocześnie nakładając obowiązek na każdy podmiot gospodarki, uznania go jako środka płatniczy. Zatem płynność instrumentów finansowych można rozumieć jako zdolność ich zamiany na gotówkę, w jak najkrótszym czasie, z jak najmniejszą stratą⁴⁶. Na rynku pieniężnym dochodzi głównie do transakcji mających na celu pozyskanie kapitału lub zapewnienie finansowej płynności poprzez korzystanie z krótkoterminowych⁴⁷ kredytów i pożyczek. Różnego rodzaju instytucje finansowe takie, jak banki komercyjne, szczególnie zdominowały

⁴⁴ Aktywa to pozyskane w przeszłości, posiadane zasoby majątkowe o określonej wartości, które wygenerują w przyszłości korzyści ekonomiczne. Setlak D., Aktywa i pasywa - definicja, przykłady i ich rodzaje, <https://6krokow.pl/aktywa-i-pasywa-definicja-przyklady-i-ich-rodzaje/>, dostęp dnia 04.07.19.

⁴⁵ Kapitał obrotowy to finansowane przez krótkoterminowe źródła aktywa bieżące, czyli takie, które mogą zostać upłynnione w okresie krótszym niż jeden rok. Pełka W., Finanse przedsiębiorstwa, <http://www.wseiz.pl/files/materialy/Finanse3.ppt>, dostęp dnia 03.07.19.

⁴⁶ Górski M., Rynkowy system finansowy, Polskie Wydawnictwo Ekonomiczne, Warszawa 2009, s. 30.

⁴⁷ Krótkoterminowe instrumenty finansowe definiowane są przez okres zapadalności poniżej jednego roku.

ten segment, odgrywając w nim główną rolę z nadzieją ulokowania nadmiaru kapitału. Podstawowymi jego instrumentami są weksle⁴⁸, bony skarbowe⁴⁹ i bony komercyjne⁵⁰.

Kolejną kategorię tworzy rynek instrumentów pochodnych, nazywany również rynkiem terminowym. Istotą jego działania jest możliwość zabezpieczenia się przed istniejącym ryzykiem wywołanym niekorzystnymi zmianami cen walut oraz papierów wartościowych. Przedmiotem obrotu są tutaj instrumenty terminowe. Walor ten jest rodzajem instrumentu finansowego, będący pochodną innego typu transakcji, w oparciu o jaką został wystawiony oraz, od której wartości zależy jego sukces. Głównymi składnikami tego sektora są kontrakty forward⁵¹, futures⁵², oraz opcje^{53 54}.

Następnym wyróżnionym elementem jest rynek depozytowo-kredytowy. Jego głównymi podmiotami są instytucje finansowe, takie jak banki komercyjne, banki spółdzielcze i inne jednostki związane z sektorem usług w tym zakresie. Za podstawowe operacje przeprowadzane w ramach tego sektora uznaje się udzielanie kredytów oraz pożyczek a także przyjmowanie depozytów. Zawierane operacje mają dużo niższą płynność w stosunku do rynku pieniężnego⁵⁵.

Rynek walutowy (ang. *Forex*) powstał w odpowiedzi na popyt zgłaszany przez użytkowników zawierających częste transakcje na arenie międzynarodowej. Jego angielska nazwa wywodzi się od stwierdzenia „*foreign exchange*” co znaczy po polsku „zagraniczna wymiana”. Należy do czołówki największych i najbardziej płynnych rynków świata. Potrzeba ciągłej wymiany walut wpływa na ryzyko związane z ich wahaniami kursu, przed którymi można się zabezpieczać dokonując transakcji w jego obrębie. Dawniej banki były głównymi

⁴⁸ Rodzaj papieru wartościowego, na mocy którego dłużnik zobligowany jest wypłacić nabywcy określoną kwotę w określonym czasie. Weksel, <https://www.biznes.gov.pl/pl/firma/kontrahenci-i-klienci/chce-nawiazac-wspolprace-z-kontrahentem/metody-zabezpieczenia-przed-nieuczciwymi-kontrahentami/weksel>, dostęp dnia 03.07.19.

⁴⁹ Bony skarbowe to inaczej krótkoterminowe papiery dłużne emitowane przez rząd. Sławiński A., Rynki finansowe, Polskie Wydawnictwo Ekonomiczne, Warszawa 2006, s. 29.

⁵⁰ Są tym samym co bony skarbowe, z tym wyjątkiem, że emitują je firmy.

⁵¹ Forward inaczej nazywana transakcją terminową zawieraną bezpośrednio między stronami, na wymianę z góry określonych dóbr, po z góry określonej cenie i w dniu wyznaczonym podczas jej zawierania. Dżuryk A., Rynki finansowe, [http://wzr.pl/read.php?plik=Rynki%20finansowe%20\[w\]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok](http://wzr.pl/read.php?plik=Rynki%20finansowe%20[w]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok), dostęp dnia 03.07.19.

⁵² Futures są notowane na rynkach giełdowych obligatoryjnymi transakcjami terminowymi, różniącymi się od transakcji forward głównie miejscem obrotu co wpływa na standardy, jakie musi spełniać. Górski M., Rynkowy system finansowy, Polskie Wydawnictwo Ekonomiczne, Warszawa 2009, s. 337-339.

⁵³ Kontrakt na mocy, którego zobowiązana jest tylko jedna strona, natomiast druga ma jedynie możliwość skorzystania. Dzielą się na opcje kupna i sprzedaży dając możliwość dokonania jednej z wymienionych transakcji w przyszłości po z góry ustalonej cenie. Zaremba A., Giełda. Podstawy inwestowania, Helion, Gliwice 2010, s. 81-83.

⁵⁴ Sławiński A., Rynki finansowe, Polskie Wydawnictwo Ekonomiczne, Warszawa 2006, s. 18, 19.

⁵⁵ Rynek finansowy w Polsce, Giełda Papierów Wartościowych, https://www.gpw.pl/pub/GPW/files/Rynek_finansowy_w_Polsce.pdf, dostęp dnia 01.07.19.

podmiotami omawianego rynku. Aktualna struktura charakteryzuje się większym zróżnicowaniem, jednak mimo to dysponują one wciąż największym kapitałem⁵⁶.

Ostatnia i najbardziej istotna z punktu widzenia tej pracy kategoria rynków finansowych, to wspomniany sektor nazywany rynkiem kapitałowym. Jest to przede wszystkim miejsce mobilizacji wolnych funduszy w celu poprawienia kondycji emitenta⁵⁷. Elementami będącymi podstawą jego funkcjonowania są papiery wartościowe rozumiane jako dokument stwierdzający prawo własności nazywany „akcją” (ang. *stock*, *share*) lub jako umowa potwierdzająca powstanie wierzytelności, nazywana wtedy „obligacją⁵⁸” (ang. *bond*). Oba z nich dają użytkownikom możliwość zarówno upłynnienia⁵⁹ jak również pozyskiwania kapitału. Rynek jest regulowany prawnie ze względu na występujące w nim transakcje o charakterze inwestycyjnym. Zgodnie z oceną gracza, dokonana inwestycja ma w zamian za wyrzeczenie się bieżącej konsumpcji przynieść pewnego rodzaju zysk. Warto podkreślić, że pojawienie się korzyści nie jest pewne, a ryzyko jego wystąpienia określa się w oparciu o poziom zmienności wartości, która stanowi podstawę transakcji.

Najważniejszymi podmiotami tego rynku są przedsiębiorstwa, instytucje finansowe jak również osoby fizyczne. Pierwsze dwa, są jego uczestnikami głównie ze względu na cel jakim jest pozyskanie kapitału na rozwój. Wszystkie jego podmioty mogą nabywać i sprzedawać zakupione dobra dostarczając tym samym pieniędzy i zwiększając jego wolumen transakcyjny⁶⁰ co pozytywnie wpływa na płynność omawianego sektora⁶¹.

Wyróżnia się dwie podstawowe kategorie rynku kapitałowego ze względu na podmiot, inicjujący transakcję:

- rynek pierwotny,
- rynek wtórny⁶².

⁵⁶ Ryłski M., *Forex – włącz emocje, wyłącz zyski. Przewodnik świadomego inwestora*, Helion, Gliwice 2009, s. 17, 18.

⁵⁷ Emitentem jest każdy podmiot finansowy, wystawiający na sprzedaż papier wartościowy we własnym imieniu w celu osiągnięcia własnych korzyści. Najbardziej rozpoznawalnym przykładem emitenta jest przedsiębiorstwo sprzedające swoje akcje. Socha A., Tuszyńska K., *Emitent*, <https://mfiles.pl/pl/index.php/Emitent>, dostęp dnia 03.07.19.

⁵⁸ Obligacja to instrument o bardzo niskim ryzyku oraz zysku nieco większym od zwykłej lokaty bankowej. Potencjalny zysk transakcji jest z góry znany. Składnik kupuje się od emitenta, który staje się dłużnikiem właściciela obligacji. Dłużnik zobowiązany jest do wykupu instrumentu w ratach powiększonych o odsetki, które to są zyskiem inwestora. Jagielnicki A., *Inwestycje giełdowe. Jak grać i wygrywać*, Helion, Gliwice 2007, s. 18.

⁵⁹ Zamiana na instrument o większej płynności, zazwyczaj rozumiane w kontekście sprzedaży, czyli wymiany na gotówkę.

⁶⁰ Wartość wszystkich dokonanych transakcji kupna i sprzedaży.

⁶¹ Dębski W., *Rynek finansowy i jego mechanizmy*, Wydawnictwo Naukowe PWN, Warszawa 2010, s. 91-94.

⁶² Kachniewski M., Majewski B., Wasilewski P., *Rynek kapitałowy i giełda papierów wartościowych*, Fundacja Edukacji i Rynku Kapitałowego, Warszawa 2008, s. 27-28.

Jak opisano wcześniej, istotą rynku kapitałowego jest możliwość pozyskania funduszy przez firmy, co z drugiej strony wiąże się z możliwością dokonywania transakcji o charakterze inwestycyjnym przez innych uczestników.

Rynek pierwotny (ang. *primary market*), jest kategorią rynku kapitałowego, na której dokonuje się wprowadzenia nowo wyemitowanych papierów wartościowych, gdzie sprzedawcą transakcji jest jednocześnie emitent⁶³. Sprzedaż udziałów, przeprowadzana na rynku pierwotnym przez spółkę po raz pierwszy nosi nazwę IPO⁶⁴, i jest to moment wejścia firmy na rynek akcji. Przyczyną powstawania takich ofert jest kusząca możliwość pozyskania dofinansowania w zamian za przekazanie prawa do części spółki, dzięki czemu emitent zyskuje kapitał a inwestor staje się współwłaścicielem⁶⁵. Zaletami rynku kapitałowego są z pewnością łatwość dostępu do funduszy, pozytywny wpływ na wiarygodność podmiotów w jego obrębie jak również możliwość wyceny firmy zapewnioną przez siły rynkowe. Głównymi wadami podjęcia decyzji o IPO przez spółkę jest ryzyko wrogiego przejęcia w drodze wykupienia akcji oraz upublicznienie wielu cennych informacji wcześniej nieznanych przez konkurencję⁶⁶.

Rynek wtórny (ang. *secondary market*) natomiast zapewnia obrót wprowadzonymi już poprzez rynek pierwotny instrumentami finansowymi pomiędzy inwestorami. Sprzedawcą w ramach tej kategorii jest kolejny właściciel instrumentu. Stał się on posiadaczem tychże walorów, poprzez dokonanie transakcji u innego gracza lub bezpośrednio u wystawcy na rynku pierwotnym. Dzięki skorzystaniu z możliwości sprzedaży papierów wartościowych, inwestor wyzbywa się generującego dochody instrumentu na rzecz absolutnej płynności finansowej⁶⁷.

To właśnie w ramach tej kategorii rynku kapitałowego przeprowadzane jest badanie skuteczności algorytmu uczenia maszynowego w odpowiedzi na pytanie dotyczące przyszłego zachowania rynku. Wykorzystana w ostatnim rozdziale strategia zakłada pozyskiwanie instrumentów poprzez interakcję z innymi graczami bez udziału spółki, która je wyemitowała.

⁶³ Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo Naukowe PWN, Warszawa 2010, s. 103, 104.

⁶⁴ IPO (ang. *Initial Public Offering*) co tłumaczy się jako „pierwsza publiczna oferta”. Zawila-Niedźwiecki J., Rostek K., Gąsiorkiewicz A., (red.), Informatyka gospodarcza, Wydawnictwo C.H. Beck, Warszawa 2010, s. 86.

⁶⁵ Banaszczyk-Soroka U., Instytucje i uczestnicy rynku kapitałowego, Wydawnictwo Naukowe PWN, Warszawa 2008, s. 10.

⁶⁶ Dżurk A., Rynki finansowe, [http://wzr.pl/read.php?plik=Rynki%20finansowe%20\[w\]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok](http://wzr.pl/read.php?plik=Rynki%20finansowe%20[w]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok), dostęp dnia 03.07.19.

⁶⁷ Górski M., Rynkowy system finansowy, Polskie Wydawnictwo Ekonomiczne, Warszawa 2009, s. 279-281.

2.2. Funkcje i podział giełdy papierów wartościowych

Giełda papierów wartościowych (ang. *stock exchange*), w skrócie GPW to podstawowy element rynku kapitałowego. Jest ona nieodłącznym i podstawowym elementem wchodzącym w skład kategorii rynków wtórnych.

Z uwagi na typ oraz sposób prowadzenia transakcji, głównymi funkcjami⁶⁸, spełnianymi przez GPW są mobilizacja oraz transformacja kapitału, jak również jego wycena.

Pierwsza z nich (mobilizacja kapitału) to zdolność przepływu dóbr od uczestników posiadających jego nadmiar do tych, którzy wykazują na nie zapotrzebowanie.

Transformacja kapitału, rozumiana jest jako przywilej natychmiastowej zamiany formy posiadanego majątku w ramach giełdy. Przykładem zastosowania tej funkcji jest konwersja pieniędzy na akcje w drodze ich zakupu.

Trzecią funkcją jaką jest wycena kapitału, w jasny sposób wskazuje na możliwość ustalania wartości spółki, kształtowanej przez popyt i podaż, które to kreowane są przez inwestorów. Pozawala ona wyznaczyć na jakim poziomie notowane jest całe przedsiębiorstwo.

Warto również wspomnieć o roli kontrolnej jaką sprawuje giełda. Wiąże się to z obowiązkami narzucanymi na spółki dotyczące podstawowych wymogów informacyjnych, które egzekwują większą ich transparentność⁶⁹.

Samą giełdę papierów wartościowych określa się jako „regularne, odbywające się w określonym czasie i miejscu, podporządkowane określonym normom i zasadom, spotkania osób pragnących zawrzeć umowę kupna-sprzedaży oraz osób pośredniczących w zawieraniu transakcji, których przedmiot stanowią zamienne papiery wartościowe, przy czym ceny owych transakcji ustalane są na podstawie układu podaży i popytu, a następnie podawane do wiadomości publicznej”⁷⁰. Z definicji wydedukować można pewne uniwersalne atrybuty, które są niezbędne by mogła ona sprawnie funkcjonować: regulamin określający zasady dokonywania transakcji oraz zarząd pełniący funkcje nadzorcze.

Z uwagi na specyfikę obracanymi na giełdzie papierami wartościowymi, wyróżnia się dwie grupy. Podstawą transakcji pierwszej z nich są fundamentalne papiery wartościowe, czyli akcje i obligacje. Ich reprezentantami są takie podmioty jak największa na świecie giełda nowojorska NYSE (ang. *New York Stock Exchange*) czy londyńska LSE (ang. *London Stock Exchange*).

⁶⁸ Kachniewski M., Majewski B., Wasilewski P., Rynek kapitałowy i giełda papierów wartościowych, Fundacja Edukacji i Rynku Kapitałowego, Warszawa 2008, s. 37-41.

⁶⁹ Dżuryk A., Rynki finansowe, [http://wzr.pl/read.php?plik=Rynki%20finansowe%20\[w\]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok](http://wzr.pl/read.php?plik=Rynki%20finansowe%20[w]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok), dostęp dnia 03.07.19.

⁷⁰ Dorosz A., Puławski M., Giełdy w gospodarce światowej, PWE, Warszawa 1991, s. 120.

Drugą natomiast grupą są giełdy, których kluczowym instrumentem obrotu są instrumenty pochodne, a w szczególności takie walory jak opcje i kontrakty futures. Ten rodzaj rynków nazywa się również terminowymi. Do ich reprezentantów należą takie giełdy jak położona w Chicago – *Chicago Board of Trade*, czy mieszcząca się w Londynie – *London International Financial Future Exchange*. Oba rodzaje giełd połączone są czasami w ramach jednego dużego podmiotu, zazwyczaj jednak ich obrót odbywa się oddzielnie⁷¹.

Ze względu na miejsce dokonywanych transakcji oraz wysokość poziomu siły regulacji giełda dzieli się na dwie kategorie⁷²:

- rynek giełdowy
- rynek pozagiełdowy.

Rynek giełdowy jest z wyżej wymienionych znacznie starszą formą gospodarczą i wyróżnia go tradycyjny, zorganizowany sposób prowadzenia zleceń odbywający się w wyznaczonej lokalizacji na parkiecie giełdowym (ang. *ring*). Charakteryzuje się silnymi przepisami normującymi czynności przebiegu wraz z wytycznymi dotyczącymi progu wejścia. Stopień regulacji podyktowany jest interesem ochrony jego konsumentów. Początkowo wytyczne narzucane przez regulamin rynku giełdowego determinowały ilość uczestników niezbędnych do jego prowadzenia. Z czasem, coraz większą popularność zyskuje handel algorytmiczny i systemy informatyczne⁷³, dzięki czemu giełdy nie wyglądają już tak jak w XX wieku, gdy maklerzy⁷⁴ przekrzykiwali się chcąc wykonać powierzone zlecenia ustępując teraz miejsca algorytmom komputerowym zarządzającym w ciszy wszystkimi transakcjami. Oprócz samego przebiegu sesji, regulamin tworzy również bariery wejścia, będące dla niektórych firm sporym wyzwaniem. Podstawowe wytyczne, które instytucja chcąc wejść na rynek giełdowy musi spełniać dotyczą⁷⁵:

- jej wielkości – podmiot ma kwalifikować się jako „duży” oraz wykazywać adekwatny rozmiar zysków wraz z wykazaniem odpowiednio wysokiej jego wartością bądź sporej ilości posiadanego kapitału (dokładne limity każda giełda ustala indywidualnie);

⁷¹ Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo Naukowe PWN, Warszawa 2010, s. 139-142.

⁷² Zawila-Niedźwiecki J., Rostek K., Gąsioriewicz A., (red.), Informatyka gospodarcza, Wydawnictwo C.H. Beck, Warszawa 2010, s. 77.

⁷³ System informatyczny to składający się z ludzi i systemów komputerowych obszar, zorientowany na uzyskiwanie określonych zadań. Zawila-Niedźwiecki J., Rostek K., Gąsioriewicz A., (red.), Informatyka gospodarcza, Wydawnictwo C.H. Beck, Warszawa 2010, s. 41.

⁷⁴ Makler papierów wartościowych to osoba, która zawodowo zajmuje się pośrednictwem w przeprowadzaniu transakcji kupna i sprzedaży wspomnianych instrumentów. Thiel S., Rynek kapitałowy i terminowy, Komisja Nadzoru Finansowego, Warszawa 2010, s. 24.

⁷⁵ Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo Naukowe PWN, Warszawa 2010, s. 152.

- ilości walorów jakie wprowadza - powinny być liczne i rozproszone. Założenie to co najmniej 1,1 mln wprowadzanych akcji, przy czym przynajmniej 2 000 akcjonariuszy powinno posiadać ich nie mniej niż 100;
- długości życia - firma musi posiadać przynajmniej 3-letnią historię na rynku;
- jawności informacji – podmioty przed wejściem na giełdę muszą stworzyć odpowiednie akty informujące o ich kondycji. Również po wkroczeniu na rynek giełdowy firmy zobligowane są regularnie dostarczać oświadczenia dotyczące, aktualnego stanu finansowego.

Tylko nieliczne instytucje potrafią przekroczyć powyższe progi, co skłoniło innych uczestników, dostrzegających potencjał związany z wejściem na giełdę by wyklarować własną dziedzinę gospodarczą, która reprezentowana jest przez drugą wspomnianą kategorię, czyli rynek pozagiełdowy.

Rynek pozagiełdowy (ang. *over the counter market, OTC-market*), określa wszystkie transakcje kupna-sprzedaży papierów wartościowych zawieranych poza regularną giełdą. Ten rodzaj handlu nie posiada sztywnych wymogów regulujących. Handel w jego obrębie jest zupełnie niezwiązany z żadnym konkretnym miejscem. Co do zasady przebiega on na własnych zasadach pomiędzy bankami, firmami maklerskimi oraz ich użytkownikami. Charakteryzuje się zazwyczaj znacznie mniejszymi obrotami w stosunku do rynku giełdowego jak również znacznie większym poziomem ryzyka. Pozytywnymi aspektami wykreowania takiego obszaru jest zmniejszenie kosztu emisji oraz stworzenie atrakcyjnej szansy pozyskania kapitału przez młode firmy. Dodatkowo, instytucje mierzą się ze znacznie niższymi wymogami informacyjnymi, które muszą spełniać w związku z chęcią rozpoczęcia działalności jak również jej kontynuowania. Powyższe aspekty z pewnością niekorzystnie wpływają to na status samej giełdy, w porównaniu do regulowanego odpowiednika. Charakterystyczne dla rynku pozagiełdowego są z pewnością obniżona płynność kapitału oraz niższa transparentność⁷⁶.

Najbardziej znaną instytucją będącą częścią sektora pozagiełdowego jest amerykański elektroniczny system obrotu NASDAQ - *National Association of Securities Dealers Automated Quotations*, który powstał w 1971 roku. Był to pierwszy w pełni elektroniczny system obrotu papierami wartościowymi, co znaczy, że obrót na niej odbywa się jedynie za pośrednictwem sieci telefonicznych i komputerowych. Z racji jej statusu nie posiada ona tradycyjnego parkietu giełdowego. W zbiorze spółek, które są na niej notowane znajdują się znane na całym świecie firmy sektora zaawansowanych technologii informatycznych takich jak Apple, Microsoft czy Google. Co ciekawe, w roku 1999, NASDAQ, reprezentująca rynek pozagiełdowy, po raz pierwszy przekroczyła poziom kapitalizacji największej na świecie NYSE, czyli głównego i

⁷⁶ Kachniewski M., Majewski B., Wasilewski P., Rynek kapitałowy i giełda papierów wartościowych, Fundacja Edukacji i Rynku Kapitałowego, Warszawa 2008, s. 29, 30.

największego reprezentanta rynku giełdowego. Jest to podmiot szczególnie istotny z punktu widzenia tej pracy ze względu na wspomnianą firmę Apple, której wielkości cen zamknięcia będą szacowane w ostatnim rozdziale.⁷⁷

Rysunek 2.2. Wejście główne do giełdy NASDAQ w dniu 08.02.18



Źródło: Burgess R., Apple, US\$1 trillion and stock market bubbles, <https://www.thestar.com.my/business/business-news/2018/08/06/apple-us1-trillion-and-stock-market-bubbles/>, dostęp dnia 04.07.19.

Powyższy rysunek przedstawia siedzibę NASDAQ z dnia 02.08.2018 roku, mieszczącą się na Manhattanie w centrum Nowego Jorku. Marka Apple widnieje na szyldzie by uczcić sukces osiągnięcia progu jej wyceny rzędu \$1 biliona. Była to pierwsza na świecie spółka, która osiągnęła ten imponujący wynik. Tego dnia cena zamknięcia pojedynczej akcji firmy Apple wynosiła \$207,5⁷⁸.

2.3. Inwestycje na giełdzie papierów wartościowych

Uczestników, inwestujących na giełdzie można de facto podzielić na dwie grupy. Pierwszą z nich są ludzie, którzy podejmują decyzje inwestycyjne robiąc to w oparciu o swoje przypuszczenia dotyczące przyszłości. Drugą natomiast grupą są ci, którzy jedynie podążają za resztą graczy starając się jak najbardziej dopasować do aktualnych trendów i bieżących sytuacji. Każda z nich posiada pozytywne uzasadnienie zastosowania i jednocześnie brak jest jednego, najlepszego rozwiązania, a gracze najczęściej mieszają obie razem.

Koncepcja podążania za innymi jest zdecydowanie łatwiejszą drogą, gdyż wykorzystywane są tylko proste narzędzia analizujące jedynie aktualną kondycję rynku by następnie odpowiednio zająć ustaloną pozycję. Jest to jednak czasochłonny proces, z racji potrzeby ciągłych analiz by bezustannie znać bieżącą sytuację rynkową.

⁷⁷ Dębski W. Rynek finansowy i jego mechanizmy, Wydawnictwo naukowe PWN, Warszawa 2010, s. 147-151.

⁷⁸ Lis M., Apple pierwszą w historii Wall Street spółką o takiej wartości. Przebiła bilion dolarów, <https://www.money.pl/gospodarka/wiadomosci/artukul/apple-pierwsza-w-historii-wall-street-spolka,244,0,2412532.html>, dostęp dnia 01.07.19.

Sposób działania uwzględniający szacowanie przyszłości wydaje się być lepszą możliwością. Ta ścieżka wymaga jednak dużej wiedzy oraz danych dotyczących rynku. Prognozy można podzielić na racjonalne i nieracjonalne. Pierwsze opierają się o logiczny ciąg wydarzeń poczynając od hipotezy kończąc na konkluzji. Jeśli w ramach tego procesu niewykorzystywane są naukowe narzędzia a kolejne kroki oparte są jedynie o wydarzenia z przeszłości mówi się o racjonalnym prognozowaniu zdroworozsądkowym. W drugiej strony, jeśli do prowadzenia logicznych rozważań gracz stosuje się do wykreowanych przez naukę zasad mowa jest o racjonalnym przewidywaniu naukowym. Każde inne domniemanie o przyszłych zdarzeniach, które wykluczają wytyczne poprzednich opisów nazywane jest nieracjonalnym⁷⁹. Przykładami rozwiązań nieracjonalnych mogą być przeczucia czy wróżby. Warto zauważyć, że nie wszystkie zjawiska możliwe są do przewidzenia, takie jak kradzieże, oszustwa czy tok myślenia i strategie innych inwestorów.

Koncepcją o efektywności rynków finansowych kieruje teza, iż poziom cen instrumentów jest odzwierciedleniem oczekiwań uczestników, dotyczących ich przyszłej wartości. Zmiany notowań są zatem częściowo determinowane przez zjawiska, których nie da się planować co przekłada się na ich nieprzewidywalność. Ten element nazywany jest również ryzykiem⁸⁰. Różnica szans na niepowodzenie wynika on również ze zmienności cen waloru, mogącej się nieoczekiwanie odchylić w górę lub dół. Wielkość potencjalnych zysków z inwestycji, zgodnie z modelem równowagi rynku kapitałowego (ang. *capital asset pricing model*, CAPM) jest wprost proporcjonalna do poziomu podejmowanego ryzyka⁸¹. Oznacza to, że chcąc zwiększyć skalę stopy zwrotu z inwestycji, trzeba liczyć się z większą szansą na poniesienie porażki⁸².

Sposób liczenia przyszłych korzyści wynikających z zaangażowania kapitału determinowany jest przez różnego rodzaju aspekty, którymi gracz stara się zarządzać w zależności od własnych preferencji. W każdym wypadku, inwestor powinien brać pod uwagę ryzyko, które przeważnie stara się minimalizować. Przykładowo podczas zakupu akcji, gracz spodziewa się dwóch źródeł potencjalnych przychodów: jedno to wynikający z różnicy kursów zysk realizowany jedynie podczas sprzedaży waloru oraz dywidenda wypłacana przez spółkę najczęściej raz do roku, której wielkość jest proporcjonalna do ilości posiadanych aktywów.

Pierwsze w jednoznaczny sposób wskazuje na możliwości generowania przychodów dokonując operacji kupna-sprzedaży w cenach różniących się od siebie. Strategia polega na

⁷⁹ Najman K., Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie, Uniwersytet Gdański, Sopot 2003, s. 29. (praca doktorska)

⁸⁰ Według F.H. Knighta, ryzyko to niepewność, którą da się zmierzyć wyznaczając jej konkretną wartość lub jej zakres. Wyraża się ją w kategoriach szansy wystąpienia pewnego zjawiska. Błaszczuk D. J., Podstawy prognozowania, symulacji i sterowania optymalnego, PWN, Warszawa 2014, s. 21, 22.

⁸¹ Górski M., Rynkowy system finansowy, Polskie Wydawnictwo Ekonomiczne, Warszawa 2009, s. 331.

⁸² Sławiński A., Rynki finansowe, Polskie Wydawnictwo Ekonomiczne, Warszawa 2006, s. 29-34.

zakupie po jak najniższej cenie, by w przyszłości sprzedać ją, gdy wartość wzrośnie, tworząc tym samym zysk będący różnicą pomiędzy nimi. Druga możliwość jest wynikającym z posiadania akcji prawem do udziału w zyskach wypracowanych przez spółkę, rozumianym również jako nagroda za udzielenie kapitału poprzez nabycie instrumentu. Jedynie niewielki segment uczestników decyduje się na zakup licząc na przyszły rozwój spółki i możliwy przychód z racji dywidendy. Znaczna większość swoje inwestycje wiąże z zyskiem wynikającym z wahań wartości cen. W ramach tej grupy członkowie podejmują różnego rodzaju strategie by uzyskać satysfakcjonujące ich dochody. Część jako podstawę swoich decyzji uznaje codzienne fluktuacje generując w ten sposób dużo transakcji charakteryzujących się najczęściej niskimi stopami zwrotu. Inni zaś preferują osiągać większy zysk czekając aż nadarzy się okazja kilka czy nawet kilkanaście lat⁸³. Jak widać, horyzont czasowy ma znaczny wpływ na osiągnane rezultaty. Biorąc pod uwagę ten aspekt wyróżnia się trzy grupy inwestycji⁸⁴: krótkoterminowe, średnioterminowe i długoterminowe.

Inwestycjami krótkoterminowymi nazywane są transakcje, których czas trwania mieści się w przedziale od kilku godzin do maksymalnie kilku dni. Uczestników tej jak i następnej kategorii „średnioterminowych inwestycji” nazywa się spekulantami, gdyż ich decyzje opierają się w dużej mierze na podstawie bieżących informacji bądź chwilowego optymizmu panującego na rynku. Spekulacje⁸⁵, czyli transakcje przeprowadzane w ramach krótkiego jak często również średniego horyzontu czasowego nie mają nic wspólnego z analizą sytuacji finansowej spółki. W tym przypadku nieistotna jest branża, ani sama korporacja, liczy się jedynie szybkość i trafność dokonywanych operacji. Ten sektor szczególnie wrażliwy jest na zjawisko podążania za tłumem wywołane pojawieniem się nowej informacji. Krótkoterminowe transakcje w porównaniu do dwóch pozostałych grup obarczone są najwyższym poziomem ryzyka. Jednak z pewnością ich atutem jest poziom potencjalnych zysków wynikający między innymi z częstotliwości przeprowadzanych operacji.

Kolejną kategorią, jaką są inwestycje średnioterminowe, określa się kilkumiesięcznym okresem trwania umowy w ramach podjętej strategii⁸⁶. Z uwagi na stosunkowo dłuższy okres przeprowadzanych analiz przed dokonaniem zakupu, uczestnicy tej grupy, bazują nie tylko na

⁸³ Zalewski G., Inwestowanie w akcje, czyli jak skorzystać ze wzrostu gospodarczego, GPW S.A., Warszawa 2018, s. 4, 13-16, 18-19.

⁸⁴ Kachniewski M., Majewski B., Wasilewski P., Rynek kapitałowy i giełda papierów wartościowych, Fundacja Edukacji i Rynku Kapitałowego, Warszawa 2008, s. 173.

⁸⁵ Spekulacje choć kojarzą się w negatywny sposób są bardzo ważnym elementem rynku zapewniającym mu dużą płynność. W sytuacji, gdyby zabrakło transakcji podejmowanych w krótkim i średnim horyzoncie czasu, okres oczekiwania lub strata, z którymi trzeba by było się liczyć decydując się na zakup bądź sprzedaż znacznie by wzrosły. Zalewski G., Inwestowanie w akcje, czyli jak skorzystać ze wzrostu gospodarczego, GPW S.A., Warszawa 2018, s. 19.

⁸⁶ Czas ich trwania definiowany jest również przez przedział od 15 do 45 sesji. Zaremba A., Giełda. Podstawy inwestowania, Helion, Gliwice 2010, s. 194.

chwilowych trendach, ale również wspierają się powierzchownymi analizami interesujących ich spółek. Wraz z wydłużaniem się horyzontu inwestycyjnego, decyzje wspomagane są przez coraz bardziej wnikliwe badania dotyczące podmiotu w jakim lokowane są pieniądze, a także środowiska go otaczającego. Poziom ryzyka podejmowany podczas gry na giełdzie zazwyczaj maleje wraz z wydłużaniem się długości transakcji niestety również niekorzystnie wpływając na wielkość generowanych zysków.

Wszystkie transakcje, których okres trwania przekracza rok nazywane są długoterminowymi. To umowy charakteryzujące się najdłuższym horyzontem czasowym. Decyzje o ich podjęciu podejmowane są w oparciu o fundamentalną analizę⁸⁷ jak również prognozy rozwoju ekonomicznego sektorów, do których spółka należy. W ramach tej grupy dywidenda staje się istotnym elementem podczas szacowania zysków. Uczestników będących częścią omawianej kategorii nie nazywa się już spekulantami, lecz inwestorami⁸⁸.

Głównymi metodami analizy przyszłej ceny akcji są analiza techniczna⁸⁹ oraz wymieniona w kontekście długoterminowych inwestycji analiza fundamentalna⁹⁰.

Badania podstawowych czynników mających wpływ na atrakcyjność wybranej spółki w odpowiednio długim czasie, w celu oszacowania ich przyszłych wartości to druga z wymienionych wyżej typów analiz. W ramach tej strategii obserwowana jest przede wszystkim kondycja finansowa firmy wraz z wpływającym na nią otoczeniem by móc ocenić jakość odzwierciedlenia rzeczywistej i giełdowej wartości akcji na podstawie jej wewnętrznych jak i zewnętrznych parametrów. Ten typ strategii wyróżnia przede wszystkim trzy etapy: analizy makroekonomicznej⁹¹, sektorowej⁹² nazywanej również branżową oraz analizy samego przedsiębiorstwa⁹³, którym inwestor jest zainteresowany. Każde z nich można przeprowadzać

⁸⁷ Analiza fundamentalna skupia się na makro i mikroekonomicznych danych dotyczących spółki. Kachniewski M., Majewski B., Wasilewski P., Rynek kapitałowy i giełda papierów wartościowych, Fundacja Edukacji i Rynku Kapitałowego, Warszawa 2008, 146.

⁸⁸ Jagielnicki A., Inwestycje giełdowe. Jak grać i wygrywać, Helion, Gliwice 2007, s. 20.

⁸⁹ Analiza techniczna to „badanie zachowania rynku, poprzez analizę wykresów i szeregów czasowych cen, w celu przewidywania ich przyszłych ruchów”. Malkiel B. G., A Random Walk down Wall Street, W.W Norton, New York 1996, s. 1.

⁹⁰ Analiza fundamentalna ma na celu oszacowanie zdolności firmy do generowania zysków. W ramach jej działania porównuje się np. poziom ceny do jej rzeczywistej wartości. Jej etapy wyróżniają analizę makroekonomiczną, sytuacyjną spółki, sektorową jak również kondycji finansowej. Zaremba A., Giełda. Podstawy inwestowania, Helion, Gliwice 2010, s. 111-114.

⁹¹ Analiza makroekonomiczna skupia się na otoczeniu spółki. Pod uwagę brane są sytuacja gospodarcza i społeczno-polityczna danego kraju a także rozpoznanie ruchów popytu i podaży. Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo naukowe PWN, Warszawa 2010, s. 211, 212.

⁹² Analiza sektorowa bada przede wszystkim obszar, w którym firma występuje. Celem jest odnalezienie takiego podmiotu, który wewnątrz panującego ustroju i sytuacji ma największe szanse na rozwój. Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo naukowe PWN, Warszawa 2010, s. 211, 212.

⁹³ Ocena spółki określa status firmy przez pryzmat jej wyników finansowych, przeprowadzanych zazwyczaj na podstawie sprawozdań przez nią prowadzonych. Oceniane są między innymi płynność, zadłużenie oraz rentowność. Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo naukowe PWN, Warszawa 2010, s. 211, 212.

oddzielnie i w dowolnej kolejności⁹⁴. Drugą podstawową strategią budowy skutecznych przewidywań przyszłości na giełdzie jest analiza techniczna. Decyzje podejmowane w oparciu o tę koncepcję kreuje się w oparciu o wykresy, tworzone bazując na takich danych jak ceny zamknięcia czy wartości skrajne w ramach jednej jak i wielu sesji. Jednym z założeń leżących u fundamentów tej techniki jest teoria mówiąca o tym, iż bieżące ceny akcji odzwierciedlają wszystkie dostępne informacje mające na nią jakikolwiek wpływ. Teza pochodzi z poglądów dotyczących koncepcji o rynkach efektywnych⁹⁵. Kolejne kluczowe założenie mówi o tym, że sytuacje z przeszłości powtarzają się. Uzasadnia to wykorzystanie historycznych danych do oceny prawdopodobieństwa przyszłych możliwości. Jednak nie ma dwóch identycznych zdarzeń co niestety w ogranicza skuteczność tego rozwiązania. Trzecią i ostatnią podstawową teorią analizy technicznej jest przekonanie o powstawaniu trendów rozumianych jako przybliżone podążanie wartości w jednym kierunku w pewnym okresie czasu. Wyróżnia się trend wzrostowy, spadkowy i jego brak⁹⁶. Dzięki zaawansowaniu dzisiejszych technik i narzędzi komputerowych wykorzystanie analizy technicznej jest stosunkowo łatwe i często wiąże się z wysoką skutecznością. Zapewnia to wysoką popularność tej strategii⁹⁷.

W praktycznej części pracy, woluntarystycznie wybrany został do badania jakości modelu jednosesyjny i dwudziestosesyjny horyzont czasowy inwestycji. Bezpośrednio wskazuje to na spekulacyjną formę decyzji podejmowanych na podstawie szacunków SSN ze względu na rodzaj strategii, odpowiednio krótkoterminowy i średnioterminowy. Zgodnie z powyższym rozdziałem, nie będą zatem w trakcie werdyktów inwestycyjnych brane pod uwagę informacje wynikające z fundamentalnej analizy spółki. Jedynym istotnym aspektem determinującym opinię o kupnie lub sprzedaży będzie program opierający predykcje wyłącznie w oparciu o badania wykresów cen zamknięcia spółki, które jest częścią analizy technicznej.

⁹⁴ Jagielnicki A., Inwestycje giełdowe. Jak grać i wygrywać, Helion, Gliwice 2007, s. 81, 82.

⁹⁵ „Teoria efektywności rynków finansowych związana jest właśnie z tą ostatnią kategorią - efektywnością informacyjną. Zgodnie z teorią, rynki są efektywne, jeśli ceny walorów odzwierciedlają w pełni wszelkie dostępne informacje”. Biliski J., Dorożyńska A. (red.), Wybrane problemy w gospodarce światowej, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2008, s. 169.

⁹⁶ Thiel S., Rynek kapitałowy i terminowy, Komisja Nadzoru Finansowego, Warszawa 2010, s. 116.

⁹⁷ Galant M., Dolan B., Forex dla bystrzaków, Helion, Gliwice 2010, s. 194-196.

3. Narzędzia wykorzystywane w trakcie budowy modelu predykcyjnego

3.1. Język programowania Python

Python jest bezpłatnym językiem programowania⁹⁸ wysokiego poziomu⁹⁹ stworzonym w roku 1991, którego twórcą jest Guida van Rossuma. Jego nazwa związana jest paradoksalnie nie ze zwierzęciem, lecz serialem komediowym „Monty Python” emitowanym w latach 70. XX wieku. Oficjalne logo zawiera jednak grafikę przypominającą węża, jak widać na rysunku 3.1.

Rysunek 3.1. Oficjalne logo języka programowania Python



Źródło: Python Software Foundation, The Python Logo, <https://www.python.org/community/logos/>, dostęp dnia 11.05.19.

Python jest językiem interpretowanym, co oznacza, że nie kompiluje się go od razu i przechowuje się na dysku w postaci kodu źródłowego¹⁰⁰. Taki plik, posiadający standardowo rozszerzenie `.py`, dopiero podczas jego uruchomienia, wczytywany jest, przetwarzany i wykonywany przez interpreter. Dzięki takiej budowie, łatwiejsze jest pisanie programów dla różnych systemów operacyjnych, gdyż obiekt bez zmiany formatu może być przekazywany pomiędzy innymi maszynami, przetwarzając dopiero w trakcie uruchamiania pliku zgodnie z indywidualnymi potrzebami¹⁰¹. Python określany również mianem języka skryptowego, gdyż bardzo łatwo można wykorzystać go do pisania zwięzłych programów (skryptów), automatyzujących pewne zadania. Nie należy jednak dewaluować przez to jego wartości. Python w pełni wspierany przez grono programistów jest językiem implementowanym w wielu znaczących i profesjonalnych aplikacjach na całym świecie. Kolejnym jego atutem jest zdecydowanie łatwość integrowania z niskopoziomowymi językami takimi jak C, C++ czy Fortran. Dzięki tej możliwości wielu naukowców wykorzystuje język Python do łączenia ze

⁹⁸ Język programowania – „język o składni i semantyce zrozumiałej dla pewnego systemu komputerowego”. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 218.

⁹⁹ „Wysoki poziom” opisując język programowania odnosi się do poziomu abstrakcji detali wykorzystanych do jego budowy. W przeciwieństwie do niskopoziomowych, ten używa często słów codziennego użytku by był bardziej zrozumiały i prostszy w użyciu dla człowieka.

¹⁰⁰ Kod źródłowy, zazwyczaj zapisywany w pliku tekstowym, to komendy programu komputerowego przy użyciu wybranego języka programowania. Programista, w sposób zrozumiały dla siebie komunikuje się z komputerem, wydając mu operacje, które powinien wykonać. Kod źródłowy kompilowany jest do kodu maszynowego, który dalej wykonuje komputer przy pomocy procesora. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 210, 212.

¹⁰¹ Dawson M., Python dla każdego. Podstawy programowania, Helion, Gliwice 2014, s. 16-19.

sobą jako spoiwo części kodu napisanych w różnych językach opracowywanych na przestrzeni wielu dekad. Struktura kodu zorientowana jest na czytelność i klarowność. Na rysunku 3.2. dla porównania przedstawiono kod napisany w trzech różnych językach: C, Python i Asembler¹⁰². Efektem końcowym każdego z zaprezentowanych programów jest wypisanie prostego powitania „Hello World”¹⁰³.

Rysunek 3.2. Porównanie struktury wybranych języków programowania

C	Python	Asembler
<pre>void main(void) { printf("Hello world!"); }</pre>	<pre>print("Hello World!")</pre>	<pre>section .data msg db "Hello world!",0xa len equ \$ - msg section .text global _start _start: mov edx,len mov ecx,msg mov ebx,1 mov eax,4 int 0x80</pre>

Źródło: Opracowanie własne

Językowi programowania Python nie brakuje również wad, a jedną z nich, jest niewątpliwie poziom optymalizacji, który przekłada się bezpośrednio na czas, potrzebny do wykonywania kodu. Dla przykładu program egzekwowany z wysoką częstotliwością oparty o wielkie zbiory danych znacznie szybciej będzie wykonywany w języku C++, któremu zdecydowanie bliżej jest do miana najszybszego.

Dzięki przejrzystej budowie i względnie łatwej do zrozumienia konstrukcji Python szybko zyskuje na popularności. Chętnie wybierany przez społeczności naukowe do tworzenia obliczeń oraz analizy danych. Odnalazł uznanie wśród wielu entuzjastów, którzy bardzo chętnie używają go przy projektach powiązanych ze sztuczną inteligencją. Jego możliwości wizualizacji i wykonywania szybkich operacji na bardzo złożonych danych można bez problemu porównywać z wieloma innymi komercyjnymi narzędziami bezpośrednio do tego dedykowanymi jak np. język R¹⁰⁴. Według trzech źródeł internetowych¹⁰⁵ Python znajduje się

¹⁰² Asembler to niskopoziomowy język programowania. Został podany w przykładzie by pokazać kontrast pomiędzy strukturami języków wyższych poziomów. Reprezentowany przez kod, który jest najbardziej zbliżony do źródłowego, bezpośrednio przekazywanego do procesora. Jedno polecenie odpowiada jednemu rozkazowi.

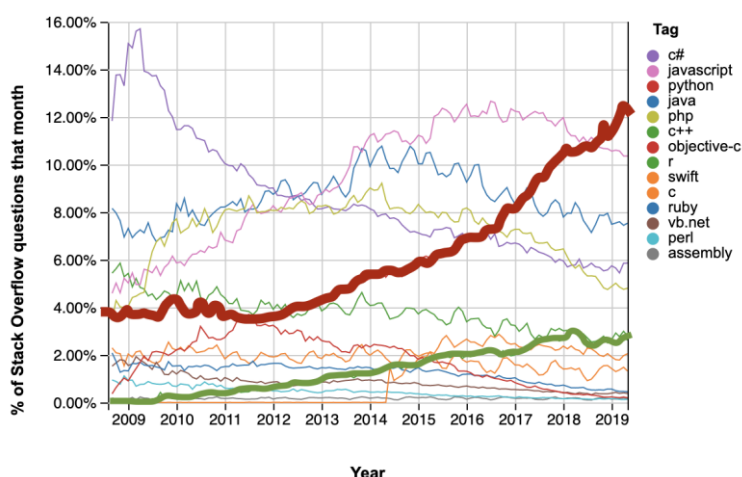
¹⁰³ „Hello World” to nazwa programu, który ma na celu jedynie wyświetlić komunikat „Hello World”. Jest to tradycyjnie pierwszy program dla osób stawiających swoje pierwsze kroki ucząc się programowania. Celem zadania jest wprowadzenie i demonstracja wybranego języka.

¹⁰⁴ Bardzo popularny stał się podział świata analityków na tych, którzy w codziennej pracy wykorzystują język R albo Python. Jednak najczęściej profesjonalści, są hybrydowi łącząc oba z nich w zależności od potrzeb.

¹⁰⁵ Saeed A., Here are the ten best programming languages to learn in 2019, codinginfinite.com/best-programming-languages-to-learn-2019/, dostęp dnia 02.06.19, Cuthbert A., 8 top programming languages & frameworks of 2019, hackernoon.com/8-top-programming-languages-frameworks-of-2019-2f08d2d21a1, dostęp dnia 02.06.19, Dwivedi V., Top 10 programming languages of the world – 2019 to begin with..., www.geeksforgeeks.org/top-10-programming-languages-of-the-world-2019-to-begin-with/, dostęp dnia 02.06.19.

w ścisłej czołówce najczęściej używanych języków programowania z czego jedno z nich plasuje go na miejscu pierwszym. Znaczący wzrost popularności w ostatnich latach jest bezpośrednio związany z rozkwitem dziedzin związanych z kategorią „data science”¹⁰⁶ czyli między innymi omawiane w tej pracy sztuczna inteligencja, uczenie maszynowe i głębokie uczenie. Poniższy wykres liniowy na rysunku 3.3 prezentuje liczbę zapytań dotyczących różnych języków programowania w latach 2009 do 2019 na stronie „stackoverflow.com”.

Rysunek 3.3. Popularność wybranych języków programowania liczona jako ilość zapytań na stronie Stackoverflow.com



Źródło: Stack Exchange Inc, Most Popular Languages (TIOBE Index for May 2017), <https://insights.stackoverflow.com/trends?tags=java%2Cc%2B%2Cpython%2Cc%23%2Cvb.net%2Cjavascript%2Cassembly%2Cphp%2Cperl%2Cruby%2Cvb%2Cswift%2Cr%2Cobjective-c>, dostęp dnia 01.07.19.

Na wykresie zaznaczono grubą linią zmiany dla języka Python oraz dla porównania, w kontekście walki o fotel lidera języków do analizy i przetwarzania danych, nieco cieńszą linią dane dla języka R. Stackoverflow to nazwa strony internetowej do dzielenia się wiedzą, uznawana przez deweloperów za pewnego rodzaju święty graal, który zawiera odpowiedź na każde pytanie z zakresu programowania¹⁰⁷.

¹⁰⁶ Dziedzina zajmująca się danymi, ich analizowaniem, przetwarzaniem i wizualizacją. Data scientist to osoba działająca w jej obrębie. Termin zyskuje swoją popularność od roku 2012 za sprawą Harvard University. Data Scientist określa osobę o kompetencjach interdyscyplinarnych posiadającą takie umiejętności jak swobodna interakcja i poruszanie się wśród danych, rozumienie biznesu. Dodatkowo związany jest z technologią i potrafi szybko dostosować się do aktualnych trendów, posiadając również kompetencje z dziedzin informatyki, statystyki, matematyki. Jego zadaniem jest dokonywanie złożonych analiz. Powinien posiadać również szereg miękkich umiejętności jak np. zdolność przeprowadzania prezentacji i tworzenia czytelnych wykresów. Dryjańska P., Data Science, https://mfiles.pl/pl/index.php/Data_science, dostęp dnia 20.06.19.

¹⁰⁷ Stronę tworzy ogromna społeczność użytkowników, którzy mogą zadawać tam pytania jak i na nie odpowiadać. Poprzez aktywną pomoc otrzymuje się punkty reputacji, dzięki czemu znana jest informacja na temat użytkownika, jego aktywności i skuteczności odpowiedzi. Strona istnieje od 2008 roku i posiada ponad 50 milionów odwiedzających miesięcznie, wraz ze średnio około 51 tysięcy deweloperów online w każdej chwili. Stack Exchange Inc, About, <https://stackoverflow.com/company>, dostęp dnia 12.05.19.

3.2. Podstawowe biblioteki do uczenia maszynowego

Python posiada całą gamę bibliotek przeznaczonych do różnych zastosowań takich jak tworzenie stron internetowych¹⁰⁸, zarządzanie graficznym interfejsem użytkownika (ang. *Graphical User Interface*, GUI¹⁰⁹), administracja systemowa, zarządzanie oprogramowaniem oraz do celów naukowych i numerycznych. Z racji tematyki uczenia maszynowego, która jest przedmiotem tej pracy, opisane zostaną jedynie biblioteki powiązane z analizą i przetwarzaniem danych. Na oficjalnej stronie języka Python jako podstawowe narzędzia z kategorii przeznaczenia „*Scientific and Numeric*” czyli naukowych i numerycznych wymienia się SciPy czy Pandas. Dodatkowo należy podkreślić również inne bardzo ważne biblioteki, które są podstawowym narzędziem pracy podczas pracy w trakcie uczenia maszynowego, jakimi są Scikit-learn, NumPy i Matplotlib.

Oprócz wymienionych dotychczas narzędzi z uwagi na wykorzystanie sztucznych sieci neuronowych w rozdziale czwartym użyta zostanie również inna biblioteka jak Keras, która w ramach Pythona domyślnie współpracuje z wspomagając się modulem TensorFlow.

Biblioteka z perspektywy języków programowania jest plikiem, dostarczającym funkcje, klasy, dane oraz ich typy, które można implementować z poziomu kodu źródłowego, czyli w trakcie pisania instrukcji przez programistę. Wyróżnia się dwa podstawowe ich rodzaje: standardowo wbudowane (ang. *standard*, *build-in*) a także takie, które należy dodatkowo instalować. Omawiany język programowania zapewnia całą listę wbudowanych bibliotek tworzących jego rdzeń. Standardowy pakiet posiada cały szereg modułów napisanych na przykład w języku C dając szeroki dostęp do wielu funkcjonalności.¹¹⁰ Wszystkie inne wymienione w pracy należy dodatkowo zainstalować by później poprzez wykonanie kilku linii kodu cieszyć się ich możliwościami.

Z uwagi na liczebność dodatków w języku Python, szczegółowo opisanych będzie jedynie kilka z wymienionych, skupiając się na tych, które odnalazły zastosowanie w praktycznej części tej pracy. Przedstawione zostaną następujące pozycje: NumPy, Pandas, Matplotlib, Scikit-learn, Keras.

Pakiet NumPy jest podstawowym narzędziem wykorzystywanym do przeprowadzania operacji numerycznych, a w szczególności operacji na macierzach. Obsługuje on wiele algorytmów oraz struktur danych niezbędnych przy przetwarzaniu danych numerycznych. Jej

¹⁰⁸ Strona internetowa to „najpowszechniejsza forma zasobów sieciowych prezentowana w hipertekście. Każda jej strona (ang. *site*) zawiera zazwyczaj jeden monotematyczny zbiór informacji w postaci dokumentu, mogącego zawierać informacje w każdej formie multimedialnej”. Gogołek W, Informatyka dla humanistów, Wydawnictwo Kropki Trzy, Warszawa 2012, s. 272.

¹⁰⁹ Interfejs graficzny, jest systemem pozwalającym na wizualną interakcję maszyny z użytkownikiem. Gogołek W, Informatyka dla humanistów, Wydawnictwo Kropki Trzy, Warszawa 2012, s. 171.

¹¹⁰ Python Software Foundation, The Python Standard Library, <https://docs.python.org/3/library/>, dostęp dnia 18.05.19.

głównym obiektem jest *ndarray* reprezentujący n-wymiarową tablicę. Moduł charakteryzuje się wysoką wydajnością, szeroką gamą funkcji matematycznych oraz narzędzi do przeprowadzania działań na macierzach. Pozwala on również na łączenie elementów kodu z takich języków jak na przykład C. Swoją popularność zawdzięcza dzięki wysoce efektywnym obiektowi *ndarray*, będący nośnikiem danych, na którym przeprowadzanie operacji przebiega znacznie szybciej od tych na podstawowych obiektach Pythona. Niskopoziomowe języki programowania znakomicie radzą sobie z odczytem obiektów NumPy dzięki czemu często wykorzystuje się go do komunikacji między aplikacjami¹¹¹.

Pandas zapewnia struktury i funkcje dedykowane do pracy z ustrukturyzowanymi danymi jak tabele, listy i serie. Nazwa jest pochodną słów „dane panelowe” (ang. *panel data*) określających n-wymiarowe ustrukturyzowane zbiory informacji. Zbudowana została w 2010 roku na wcześniej omawianej bibliotece NumPy. Jej podstawowymi elementami są obiekt *DataFrame*, będący reprezentacją tabel, złożonych z wierszy i kolumn opisanych etykietami oraz *Series*, który jest jednowymiarową tablicą często wykorzystywaną do zapisu informacji rozproszonych w czasie. Pandas łączy wysoką wydajność obiektu *ndarray* biblioteki, która jest jej fundamentem oraz ogromne możliwości manipulowania i przetwarzania danych, reprezentowanych za pomocą tabel.

Następnym szeroko wykorzystywanym pakietem jest Matplotlib, zajmujący pierwsze miejsce pod względem popularności bibliotek do wizualizacji danych. Jest to przeznaczony do tworzenia wysokiej jakości, interaktywny i kompatybilny z wieloma środowiskami moduł dla języka Python. Posiada ogromną paletę możliwości przy konstrukcji wszelkiego rodzaju wykresów. Nowsze jego edycje wspierają już tylko najbardziej aktualny rodzaj wersji języka.

Z perspektywy uczenia maszynowego, projekt Scikit-learn jest jednym z najważniejszych narzędzi dostępnym dla języka programowania Python. Korzystanie z programu nie wymaga żadnych opłat oraz daje możliwość pełnego dostępu do kodu źródłowego z możliwością jego edycji. Jest on komercyjnym rozwiązaniem zbudowanym jak wiele innych w oparciu o bibliotekę NumPy. Dzięki zaangażowaniu blisko 1500 osób, stał się podstawą projektów wykorzystujących uczenie maszynowe ze względu na zaawansowane możliwości jakie zapewnia. Tworzą go zaawansowane typy modeli przeznaczonych do takich zadań jak klasyfikacja, analiza skupień, regresja oraz do wstępnego przetwarzania danych. Scikit-learn znacząco wpłynął na wzrost produktywności języka Python¹¹².

¹¹¹ McKinney W., Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i Numpy oraz środowiska Ipython, Helion, Gliwice 2018, s. 14–18.

¹¹² McKinney W., Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i Numpy oraz środowiska Ipython, Helion, Gliwice 2018, s. 17-21.

Ostatnim poruszonym jest pakiet Keras, który jest narzędziem *open-source* języka Python umożliwiającą łatwą budowę i trenowanie dowolnie wybranych modeli wykorzystywanych do uczenia głębokiego. Pierwotnie był on niezależną biblioteką, lecz w roku 2017 team TensorFlow Google postanowił go wspierać, przez co aktualnie rozpatrywany jest jako interfejs większej biblioteki jaką jest właśnie TensorFlow. Celem stworzenia pakietu Keras było umożliwienie szybkiego przeprowadzania eksperymentów na sieciach neuronowych. Jednymi z podstawowych atutów omawianego pakietu są z pewnością przyjazny interfejs programistyczny zapewniający szybkie prototypowanie modeli, obsługa SSN o dowolnej architekturze oraz możliwość uruchamiania tego samego kodu na procesorach zarówno CPU (ang. *Central Processing Unit*), GPU (ang. *Graphics Processing Unit*¹¹³) jak i TPU (ang. *Tensor Processing Unit*¹¹⁴). Keras umożliwia tworzenie sekwencyjnych modeli sieci neuronowych, co znajdzie zastosowanie w projekcie. Wykorzystany zostanie najbardziej podstawowy typ warstwy „dense layer”, ze względu na wykorzystanie łączenia neuronów na zasadzie „każdy z każdym”¹¹⁵.

Rysunek 3.4 Zbiór grafik reprezentujących wybrane biblioteki języka programowania Python



Źródło: Opracowanie własne na podstawie: Harasymczuk M., Biblioteki w zastosowaniach naukowych i inżynierskich, <http://python.astrotech.io/numerical-analysis/libraries.html>, dostęp dnia 05.04.19.

Przedstawiony wyżej rysunek 3.4. przedstawia logo każdej wymienionej w tym rozdziale biblioteki. Wszystkie wymienione narzędzia posiadają rozbudowaną dokumentację opisującą sposób zastosowania i możliwości wszystkich dostępnych wariantów do użycia. Dzięki

¹¹³ Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s. 122.

¹¹⁴ Procesor istnieje od roku 2016 i został stworzony przez firmę Google specjalnie do przeprowadzania operacji na sieciach neuronowych uczenia maszynowego

¹¹⁵ Chollet F., Deep learning. Praca z językiem Python i biblioteką Keras, Helion, Gliwice 2018, s. 75-77.

wymienionym modułom Python stał się silnym narzędziem do analizy wykorzystywanym przez zarówno profesjonalistów jak i adeptów programowania.

Wyróżnia się różne sposoby instalacji dodatkowych pakietów. Jedną z nich jest standardowa ścieżka, poprzez pobranie instalatora ze strony głównej producenta. W tym wypadku należy odwiedzić witrynę internetową związaną z biblioteką, zapewniającą pobieranie pakietu, a następnie rozpocząć proces instalacyjny poprzez odpowiednio uruchomienie pobranych obiektów i wykonanie wyświetlonych w trakcie ich działania instrukcji.

Inną metodą jest nieco bardziej zaawansowana ścieżka, wykorzystująca użycie aplikacji systemowych oraz pakietu instalacyjnego języka Python, zapewnionego w każdej jego wersji powyżej 2.7.9. oraz 3.4.0.¹¹⁶ pobranej z jego głównej strony. Proces instalacji rozpoczyna się od uruchomienia wiersza poleceń (ang. *command line*)¹¹⁷ a następnie wpisania kolejnych linijek, zatwierdzając wybór klawiszem „Enter”.

By zainstalować wybrany pakiet w wierszu poleceń, należy skorzystać z podanej formuły:

`pip install nazwa_pakietu`

Gdy biblioteka znajduje się na komputerze można ją z łatwością aktualizować przy użyciu komendy:

`pip install --upgrade nazwa_pakietu`¹¹⁸

Aby skorzystać z jakiegokolwiek modułu, wymagane jest przeprowadzenie kilku dodatkowych kroków już z poziomu środowiska, w którym używa się Pythona. Omawiany język programowania zapewnia bardzo prostą metodę użytkowania dowolnych pakietów dzięki konwencjom, tworzących jego całość. Jeżeli podczas programowania, pojawi się potrzeba użycia funkcjonalności, którą zapewnia jedna z bibliotek należy ją zaimportować by później zgodnie z nazewnictwem zaproponowanym przez producenta wykorzystywać jej opcje.

Poniższy wiersz przedstawia standardową formułę łączenia się z biblioteką „nazwa_biblioteki” w celu dalszego użytkowania jej podzespołów:

¹¹⁶ Wyróżnia się dwie wersje Pythona: 2.x oraz 3.x, które nie są ze sobą kompatybilne. Druga wymieniona jest nowsza i to właśnie z niej będę korzystał podczas budowy projektu. Oba warianty nieco różnią się między sobą, a ich wybór zależy od upodobań programisty, natomiast wersję 3.x uznaje się za wiodącą i domyślnie proponowaną.

¹¹⁷ W zależności od używanego systemu operacyjnego nazwa aplikacji wiersza poleceń będzie się różniła. Przykładowo w systemie operacyjnym Windows nazywa się „cmd.exe” a w systemie MacOS nosi nazwę „Terminal”.

¹¹⁸ Raschka S., Python. Uczenie maszynowe, Helion, Gliwice 2018, s. 36-37.

import nazwa_biblioteki as nb

Działanie powyższego zwrotu zapewnia sposobność wykorzystywania elementów modułu w kolejnych liniach kodu występujących po przeprowadzeniu importu. Znakomitą przywilejem jaki zapewnia Python jest możliwość przypisania komponentu „nazwa_biblioteki” do zmiennej „nb”, dzięki czemu skorzystanie z obiektów staje się znacznie prostsze. By użyć konkretnych elementów wybranej biblioteki wystarczy wpisać „nb.nazwa_elementu” co zapewnia do niego dostęp bez ograniczeń. Pamiętać należy, by korzystać z podkategorii modułów zgodnie z ich przeznaczeniem opisanym w dokumentacji.

3.3. Miary do oceny wyników i ich porównywania

Celem modelu sztucznej sieci neuronowej budowanej w rozdziale czwartym tej pracy jest tworzenie jak najlepszych prognoz cen zamknięcia, czyli kursu ostatniej transakcji w trakcie trwania sesji giełdowej. Idealną sytuacją byłoby, gdyby prognozy algorytmu w żadnym stopniu nie różniły się od rzeczywistych przyszłych wartości. Jednak szanse zaistnienia takiej sytuacji są bardzo niskie, w związku z czym tworząc jakąkolwiek prognozę należy liczyć się z możliwością wystąpienia błędów. Z uwagi na typ szacowanych wartości, jest on rozumiany jako różnica występująca między wartością teoretyczną, zapewnioną przez model, a rzeczywistą wartością empiryczną. Zatem miara, od której bezpośrednio zależy skuteczność algorytmu ująć można następującym równaniem:

$$e_t = \hat{y}_t - y_t$$

gdzie:

e_t - szacowany błąd (ang. error) w okresie t ,

\hat{y}_t - oczekiwana wielkością y , w okresie t ,

y_t - rzeczywista wielkość wystąpienia, w tym samym okresie t .

Z powyższego wzoru wynika, że jakość prognozy jest odwrotnie proporcjonalna do błędu, który z niej wynika.

Same błędy spowodowane być mogą wieloma czynnikami, determinowanymi między innymi niewłaściwym doбором danych, budową samej sieci¹¹⁹ czy zmianą warunków wpływających na analizowane wartości¹²⁰. Ocena siły i kierunku wpływu wszystkich czynników zniekształcających percepcję modelu jest skomplikowanym procesem,

¹¹⁹ Proces doboru parametrów podczas budowy sieci opisany zostanie w rozdziale czwartym.

¹²⁰ W szczególności na rynku kapitałowym akcji czynniki oddziałujące na proces kreowania się cen mogą być nieprzewidywalne.

wykraczającym poza zakres tej pracy. Porównane jednak będą między sobą modele o różnej budowie, by spośród wybranych wyłonić ten, który najlepiej rokuje.

Pojedyncza wielkość wartości błędu, nie pozwala trafnie ocenić generalnej skuteczności przebiegu modelu. Krytyka algorytmu oparta na badaniu jednej pomyłki byłaby nierozsądna, gdyż daje to jedynie bardzo wąski pogląd na jakość jego przebiegu w ogóle. W związku z tym, do kreowania osądu stosuje się wskaźniki, które biorą pod uwagę więcej niż jeden błąd najczęściej sumując je w pewien sposób, uśredniając je następnie. Jedną z podstawowych miar uwzględniającą szereg różnic jest średni absolutny błąd prognozy¹²¹ (ang. *Mean Absolute Error*, MAE), którego równanie wygląda następująco:

$$MAE = \frac{\sum_{t=1}^n |\hat{y}_t - y_t|}{n} = \frac{\sum_{t=1}^n |e_t|}{n}$$

gdzie:

n – liczba wszystkich błędów branych pod uwagę.

Jest to bezwzględna miara określająca przeciętne wielkości różnic między szacunkami a ich prawdziwymi wartościami. W tym wypadku nie są brane pod uwagę kierunki błędów a jedynie ich rozmiary, przez co nie informuje ona o występowaniu tendencji do zaniżania czy zawyżania prognoz¹²².

By zyskać pojęcie na temat zwrotów pomyłek skorzystać można z innej metody jaką jest średni błąd prognozy (ang. *Mean Error*, ME). Przedstawia się go w następujący sposób:

$$ME = \frac{\sum_{t=1}^n (\hat{y}_t - y_t)}{n}$$

Poprzez zestawienie wzorów na ME z MAE, można zauważyć różnicę zastosowania wartości bezwzględnej w liczniku równania. Skutkiem użycia zwykłych nawiasów jest możliwość generowania ujemnych bądź dodatnich rezultatów przez ME mówiących odpowiednio o generalnym niedoszacowaniu lub przeszacowaniu prognoz. Głównym jednak problemem tej metody jest jej zerowa wartość w momencie, gdy błędy są symetryczne. Oznacza to, że $ME = 0$ niekoniecznie wskazuje na brak jakichkolwiek pomyłek.

¹²¹Błąd względny jest postaci bezwymiarowej, określony przy pomocy stosunku różnicy pomiędzy wartością prognozy a jej rzeczywistym odpowiednikiem do odpowiadającej, tej samej wartości rzeczywistej. Najczęściej wyrażany w procentach. Zawada J., Wybrane zagadnienia z podstaw metrologii, Wydawnictwo Politechniki Łódzkiej, Łódź 2002, s. 32.

¹²² Kelleher J. D., Namee B. M., D'Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 444.

Badając rezultaty modeli istotna jest wiedza o skali rozbieżności między prognozowanymi, a empirycznymi argumentami występującymi w zbiorze. Inwestując gracz z pewnością nie chce popełnić ponadprzeciętnych błędów¹²³. Zatem pożądana jest miara inaczej traktująca małe i duże odchylenia. W celu uzyskania informacji o rozkładzie stosuje się średni błąd kwadratowy (ang. *Mean Squared Error*, MSE), którego równanie wygląda jak niżej:

$$MSE = \frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}$$

Wykorzystanie pierwiastka kwadratowego skutkuje nałożeniem kar na duże wartości. Im rozmiar błędu jest większy tym większy wpływ ma na wynik opisywanej metody. Niestety przez wykorzystanie potęgi jednocześnie wynik przestaje być interpretowalny, ponieważ jednostki, w których mierzone są obserwacje tracą sensowność¹²⁴.

Metodą posiadającą pozytywne cechy MSE, jednocześnie redukująca niuans związany z jego interpretacją, stając się jedną z najważniejszych wskaźników określających miarę jakości prognoz jest pierwiastek średniego błędu kwadratowego (ang. *Root Mean Squared Error*, RMSE). Jego wzór wygląda następująco:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}$$

RMSE informuje jak przeciętnie model myli się przy każdej prognozie. W przypadku obydwu powyższych wzorów MSE oraz RMSE ztraca się informację o kierunku błędów jedynie skupiając się na ich rozmiarach¹²⁵.

Wyniki powyższych metod zyskują dodatkową wartość, gdy istnieje możliwość porównania ich z innymi miarami, jak na przykład średnia i odchylenie standardowe. Ich wartości mogą być punktami odniesienia do oceny modelu na podstawie wymienionych wcześniej równań, zyskując jednocześnie większą świadomością rozproszenia danych¹²⁶.

Średnią (ang. *average*, *mean*) liczy się w następujący sposób:

¹²³ Ponadprzeciętne błędy, inaczej nazywane grubymi czy nadmiernymi to „błędy wynikające z nieprawidłowego wykonania pomiaru. Wynik obciążony błędem grubym jest niewiarygodny i nie powinien być brany pod uwagę. Niestety, wykrycie błędu grubego umożliwiają najczęściej tylko pomiary wielokrotne”. Zawada J., Wybrane zagadnienia z podstaw metrologii, Wydawnictwo Politechniki Łódzkiej, Łódź 2002, s. 34.

¹²⁴ Na przykład przy wyznaczaniu jakości szacunków cen zamknięcia w \$ przy użyciu MSE jego wynik wyrażany jest w \$² (dolar podniesiony do kwadratu) przez co ztraca logiczne znaczenie.

¹²⁵ Kelleher J. D., Namee B. M., D’Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 442-444.

¹²⁶ Najman K., Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie, Uniwersytet Gdański, Sopot 2003, s. 196-203. (praca doktorska)

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

gdzie:

x_i – kolejne wartości argumentów,

n – ilość wszystkich argumentów wziętych pod uwagę.

Określa ona przeciętną wartość w analizowanym zbiorze.

Błąd standardowy (ang. *standard error*) to wskaźnik, który wyznacza jak przeciętnie różnią się obserwacje od ich średniej, informując o ich zmienności. Jego równanie wygląda jak niżej:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

Budowa jest analogiczna do RMSE jednak skupia się ona na rozproszeniu cechy a nie błędów jej szacunku¹²⁷. Obszar, który powstaje wokół średniej o szerokości dwóch błędów standardowych nazywany jest typowym obszarem zmienności:

$$\bar{x} - \sigma \leq x_{typ} \leq \bar{x} + \sigma$$

Powyższy przedział określa typową zmienność argumentów x ¹²⁸.

Do uzyskania względnej wielkości błędów wykorzystany zostanie również średni względny błąd procentowy (ang. *Mean Absolute Percentage Error*, MAPE):

$$MAPE = \frac{\sum_{t=1}^n \frac{|\hat{y}_t - y_t|}{y_t} * 100}{n}$$

Powyższa miara określa przeciętny stosunek popełnionych błędów przy wyznaczaniu prognozy w stosunku do odpowiadających im rzeczywistych wartości. MAPE wyraża się w procentach¹²⁹.

Oceniając przydatność użycia sieci, warto porównać jej wyniki z innymi algorytmami do kreowania szacunków dotyczących przyszłości. W rozdziale praktycznym obok sieci neuronowych zbudowane zostaną również prognozy naiwne oraz prognozy metody średnich

¹²⁷ Maddala G. S., *Ekonometria*, Wydawnictwo Naukowe PWN, Warszawa 2008, s. 57-58.

¹²⁸ Pułaska-Turyna B., *Statystyka dla ekonomistów*, Difin, Warszawa 2005, s. 71-78.

¹²⁹ Błaszczuk D. J., *Podstawy prognozowania, symulacji i sterowania optymalnego*, PWN, Warszawa 2014, s. 34-35.

kroczących. Dzięki zastosowaniu tej strategii analityk bliżej będzie odpowiedzi czy uzasadnione w przewidywaniu giełdowych cen zamknięcia firmy Apple jest użycie tak skomplikowanych narzędzi jak sztuczne sieci neuronowe.

Istnieją różne typy prognoz naiwnych zależnych od takich składników jak trend czy sezonowość. Ponieważ często spotkać się można z występowaniem trendu w cenach akcji, przy jednoczesnym braku sezonowości, zastosowana będzie bezwzględna prognoza naiwna¹³⁰ uwzględniająca jedynie tendencję w zbiorze, której równanie przedstawiono poniżej:

$$\hat{y}_{t+1} = y_t + (y_t - y_{t-1})$$

Ze wzoru wynika, że jutrzejsza prognoza zmieni się w stosunku do dzisiejszej wartości o tyle samo co zmieniła się wartość dzisiejsza w stosunku do wczorajszej. Jak można wnioskować ten sposób bierze pod uwagę kierunek zmian analizowanej wartości¹³¹.

Również zastosowana będzie metoda wykorzystująca prostą średnią kroczącą (ang. *Simple Moving Average*, SMA), której działanie oparte jest o ideę, iż prognozę ceny jutrzejszej wyznacza średnia arytmetyczna wartość ostatnich n sesji. Równanie SMA wygląda następująco¹³²:

$$SMA = \frac{\sum_{t=-1}^n x_t}{n}$$

Do obliczeń zawsze bierze się kursy poprzedzających n sesji.

Każda z wyżej wymienionych metod, zarówno stosowana do oceny jakości jak również do wyznaczania samych prognoz niesie w sobie inną informację. By skuteczność ich stosowania była satysfakcjonująca w trakcie przeprowadzania diagnozy wykorzystane zostaną w następujący sposób:

- najpierw wraz z prognozami zapewnionymi przez sieci neuronowe, powstaną również dwie inne predykcje przez wykorzystanie dwóch wspomnianych sposobów: prostej średniej kroczącej (dla wszystkich strategii inwestycyjnych) oraz naiwnej prognozy (jedynie dla strategii krótkookresowej),
- następnie wyznaczone dla wszystkich szacunków zostaną ME, MAE, RMSE oraz MAPE.
- na koniec dla najlepszego modelu przeprowadzona zostanie prosta symulacja gry giełdowej. Założenia zastosowane w jej trakcie zostaną opisane w rozdziale czwartym.

¹³⁰ Prognoza metodą naiwną zakłada, że obserwacje będą zmieniały się w ten sam sposób jak w ostatnim znanym okresie. Salmonowicz H. (red.), Koniunktura w gospodarce światowej a rynki żeglugowe i portowe, Kreos, Szczecin 2009, s. 244.

¹³¹ Prognozowanie naiwne, <http://www.prognozowanie.info/prognozowanie-naiwne/>, dostęp dnia 07.06.19.

¹³² Jagielnicki A., Inwestycje giełdowe. Jak grać i wygrywać, Helion, Gliwice 2007, s. 91, 92.

4. Wykorzystanie SSN do szacowania przyszłej wartości akcji firmy Apple

4.1. Etapy budowy modelu uczenia maszynowego

W każdym badaniu można wyróżnić kolejne etapy, których znaczenie z perspektywy wyznaczonego celu jest kluczowe. Tak jak wiele innych przedsięwzięć naukowych, budowa programów wykorzystujących uczenie maszynowe podzielona jest na etapy, którymi twórca programu powinien podążać. Formalną nazwą określającą wszystkie te fazy, od rozpoczęcia aż do zakończenia jest cykl życia projektu (ang. *project life cycle*¹³³). Celem ich zastosowania jest ułatwienie pracy poprzez wytyczenie jasnej ścieżki. Takie podejście stwarza możliwość podziału pracy na partie oraz wykluczają chaos co przekłada się na zwiększenie jakości osiąganych efektów. Wyróżnia się sześć głównych kroków w trakcie pracy nad stworzeniem oraz wprowadzeniem odpowiedniego algorytmu opartym o uczenie maszynowe¹³⁴:

- zrozumienie problemu,
- pobranie danych,
- przygotowanie danych,
- modelowanie,
- ocena jakości,
- zastosowanie modelu.

Zrozumienie problemu to pierwszy krok na ścieżce do stworzenia dobrego algorytmu. Prawdopodobnie jest to główny proces, będący początkiem każdego większego przedsięwzięcia. To właśnie w tym momencie indywidualny twórca lub cały zespół kreuje pytania, stawia hipotezy i wyznacza horyzont zdarzeń. Często to zadanie wspomagane jest przez specjalistów z dziedziny, w obrębie której znajduje się problem. Zasadniczo etap ten można podzielić na dwa, podkreślając w ten sposób istotność każdego z zadań do wykonania.

Pierwszy z nich to zrozumienie samej kwestii, której rozwiązania się oczekuje, a drugi, to dokładne zapoznanie się z samymi danymi, uznanymi za istotne w oparciu o uprzednią analizę zagadnienia. Czasami zdarza się, że brakuje informacji, będących kluczem do rozwiązania. W takim wypadku celem tego zadania jest również zdecydowanie o tym w jaki sposób, gdzie oraz jak długo powinno się je zbierać by możliwe było udzielenie satysfakcjonującej odpowiedzi na postawione pytania¹³⁵.

Gdy już problem został zrozumiany i stworzono plan jego rozwiązania, przechodzi się do kolejnego etapu, jakim jest pobranie danych. Ich cenne źródła wyliczać można bez końca.

¹³³ Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019, s 400-401.

¹³⁴ Kelleher J. D., Namee B. M., D’Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 13-15.

¹³⁵ Kelleher J. D., Namee B. M., D’Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 21-25.

Wiele instytucji takich jak urzędy statystyczne, banki państwowe czy prywatne firmy udostępniają zapisane rekordy publicznie w Internecie. Czasami wartości, które inwestor chce pozyskać, można otrzymać jedynie w zamian za ustalonej wielkości opłatę. Przykładem na to są aktualne wielkości indeksów i cen udostępniane przez giełdy. By pozyskać najświeższe informacje ich dotyczące, czyli takie, których opóźnienie w dostawie nie przekracza 15 minut użytkownik jest zobligowany do wykupienia abonamentu¹³⁶. Poza „danymi czasu rzeczywistego”¹³⁷, reszta najczęściej dostępna jest za darmo. Sytuacja jest bezpośrednim przykładem wartości jednego z podstawowych atrybutów informacji, ich aktualności, uzyskanie, którego nierzadko łączy się z opłatami. To jak trudno jest skompletować wszystkie potrzebne wartości zależy od poziomu skomplikowania zagadnienia¹³⁸.

Będąc w posiadaniu pożądaných z perspektywy projektu plików należy je dostroić do wymogów determinowanych przez model. W rzeczywistości rzadko kiedy uzyskane rekordy spełniają wymogi dla osiągnięcia optymalnej skuteczności algorytmu. Zatem celem następnego etapu jest przygotowywanie danych (ang. *data processing, data preparation*). Ta część projektu stwarza wiele możliwości. Z uwagi na dziesiątki dostępnych metod, w tej pracy wymienione zostaną jedynie wybrane przykłady.

Jedną z pierwszych czynności, jaką należy wykonać jest wyznaczenie interesującej części informacji ze wszystkich uzyskanych, weryfikując jednocześnie ich kompletność. W tym momencie analityk decyduje o uzupełnianiu lub usuwaniu obserwacji wybrakowanych. Zadaje sobie pytania dotyczące rozkładu cech, ich ilości sprawdzając również obecność wartości skrajnych. Przy decyzji o uzupełnieniu pól, wybrana musi zostać metoda egzekwowania nowych miar. Samych sposobów uzupełniania jest wiele i różnią się zastosowaniem. Przykładowo jednym z prostych rozwiązań jest wstawienie w puste pola wartości średniej wszystkich obserwacji. Jeśli błędnie skompletowanych obiektów będzie tylko niewielka część a ich usunięcie nie wpłynie negatywnie na poziom informacji jaki ze sobą niosą, prawdopodobnie programista skłonny będzie się ich pozbyć. Trzeba jednak wiedzieć o tym, że

¹³⁶ Dla giełdy papierów wartościowych w Warszawie, opłata roczna za dane czasu rzeczywistego dla wszystkich klas aktywów oraz indeksów, w celu wykorzystania do handlu automatycznego wynosi 65 tys. zł (cennik obowiązujący od dnia 01.01.2019 r.). Dane rynkowe, gpw.pl/dane-rynkowe, dostęp dnia 01.07.19.

¹³⁷ Termin używany na stronie głównej GPW w Warszawie gpw.pl

¹³⁸ Bell J., *Machine learning. Hands-on for Developers and Technical Professionals*, Wiley, Indianapolis 2015, s. 17-22.

każda manipulacja danymi wpływa na wynik całego przedsięwzięcia.¹³⁹ Inną operacją, którą należy rozważyć jest zamiana typów danych¹⁴⁰.

Z perspektywy statystyki wyróżnia się ich dwie podstawowe kategorie: ilościowe oraz jakościowe¹⁴¹. Pierwsza z nich odnosi się wielkości określanych liczbowo, mierzalnych oraz takich, które da się policzyć. Ich przykładem są zmienne reprezentujące wartości ceny, ilość dokonanych transakcji czy czas trwania sesji. Natomiast druga kategoria reprezentuje atrybuty w sposób opisowy dotyczące jakości cechy. Wartościami zaliczającymi się do tego zbioru będą kolor samochodu, poziom wykształcenia, płeć oraz grupa krwi. Modele zazwyczaj lepiej radzą sobie z operacjami na numerach, co zmusza do konwersji ich typów. Jedną z podstawowych czynności jest kodowanie wartości jakościowych na liczbowe (ang. *dummy variables*). Przykładowo, dla kategorii płciowej wyróżniającej dwa rodzaje: kobieta i mężczyzna lub gdy opisuje się dni świąteczne (święto / dzień powszedni), konwertuje się na binarną reprezentację 0 oraz 1. Gdy w zbiorze występuje więcej etykiet, reprezentowane mogą być jako kolejne numery. Podczas późniejszej oceny jakości projektu, wykonuje się inwersję cech, używając przy tym ustalonego w trakcie ich przygotowywania klucza.

Inną funkcją wykorzystywaną podczas przetwarzania danych jest ekstrakcja wartości (ang. *feature extraction*). Tutaj używane są na przykład elementy uczenia bez nadzoru, opisywane w rozdziale pierwszym. Stosuje się je w sytuacjach, gdy rozmiar surowych danych jest znaczący a wiele z jego wartości zbędnych. Jednym z rozwiązań jest na przykład redukcja wymiarów danych kompresując je w dużo bardziej treściwy i wielokrotnie mniejszy zestaw by model lepiej radził sobie w trakcie uczenia.

Bardzo często zdarza się, że rozpiętość wartości danych jest duża, oraz trudno jest wyznaczyć ich limity. Podczas trenowania algorytmów zadaniem jest znalezienie optymalnych wartości wag z perspektywy obranego celu. By uniknąć chaosu i ich nagłych skoków w trakcie ćwiczenia stosunkowy poziom ich zmian jest niewielki. W porównaniu do dużych wartości wejściowych oraz braku ich limitów proces ten jest często nieefektywny. Aby poradzić sobie z tym problemem wykonuje się normalizację danych (ang. *data normalization*). Oznacza to, że wartości przetwarza się tak by mieściły się w konkretnym przedziale liczbowym bez zmiany ich stosunku do siebie i relatywnego rozmieszczenia. Są dwie standardowe metody tej techniki.

¹³⁹ Podany został przykład manipulacji cechami wybrakowanymi. Jednak warto się jednak zastanowić, dlaczego tych wartości brakuje, taki stan rzeczy również informuje o przebiegu zdarzenia. Odpowiedź na to pytanie, odkryć może nowe zależności dając zupełnie inną perspektywę.

¹⁴⁰ Kelleher J. D., Namee B. M., D'Arcy A., *Fundamentals of Machine Learning for Predictive Data Analytics*, The MIT Press, London 2015, s. 55-56, 67-70.

¹⁴¹ Makać W., D. Urbanek-Krzysztofiak, *Metody opisu statystycznego*, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2010, s. 11-14, 17-20.

Pierwsza to normalizacja tak by dane mieściły się zazwyczaj w zakresie od 0 do 1. Taki stan uzyskuje się dzięki równaniu:

$$\mathbf{x}' = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$$

Gdzie \mathbf{x} to wartość wpasowywana w przedział, a $\min(\mathbf{x})$ oraz $\max(\mathbf{x})$ są odpowiednio minimalną i maksymalną wartością argumentu \mathbf{x} w bazie danych.¹⁴²

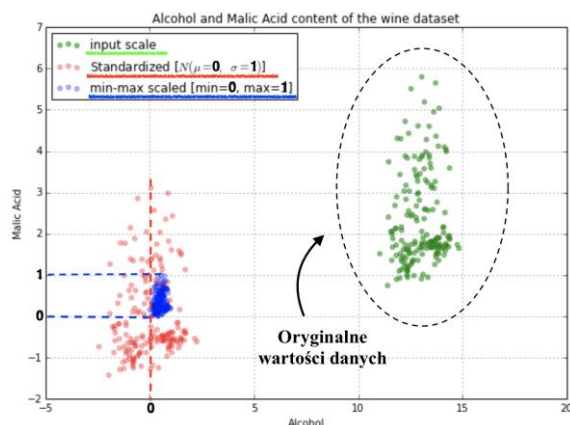
Alternatywną procedurą jest takie przeistoczenie by ich średnia wartość wynosiła $\bar{\mathbf{x}} = 0$ a wariancja $\sigma^2 = 1$ i odchylenie standardowe¹⁴³ $\sigma = 1$. Oznacza to, że środek danych po naniesieniu ich na oś współrzędnych znajduje się w punkcie 0 osi ox . Równanie tej operacji wygląda następująco:

$$\mathbf{z} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma}$$

W tym wzorze \mathbf{x} to wartość transformowana, $\bar{\mathbf{x}}$ to średnia wartość dla zbioru, a σ to odchylenie standardowe tego samego wzoru.¹⁴⁴

By lepiej zrozumieć zasadę działania, poniżej znajduje się przykład zastosowania wymienionych czynności.

Rysunek 4.1. Wizualizacja metod wstępnego przetwarzania danych



Źródło: Opracowanie własne na podstawie: IVlad, Linear Regression: Normalization Vs Standardization, <https://stackoverflow.com/questions/32108179/linear-regression-normalization-vs-standardization>, dostęp dnia 15.04.19.

¹⁴² Kelleher J. D., Namee B. M., D'Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 92-94.

¹⁴³ Odchylenie standardowe (σ) jest pierwiastkiem kwadratowym z wariancji (σ^2) więc w przypadku, gdy jedna z tych wartości równa jest 1, obie równe są 1 ($1^2 = 1 = \sqrt{1}$). Odchylenie standardowe interpretuje się jako przeciętne rozproszenie jakiejś cechy wokół jej średniej wartości. Gdy $\sigma = 0$ wtedy wszystkie wartości równe są wartości przeciętnej (średniej). Wartość ta informuje o wielkości różnic (rozproszeniu) pomiędzy obserwacjami.

¹⁴⁴ Scikit-learn developers, 5.3. Preprocessing data. <https://scikit-learn.org/stable/modules/preprocessing.html>, dostęp dnia 02.06.19.

Rysunek 4.1 Jest wizualizacją omawianych dwóch przykładów metod normalizowania danych. Dany jest wykres punktowy opisujący zależność pewnych dwóch zmiennych. Zielonym kolorem przedstawione są wartości oryginalne. Czerwoną barwą opisane zaś zostały dane znormalizowane drugim podanym sposobem, by ich średnia wynosiła 0 a odchylenie 1, natomiast niebieskim oznaczono efekt działania transformacji danych by mieściły się w wyznaczonym przedziale. Istotnym jest, że stosunek odległości pomiędzy punktami pozostaje ten sam i jedynie skala ulega zmianie.

Ostatnim bardzo ważnym etapem przed modelowaniem jest podział danych na trenujące i testowe (ang. *train/test split*). Jest to sposób na przeprowadzenie symulacji¹⁴⁵ pracy w rzeczywistości. Proces ten polega na wybraniu części obserwacji, na których komputer będzie się „uczył”, oraz tych, które posłużą do sprawdzenia jakości jego efektów na zupełnie nowych dla algorytmu danych. W prawdziwym świecie, zadaniem modelu uczenia maszynowego jest generowanie jak najlepszych rozwiązań dla analogicznych, do poznanych w procesie nauki problemów. Podstawowym jego atutem będzie umiejętność radzenia sobie z wystąpieniami, z którymi dotychczas nie miał do czynienia. Realny poziom skuteczności otrzymuje się zatem dzięki „okłamaniu” systemu, o ilości dostępnych danych separując pewną ich część w celu porównania później szacunków odpowiadających wystąpień (z jego perspektywy nieznanych), dla których posiada się prawidłowe odpowiedzi. Jest kilka różnych sposobów ich podziału by otrzymać jak najlepsze wyniki. Najpopularniejszym, jest podzielenie w sposób losowy danych w proporcjach 80% do trenowania oraz 20% do testów. Podane wartości wybiera się arbitralnie. Ważne jest by wartości dzielić w sposób losowy by zminimalizować szanse na wystąpienie niechcianych tendencji w obu grupach¹⁴⁶.

W trakcie przetwarzania danych można łączyć ze sobą wybrane metody. O kolejności przebiegu i wyborze operacji decyduje analityk w zależności od pożądaných efektów¹⁴⁷.

Czwarty etap nazywany jest modelowaniem. To moment, narodzin modelu i jego przygotowań by mógł stawić czoła problemom, które na niego czekają. Fazę zaczyna się od pobrania właściwego algorytmu, by następnie odpowiednio go dostroić zgodnie z wyznaczonymi założeniami. W momencie, gdy wszystkie dotychczasowe procesy ukończone

¹⁴⁵ Symulacja to odtwarzanie pewnego systemu lub jego działania bez rzeczywistego jego uruchamiania. Często wykorzystuje się w tym celu modele matematyczne, które opisują działanie jakiegoś zjawiska w określonym czasie. Gajda J. B., Prognozowanie i symulacje w ekonomii i zarządzaniu, C. H. Beck, Warszawa 2017, s. 16.

¹⁴⁶ Kelleher J. D., Namee B. M., D’Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015, s. 98-100.

¹⁴⁷ W trakcie planowania projektu należy wybrać logiczną ścieżkę tak by kolejne operacje miały sens i nie degenerowały późniejszych etapów. Na przykład, dobrą praktyką jest najpierw podzielić zestaw danych na treningowe i testowe, a następnie przeprowadzić normalizację ustawiając klucz transformacji w oparciu jedynie o dane treningowe. W ten sposób uzyskuje się rezultaty zbliżone do realnych nie podając dodatkowych informacji o obserwacjach testowych, które posłużą do sprawdzenia skuteczności na nieznanach wartościach.

zostały pomyślnie, nadszedł czas by zainicjować etap nauki, wprowadzając według zasad uprzednio przygotowane dane trenujące. Maszyna rozpoczęła intrygujący proces samodoskonalenia, czyli mozolne poszukiwanie optymalnych parametrów poprzez kolejne iteracje. Czas trwania cyklu jest bardzo zróżnicowany i w dużej mierze zależy od poziomu jego zaawansowania, złożoności problemu, ilości obserwacji oraz ich atrybutów. Długość „nauki” może zaczynać się od setnych części sekundy, by trwać nawet do 200 dni¹⁴⁸. Po zakończeniu nauki, często wraca się do tego etapu, by dostroić model w oparciu o informacje zwrotne z następnej części jaką jest ocena jakości przebiegu (ang. *model evaluation*).

Teraz w zależności od powierzonego zadania wykonywane są pomiary skuteczności i efektywności. Sprawdzane są takie rzeczy jak czas trwania, postępy w nauce w oparciu o wybrane wskaźniki błędów oraz stopień jego dopasowania do danych. Na tym etapie wykorzystuje się zbiór wartości testowych, by zweryfikować, jak maszyna radzi sobie z nowymi obserwacjami. Porównuje się odpowiedzi modelu z etykietami znanymi w porcji testowej co w rzetelny sposób weryfikuje uzyskane efekty. Jest wiele podstawowych miar błędów wykorzystywanych odpowiednio do podjętego problemu. Dla klasyfikacji może być to stosunek prawidłowego przyporządkowania do wszystkich udzielonych odpowiedzi, natomiast dla prognoz ceny, użyta może zostać różnica między wielkością szacunku i oczekiwaną. Opierając decyzję o wartości błędów zdecydować się można na powrót do dowolnego z czterech etapów budowy projektu. Zdarzyć się również może, że niezbędnym będzie redefinicja całej obranej ścieżki zaczynając tym samym pracę od nowa.

Jeśli efekty działań są satysfakcjonujące, płynnie przechodzi się do wdrożenia (ang. *deployment*). Warto przed wprowadzeniem algorytmu w jego docelowy obszar pracy, przeprowadzić również proces trenowania wprowadzając wyselekcjonowane w korku trzecim dane do testowania, tak by nie tracić cennej wartości jaką niosą ze sobą dodatkowe obserwacje. Wszystkie dotychczasowe rundy kierować miały do osiągnięcia założonego celu. Zbudowanie modelu wykorzystującego uczenie maszynowe podyktowane było wizją szansy usprawnienia pewnych zadań. Po wykonaniu wszystkich kroków powołuje się wykreowany model do wspomagania rozwiązań postawionych problemów.

¹⁴⁸ Firma Google w roku 2012 rozpoczęła projekt stworzenia sztucznej inteligencji posiadającej umiejętność wysoce skutecznego rozpoznawania zdjęć. Wykorzystana została głęboka sieć neuronowa, której liczba warstw na początku wynosiła 7 by po kilku latach w drodze doskonalenia skończyć na blisko 300-stu. Do szkolenia użyty został zestaw zawierający 1 milion oznaczonych obrazków, posiadających około 15 tys. atrybutów. Czas trwania „nauki” jak informuje Google trwał 6 miesięcy i zakończony został sukcesem. Gupta A., Revisiting the Unreasonable Effectiveness of Data, <https://ai.googleblog.com/2017/07/revisiting-unreasonable-effectiveness.html>, dostęp dnia 02.07.19.

Oczekując najlepszych rezultatów z podczas działania wdrożonego modelu „ufaj, ale sprawdzaj¹⁴⁹”, by nie dopuścić do porażki lub przynajmniej zminimalizować jej szanse. Jakość modelu prawdopodobnie będzie zmienna w czasie, a dostarczone dane okazać się mogą niewystarczające by utrzymać jej wartość na satysfakcjonującym poziomie. Podczas pierwszego etapu planowania, nie można zapomnieć o ustaleniu metod pomiarów skuteczności i udoskonalania, wdrożonego programu by osiągnąć efekty były stale zadowalające.

4.2. Wybór konstrukcji modelu sztucznych sieci neuronowych

Istnieje prawdopodobnie nieskończona liczba różnych sieci neuronowych, możliwych do skonstruowania ze względu na choćby możliwość dodawania kolejnych komórek i warstw. Jednak chaos, który powstałby w trakcie trwania tego procederu skutkowałby brakiem możliwości zapanowania nad modelem a w efekcie utraceniem sensownego poziomu istotności z jego użytkowania. By uniknąć zamieszania w procesie tworzenia i wykorzystania sztucznych sieci należy odpowiedzieć na kilka podstawowych pytań:

- ile warstw ukrytych posiadać ma sieć,
- ile neuronów zawierać będzie każda z warstw,
- jakie funkcje aktywacji zostaną zastosowane,
- jaką wybrać funkcję straty,
- jaką metodą optymalizować wagi by uzyskać minimalną wartość funkcji straty.

Powyższe kwestie mają duży wpływ na przebieg działania programu, tworząc jego najważniejsze komponenty. Brakuje jednoznacznych odpowiedzi na wymienione pytania, natomiast istnieją wskazówki, którymi twórca może się kierować by osiągnąć sukces.

Rozważania rozpoczną się od wizualizacji różnic między zakresem pól decyzyjnych w zależności od liczby warstw sieci by lepiej zrozumieć istotność tego parametru.

¹⁴⁹ Rosyjskie powiedzenie spopularyzowane przez Ronalda Reagana, amerykańskiego polityka. Przez niektórych stwierdzenie uznawane za jedną z podstawowych zasad biznesu. The New York Times Company, Trust But Verify, <https://www.nytimes.com/1987/12/10/opinion/trust-but-verify.html>, dostęp dnia 23.06.19.

Rysunek 4.2. Różnice pomiędzy sieciami o różnej liczbie warstw ukrytych na przykładzie problemu klasyfikacji

TABLE I. FEATURE OF MULTI LAYER PERCEPTRON

Structure		Area	
Number of Layer	Diagram	Description	Example with XOR Problem
1		Half plane	
2		Overlapped area between lines	
3		Enclosed area for multiple regions	

Źródło: Hyoung U. H., Jong-Kook K., An evolutionary genetic neural network for problems without prior knowledge, Table 1, <https://www.semanticscholar.org/paper/An-evolutionary-genetic-neural-networks-for-without-Ha-Kim/79fb50f1aad2b4d6f3ff43129540ea37b30f9829/figure/6>, dostęp dnia 01.07.19.

Jak widać na rysunku 4.1 oraz zgodnie z opisem perceptronu w rozdziale pierwszym, pojedynczy neuron jest w stanie rozwiązać jedynie problemy separowane liniowo. Wizualizacja jego skuteczności widoczna jest na przykładzie numer jeden po prawej stronie rysunku. W takim przypadku obiekt przypisywany jest do jednej z dwóch kategorii, znajdujących się po przeciwnych stronach prostej. Opisana konstrukcja nie poradzi sobie z zadaniami, w których obszary kategorii nachodzą na siebie jak i innych nie dających się w sposób satysfakcjonujący oddzielić prostą linią. Dodając pierwszą warstwę model „rozwiąże” wszystkie problemy, w których obiekty da się rozdzielić przy pomocy otwartych lub zamkniętych obszarów decyzyjnych wypukłych i jednospójnych¹⁵⁰. W ten sposób tylko częściowo rozwiązane zostały wymienione ograniczenia sieci jednowarstwowych, pozostawiając wciąż problem separacji danych nachodzących na siebie jako aktualny. Aby wyeliminować również to ograniczenie, dodaje się kolejną warstwę tworząc dwuwarstwową sztuczną sieć neuronową. Czym więcej neuronów zastosowanych jest podczas jej budowy, tym obszary decyzyjne stają się coraz bardziej skomplikowane. Sieć zbudowana z dwóch ukrytych warstw potrafi dostroić się praktycznie do każdego problemu klasyfikacji, jak i również dopasować się do dowolnej linii regresji. Wielkości i kształty reprezentujące pola, będące podstawą do podejmowania szacunków przez model determinowane są przez liczbę wykorzystanych neuronów w warstwach. Manipulacja liczbą powłok wewnętrznych wpływa na poziom abstrakcji w kolejnych jej wystąpieniach rosnąc proporcjonalnie do ich ilości¹⁵¹.

¹⁵⁰ Najman K., Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie, Uniwersytet Gdański, Sopot 2003, s. 176. (praca doktorska)

¹⁵¹ Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, Cambridge 2006, s. 257-258.

Gdy ustalono już złożoność warstwową sieci kolejnym pytaniem jest, ile neuronów w każdej z nich. Zgodnie z powyższym akapitem, determinują one obszar decyzyjny, który bezpośrednio wpływa na jakość modelu. Analitykowi zależy by program był jak najmniej skomplikowany spotykając się jednocześnie z oczekiwanymi rezultatami. Zatem w przypadku dwóch sieci, o porównywalnej skuteczności, różniącym się jedynie wolumenem warstw, tę mniej złożoną rozważa się jako lepszą. Taka sama zasada odnosi się do liczby neuronów. W pierwszej powłoce ilość komórek adekwatna ma być do rozmiaru obiektu, który jest podstawą by szacować wartość wyjściową. Ustalenie pożądanego numeru definiującego liczebność komórek ostatniej warstwy jest również stosunkowo łatwym zadaniem, gdyż wartość ta odpowiadać powinna oczekiwanemu kształtowi informacji zapewnionej przez algorytm¹⁵². Przykładowo, gdy na podstawie wzrostu, wagi oraz wieku analityk chce by model przyporządkował obiekt jako kobietę lub mężczyznę dobrym pomysłem jest inicjacja warstwy wyjściowej z trzema neuronami odpowiadającym kolejno parametrom. W warstwie wyjściowej uzasadnione będzie skorzystanie z dwóch neuronów reprezentujących dwie możliwości przyporządkowania. Jeśli zdecydowano się dodać warstwy ukryte, najprostszym sposobem ustalenia liczby neuronów dla warstwowych sieci neuronowych jest użycie takiej samej wartości co w warstwie wejściowej. Zatem dla zaproponowanego klasyfikatora powłoki ukryte początkowo składać się będą z trzech komórek¹⁵³.

Przechodząc do kolejnego etapu istotne jest ustalenie typów funkcji aktywacji. Decydują one czy kolejny neuron z nimi połączony ma zostać aktywowany, wprowadzając również możliwość nieliniowych zachowań wygenerowanych prostych rysujących obszary decyzyjne. Najpopularniejsze z nich zostały szczegółowo opisane w rozdziale pierwszym. Typ funkcji dla każdej z warstw wybiera się arbitralnie, testując kombinacje swoich faworytów w kolejnych wersjach modelu porównując ich przebieg między sobą. Standardowym wyborem dla problemów, gdzie celem jest znalezienie linii najlepszego dopasowania będzie wielowarstwowy perceptron z tangensem hiperbolicznym w warstwach ukrytych oraz liniową funkcją jako ostatnią. Generalnie sigmoidalna funkcja spotyka się z coraz mniejszym entuzjazmem w porównaniu do tangensa hiperbolicznego w ogóle. Podany został przykład liniowej funkcji ze względu na ciągłość jej wartości wyjściowej. Jest to standardowa procedura przy zadaniach regresji¹⁵⁴. W modelach rozwiązujących problemy klasyfikacji zazwyczaj stosuje się nieliniowe funkcje wybierając sigmoidalną bądź tangens hiperboliczny hiperbolicznego dla warstwy wyjściowej.

¹⁵² Hagan M. T. [i. in.], *Neural Network Design*, Martin Hagan, Stwillwater 2014, s. 2-19.

¹⁵³ Najman K., *Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie*, Uniwersytet Gdański, Sopot 2003, s. 177. (praca doktorska)

¹⁵⁴ Hagan M. T. [i. in.], *Neural Network Design*, Martin Hagan, Stwillwater 2014, s. 22-8-22-13.

Ostatnie dwa aspekty, determinujące przebieg działania sieci to rodzaj funkcji celu oraz optymalizator. Wymienione zostały wspólnie ze względu na ich bliską współpracę w trakcie trenowania modelu. Funkcja celu, inaczej nazywana również funkcją straty (ang. *loss function*) jest równaniem, będącym miarą sukcesu, odmierzaną w kolejnych iteracjach procesu nauki. Nadrzędnym zadaniem sieci jest minimalizowanie zwracanej przez funkcję wartości. Wybrany rodzaj powinien być jak najbardziej skorelowany z intencją zadania by nie wykonywane były zbędne czynności wpływając negatywnie na poziom zadowolenia z rezultatów. Dla modeli innych niż klasyfikacyjne najczęściej wybierana jest średni błąd kwadratowy¹⁵⁵. Miara reprezentuje przeciętną wielkość różnic pomiędzy wartością szacowaną, a oczekiwaną podniesioną do kwadratu. To najpopularniejsza miara dopasowania wykorzystywana do obliczania jakości zgrania obserwacji z szacunkiem modelu. Jest wiele innych jej odpowiedników jak na przykład absolutna suma wszystkich różnic, lecz niekorzystnie dla niej świadczy brak czułości na rozbieżność wielkości błędów. Miary te spisują się dla wszystkich modeli szacujących wartości ciągle. W przypadku klasyfikatorów jednym z najlepszych wyborów jest entropia krzyżowa (ang. *cross entropy*). Niestety działa ona tylko przy założeniu, że klasy są dwie. Gdy w zbiorze rozwiązań znajdują się więcej niż dwie grupy stosuje się wtedy najczęściej kategoryzacyjną entropię krzyżową. Warto wiedzieć, że występuje możliwość pisania własnych równań funkcji straty, natomiast należy korzystać z tego przywileju rozważnie, gdyż podstawowe rodzaje dla większości problemów są wystarczające¹⁵⁶.

Optymalizator (ang. *optimizer*) to sposób w jaki poszukuje się najlepszych parametrów wag by funkcja straty zwracała minimalne wartości. Są to kluczowe składowe dla procesu nauki, gdyż dostarczają niezbędnych informacji na temat jakości efektów dobranych sił połączeń oraz wskazują, jak należy je zmienić by rezultaty zbliżały się do sukcesu. Najpopularniejszą metodą wyznaczania nowych wag jest metoda gradientu prostego (ang. *gradient descent*). To jeden ze sposobów w jaki sieć optymalizuje wagi, aby w efekcie wyniki zwracały najmniejszą wartość funkcji straty, co równoznaczne jest z rosnącą jakością modelu.

Nie ma jednego przepisu wskazującego na najlepszą strukturę sieci oraz opisującej atrybuty w niej wykorzystane. Na wszystkie powyższe pytania brakuje jednoznacznej odpowiedzi, natomiast powstało wiele wskazówek, pomocnych z punktu widzenia kreowania sztucznej sieci neuronowej. Do optymalnej budowy dochodzi się poprzez wykonanie sekwencji prób, na których podstawie edytuje się sieć neuronową aż do momentu, gdy jej efekty są zadowalające¹⁵⁷.

¹⁵⁵ Jego szczegółowy opis znajduje się w kolejnym podrozdziale.

¹⁵⁶ Chollet F., Deep learning. Praca z językiem Python i biblioteką Keras, Helion, Gliwice 2018, s.74.

¹⁵⁷ Zocca V., Spacagna G., Slater D., Roelants P., Deep learning. Uczenie głębokie z językiem Python, Helion, Gliwice 2017, s. 110.

4.3. Utworzenie oraz trenowanie modelu predykcyjnego SSN

Celem niniejszego rozdziału jest budowa sztucznej sieci neuronowej, która następnie wykorzystana zostanie do zbadania hipotezy o możliwym jej wykorzystaniu do tworzenia satysfakcjonującej prognozy cen na giełdzie rynku kapitałowego. Operacje będą dokonywane przy użyciu notowań wartości zamknięcia, czyli ostatniej ceny wypracowanej w ciągu dnia. Badany indeks dotyczy amerykańskiej korporacji Apple, zajmującej się przede wszystkim produkcją wysokiej klasy urządzeń elektronicznych oraz oprogramowania, mającej swoją siedzibę w stanie Kalifornia na zachodnim wybrzeżu USA. Wybór podyktowany jest wysokim poziomem istotności firmy na światowym rynku, co zapewnia jej stabilność jak również bogatą historią danych dotyczących jej nieustannej wyceny akcji od debiutu na nowojorskiej giełdzie papierów wartościowych NASDAQ w 1980 roku. Z racji niewielkich zmian oraz relatywnie niskich cen w ciągu pierwszych blisko dwudziestu lat istnienia jako spółka akcyjna wybrane zostały jedynie wartości z przedziału od stycznia roku 2000 do stycznia roku 2018. W tym okresie czasu zanotowano 4 528 aktywnych dni na giełdzie zakończonych sukcesem. Dane determinowane tym kryterium wyboru, będą przedmiotem badań w tym rozdziale.

Respektując wcześniej opisane zasady budowy modelu wykorzystującego uczenie maszynowe, w tym podrozdziale przeprowadzone zostaną następujące kroki:

- import danych,
- przygotowanie danych,
- utworzenie modelu,
- trenowanie modelu.

W kolejnym podrozdziale powstaną prognozy przygotowanych SSN w dwóch różnych horyzontach czasowych, na których podstawie przedstawiona zostanie ocena ich skuteczności oraz wnioski.

Tworząc kod w języku Python, zazwyczaj pierwszym krokiem jest import podstawowych bibliotek wykorzystywanych w trakcie pracy. Zgodnie z konwencją pobrane zostają takie pakiety jak Pandas, NumPy oraz Matplotlib. Następnie przypisany zostaje wybór pożądanego przedziału czasowego by następnie korzystając z kolejnej biblioteki jaką jest `pandas_datareader` wczytać wartości do pliku bezpośrednio ze strony Yahoo Finance¹⁵⁸.

¹⁵⁸ Amerykańska popularna strona internetowa poświęcona dziedzinie finansów. Jest elementem tworzącym główny człon jakim jest Yahoo.com.

Rysunek 4.3. Import danych

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 %matplotlib inline

1 start_date = '2000-01-01'
2 end_date = '2018-01-01'

1 import pandas_datareader as pdr
2
3 df = pdr.DataReader(name='AAPL', data_source='yahoo', start=start_date, end=end_date)
4 df.head(5)
```

Źródło: Opracowanie własne.

Na koniec wyświetlonych zostaje pierwszych pięć rzędów obiektu DataFrame by wstępnie zapoznać się z danymi.

Rysunek 4.4. Widok zaimportowanych danych

	High	Low	Open	Close	Volume	Adj Close
Date						
2000-01-03	4.017857	3.631696	3.745536	3.997768	133949200.0	2.655498
2000-01-04	3.950893	3.613839	3.866071	3.660714	128094400.0	2.431611
2000-01-05	3.948661	3.678571	3.705357	3.714286	194580400.0	2.467196
2000-01-06	3.821429	3.392857	3.790179	3.392857	191993200.0	2.253689
2000-01-07	3.607143	3.410714	3.446429	3.553571	115183600.0	2.360442

Źródło: Opracowanie własne.

Jak widać na powyższym rysunku w tabeli znajdują się informacje opisujące wartości skrajne notowań, wolumen transakcji czy cena otwarcia. Ponieważ interesująca z perspektywy projektu jest jedynie wartość cen zamknięcia, do analizy wybrana zostaje jedynie kolumna „Close”, poprzez przypisanie jej danych do nowej zmiennej. Aby upewnić się, że czynność przebiegła pomyślnie, wyświetlone zostaną początkowe jej elementy.

Rysunek 4.5. Wybór pożądanych wartości

```
1 df_close = df[['Close']].copy()
2 df_close.head(5)
```

	Close
Date	
2000-01-03	3.997768
2000-01-04	3.660714
2000-01-05	3.714286
2000-01-06	3.392857
2000-01-07	3.553571

Źródło: Opracowanie własne.

Przy użyciu pakietu Matplotlib dane zawarte w nowo zdefiniowanej bazie danych zwizualizowane są na wykresie poniżej. Reprezentuje on wykres cen zamknięcia akcji firmy Apple wyrażony w dolarach amerykańskich, od 2000 do 2018 roku.

Rysunek 4.6. Wizualizacja danych, ceny zamknięcia firmy Apple



Źródło: Opracowanie własne.

Po zaimplementowaniu wszystkich interesujących obserwacji przechodzi się do procesu przetwarzania danych. Wiadomo, że badany szereg czasowy jest kompletny i rzetelny ze względu na źródło jego pochodzenia, zatem na tym etapie przeprowadzone zostaną jedynie takie działania jak podział danych na do trenowania i testowe oraz ich normalizacja. W momencie, gdy powyższe operacje zakończone zostaną sukcesem, wykorzystując obiekt klasy „generator” zgodnie z założeniami projektu przygotowane będą odpowiednie porcje danych w celu realizacji uczenia nadzorowanego.

Najpierw istotnym jest poznać wielkość zestawu obserwacji by następnie arbitralnie wybrać wielkości jakie określać będą rozmiary poszczególnych pakietów „train” i „test”. W rozdziale trzecim jako metodę podziału opisano losowy wybór danych ze zbioru najczęściej w proporcjach 8/2. Jednakże w tym wypadku, przywołany sposób byłby nierozsądny, z uwagi na typ danych jakim jest szereg czasowy. Zgodnie z jego budową, każda kolejna obserwacja ma swoje miejsce w czasie, przy czym wartości bezpośrednio od siebie zależą. Nie można wybrać zatem losowo pozycji to testowania z racji utraty ich istotnej cechy jaką jest ciąg przyczynowo-skutkowy. Zatem by w racjonalny sposób podzielić wartości, za zbiór testowy posłuży wybrana partia najnowszych notowań ułożona w sposób chronologiczny, natomiast danymi trenującymi będą wszystkie pozostałe poprzedzające je wielkości. Jej rozmiar determinuje wybrana subiektywnie liczba 128, uznana za wystarczającą porcję do przeprowadzenia wielokrotnej predykcji jednocześnie pozostawiając liczny zestaw danych trenujących. Zatem w tym przypadku dane, na których model będzie się „uczył” stanowią blisko 97% całości.

Po wykonaniu podziału danych, przypisując odpowiednio części do zmiennych „train” i „test”, wyświetlony zostaje aktualny rozkład danych uwzględniając ich fragmentację.

Rysunek 4.7. Podział danych oraz ich wizualizacja

```
1 # Train set
2 train = df_close.iloc[:4400] # pandas.core.series.Series (4400,) **
3 train_index = train.index # pandas.core.indexes.datetimes.DatetimeIndex
4
5 # Test set # 128 sessions
6 test = df_close.iloc[4400:] # pandas.core.series.Series (128,) **
7 test_index = test.index # pandas.core.indexes.datetimes.DatetimeIndex
8
9 plt.figure(figsize=(15, 4))
10 plt.autoscale(tight=True, axis='x')
11 plt.grid(True)
12 plt.plot(train, lw=2, color='blue')
13 plt.plot(test, lw=2, color='orange')
14 plt.legend(['Train', 'Test'])
15 plt.xlabel('Date')
16 plt.ylabel('Price [USD]')
17 plt.title('AAPL Stock Close Price since 01-01-2000 untill 01-01-2018')
```



Źródło: Opracowanie własne.

Kolejnym działaniem zastosowanym w ramach przetwarzania danych jest ich normalizacja. Z jej wykorzystaniem, algorytm w bardziej efektywny sposób poszukiwać będzie optymalnych wag dla połączeń neuronowych. W tym wypadku, przekształcenie wykorzystane zostanie jedynie dla danych osi y (pionowej) czyli wielkości cen. Istotnym jest przeprowadzenie obróbki analizowanych danych zgodnie z kolejnością przedstawioną w tym rozdziale, ponieważ wtedy istnieje możliwość ich normalizowania tworząc klucz dostosowany jedynie do danych „train”. Bezspornie wszystkie obserwacje zarówno do trenowania jak i do testów, przekształcone zostaną za pomocą tej metody. Ważne jest natomiast, że gdy dzięki równaniu tworzony zostaje pewien pryzmat przez, który wartości zostaną przepuszczone, programiście zależy by nie przekazać żadnych wskazówek dotyczących rozkładu danych wykorzystywanych do sprawdzenia. Jak widać na rysunku 4.8, w tym celu metoda „fit” będąca częścią klasy „MinMaxScaler” użyta zostanie jedynie na danych wyselekcjonowanych do trenowania. Po zmianie formatu wartości cen, wyświetlona zostaje nowo powstała początkowa ich część.

Rysunek 4.8. Normalizacja danych

```
1 from sklearn.preprocessing import MinMaxScaler

1 # From Pandas TimeSeries into NumPy array
2 # Train
3 train = train.values # numpy.ndarray (4400,)
4 train = train.reshape((len(train), 1)) # numpy.ndarray (4400, 1)
5
6 # Test
7 test = test.values # numpy.ndarray (128,)
8 test = test.reshape((len(test), 1)) # numpy.ndarray (128, 1)

1 # Fit the train data only. You don't want to provide any clue about test set condition.
2 scaler = MinMaxScaler()
3 scaler.fit(train)

1 # Transform the data then export to Pandas TimeSeries as began
2 # Train
3 train_norm = scaler.transform(train) # numpy.ndarray (4400, 1)
4 train_norm = train_norm.tolist() # list of single value lists [[x], [y], [z]]
5 train_norm = np.squeeze(train_norm) # numpy.ndarray array([x, y, z])
6 train_norm = pd.Series(train_norm, train_index) # pandas.core.series.Series (4500,) **
7
8 # Test
9 test_norm = scaler.transform(test) # numpy.ndarray (128, 1)
10 test_norm = test_norm.tolist() # list of single value lists [[x], [y], [z]]
11 test_norm = np.squeeze(test_norm) # numpy.ndarray array([x, y, z])
12 test_norm = pd.Series(test_norm, test_index) # pandas.core.series.Series (128,) **

1 train_norm.head(5)
```

Źródło: Opracowanie własne.

Jak porównać można na rysunku 4.8 wielkość dla dnia trzeciego stycznia, 2000 roku wynosi oryginalnie \$ 3.997768, natomiast po jej normalizacji równa się 0.019725.

Rysunek 4.9. Znormalizowane wartości badanego szeregu

Date	
2000-01-03	0.019725
2000-01-04	0.017553
2000-01-05	0.017898
2000-01-06	0.015827
2000-01-07	0.016862

Źródło: Opracowanie własne.

Gdy już format danych jest satysfakcjonujący nadszedł czas by przygotować odpowiednio porcje informacji, by model sieci neuronowej był w stanie sprostać oczekiwaniom analityka. Działanie schematu opiera się o dwudziestosesyjny pakiet danych, które uznane zostały za optymalną informację wystarczającą do szacowania dwudziestej pierwszej wartości ceny następującej zaraz po nich. Zatem model składał się będzie z dwudziestu neuronów w warstwie wejściowej i jednego w ostatniej jego powłoce. Pasujące do tej sytuacji paczki powinny zawierać dwadzieścia wartości, opisywanych pojedynczą etykietą. Jest to rodzaj uczenia nadzorowanego opisany w rozdziale pierwszym. Wykorzystując bibliotekę Keras importowany zostaje obiekt, „TimeseriesGenerator”, który zgodnie z zastosowaniem opisanym w dokumentacji, po poprawnym wprowadzeniu atrybutów „zapakuje” odpowiednio wartości w pakiety danych gotowych do użycia przez sieć. Zgodnie z charakterystyką modułu Keras w rozdziale trzecim w Pythonie docelowo współpracuje ona z biblioteką TensorFlow, co widać na rysunku 4.9 w informacji wyświetlonej podczas jej wczytywania.

Najistotniejsze w trakcie tworzenia generatora jest wprowadzenie takich atrybutów jak źródło danych, ich etykiet oraz oczekiwany kształt paczki, który w tym przypadku wynosi 20.

Po wykonaniu poniższego kodu, do zmiennej „generator” przypisane zostaną przygotowane wartości gotowe by rozpocząć proces „nauki”.

Rysunek 4.10. Utworzenie obiektu generatora

```
1 from keras.preprocessing.sequence import TimeseriesGenerator
Using TensorFlow backend.

1 n_input = 20
2
3 # All the data and targets are placed in the same set: train_norm
4 generator = TimeseriesGenerator(data=train_norm,
5                                targets=train_norm,
6                                length=n_input,
7                                batch_size=1)
```

Źródło: Opracowanie własne.

Po przygotowaniu danych nadszedł czas na budowę oraz trenowanie modelu. Przede wszystkim, należy zaimportować interesujące biblioteki dedykowane do budowy sztucznych sieci neuronowych. Pierwszy wiersz, pobiera z kategorii „models” jeden z typów oznaczony jako „Sequential”. Odnosi się on do strukturalnej budowy, wyznaczonej przez kolejne warstwy, tak jak zostało opisane w poprzednich rozdziałach. Obiekt „Dense” podkategorii „layers”, jest jednym z typów połączeń, którego nazwa w języku polskim odpowiada słowu „gęsty”. Oznacza to, że neurony między sobą, będą połączone na zasadzie „każdy z każdym”. Na koniec importowany jest również cały moduł „layers”.

Rysunek 4.11. Import biblioteki Keras

```
1 from keras.models import Sequential
2 from keras.layers import Dense
3 from keras import layers
```

Źródło: Opracowanie własne.

W poniższym kroku tworzona jest dokładna struktura sieci, uwzględniając takie podzespoły jak ilość warstw, liczby ich neuronów oraz funkcji aktywacji dla każdej z nich. Oprócz tego, przed rozpoczęciem procesu nauki, trzeba również zdefiniować optymalizator oraz funkcję celu.

Proces rozpoczynany jest od importu biblioteki „time” by zmierzyć czas trwania nauki. Później tworzona jest pętla, wykonująca się trzykrotnie, by zminimalizować ryzyko szansy na błąd przy wyznaczaniu parametrów w tym samym modelu. Każda kolejna iteracja, od nowa inicjuje całą sieć, z tymi samymi podzespołami. Pozwoli to porównać pracę optymalizatora i wybrać tą, której przebieg jest najbardziej satysfakcjonujący. Kreację samej sieci zaczyna się od stworzenia obiektu „Sequential”, określającego jej jednokierunkowy przepływ danych. Obiekt przypisany zostaje do zmiennej „model” by zwiększyć czytelność kodu. Następnie dodawane są warstwy, zaczynając od pierwszej, aż do ostatniej zgodnie z ich ustawieniem. W przypadku początkowej powłoki niezbędne jest ustalenie ilości parametrów wejściowych

poprzez argument „input_dim”, jak również ilość parametrów w kolejnej warstwie. Zgodnie z założeniami wartość dla zmiennej „n_input” podanej w warstwie wynosi 20. Dodając kolejne wiersze z funkcją „add” wprowadzane są następne warstwy ukryte. Ostatnia z nich automatycznie definiuje wartość wyjściową. Jak widać zgodnie z podstawową wskazówką zakładania sieci każda z warstw posiada tyle samo komórek co ilość wartości wejściowych. Wyjątkiem jest powłoka, zwracająca wartość wyjścia. Ta posiada jedną komórkę zgodnie z oczekiwaniem. Do wszystkich warstw przypisane zostały funkcje aktywacji. Następnie używając funkcji „compile” dla klasy „Sequential” zatwierdza się wybór optymalizatora i funkcji straty. Gdy model jest gotowy, rozpocząć można etap poszukiwań optymalnych wag SSN, korzystając z narzędzia „fit_generator”. Zaproponowany sposób wprowadzania danych podyktowany jest specyfiką budowy obiektu generującego paczki sukcesywnie w trakcie procesu „uczenia”. Istotne jest podanie wartości dla argumentu „epochs”, twierdzącej o ilości iteracji, doskonalenia parametrów. Na koniec model jest zapisywany, wraz z rekordami jego funkcji straty co przedstawione zostało na rysunku 4.12.

Rysunek 4.12. Utworzenie oraz trenowanie modelu

```
1 import time
2 elapsed_all = {}
3 for i in range(3):
4     model = Sequential()
5
6     # n_input = 20
7     model.add(Dense(20, activation='tanh', input_dim=n_input))
8     model.add(Dense(20, activation='tanh'))
9     model.add(Dense(1, activation='linear'))
10
11     model.compile(optimizer='sgd', loss='mse')
12
13     start = time.time()
14
15     model.fit_generator(generator=generator,
16                       epochs=500,
17                       verbose=2)
18
19     end = time.time()
20     elapsed = end - start
21     elapsed_all.append(elapsed/60)
22
23     # Model save
24     model.save("my_MODEL1-tanh-tanh-linear-sgd_gen" + str(i) + ".h5")
25
26     # Loss history save
27     df_loss = pd.DataFrame(np.array(model.history.history['loss']))
28     df_loss.to_csv("my_MODEL1-tanh-tanh-linear-sgd_gen_LOSS" + str(i) + ".csv")
29
```

Źródło: Opracowanie własne.

Wykonanie powyższej komórki, skutkuje inicjacją modelu oraz przeprowadzeniem procesu jego trenowania by, po zakończonych sukcesem wymienionych etapach zapisać efekty w oddzielnych plikach. W tym konkretnym wypadku powstanie SSN o dwóch ukrytych warstwach, dwudziestu parametrach wejściowych, dwudziestu parametrach w każdej warstwie ukrytej oraz jednym parametrem wyjściowym. Oprócz tego, sieć w ostatniej warstwie stosuje liniową funkcję aktywacji oraz tangensa hiperbolicznego we wszystkich pozostałych. W trakcie ustalania optymalnych wag stosuje ona metody gradientu prostego (SGD) by minimalizować przeciętny błąd kwadratowy (MSE) będący jej funkcją straty. Stosowanie ostatnich linijek kodu nie jest konieczne, lecz bardzo pomocne. Dzięki nim programista jest w stanie dowolnie korzystać ze swojego projektu, wykonując na nim różnorakie operacje bez konieczności

ponownej „nauki”, co w tym wypadku trwa około 45 minut¹⁵⁹ dla każdej z trzech iteracji. Jedną z często stosowanych operacji na uprzednio przygotowanym i zapisanym modelu, jest trenowanie go w przyszłości wprowadzając odpowiednio nowe dane.

W oparciu o kod zaprezentowany na rysunku 4.12 poprzez jedynie zmianę parametrów „activation” oraz nazwy zapisu pliku, stworzone zostaną w sumie dwa dodatkowe modele sztucznych sieci neuronowych. Wszystkie z nich różnić się będą wyłącznie rodzajem wykorzystanej funkcji aktywacji. Łącznie subiektywnie wybrane zostały następujące ich rodzaje:

- tangens hiperboliczny (tanh),
- sigmoidalna (sigmoid),
- dodatnia liniowa (ReLU).

W rezultacie wykonania kodu powstaną trzy różne rozwiązania dla trzech różnych funkcji aktywacji dając w sumie dziewięć różnych plików z modelami. Jak opisano wcześniej, strategia ta jest podyktowana słabościami zastosowanego optymalizatora, który niekoniecznie za każdym razem wybierze najlepsze parametry. Chcąc zminimalizować szansę przypadkowego błędu, wykonane zostają trzy próby dla kolejnych typów.

4.4. Porównanie i interpretacja wyników uczenia maszynowego

Na początku, przy użyciu funkcji biblioteki Pandas wczytana zostaje przygotowana wcześniej tabela z podstawowymi danymi na temat niektórych parametrów sieci, ścieżki prowadzącej do folderu wraz z jego nazwą jak również tytułu pliku zawierającego wytrenowany model. Rysunek 4.13 przedstawia widok ze wszystkimi szczegółami niezbędnymi do wygodnego przeprowadzenia prognoz, ich sprawdzenia wraz z wizualizacją.

¹⁵⁹ Dla laptopa z procesorem 2,3 GHz Intel Core i5 oraz pamięcią 8 GB 2133 MHz LPDDR3, (MacOS Mojave).

Rysunek 4.13. Wczytanie gotowych modeli z zapisanego pliku

```

1 data = pd.read_csv('activation_function_csv.csv', sep=';', encoding='utf-8', index_col='index')
2 data.dropna(inplace=True)
3 data

```

	optimizer	path	folder_name	model_name	loss_history	generator	hidden_layers	neurons
0.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-relu-linear-sgd/	my_MODEL1-relu-linear-sgd_gen0.h5	my_MODEL1-relu-linear-sgd_gen_LOSS0.csv	1.0	2.0	20.0
1.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-relu-linear-sgd/	my_MODEL1-relu-linear-sgd_gen1.h5	my_MODEL1-relu-linear-sgd_gen_LOSS1.csv	1.0	2.0	20.0
2.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-relu-linear-sgd/	my_MODEL1-relu-linear-sgd_gen2.h5	my_MODEL1-relu-linear-sgd_gen_LOSS2.csv	1.0	2.0	20.0
3.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-sigmoid-sigmoid-linear-sgd/	my_MODEL1-sigmoid-sigmoid-linear-sgd_gen0.h5	my_MODEL1-sigmoid-sigmoid-linear-sgd_gen_LOSS0...	1.0	2.0	20.0
4.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-sigmoid-sigmoid-linear-sgd/	my_MODEL1-sigmoid-sigmoid-linear-sgd_gen1.h5	my_MODEL1-sigmoid-sigmoid-linear-sgd_gen_LOSS1...	1.0	2.0	20.0
5.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-sigmoid-sigmoid-linear-sgd/	my_MODEL1-sigmoid-sigmoid-linear-sgd_gen2.h5	my_MODEL1-sigmoid-sigmoid-linear-sgd_gen_LOSS2...	1.0	2.0	20.0
6.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-tanh-tanh-linear-sgd/	my_MODEL1-tanh-tanh-linear-sgd_gen0.h5	my_MODEL1-tanh-tanh-linear-sgd_gen_LOSS0.csv	1.0	2.0	20.0
7.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-tanh-tanh-linear-sgd/	my_MODEL1-tanh-tanh-linear-sgd_gen1.h5	my_MODEL1-tanh-tanh-linear-sgd_gen_LOSS1.csv	1.0	2.0	20.0
8.0	sgd	/Users/patryklaskowski/Desktop/ann_models/	MODEL1-tanh-tanh-linear-sgd/	my_MODEL1-tanh-tanh-linear-sgd_gen2.h5	my_MODEL1-tanh-tanh-linear-sgd_gen_LOSS2.csv	1.0	2.0	20.0

Źródło: Opracowanie własne.

Jak widać na powyższym rysunku liczba sieci zgadza się z oczekiwaną wartością (9). W tabeli zawarte są informacje na temat ilości neuronów oraz liczby ukrytych warstw. W kolumnie „optimizer” znajdują się dane reprezentujące typ wykorzystanego optymalizatora. Porównując nazwy plików można zauważyć, że nie różnią się one od wyznaczonych w ramach kodu na rysunku 4.12.

Gdy struktura tabeli jest znana, rozpoczęta zostaje procedura wczytywania modelu do pliku, w którym zanalizowane zostaną algorytmy. Biblioteka Keras, zapewnia specjalną metodę do pobierania modelu sieci neuronowych. Istotny jest odpowiedni dobór typu pliku podczas zapisu algorytmu, w innym wypadku, nie będzie możliwy jego odczyt. Poniższy rysunek 4.14 przedstawia kod w języku Python, automatycznie importujący wszystkie dotychczas zbudowane modele sieci neuronowych poprzez wykorzystanie danych z tabeli na rysunku 4.13.

Rysunek 4.14. Wczytywanie zapisanych modeli sztucznych sieci neuronowych

```

1 from keras.models import load_model

2 # Create dictionary with models names as a key and its Keras equivalent with loss history based on MSE
3 # full_model_path = folder_path + folder_name + model_name
4
5 MODELS = {}
6 for i in range(len(data)):
7     model = load_model(data.iloc[i][path] + data.iloc[i][folder_name] + data.iloc[i][model_name])
8     model_loss = pd.read_csv(data.iloc[i][path] + data.iloc[i][folder_name] + data.iloc[i][loss_history],
9                             index_col='Unnamed: 0')
10    print(NAMES[i], end=", ")
11    MODELS[NAMES[i]] = [model, model_loss]

```

Źródło: Opracowanie własne.

Na rysunku 4.14 utworzona zostaje główna zmienna „MODELS” zawierająca przykładowe nazwy wraz z odpowiadającymi im algorytmami SSN. Zastosowane zostały następujące nazwy dla kolejnych modeli: model_1_1, model_1_2, model_1_3, model_2_1,

model_2_2, model_2_3, model_3_1, model_3_2, model_3_3. Pierwsza grupa, o numerze od 1_1 do 1_3 to model o pozytywnej liniowej funkcji aktywacji (ReLU). Te nazwy, których pierwsza z dwóch cyfr jest równa dwa, należą do sieci o sigmoidalnej funkcji aktywacji. Ostatnim z zaproponowanych w poprzednim podrozdziale typów sieci jest SSN zawierająca kolejno dwie warstwy z tangensem hiperbolicznym i funkcją liniową w ostatniej z nich.

Ponieważ sprawdzanych jest dziewięć modeli, podzielonych na trzy równe grupy, rozsądnym rozwiązaniem jest określić najpierw strategię, którą analityk będzie się kierował.

Rozważane będą dwa różne horyzonty czasowe dla każdej z sieci: jednosesyjny oraz dwudziestosesyjny. Powyższe warianty wybrane zostały subiektywnie.

Pierwszy z nich reprezentuje jedynie jednodniową prognozę. Jego szacunek oparty jest wyłącznie o rzeczywiste wartości notowań, przez co powinien być najmniej obciążony błędem. Przede wszystkim stworzone zostaną jednodniowe prognozy dla każdego z dziewięciu modeli, następnie wizualizując uzyskane efekty, uwzględniając ich absolutny błąd. W ten sposób, oceniona zostanie wstępna jakość modeli, dając ogólne poczucie o ich skuteczności. Zaraz po tym wyznaczone zostaną prognozy dla pięciu kolejnych jednodniowych okresów. Wraz z nimi, również wyznaczone będą takie miary jak średni absolutny błąd (MAE), średni absolutny błąd procentowy (MAPE), czy pierwiastek średniego błędu kwadratowego (RMSE). Wszystkie wyniki zestawione zostaną ze sobą wraz z aktualnymi wartościami dla odpowiadających sesji. By wyłonić sieć najlepszą z powstałych, przewidującą jednodniowe zmiany, użytych zostanie w sumie czterdzieści predykcji wykonanych kolejno dzień po dniu na następną sesję. Poprzez porównanie średnich błędów popełnianych przez sieci, wytypowana zostanie ta, która charakteryzuje się najmniejszymi ich wartościami z pośród reszty. Gdy wybrany już zostanie model o najwyższej skuteczności sprawdzona zostanie jego funkcja straty wygenerowana w trakcie procesu „nauki”, by lepiej rozumieć jego działanie oraz zyskać dodatkowe informacje dotyczące doboru jego parametrów. W celu uzasadnienia zastosowania SSN do predykcji jednodniowych zmian zestawiona zostanie jedna o najwyższej jakości wraz z rezultatami prognozy naiwnej oraz prognozy stosującej prostą średnią kroczącą.

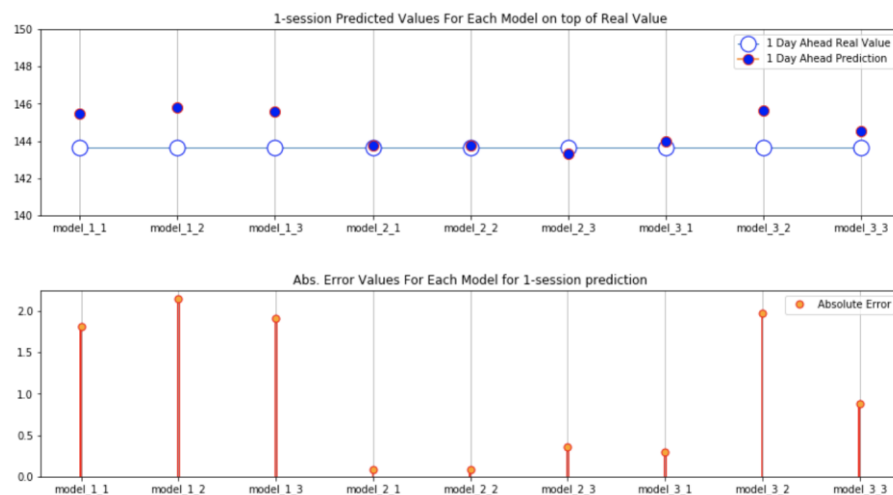
Drugą metodą sprawdzenia jakości sieci jest przeprowadzenie analizy w oparciu o dwudziestosesyjny horyzont inwestycyjny. Zadaniem modelu jest zbudować serię prognoz do dwudziestu dni w przyszłość. Ponieważ giełda NASDAQ działa jedynie od poniedziałku do piątku, wyłączając w ten sposób weekendy, dwudziestodniowa prognoza jest w zasadzie próbą SSN na przestrzeni całego miesiąca. Imponującym wynikiem będzie, gdy szacunki w tak dalekiej perspektywie okażą się satysfakcjonujące. Schemat działania będzie nieco podobny do zastosowanego w trakcie analizy pierwszego wariantu czasowego. Zatem przeprowadzona zostanie najpierw dwudziestodniowa prognoza używając w tym celu wszystkich modeli.

Pozwoli to ocenić wstępny zakres błędów ustalając początkowych faworytów. Następnie analogicznie do pierwszej metody, wyznaczone będą średnie błędy dla czterdziestu kolejnych, dwudziestosesyjnych prognoz. Dzięki tym statystykom, wybrany zostanie najlepszy z modeli w kategorii średniego horyzontu inwestycyjnego. Dla najefektywniejszej przeprowadzona będzie również analiza absolutnych błędów na przestrzeni dwudziestu sesji by potem porównać jego przebieg z prostą średnią kroczącą.

W najlepszym wypadku, wytyczony zostanie tylko jeden najskuteczniejszy model, ten sam dla obu kategorii, który będzie podstawowym komponentem w trakcie symulacji inwestycyjnej pod koniec tego rozdziału.

Zgodnie z zaplanowaną strategią, najpierw przeprowadzone zostają jednodniowe prognozy dla każdej sieci.

Rysunek 4.15. Porównanie jednodniowych prognoz wraz z błędami



Źródło: Opracowanie własne.

Powyższa grafika składa się z dwóch wykresów. Górny przedstawia jaki jest stosunek prognoz do rzeczywistej wartości. Widocznie najlepszą wydaje się być sieć numer 2.x zbudowana z sigmoidalnej funkcji aktywacji. Na drugim miejscu znajduje się model 3.x z tangensem hiperbolicznym w warstwach ukrytych. Najgorzej w tym wypadku poradziła sobie sieć numer 1.x. Jak widać w ramach każdego z trzech rodzajów prognozy są do siebie zbliżone. Zauważalna jest tendencja do przeszacowywania wybiegających o jedną sesję na przód cen zamknięcia. Najgorszy wynik odchylił się o nieco więcej niż \$2 w stosunku do realnej ceny wynoszącej około \$144 co daje 1.38% błędu. Najniższej jakości omawiany szacunek zaproponowany jest przez drugą iterację pierwszego typu sieci, złożonej jedynie z pozytywnych liniowych funkcji aktywacji (ReLU). W dolnej części rysunku 4.15 pokazana jest wielkość absolutnego błędu dla prognoz przedstawionych bezpośrednio nad nimi. W trakcie tej

próby najlepsze wydają się być ex aequo pierwsza i druga iteracja modelu drugiego z zawyżonym szacunkiem o około \$0.1, co jest imponującym wynikiem.

Pojedyncza prognoza nie może być podstawą rzetelnej oceny wyników przygotowanych algorytmów ze względu na możliwe losowe zniekształcenia, przeprowadzona zostanie bardziej złożona analiza uwzględniająca 40 kolejnych prognoz. Najpierw jednak by zapoznać się ze sposobem działania następnego etapu, poniżej przedstawiona zostaje tabela składająca się z pięciu jednosesyjnych prognoz kolejnych notowań wraz z miarami określającymi ich błędy.

Rysunek 4.16. Pięć jednosesyjnych prognozy wraz z ich średnimi błędami

	model_1_1	model_1_2	model_1_3	model_2_1	model_2_2	model_2_3	model_3_1	model_3_2	model_3_3	Actual Values
0	145.496222	145.823866	145.595707	143.767237	143.761152	143.319364	143.975688	145.653066	144.564211	143.679993
1	143.838460	143.286948	143.858251	143.340515	143.591998	142.716496	143.679100	143.481424	144.222898	144.020004
2	143.793448	143.683965	143.884369	143.000358	143.322018	142.369865	143.173683	143.655757	143.087367	143.500000
3	144.113286	142.639984	143.657292	143.534991	142.406794	141.610162	143.007877	143.673717	143.228248	144.089996
4	143.695710	144.723913	144.616493	143.539393	142.667174	141.782146	143.428643	143.788712	143.319170	142.729996
MAE_1	0.656045	1.300965	0.956207	0.526156	0.486634	1.244391	0.548737	0.828481	0.590134	0.000000
MAPE_1	0.457479	0.906524	0.667102	0.366594	0.337975	0.865750	0.382080	0.577282	0.410938	0.000000
RMSE_1	0.932835	1.499710	1.232069	0.580020	0.782122	1.425089	0.627571	1.048953	0.645447	0.000000
Mean_1	144.187425	144.031735	144.322422	143.436499	143.149827	142.359606	143.452998	144.050535	143.684379	143.603998
Std_1	0.668913	1.122806	0.715189	0.256557	0.526106	0.623496	0.346719	0.807265	0.593627	0.487796

Źródło: Opracowanie własne.

Rysunek 4.16 pokazuje działanie procesu, na którego podstawie wyznaczona zostanie najlepsza wśród stworzonych w poprzednim podrozdziale sztuczna sieć neuronowa w perspektywie krótkiego horyzontu inwestycyjnego. Nazwy kolumn reprezentują kolejno dziewięć modeli wraz z rzeczywistymi wartościami dla szacowanych okresów. Tabela składa się z dziesięciu rzędów. Pierwsze pozycje dla każdej SSN to ich prognozy na daną sesję. Uwzględnione zostały takie miary błędów jak MAE, MAPE, RMSE, wraz z dodanym w nazwie indeksem „_1” odnoszącym się do jedno-sesyjnej predykcji. Ich miary dla kolumny „Actual Values” wynoszą 0 odzwierciedlając ich idealne wielkości. Ostatnie dwa rzędy opisują odpowiednio średnią wartość proponowanych cen oraz to jak przeciętnie odchylają się one od średniej, dając wskazówki na temat rozproszenia. Jak wnioskować można ze średnich wartości błędów, sieci uznane wcześniej za najlepsze w oparciu o pojedynczą predykcję, wciąż plasują się w czołówce najskuteczniejszych. Najgorszym dotychczas jednoznacznie jest model „model_1_2”, który jak sądzić można po wielkości odchylenia standardowego (Std_1) charakteryzuje się największymi wahaniami wokół średniej, która w jego przypadku wynosi \$144.03. Oznacza to, że jego typowy obszar (Mean_1 +/- Std_1) określany jest przez \$144.03 +/- \$0.12, gdy w rzeczywistości wynosi on \$143.6 +/- \$0.49.

Jak widać większa ilość przeprowadzonych prognoz powinna pozwolić na wyższym poziomie pewności ocenić zwycięski model dla prognoz jednej sesji. Przeprowadzonych zostaje zatem czterdzieści prognoz analogicznie jak w przykładzie na rysunku 4.16.

Rysunek 4.17. Średnie błędy dla czterdziestu kolejnych jednosesyjnych prognoz

	model_1_1	model_1_2	model_1_3	model_2_1	model_2_2	model_2_3	model_3_1	model_3_2	model_3_3	Actual Values
ME	-0.192396	-0.619164	-0.329337	-1.715588	-1.925657	-2.208954	-1.496687	-0.709857	-0.881540	0.00000
MAE	1.357769	1.573945	1.482956	2.333516	2.388735	2.513182	1.872955	1.450672	1.545059	0.00000
MAPE	0.881933	1.026922	0.965531	1.512104	1.548537	1.634736	1.212220	0.941196	1.001185	0.00000
MSE	3.847335	4.234214	4.147071	7.993833	8.138724	8.978356	5.583746	3.872856	4.358480	0.00000
RMSE	1.961463	2.057720	2.036436	2.827337	2.852845	2.996391	2.362995	1.967957	2.087697	0.00000
Mean	152.297354	151.870587	152.160413	150.774163	150.564094	150.280796	150.993064	151.779893	151.608210	152.48975
Std	5.911487	5.889121	5.855412	5.549474	5.578064	5.620204	5.481779	5.589466	5.667771	5.91365

Źródło: Opracowanie własne.

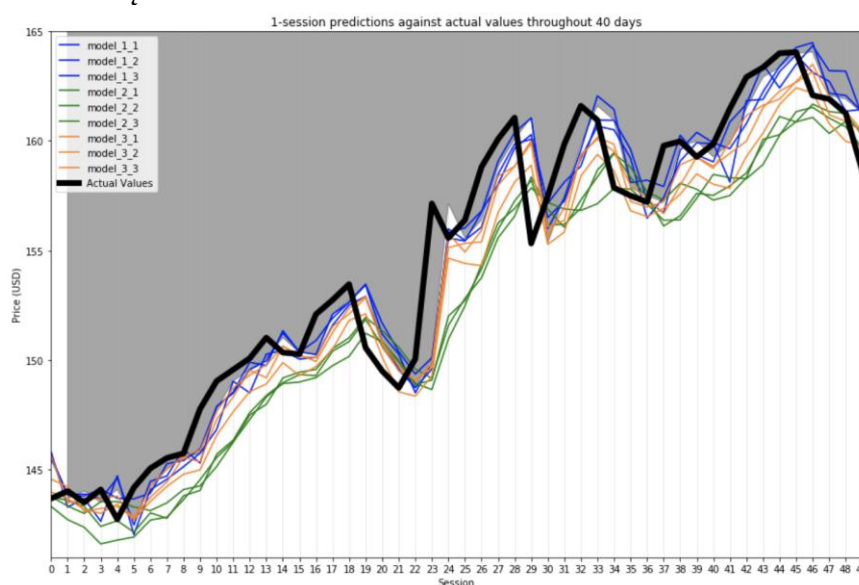
Powyższy rysunek prezentuje wybrane miary błędów wraz ze średnimi wielkościami cen oraz ich przeciętnymi zmianami. Ponieważ liczbową prezentacją wszystkich prognoz w tabeli zakłóciłaby odczyt wypisane zostały jedynie wyniki metod określające ich skuteczność. Z uwagi na istotność znaczenia informacji o tendencji do przeszacowania lub niedoszacowania prognoz przez model dodana została miara średniego błędu (ME). Ujemne wartości w pierwszym rzędzie tabeli na rysunku 4.17. wskazują na jednoznaczną preferencję do zaniżania notowań w każdym modelu. Absolutne średnie miary nie wskazują wyraźnie na jeden model, wyłaniając jednak trzy najbardziej skłonne do prawidłowych prognoz, czyli „model_1_1”, „model_1_3” oraz „model_3_2”. Wraz ze zwiększającą się liczbą prognoz zmieniają się kolejność najbardziej skutecznych modeli. W starciu ze średnimi pomyłkami na przestrzeni czterdziestu sesji model początkowo uważany za najlepszy, nie znajduje się już nawet w najlepszej trójce. Porównując wartości średnich błędów zdecydowanie najlepiej z nich radzi sobie model pierwszy, który niedoszacowuje jedynie na poziomie \$0.19. Dla porównania drugi z wyłonionych faworytów miarę ME ma na poziomie \$0.33 będąc prawie dwa razy gorszym. Istotne jest ustalić który z nich popełnia najmniej ponadprzeciętnych błędów. Ponieważ sama interpretacja parametru nie ma znaczenia w tym momencie, najlepszym wskaźnikiem będzie średni błąd kwadratowy (MSE). Z danych wynika, iż „model_1_3” sprawuje się najgorzej ze wszystkich trzech. Na bardzo porównywalnym poziomie są zarówno „model_1_1” i „model_3_2” z MSE odpowiednio 3.85 oraz 3.87. Zestawiając ze sobą jedynie te dwie sztuczne sieci neuronowe wszystkie miary wskazują na niekorzyść drugiego, pozostawiając „model_1_1” jako ostatni i najsukuteczniejszy z badanych. Zatem, znaczący wpływ na decyzje o odrzuceniu modelu „model_3_2” ma na pewno miara średniego błędu zdradzająca jego prawie czterokrotnie wyższą tendencję do niedoszacowania prognoz. Pierwsza w tabeli SSN jednoznacznie zostaje wybrana jako najsukuteczniejsza w oparciu o krótki, jedno-sesyjny horyzont inwestycyjny.

Słowem podsumowania pierwszej części badania, sieć o:

- dwudziestu neuronach wejściowych,
- dwóch warstwach ukrytych, po dwadzieścia neuronów w każdej,
- liniowej funkcji aktywacji w ostatniej powłoce z jednym neuronem wyjściowym,
- optymalizatorze stochastycznego gradientu prostego,
- funkcji straty średniego błędu kwadratowego,
- 500 epokach,

lepiej w jednosesyjnej prognozie cen zamknięcia firmy Apple poradzi sobie SSN o pozytywnej liniowej funkcji aktywacji w warstwach ukrytych niż sigmoidalnej czy tangensa hiperbolicznego.

Rysunek 4.18. Wykres pięćdziesięciu kolejnych jednodniowych prognoz wraz z ich rzeczywistą wartością



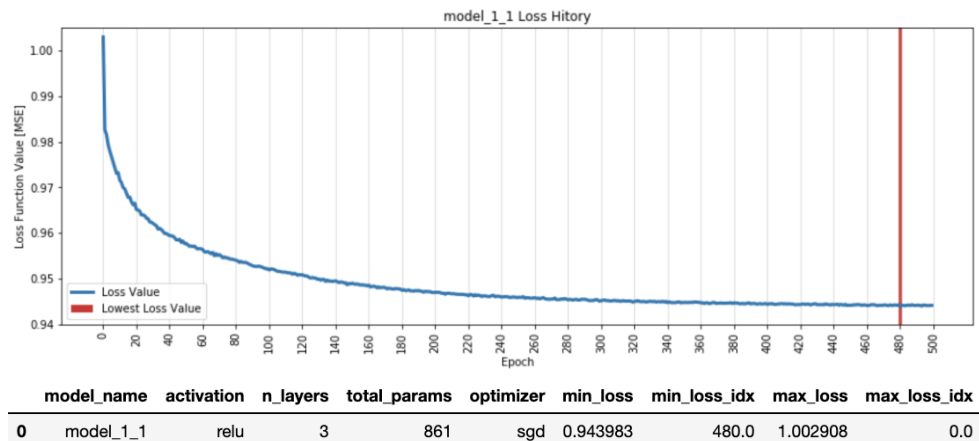
Źródło: Opracowanie własne.

Powyższy rysunek jest wizualizacją przebiegu analizowanych dotychczas prognoz dla wszystkich dziewięciu modeli. Sieci podzielone zostały kolorami ze względu na funkcje aktywacji zawarte w ukrytych warstwach. Jak widać niebieska linia wypracowana przez model 1.x najbliższej jest realnym wartościom zaznaczonych czarną linią na wykresie. Zielona sieć pretendująca z początku do miana najlepszej radzi sobie zauważalnie najgorzej. W danych zaistniała interesująca zależność, która również przedstawiona została na rysunku 4.18 Mianowicie, jeśli prosta reprezentująca empiryczne, rzeczywiste kursy cen zamknięcia przesunięta zostanie o jedną sesję w prawo wzdłuż osi x prawie idealnie dopasowuje się do prognoz zapewnionych przez pierwszy model. Ta sytuacja zobrażowana jest przez czarną-szarą płaszczyznę będącą dokładnie polem powierzchni nad wartościami rzeczywistymi. Jest ona reprezentacją prostej zaznaczonej kolorem czarnym, przesuniętej o jedną sesję w prawo. Dolna

granica czarno-szarej płaszczyzny przedstawia zatem prawdziwe ceny zamknięcia przesunięte jedynie w czasie. Widocznie lepiej prognozy modelu dopasowane są do jej zmodyfikowanego o jedną sesję pola niż do jej pierwotnego położenia, praktycznie dokładnie odwzorowując jej kształt.

Zgodnie z wyznaczoną strategią analizy modeli sztucznych sieci neuronowych, następnym krokiem po wybraniu tej, która wśród zbudowanych charakteryzuje się największą skutecznością jednosesyjnej prognozy, jest analiza działania wartości jej funkcji straty.

Rysunek 4.19. Wielkości funkcji straty dla modelu „model_1_1” na przestrzeni 500 epok



Źródło: Opracowanie własne.

Rysunek 4.19 jest wizualizacją poziomu średniego błędu kwadratowego (MSE) sieci „model_1_1” w trakcie trwania całego jej procesu „nauki”. Jak widać, maksymalna wartość (1.002908) osiągnięta została w zerowej epoce. Ponieważ wartość funkcji straty, determinowana jest przez wielkości wag sieci, które pierwszej kolejności najczęściej wybierane są w sposób losowy, taki stan rzeczy jest zrozumiały i często spotykany. Dopiero gdy ustalony zostanie punkt wyjściowy, optymalizator zyskuje punkt odniesienia, wykorzystywany do obliczania kolejnych, lepszych kombinacji parametrów tak, by z każdą kolejną epoką poziom błędu spadał. Największe postępy model popełniał w ciągu pierwszych 200 epok, by później fluktuując pomiędzy wartościami 0.94 i 0.95 osiągnąć minimalną wartość błędu (0.943983) w 480. epoce, co zaznaczone zostało czerwoną linią pionową. Dodatkowo u podstawy rysunku 4.19 widnieje tabela mówiąca o nazwie modelu, typach jego funkcji aktywacji w ukrytych warstwach, ich liczbie¹⁶⁰, sumie wszystkich neuronów, typie optymalizatora oraz o skrajnych wartościach funkcji straty.

Po wyłonieniu zwycięskiej SSN w kategorii krótkiego horyzontu inwestycyjnego, wartościowe będzie porównanie jej skuteczności z innymi znacznie prostszymi metodami

¹⁶⁰ Parametr „n_layers” = 3 ponieważ uwzględnia on dwie warstwy ukryte oraz jedną warstwę wyjściową.

prognozowania. Wykonanie zaproponowanego zestawienia pozwoli odpowiedzieć na pytanie o zasadność stosowania tak złożonych rozwiązań jak sztuczna sieć neuronowa.

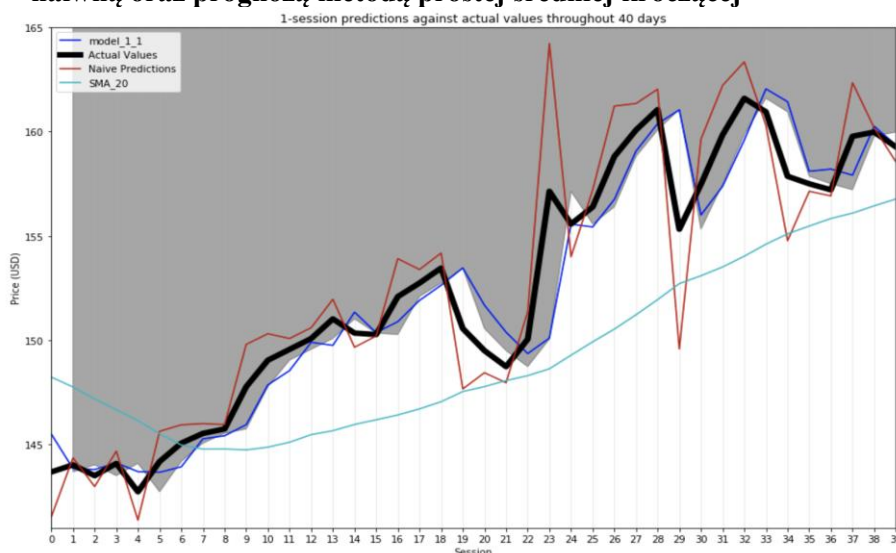
Rysunek 4.20. Wartości błędów czterdziestu kolejnych jednodniowych sesji modelu „model_1_1” wraz z prognozą naiwną oraz prognozą metodą prostej średniej kroczącej

	model_1_1	Naive Predictions	SMA_20	Actual Values
MAE_1	1.357769	1.432000	4.178326	0.00000
MAPE_1	0.881933	0.931483	2.711159	0.00000
RMSE_1	1.961463	1.997227	4.799065	0.00000
Mean_1	152.297354	152.825750	149.275600	152.48975
Std_1	5.911487	6.548029	3.900879	5.91365

Źródło: Opracowanie własne.

Rysunek 4.20 zawiera wyniki wybranych miar błędów dla czterdziestu kolejnych jednodniowych prognoz. Na pierwszy rzut oka średnia krocząca, znacząco wyróżnia się wielkościami popełnionych błędów. Już jedynie porównanie średniego błędu absolutnego daje podstawy by odrzucić SMA ze względu blisko trzy razy większe przeciętne błędy. Model SSN w porównaniu do prognoz naiwnych radzi jest nieco skuteczniejszy, np. popełniając mniejszy błąd MAPE wskazujący na skromniejszą liczbę popełnianych ponadprzeciętnych pomyłek.

Rysunek 4.21. Wykres czterdziestu kolejnych prognoz modelu „model_1_1” wraz z prognozą naiwną oraz prognozą metodą prostej średniej kroczącej



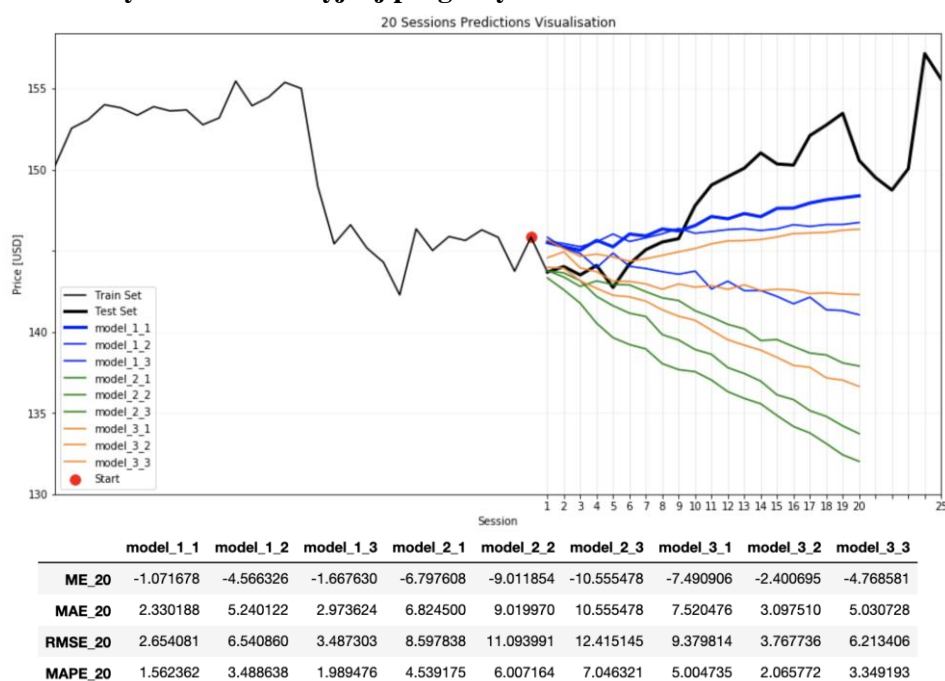
Źródło: Opracowanie własne

Powyższy wykres przedstawiony na rysunku 4.21 jest wizualizacją wartości zamieszczonych na rysunku 4.20 procesu przeprowadzanych jednodniowych prognoz. Analogicznie do wykresu na rysunku 4.18 zestawione zostały z rzeczywistymi empirycznymi wartościami zaznaczonymi czarnym kolorem. Jak wskazywały na to miary błędów, prosta średnia krocząca opisana kolorem niebieskim radzi sobie najgorzej. Źródła różnicy między

wynikami błędu MSE, dla SSN i naiwnej prognozy są widoczne. Wyniki prognoz naiwnych charakteryzują się większą ilością znaczących błędów. Zaznaczona jest również niepokojąca zależność pomiędzy przesuniętymi w czasie wartościami prawdziwymi a prognozami modelu sieci neuronowej.

Drugim analizowanym horyzontem czasowym jest prognoza dwudziestu przyszłych sesji. W tym wypadku oczekiwane miary błędów są znacznie wyższe ze względu na specyfikę powstawania kolejnych prognoz. W poprzednim wypadku każda prognoza całkowicie oparta była o rzeczywiste ceny zamknięcia. Gdy budowana jest predykcja więcej niż jednej sesji kolejne wartości opierają się na coraz większej ilości przewidzianych już, nieco błędnych notowań, co skutkuje najczęściej zwiększającą się z każdą kolejną sesją wielkością oczekiwanego błędu.

Rysunek 4.22. Efekty dwudziestosesyjnej prognozy SSN



Źródło: Opracowanie własne.

Rysunek 4.22 jest graficzną wizualizacją przebiegu procesu prognozowania dwudziestu przyszłych sesji przez każdą z opracowywanych sztucznych sieci neuronowych. Cienką linią w kolorze czarnym przedstawiono ostatnich kilkadziesiąt wartości, na których modele się „uczyły”, natomiast pogrubioną, część zbioru testowego. Odpowiednio barwami niebieską, zieloną i pomarańczową oznaczono predykcje modeli 1.x (ReLU), 2.x (Sigmoid) i 3.x. (Tanh). Dodatkowo pogrubiona została linia reprezentująca sieć „model_1_1”, której wartości dotychczas najbliższe są rzeczywistym. Na wykresie zauważalna jest tendencja do niedoszacowania co potwierdzają wartości średniego błędu (ME_20) przedstawione w

pierwszym rzędzie tabeli. Znacząca jest rozbieżność między prognozami modeli oraz jak przypuszczano jakość modeli spada wraz ze zwiększaniem się horyzontu czasowego. Najgorszy z szacunków jest prawie pięciokrotnie gorszy od najlepszego porównując ich średnie absolutne błędy. Traktując modele grupami kolorów, zdecydowanie najskuteczniejsze w tej próbie są modele 1.x, obejmując pierwsze dwie pozycje w kategorii najlepszych sieci. Na drugim miejscu jako rodzaj sieci plasuje się jeden z trzeciej grupy. Najslabsze efekty pokazuje pomarańczowy, z sigmoidalnymi funkcjami aktywacji.

By zmniejszyć prawdopodobieństwo wystąpień losowych wpływających na wyniki, przeprowadzono 40, dwudziestosesyjnych prognoz i obliczono ich średnie miary uśrednionych błędów. Wyniki przedstawione są w poniższej tabeli.

Rysunek 4.23. Miary błędów dla czterdziestu prognoz dwudziestosesyjnych

	model_1_1	model_1_2	model_1_3	model_2_1	model_2_2	model_2_3	model_3_1	model_3_2	model_3_3	Real Values
ME_20	-1.852234	-6.397569	-3.430177	-9.433626	-11.511711	-11.707941	-11.291983	-5.136537	-7.001502	0.000000
MAE_20	3.263919	6.649052	4.133390	9.529556	11.570170	11.743708	11.327897	5.290613	7.133974	0.000000
MSE_20	16.410836	63.463132	26.520395	118.431557	172.241088	174.660829	162.969712	37.085877	66.147639	0.000000
RMSE_20	3.805536	7.579296	4.782804	10.778049	13.057429	13.162985	12.684274	5.976994	8.057230	0.000000
MAPE_20	2.083704	4.234974	2.634886	6.048415	7.348549	7.465544	7.182767	3.350670	4.523337	0.000000
Mean_20	154.356167	149.810831	152.778224	146.774775	144.696690	144.500459	144.916418	151.071864	149.206899	154.967167
Std_20	1.266020	1.309389	0.381502	2.733277	3.795579	3.711726	3.343731	0.725711	1.402418	6.238761

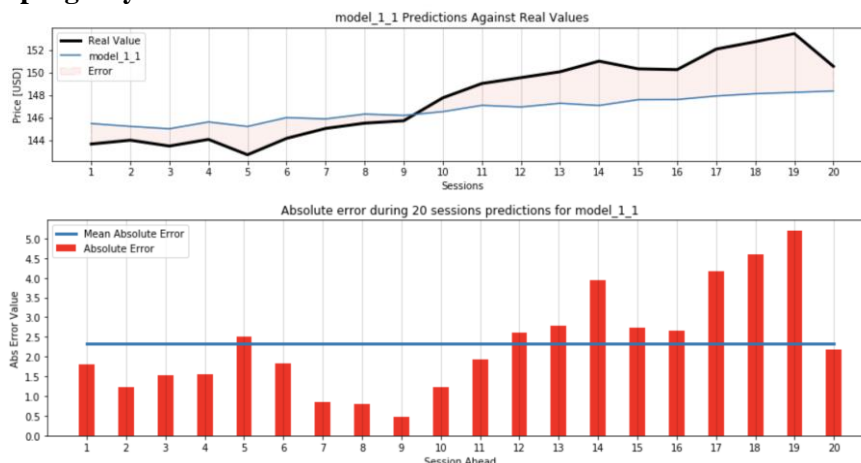
Źródło: Opracowanie własne.

Jak wynika z rysunku 4.23, sztuczna sieć neuronowa o nazwie „model_1_1” wciąż wyznacza najbliższe prawdzie prognozy, osiągając zdecydowanie najniższe miary błędów. Na przestrzeni czterdziestu kolejnych, dwudziestosesyjnych prognoz wciąż zauważalna jest tendencja do zaniżania notowań cen zamknięcia, o czym świadczą ujemne wartości ME_20. Wnioskując po odchyleniu standardowym, zmienność wszystkich SSN jest zdecydowanie zbyt niska w porównaniu do wyników rzeczywistych, których miary przedstawiono w ostatniej kolumnie. Świadczy to o znacznym spłaszczeniu linii prognozy, co widoczne jest na wykresie, wskazując na nieumiejętność uwzględniania silnych wahań. Zatem pierwsza sieć, o liniowych pozytywnych funkcjach aktywacji we wszystkich dwóch warstwach ukrytych i liniowej w warstwie wyjściowej radzi sobie najlepiej wśród pozostałych ośmiu, myśląc się przeciętnie o około 2% w stosunku do wartości rzeczywistej. Uzasadnione będzie stwierdzenie, że model charakteryzowany funkcjami ReLU, jest najskuteczniejszy w ogóle, uzyskując najlepsze wyniki również jako grupa.

Jak już wspomniano głównym problemem prognozy o więcej niż jedną sesję w przód jest fakt, że kolejne przypuszczenia dotyczące kursów oparte są o już obarczone błędem wcześniejsze prognozy. Wynika to z budowy sieci, która to opiera swój wynik reprezentowany przez jeden neuron, dwudziestoma historycznymi obserwacjami. Ponieważ modele zbudowanych sieci, nie są w stanie zwrócić więcej niż jednego wyniku na raz, w celu prognozy

dwudziestodniowej, zmuszone są w stanie powtarzać swoje obliczenia dwadzieścia razy, kolejno dokładając pojedynczą prognozę do zbioru, w oparciu o który przeprowadzać będzie następną.

Rysunek 4.24. Absolutny błąd modelu „model_1_1” w trakcie pierwszej dwudziestosesyjnej prognozy

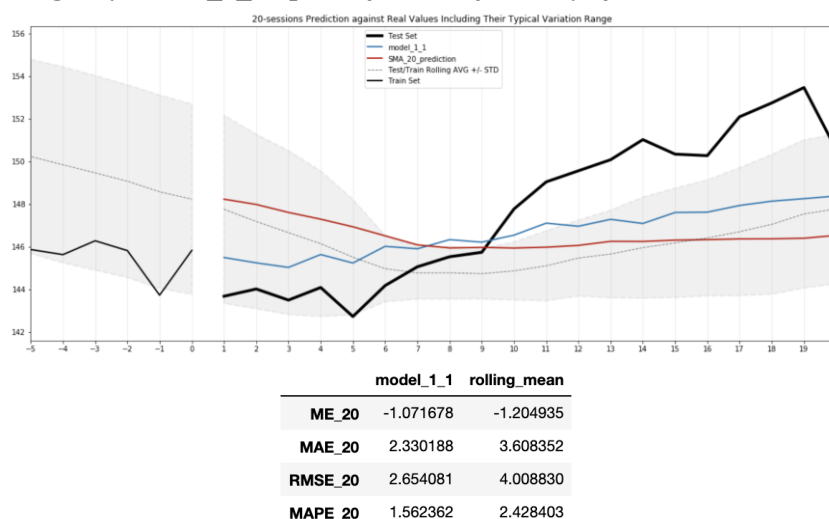


Źródło: Opracowanie własne.

W dolnej części rysunku 4.24 na wykresie słupkowym widać wzrostową tendencję absolutnych błędów prognoz modelu „model_1_1”.

Ostatnim krokiem w trakcie oceny modelu jest jego zestawienie z inną metodą prognozowania, która nosi nazwę prostej średniej kroczącej.

Rysunek 4.25. Prognozy model_1_1 i prostej średniej kroczącej



Źródło: Opracowanie własne.

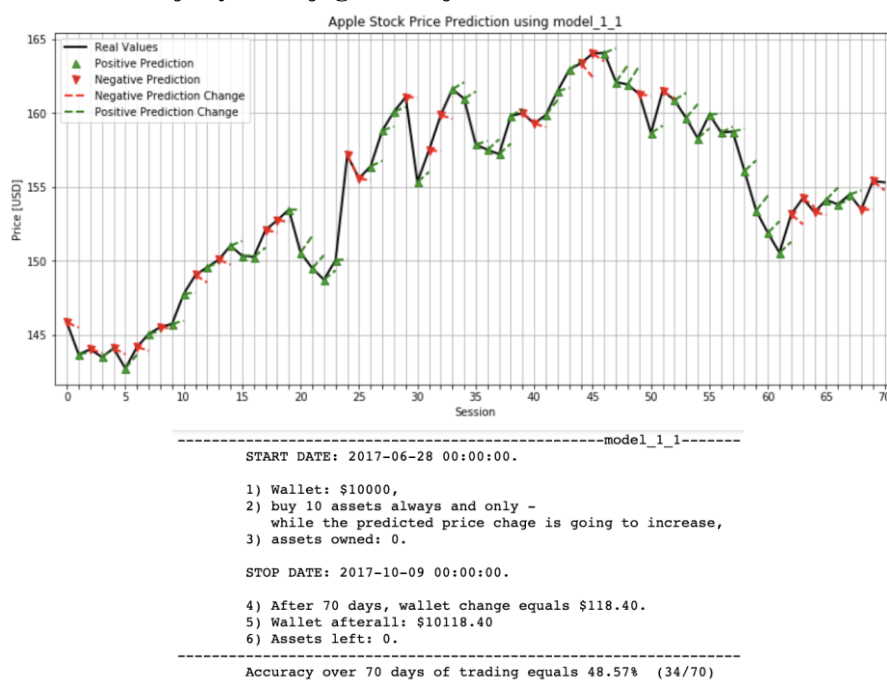
Na rysunku 4.25 zwizualizowany został efekt działania dwóch metod prognostycznych dla dwudziestosesyjnego horyzontu prognostycznego. Przedstawione zostały razem z rzeczywistymi wartościami notowań i ich typowym przedziałem zmienności. Powstały „tunel” to pole, reprezentujące obszar wokół średniej ruchomej dla prawdziwych cen zamknięcia firmy

Apple o szerokości dwóch odchyłeń standardowych. Jak widać na wykresie praktycznie wszystkie wartości zarówno jednej, jak i drugiej metody budowy prognoz nie wychodzą poza typowy obszar szacowanych cen zamknięcia. Oba z wymienionych sposobów mają tendencję do zaniżania, co niewyraźnie widoczne jest na wykresie i może być wynikiem losowego zdarzenia związanego z tą konkretną próbką. Dodatkowo średni błąd kwadratowy wraz z średnim absolutnym błędem procentowym wskazują przewagę SSN nad prostą średnią kroczącą.

By sprawdzić działanie modelu w praktyce, przeprowadzona zostanie prosta symulacja giełdowa, której celem jest obserwacja skuteczności prognoz w praktyce. Egzekwowane będą jednosesyjne predykcje będące podstawą decyzji na temat transakcji kupna-sprzedaży w trakcie siedemdziesięciu sesji. Z uwagi na dużą ilość zasad i zależności panujących na realnej giełdzie, podczas kreowania jej pozornego odpowiednika wprowadzone zostały pewne założenia, upraszczające przebieg badania. Reguły wykorzystane w symulacji prezentują się następująco:

- podmiot zaczyna z portfelem o wartości \$10 000,
- jedynie, jeśli prognoza na następny dzień zapowiada wzrost dokonywany jest zakup,
- transakcje przeprowadzane są zawsze dziesięcioma akcjami,
- każdego dnia po zakupie, program sprzedaje nabyte dnia poprzedniego akcje realizując zysk lub stratę,
- cena waloru zmienia się jedynie raz w ciągu dnia.

Rysunek 4.26. Wizualizacja symulacji giełdowej



Źródło: Opracowanie własne.

Rysunek 4.26 przedstawia przebieg przeprowadzonej symulacji giełdowej w okresie siedemdziesięciu sesji począwszy od dnia 28.06.17 opartej o notowania cen zamknięcia firmy Apple. Na wykresie zaznaczone są zielonymi i czerwonymi strzałkami jednodniowe prognozy odpowiednio wzrostów oraz spadków. Przerywaną linią uwzględniono wielkości ich zmian wraz z kierunkiem. Jak wynika z założeń, każdy punkt oznaczony kolorem zielonym, odpowiada sesji w trakcie, której dokonano zakupu pakietu 10 akcji by sprzedać tę samą ilość następnego dnia. Z opisu przebiegu symulacji, zawartej w dolnej części rysunku 4.26, wynika, iż po zakończonych w dniu 09.10.17 siedemdziesięciu sesjach, wartość portfela wzrosła o \$118.4. W trakcie trwania tego okresu program dokonał trafnej decyzji o zakupie 34 razy z pośród 70 prób co oznacza około 50% skuteczności.

Z powyższych analiz, wynika, że sztuczna sieć neuronowa może zostać wykorzystana do wspierania decyzji giełdowych. Nie jest zaskoczeniem, iż zwiększając horyzont prognoz, również zmniejsza się skuteczność zbudowanego modelu. W przypadku inwestycji o charakterze spekulacyjnym, zaprezentowany model określony mianem najskuteczniejszego może istotnie przyczynić się do sukcesu inwestycyjnego. Widoczny jest potencjał zwiększenia jakości modeli poprzez uwzględnienie zaobserwowanego zjawiska „przesuniętych” wartości w czasie. Kolejnym etapem badań może być obserwacja zmiany skuteczności modelu zwiększając liczbę jego warstw ukrytych lub manipulując ilością neuronów każdej z nich. Możliwości dalszych usprawnień SSN są dziesiątki i jedynie od poziomu zaangażowania i wytrwałości analityka zależy ich dalsza eksploracja.

Zakończenie

Celem niniejszej pracy było zweryfikowanie hipotezy o możliwości wspierania decyzji inwestycyjnych na giełdzie papierów wartościowych poprzez zastosowanie prognoz uzyskanych przy pomocy sztucznych sieci neuronowych. Tak sformułowany problem badawczy, jeszcze do niedawna z pewnością wprowiłby jedynie w stan niedowierzania z góry odrzucając szanse na powodzenie. Ponieważ, na przestrzeni ostatnich lat, dziedziny związane ze sztuczną inteligencją rozwinęły się w tak znaczący sposób, że dziś realnie brane są pod uwagę ogromne potencjalne korzyści, które mogą być wynikiem w razie powodzenia.

W strukturze pracy wyróżnia cztery części:

- wprowadzenie do uczenia maszynowego i sztucznych sieci neuronowych,
- wyjaśnienie działania rynków finansowych, a w szczególności rynku kapitałowego,
- opis narzędzi stosowanych w badaniu,
- przeprowadzenie analizy wraz z jej oceną.

Pierwsze strony pracy zawierają wyjaśnienia najważniejszych etapów na ścieżce do znanej dzisiaj powszechnie sztucznej inteligencji. Dzięki zapoznaniu się z podstawowymi elementami prowadzącymi do takich dziedzin jak uczenie maszynowe i sztuczne sieci neuronowe, łatwiej jest zrozumieć pewne ograniczenia omawianych, które w dużej mierze wynikają również z ich krótkiej historii. Po ogólnym wprowadzeniu do zagadnień, w którym maszyny powoli zyskują umiejętności „uczenia się”, następuje zapoznanie się z głównym narzędziem pracy, czyli sztuczną siecią neuronową. W efekcie ma to pozwolić na w pełni świadomie dobieranie później najbardziej pożądanych parametrów w części praktycznej.

Znając rys historyczny omawianych zdarzeń wraz z ich podstawowymi rodzajami, ważne z perspektywy omawianego tematu jest również poznanie zasad i zależności panujących w świecie finansów. Przede wszystkim, by zrozumieć ich działanie oraz strukturę, wprowadzony zostaje generalny opis rynków finansowych w ogóle, kolejno skupiając się już jedynie na coraz większych szczegółach rynku kapitałowego. Dzięki zastosowaniu takiej strategii, orientując się wśród głównych elementów dziedziny jaką jest rynek finansowy, znacznie łatwiej jest zrozumieć działanie rynków kapitałowych, a w szczególności jednego z ich głównych elementów, czyli giełdy papierów wartościowych.

W ostatnim rozdziale przygotowującym, do świadomego uczestniczenia w trakcie badania skuteczności prognoz uzyskanych w wyniku zastosowania sztucznych sieci neuronowych na rynku wtórnym akcji, wprowadzone i opisane są pojęcia związane z narzędziami, które wykorzystane zostały w trakcie badania. Wyróżnione są tutaj pojęcie języka programowania Python, jego komponenty jak również kryteria oceny skuteczności uzyskanych efektów. Ważne jest, by po przeczytaniu tego podrozdziału, miary błędów stosowane w analizie były

zrozumiałe, co znacznie lepiej wpłynie na ostateczny odbiór badania, przede wszystkim na nich opierającego swoje wnioski.

W momencie, gdy nie są już obce pojęcia z dziedziny sztucznej inteligencji, rynków finansowych oraz związane z programowaniem narzędzia, następuje bezpośrednie przejście do praktycznego rozdziału zawierającego esencję badania postawionej hipotezy. Z uwagi na ograniczenia nakładane przez wytyczne dotyczące niniejszej pracy, jedynie bardzo niewielka część wszystkich możliwości została wykorzystana by móc ustosunkować się do postawionego zagadnienia. Prognozy sztucznych sieci neuronowych przebadane zostały jedynie przez pryzmat ich głównych, dobranych arbitralnie czynników. Co więcej, spośród bliżej nieokreślonej ilości możliwych horyzontów czasowych sprawdzone zostały jedynie dwa z nich, odpowiednio jednosesyjny i dwudziestosesyjny. Mimo analizy jedynie wierzchołka całej góry lodowej, zastosowanie SSN na giełdzie papierów wartościowych rokuje dobrze, co wnioskować można chociażby po zysku wypracowanym w trakcie prostej symulacji giełdowej nawet przy skuteczności na poziomie 48.5%. Pomimo skromnych możliwości, osiągnięte efekty są imponujące i napawają nadzieją na dalsze sukcesy.

Poświęcając dalej uwagę poniższemu problemowi prognoz uzyskanych w wyniku zastosowania sieci neuronowych, istotna jest również poza typem funkcji aktywacji manipulacja innymi głównymi komponentami jak na przykład:

- ilość neuronów w warstwach – oprócz zmian ich liczby w warstwach ukrytych, również warto byłoby sprawdzić efekty zmieniając wielkość warstwy wejściowej jak i wyjściowej;
- rodzaj połączeń między warstwami - z pewnością interesujące byłoby wykorzystanie choćby omawianych w pierwszym rozdziale sieci rekurencyjnej;
- ilość warstw ukrytych.

Zatem, nawet przy tak ograniczonym zakresie badawczym pracy, przy sprawdzeniu tylko jednej z możliwych manipulacji struktury sztucznej sieci neuronowej, przeprowadzone próby szczególnie jednosesyjne, w satysfakcjonujący sposób pomóc mogą w podejmowaniu decyzji. Ciekawe jak rewizja pozostałych składników budowy SSN wpłynęłyby na jakość ich prognoz za każdym razem, przy zastosowaniu najbardziej optymalnej i dopasowanej do problemu struktury.

Bibliografia

1. Ampuła D., Proces predykcji zdarzeń i zjawisk, Wojskowy Instytut Techniczny Uzbrojenia, Warszawa 2012.
2. Apanowicz J., Metodologia ogólna, Wydawnictwo Diecezji IV Pelplińskiej, Gdynia 2002.
3. Banaszczyk-Soroka U., Instytucje i uczestnicy rynku kapitałowego, Wydawnictwo Naukowe PWN, Warszawa 2008.
4. Bell J., *Machine learning. Hands-on for Developers and Technical Professionals*, Wiley, Indianapolis 2015.
5. Biliski J., Dorożyńska A. (red.), Wybrane problemy w gospodarce światowej, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2008.
6. Błaszczuk D. J., Podstawy prognozowania, symulacji i sterowania optymalnego, PWN, Warszawa 2014.
7. Bojańczyk M., Regresja i korelacja na światowych rynkach – w pułapce metod ilościowych, Oficyna Wydawnicza SGH, Warszawa 2013.
8. Chollet F., Deep learning. Praca z językiem Python i biblioteką Keras, Helion, Gliwice 2018.
9. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, Cambridge 2006.
10. Cichosz P., Systemy uczące się, Wydawnictwo Naukowo-Techniczne, Warszawa 2000.
11. Dawson M., Python dla każdego. Podstawy programowania, Helion, Gliwice 2014.
12. Dębski W., Rynek finansowy i jego mechanizmy, Wydawnictwo Naukowe PWN, Warszawa 2010.
13. Dorosz A., Puławski M., Giełdy w gospodarce światowej, PWE, Warszawa 1991.
14. Flasiński M., Wstęp do sztucznej inteligencji, Polskie Wydawnictwo Naukowe PWN, Warszawa 2011.
15. Gajda J. B., Prognozowanie i symulacje w ekonomii i zarządzaniu, C. H. Beck, Warszawa 2017.
16. Galant M., Dolan B., Forex dla bystrzaków, Helion, Gliwice 2010.
17. Geron A., Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly, Sebastopol 2019.
18. Gogołek W., Informatyka dla humanistów, Wydawnictwo Kropki Trzy, Warszawa 2012.
19. Górski M., Rynkowy system finansowy, Polskie Wydawnictwo Ekonomiczne, Warszawa 2009.
20. Hagan M. T. [i. in.], Neural Network Design, Martin Hagan, Stwillwater 2014.
21. Hearty J., Zaawansowane uczenie maszynowe z językiem Python, Helion, Gliwice 2017.
22. Jagielnicki A., Inwestycje giełdowe. Jak grać i wygrywać, Helion, Gliwice 2007.
23. Kachniewski M., Majewski B., Wasilewski P., Rynek kapitałowy i giełda papierów wartościowych, Fundacja Edukacji i Rynku Kapitałowego, Warszawa 2008.
24. Kaczmarek-Templin B., Web 2.0 – treści cyfrowe a wirtualne światy gier. Wybrane problemy, WSS, Warszawa 2014.
25. Kasperski M., Sztuczna Inteligencja, Helion, Gliwice 2003.

26. Kelleher J. D., Namee B. M., D'Arcy A., Fundamentals of Machine Learning for Predictive Data Analytics, The MIT Press, London 2015.
27. Kisielewicz A., Sztuczna inteligencja i logika, Wydawnictwo WNT, Warszawa 2015.
28. Maddala G. S., Ekonometria, Wydawnictwo Naukowe PWN, Warszawa 2008.
29. Makać W., D. Urbanek-Krzysztofiak, Metody opisu statystycznego, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2010.
30. Malkiel B. G., A Random Walk down Wall Street, W.W Norton, New York 1996.
31. McKinney W., Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i Numpy oraz środowiska Ipython, Helion, Gliwice 2018.
32. Najman K. M., Najman K., Samouczące się sztuczne sieci neuronowe w grupowaniu klasyfikacji danych, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2013.
33. Najman K., Sztuczne sieci neuronowe w prognozowaniu dynamicznych szeregów czasowych na przykładzie Giełdy Papierów Wartościowych w Warszawie, Uniwersytet Gdański, Sopot 2003. (praca doktorska).
34. Patterson J., Gibson A., Deep learning. Praktyczne wprowadzenie, Helion, Gliwice 2017.
35. Pułaska-Turyna B., Statystyka dla ekonomistów, Difin, Warszawa 2005.
36. Raschka S., Python. Uczenie maszynowe, Helion, Gliwice 2018.
37. Ryłski M., Forex – włącz emocje, wyłącz zyski. Przewodnik świadomego inwestora, Helion, Gliwice 2009.
38. Salmonowicz H. (red.), Koniunktura w gospodarce światowej a rynki żeglugowe i portowe, Kreos, Szczecin 2009.
39. Samuel A. L., Some studies in machine learning using the game of checker, IBM Journal of Research and Development, New York 1959.
40. Sławiński A., Rynki finansowe, Polskie Wydawnictwo Ekonomiczne, Warszawa 2006.
41. Szeliga M., Data Science i uczenie maszynowe, Wydawnictwo Naukowe PWN, Warszawa 2017.
42. Śnieżyński B., Uczenie się pojęć przez agentów jako metoda generowania strategii, Wydawnictwa AGH, Kraków 2013.
43. Thiel S., Rynek kapitałowy i terminowy, Komisja Nadzoru Finansowego, Warszawa 2010.
44. Wrycza S., Maślankowski J.(red.), Informatyka ekonomiczna: Teoria i zastosowania, Wydawnictwo Naukowe PWN, Warszawa 2019.
45. Zalewski G., Inwestowanie w akcje, czyli jak skorzystać ze wzrostu gospodarczego, GPW S.A., Warszawa 2018.
46. Zaremba A., Giełda. Podstawy inwestowania, Helion, Gliwice 2010.
47. Zawada J., Wybrane zagadnienia z podstaw metrologii, Wydawnictwo Politechniki Łódzkiej, Łódź 2002.
48. Zawila-Niedźwiecki J., Rostek K., Gąsioriewicz A., (red.), Informatyka gospodarcza, Wydawnictwo C.H. Beck, Warszawa 2010.
49. Zocca V., Spacagna G., Slater D., Roelants P., Deep learning. Uczenie głębokie z językiem Python, Helion, Gliwice 2017.

Netografia

1. Burgess R., Apple, US\$1 trillion and stock market bubbles, <https://www.thestar.com.my/business/business-news/2018/08/06/apple-us1-trillion-and-stock-market-bubbles/>, dostęp dnia 04.06.19.
2. Cuthbert A., 8 top programming languages & frameworks of 2019, hackernoon.com/8-top-programming-languages-frameworks-of-2019-2f08d2d21a1, dostęp dnia 02.06.19
3. Dane rynkowe, gpw.pl/dane-rynkowe, dostęp dnia 01.07.19.
4. Data mining tasks – data mining, <https://www.wisdomjobs.com/e-university/data-mining-tutorial-199/data-mining-tasks-1871.html>, dostęp dnia 11.05.19.
5. Dryjańska P., Data Science, https://mfiles.pl/pl/index.php/Data_science, dostęp dnia 20.06.19.
6. Dude S., Curse of dimensionality, <https://bigsnarf.wordpress.com/2013/06/14/curse-of-dimensionality/>, dostęp dnia 28.06.19
7. Dwivedi V., Top 10 programming languages of the world – 2019 to begin with..., www.geeksforgeeks.org/top-10-programming-languages-of-the-world-2019-to-begin-with/, dostęp dnia 02.06.19.
8. Dżuryk A., Rynki finansowe, [http://wzr.pl/read.php?plik=Rynki%20finansowe%20\[w\]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok](http://wzr.pl/read.php?plik=Rynki%20finansowe%20[w]%20-%20Andrzej%20D%C5%BCuryk.pdf&sum=675&from=dok), dostęp dnia 13.06.19.
9. Genc O., Notes on Artificial Intelligence, Machine Learning and Deep Learning for curious people, <https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2>, dostęp dnia 03.04.19.
10. Gupta A., Revisiting the Unreasonable Effectiveness of Data, <https://ai.googleblog.com/2017/07/revisiting-unreasonable-effectiveness.html>, dostęp dnia 02.07.19.
11. Harasymczuk M., Biblioteki w zastosowaniach naukowych i inżynierskich, <http://python.astrotech.io/numerical-analysis/libraries.html>, dostęp dnia 05.04.19.
12. Hyoung U. H., Jong-Kook K., An evolutionary genetic neural network for problems without prior knowledge, Table 1, <https://www.semanticscholar.org/paper/An-evolutionary-genetic-neural-networks-for-without-Ha-Kim/79fb50f1aad2b4d6f3ff43129540ea37b30f9829/figure/6>, dostęp dnia 01.07.19.
13. IVlad, Linear Regression: Normalization Vs Standardization, <https://stackoverflow.com/questions/32108179/linear-regression-normalization-vs-standardization>, dostęp dnia 15.04.19.
14. Kawaguchi K., The McCulloch-Pitts Model of Neuron, <http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node12.html>, dostęp dnia 01.07.19.
15. Lis M., Apple pierwszą w historii Wall Street spółką o takiej wartości. Przebiła bilion dolarów, <https://www.money.pl/gospodarka/wiadomosci/artykul/apple-pierwsza-w-historii-wall-street-spolka,244,0,2412532.html>, dostęp dnia 01.07.19.
16. Mohhamadi M., A conceptual structure of a deep reinforcement learning system, https://www.researchgate.net/figure/A-conceptual-structure-of-a-deep-reinforcement-learning-system_fig3_319121340, dostęp dnia 23.04.19.
17. Pełka W., Finanse przedsiębiorstwa, <http://www.wseiz.pl/files/materialy/Finanse3.ppt>, dostęp dnia 03.07.19.

18. Prognozowanie naiwne, <http://www.prognozowanie.info/prognozowanie-naiwne/>, dostęp dnia 07.06.19.
19. Python Software Foundation, The Python Logo, <https://www.python.org/community/logos/>, dostęp dnia 11.05.19.
20. Python Software Foundation, The Python Standard Library, <https://docs.python.org/3/library/>, dostęp dnia 18.05.19.
21. Rynek finansowy w Polsce, Giełda Papierów Wartościowych, https://www.gpw.pl/pub/GPW/files/Rynek_finansowy_w_Polsce.pdf, dostęp dnia 01.07.19.
22. Saeed A., Here are the ten best programming languages to learn in 2019, codinginfinite.com/best-programming-languages-to-learn-2019/, dostęp dnia 02.06.19
23. Scikit-learn developers, 5.3. Preprocessing data. <https://scikit-learn.org/stable/modules/preprocessing.html>, dostęp dnia 02.06.19.
24. Setlak D., Aktywa i pasywa - definicja, przykłady i ich rodzaje, <https://6krokow.pl/aktywa-i-pasywa-definicja-przyklady-i-ich-rodzaje/>, dostęp dnia 04.07.19.
25. Socha A., Tuszyńska K., Emitent, <https://mfiles.pl/pl/index.php/Emitent>, dostęp dnia 03.05.19.
26. Stack Exchange Inc, About, <https://stackoverflow.com/company>, dostęp dnia 12.05.19.
27. Technical Site, Bramki logiczne, <http://technical.elektronowolt.net/?page=bramki>, dostęp dnia 01.07.19.
28. The New York Times Company, Trust But Verify, <https://www.nytimes.com/1987/12/10/opinion/trust-but-verify.html>, dostęp dnia 23.06.19.
29. Weksel, <https://www.biznes.gov.pl/pl/firma/kontrahenci-i-klienci/chce-nawiazac-wspolprace-z-kontrahentem/metody-zabezpieczenia-przed-nieuczciwymi-kontrahentami/weksel>, dostęp dnia 03.07.19.

Spis rysunków

Rysunek 1.1. Model uproszczonego neuronu (model MCP).....	8
Rysunek 1.2. Schemat neuronu matematycznego	9
Rysunek 1.3. Ewolucja sztucznej inteligencji	10
Rysunek 1.4. Schemat procesu uczenia nadzorowanego dla programu wykrywającego spam	11
Rysunek 1.5. Schemat przedstawiający uczenie ze wzmocnieniem.....	12
Rysunek 1.6. Przykład działania algorytmu metody analizy skupień	13
Rysunek 1.7. Porównanie danych w różnych wymiarach	14
Rysunek 1.8. Idea wykorzystania uczenia nadzorowanego wykorzystana w projekcie niniejszej pracy.....	15
Rysunek 1.9. Jednowarstwowa sieć neuronowa	16
Rysunek 1.10. Porównanie perceptronów o różnej strukturze	17
Rysunek 1.11. Trójwarstwowa sztuczna sieć neuronowa.....	18
Rysunek 1.12. Skokowa funkcja liniowa.....	19
Rysunek 1.13. Liniowa funkcja aktywacji.....	20
Rysunek 1.14. Liniowa funkcja aktywacji ReLU	21
Rysunek 1.15. Logistyczna funkcja aktywacji	22
Rysunek 1.16. Tangens hiperboliczny	22
Rysunek 1.17. Przykład prostego perceptronu z jednym neuronem.....	23
Rysunek 1.18. Ogólny model prostej sieci rekurencyjnej	25
Rysunek 2.1. Kategorie rynku finansowego z punktu widzenia rodzaju instrumentu finansowego	27
Rysunek 2.2. Wejście główne do giełdy NASDAQ w dniu 08.02.18	34
Rysunek 3.1. Oficjalne logo języka programowania Python.....	39
Rysunek 3.2. Porównanie struktury wybranych języków programowania	40
Rysunek 3.3. Popularność wybranych języków programowania liczona jako ilość zapytań na stronie Stackoverflow.com	41
Rysunek 3.4. Zbiór grafik reprezentujących wybrane biblioteki języka programowania Python	44
Rysunek 4.1. Wizualizacja metod wstępnego przetwarzania danych	54
Rysunek 4.2. Różnice pomiędzy sieciami o różnej liczbie warstw ukrytych na przykładzie problemu klasyfikacji.....	58
Rysunek 4.3. Import danych	62
Rysunek 4.4. Widok zaimportowanych danych	62
Rysunek 4.5. Wybór pożądaných wartości.....	62
Rysunek 4.6. Wizualizacja danych, ceny zamknięcia firmy Apple.....	63
Rysunek 4.7. Podział danych oraz ich wizualizacja	64
Rysunek 4.8. Normalizacja danych	65
Rysunek 4.9. Znormalizowane wartości badanego szeregu	65
Rysunek 4.10. Utworzenie obiektu generatora	66
Rysunek 4.11. Import biblioteki Keras	66
Rysunek 4.12. Utworzenie oraz trenowanie modelu	67
Rysunek 4.13. Wczytanie gotowych modeli z zapisanego pliku.....	69
Rysunek 4.14. Wczytywanie zapisanych modeli sztucznych sieci neuronowych.....	69
Rysunek 4.15. Porównanie jednodniowych prognoz wraz z błędami	71
Rysunek 4.16. Pięć jednodniowych prognozy wraz z ich średnimi błędami	72
Rysunek 4.17. Średnie błędy dla czterdziestu kolejnych jednodniowych prognoz.....	73
Rysunek 4.18. Wykres pięćdziesięciu kolejnych jednodniowych prognoz wraz z ich rzeczywistą wartością.....	74
Rysunek 4.19. Wielkości funkcji straty dla modelu „model_1_1” na przestrzeni 500 epok ...	75
Rysunek 4.20. Wartości błędów czterdziestu kolejnych jednodniowych sesji modelu „model_1_1” wraz z prognozą naiwną oraz prognozą metodą prostej średniej kroczącej	76

Rysunek 4.21. Wykres czterdziestu kolejnych prognoz modelu „model_1_1” wraz z prognozą naiwną oraz prognozą metodą prostej średniej kroczącej	76
Rysunek 4.22. Efekty dwudziestosesyjnej prognozy SSN.	77
Rysunek 4.23. Miary błędów dla czterdziestu prognoz dwudziestosesyjnych.....	78
Rysunek 4.24. Absolutny błąd modelu „model_1_1” w trakcie pierwszej dwudziestosesyjnej prognozy.....	79
Rysunek 4.25. Prognozy model_1_1 i prostej średniej kroczącej	79
Rysunek 4.26. Wizualizacja symulacji giełdowej	80
§	