

Uniwersytet Jagielloński w Krakowie
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Paweł Łabno

Nr albumu: 1138170

Implementacja Sztucznej Inteligencji dla gry *Wsiąść do Pociągu*

Praca magisterska
na kierunku Informatyka Stosowana

Praca wykonana pod kierunkiem
dr Przemysława Witaszczyka
Zakład Technologii Gier

Kraków 2018

Spis treści

1	Wprowadzenie	3
1.1	Wprowadzenie do pracy	3
1.2	Definicja pojęć	3
1.3	Zasady gry <i>Wsiąść do pociągu</i>	4
2	Przebieg badania	8
2.1	Analiza zachowań gracza	8
2.2	Model doświadczenia	8
3	Model algorytmiczny	10
3.1	Wprowadzenie do eksperymentu	10
3.2	Opis algorytmu	10
3.3	Wyniki modelu algorytmicznego	14
3.4	Podsumowanie	16
4	Model sieci neuronowej	18
4.1	Wprowadzenie do eksperymentu	18
4.2	Model danych wejściowych	19
4.3	Model danych wyjściowych	20
4.4	Struktura sieci oraz zbiór danych szkolących	20
4.5	Proces weryfikacji uczenia	21
4.6	Wykorzystanie sieci neuronowej	21
4.7	Wyniki i podsumowanie	22
4.8	Wnioski na temat uczenia	24
5	Optymalizacja wyników i generalizacja sieci	26
5.1	Wprowadzenie do eksperymentu	26
5.2	Przebieg doświadczenia	26
5.3	Omówienie wyników	27
5.4	Podsumowanie	29
6	Dowód postępów działania sieci w procesie uczenia	30
6.1	Wprowadzenie do eksperymentu	30
6.2	Model eksperymentu	30
6.3	Wyniki i podsumowanie	30

7	Podsumowanie pracy	33
8	Słownik	34
9	Bibliografia i odniesienia	36

Rozdział 1

Wprowadzenie

1.1 Wprowadzenie do pracy

Wraz z rozwojem możliwości technicznych komputerów człowiek szukał kolejnych zastosowań dla swojego wynalazku. Doskonałym celem do tego było sprawienie by życie człowieka stało się prostsze. Początkowo odbywało się to za pomocą przemyślanych i sprawdzonych algorytmów wykorzystywanych do celów naukowych oraz wojskowych. Pojawiały się jednocześnie coraz bardziej skomplikowane problemy, dla których przygotowanie działającego algorytmu stanowiło wyzwanie. W domenie problemów niealgorytmicznych zastosowanie znalazły sieci neuronowe, oparte na próbach modelowania pracy ludzkiego mózgu. Niestety wymagają one szczególnie zaprojektowanych danych uczących.

Zastosowanie sieci neuronowych w procesie podejmowania decyzji w ostatnich latach staje się coraz bardziej popularne i powszechnie wykorzystywana - np. w transporcie, finansach czy medycynie. Pojawiły się wreszcie próby wykorzystania sieci neuronowych w sytuacjach, dla których nie można z łatwością ocenić czy otrzymany wynik jest prawidłowy - przykładowo w grach. Chodzi o sytuacje, gdy nie ma zdefiniowanej jednocześnie funkcji celu, a gra dostarcza jedynie zestawu reguł świata oraz np. systemu punktów. Przykładem takiego uczenia jest reinforcement learning.

Niniejsza praca zawiera opis przygotowania implementacji sztucznej inteligencji dla gry *Wsiąść do Pociągu* w oparciu o zaprojektowany algorytm oraz sztuczną sieć neuronową.

1.2 Definicja pojęć

W celu lepszego zrozumienia treści pracy warto przybliżyć kilka terminów przedstawionych poniżej. Kluczowe pojęcia, pojawiające się w treści pracy dotyczące samej gry znalazły się w Rozdziale 8.

Gra W literaturze pojawia się wiele definicji gry. Można uznać, że jest to czynność o rywkowym charakterze, w której uczestniczy jeden lub wielu graczy. Grę można również

opisać jako model matematyczny charakteryzujący się określonymi zasadami oraz zbiorem możliwych operacji na tym modelu.

Teoria gier W przypadku kiedy w rozgrywce uczestniczy więcej niż jeden gracz możemy rozważać zachowania każdego z uczestników. Możemy również założyć, że każdy z graczy chce uzyskać jak najbardziej korzystny dla siebie wynik. Teoria gier stara się znaleźć optymalne strategie dla graczy.

Klasyfikacja gry Gra *Wsiąść do Pociągu* jest grą wieloosobową dopuszczającą od dwóch do pięciu graczy, niesymetryczną z niepełną informacją. Decyzje podejmowane są w sposób sekwencyjny po jednym ruchu na uczestnika rozgrywki.

Problem klasyfikacji W grze *Wsiąść do Pociągu* pojawia się problem decyzyjny w postaci klasyfikacji. Klasyfikacja jest problemem przyporządkowania danego zestawu danych do jednej lub więcej klas ze zbioru co najmniej dwóch klas.

Gra z niepełną informacją Określenie gra z niepełną informacją oznacza, że gracze podejmują swoje decyzje nie wiedząc jakie cele mają inni uczestnicy rozgrywki lub jaki jest stan środowiska gry. W grze *Wsiąść do pociągu* polega to m.in. na braku wiedzy jakie bilety realizują konkurenci (oraz czy zostały ukończone).

Uczenie maszynowe Jako definicję można wykorzystać publikację *A Few Useful Things to Know about Machine Learning* autorstwa Pedro Domingos z *University of Washington* Angielska: *Machine learning algorithms can figure out how to perform important tasks by generalizing from examples.*

Polskie tłumaczenie: *Algorytmy uczenia maszynowego potrafią dowiedzieć się, jak wykonywać ważne zadania, generalizując je na podstawie przykładów.*

Głębokie sieci neuronowe Głębokie sieci neuronowe są szczególnym przypadkiem uczenia maszynowego. Sieć składa się z więcej niż jednej warstwy ukrytej, a reprezentacja wewnętrzna neuronów przeważnie nie jest odwzorowaniem liniowym.

1.3 Zasady gry *Wsiąść do pociągu*

W celu zrozumienia pracy należy także zrozumieć zasady gry (model) *Wsiąść do Pociągu* będącej tematem badania. Pełna treść polskiej instrukcji do gry jest dostępna pod wpisem nr. 2 w bibliografii pracy.

1.3.1 Omówienie celu gry

W trakcie gry gracze budują imperium kolejowe na terenie Stanów Zjednoczonych (model planszy przedstawiony na Rysunku 1.1). Gracze gromadzą zasoby potrzebne do budowy połączeń między poszczególnymi miastami. Dodatkowo, każdy z graczy otrzymuje w trakcie

rozgrywki dodatkowe wyzwania połączenia dwóch wskazanych miast swoją siecią.

Celem rozgrywki jest uzyskanie jak największej liczby punktów. Gracz może zdobyć punkty za zarezerwowanie połączeń (Tablica 1.1) oraz za ukończenie biletów (korzystając z połączeń jednego gracza można dotrzeć z jednego wskazanego miasta do drugiego).



Rysunek 1.1: Plansza wykorzystana w rozgrywce

Długość połączenia (w wagonach)	Ilość punktów
1	1
2	2
3	4
4	7
5	10
6	15

Tablica 1.1: Ilość punktów otrzymywanych za zrealizowanie połączenia

1.3.2 Faza przygotowania rozgrywki

Po rozłożeniu planszy następuje przygotowanie zasobów do gry:

1. Przygotowanie talii

Gracze tasują dostępne karty wagonów oraz biletów. Rozkładają na planszy pierwszych 5 kart wagonów - widocznych awerssem dla graczy. Jeśli są wśród nich co najmniej 3 lokomotywy należy wymienić cały zestaw kart.

2. Losowanie biletów

Każdy z graczy pobiera z talii biletów po trzy karty. Następnie wybiera które z nich zachować a które odłożyć na spód talii. Musi zachować conajmniej dwa bilety. *Jako bezpieczną taktykę gry przyjmuje się zachowanie jedynie dwóch kart biletów*

3. Losowanie kart wagonów

Każdy z graczy pobiera z talii biletów po cztery karty. Nie ma prawa ich odrzucić lub zamienić.

Gdy zostaną wykonane powyższe kroki uczestnicy gry mogą rozpocząć rozgrywkę.

1.3.3 Rodzaje decyzji podejmowanych przez gracza

W trakcie swojej tury gracz może wybrać jedną z trzech dostępnych akcji:

Rezerwacja połączenia

Warunki rezerwacji połączenia:

1. Gracz musi posiadać odpowiednią ilość kart wagonów odpowiedniego koloru (karta lokomotywy zastępuje dowolny kolor).
2. Gracz musi posiadać co najmniej tyle wolnych wagonów ile wynosi długość połączenia które chce zarezerwować.
3. Gracz nie zarezerwował już ścieżki (w przypadku połączeń podwójnych) w rozważanym połączeniu.
4. Pozostaje co najmniej jedna ścieżka wolna w połączeniu.
5. Połączenie ma dwie ścieżki, jedna z nich jest zajęta, druga wolna. W rozgrywce uczestniczy co najmniej 4 graczy.

Decyzja zabroniona:

Gracz nie posiada wagonów lub kart wagonów potrzebnych do zarezerwowania dowolnego połączenia na mapie.

Dobranie kart wagonów

Warunki dobrania kart wagonów:

1. Gracz może zdecydować o dobraniu do dwóch wagonów w zależności od tego jakie wagony chce dobrać.

2. Gracz może zabrać jako pierwszy wagon - kartę lokomotywy (Joker) z puli planszy. Nie dobiera wtedy drugiego wagonu.
3. Gracz może zabrać z talii lub puli dwie karty. Może być jedna karta z talii oraz jedna karta z planszy.

Decyzja zabroniona:

Skończyła się talia kart wagonów, na planszy nie znajdują się żadne kart wagonów a stos kart odrzuconych jest pusty.

Dobranie kart biletów

Gracz w ramach decyzji o dobranie biletów losuje 3 bilety z talii i decyduje, które zachowa, a które odrzuci. Zachować musi co najmniej 1 bilet.

Decyzja zabroniona:

W talii kart biletów nie pozostał żaden bilet.

1.3.4 Uzupełnianie kart wagonów na planszy

Gdy jeden z graczy dobierze kartę wagonów znajdującą się na planszy na jej miejsce pojawia się pierwsza karta z talii. Jeżeli w dowolnym momencie rozgrywki na planszy znajdują się co najmniej trzy karty lokomotyw to cały zestaw kart zostaje odrzucony, a na ich miejsce dobierane jest pięć kart z talii kart wagonów.

Jeżeli w danym momencie talia kart wagonów jest pusta należy potasować stos kart odrzuconych a następnie położyć go na miejscu talii kart wagonów. Jeżeli odbywa się to w trakcie uzupełniania kart na planszy to karty należy wykorzystać do uzupełnienia kart wagonów na planszy.

1.3.5 Koniec gry

Rozgrzywka kończy się w momencie, gdy jednemu z graczy pozostają co najwyżej dwa wagoniki. Po tym zdarzeniu każdy z graczy ma jeszcze jeden ruch do wykonania. Gdy wszyscy gracze wykonają swój ostatni ruch przechodzi się do fazy podliczenia punktów.

Podliczenie przebiega następująco:

1. Podliczenie punktów za połączenia (według Tablicy 1.1)
2. Dodanie punktów za każdy zrealizowany bilet (według oznaczenia na bilecie)
3. Odjęcie punktów za każdy niezrealizowany bilet (według oznaczenia na bilecie)

Dla usprawnienia procesu gry w eksperymentach pominięto zasadę bonusowych 10 punktów dla gracza posiadającego najdłuższą nieprzerwaną ścieżkę.

Rozdział 2

Przebieg badania

2.1 Analiza zachowań gracza

Pierwszym etapem pracy nad sztuczną inteligencją było przeanalizowanie zachowań graczy pod kątem przygotowania algorytmu, który wykorzystano do stworzenia zbioru uczącego. Sieć neuronowa wymaga wielu przykładów rozgrywki. Zostały one dostarczone poprzez rozegranie wielu partii gry algorytmem, który rozwiązuje grę. Jako rozsądne kroki w celu przeprowadzenia analizy przyjąłem:

1. Analiza rozgrywek z rzeczywistym graczem
2. Analiza rozgrywek z graczem komputerowym (dostępnym z grą w wersji cyfrowej)
3. Przeszukanie sieci internet o tematyce sztucznej inteligencji w grach

Jako kluczowe w dalszej analizie przyjąłem następujące zachowanie gracza komputerowego:

1. Gracz kieruje rozgrywkę dla siebie - zależy mu jedynie na jak największej ilości punktów (w danym momencie oraz ogólnie)
2. Gracze nie przeszkadzają sobie nawzajem (uczestnicy rozgrywki nie podejmują decyzji mających na celu utrudnienie gry innym uczestnikom, wykluczając sytuację, gdy dla danego gracza decyzja blokująca jest jednocześnie najbardziej korzystną w kontekście zdobyczy punktowej)
3. Uczestnicy starają się najpierw ukończyć posiadane bilety, a następnie gromadzą punkty

2.2 Model doświadczenia

Po określeniu wstępnych założeń dotyczących działania sztucznej inteligencji w grze przystąpiłem do przygotowania doświadczenia badającego jakość algorytmu oraz uczenia sieci za pomocą wygenerowanych danych. W pierwszym kroku określiłem cechy(features), które posłużyły mi do oceny sztucznej inteligencji. Zostały one wpisane do Tablicy 2.1.

Dla uzyskania miarodajnych wyników dla każdej próby postanowiłem przeprowadzić 1000

Parametr	Opis
MAX	Maksymalna liczba punktów zdobyta przez gracza
MIN	Najmniejsza liczba punktów zdobyta przez gracza
AVG	Sredni wynik punktowy
MED	Mediana punktów
AVG DONE TCK	Srednia liczba zrealizowanych biletów
AVG FAIL TCK	Srednia liczba niezrealizowanych biletów
110 (%)	Liczba graczy z co najmniej 110 punktami
120 (%)	Liczba graczy z co najmniej 120 punktami
140 (%)	Liczba graczy z co najmniej 140 punktami
FAIL (%)	Liczba gier w których gracz dokonał ruchu zabronionego

Tablica 2.1: Parametry rozgrywki

rozgrywek w sumie, po 250 w każdym z poszczególnych modeli według konfiguracji zawartej w Tablicy 2.2.

Liczba graczy	Liczba rozgrywek
2	250
3	250
4	250
5	250

Tablica 2.2: Konfiguracja danych testowych

W trakcie pracy dodałem dwa eksperymenty do doświadczenia - weryfikację procesu uczenia oraz jakość działania sieci w zależności od przygotowanego modelu danych. Każde z tych badań odbywało się na mniejszej próbce składającej się z 60 rozgrywek dla każdej ilości graczy (sumarycznie 240 rozgrywek na próbkę). Opis przygotowania zbioru danych uczących zawarto w rozdziałach 4.4 (strona 20) oraz 5.1 (strona 26).

Rozdział 3

Model algorytmiczny

3.1 Wprowadzenie do eksperymentu

Celem drugiej części eksperymentu był dowód, że komputer może rozgrywać samodzielnie partię gier *Wsiąść do Pociągu*. Symulacja i zapis stanu rozgrywek za pomocą modelu algorytmicznego pozwoli zebrać dane, które w opracowanej formie posłużą jako zbiór danych uczących sieć neuronową. Opracowanie danych ma posłużyć również do ustalenia jakościowych wyników algorytmu, które posłużą do oceny jakości działania sieci. W dalszej pracy model algorytmiczny będzie opisywany jako *Algorytm* lub *Algorytm klasyczny*. W trakcie przygotowania algorytmu rozwiązującego grę posłużono się dyskusją wskazaną w punkcie 3. Bibliografii. Ostateczna forma algorytmu została zbudowana w ramach pracy.

3.2 Opis algorytmu

W pracy przygotowano kilka szczegółowych algorytmów:

Decyzja ogólna Jedna z trzech dozwolonych przez zasady gry decyzji (plus pominięcie tury w szczególnych przypadkach).

Decyzja bilet Decyzja polegająca na wyborze najlepszego podzbioru biletów z przekazanych jako parametr algorytmu.

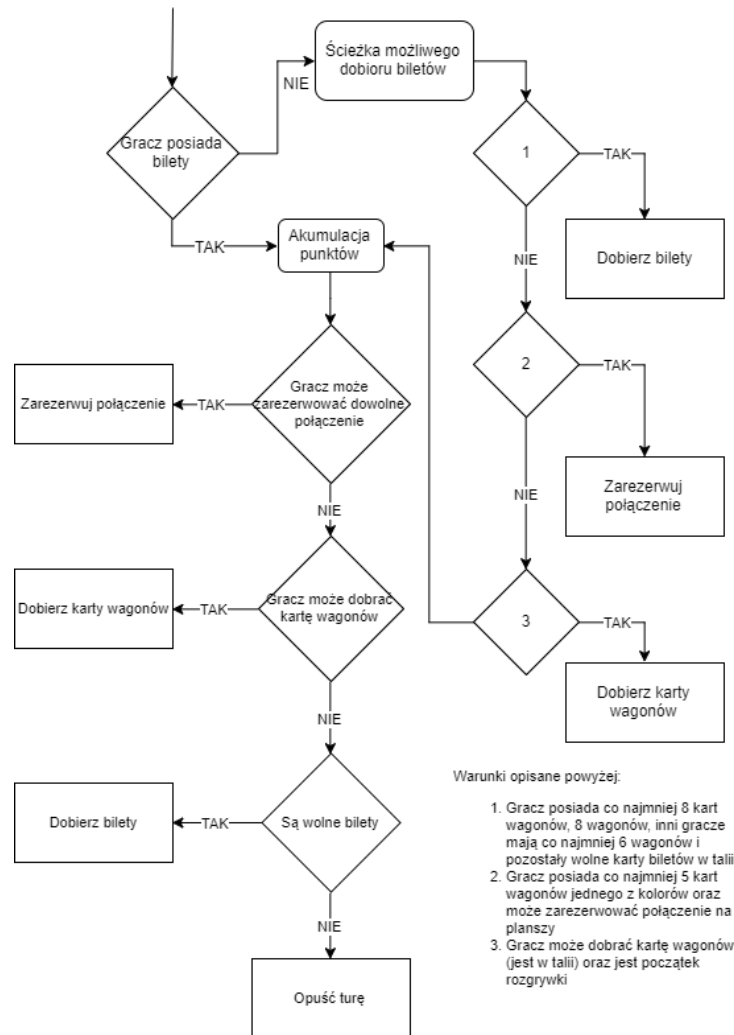
Karta wagonów Decyzja polegająca na wyborze, które karty wagonów gracz powinien zabrać z planszy.

Rezerwacja połączenia Decyzja, której efektem jest wybór połączenia, które powinno zostać zarezerwowane przez gracza w pierwszej kolejności.

Przygotowanie tury Wyznaczenie kierunku w którym powinien się gracz poruszać w tym wyznaczenie dostępnych i brakujących połączeń.

3.2.1 Decyzja ogólna

Wykorzystany w pracy algorytm przedstawiono na Rysunku 3.1 w formie schematu blokowego.



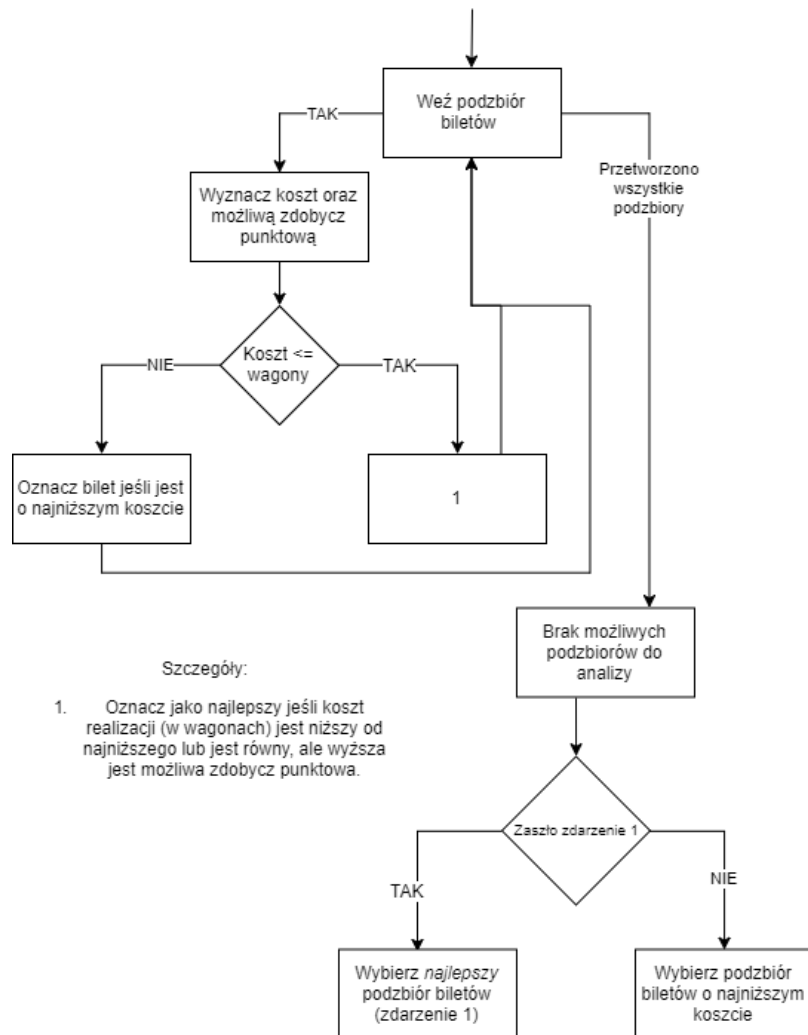
Rysunek 3.1: Schemat blokowy algorytmu podejmującego decyzje o typie wykonywanego ruchu

3.2.2 Poddecyzja - bilet

Wykorzystany w pracy algorytm wyboru biletów przedstawiono na Rysunku 3.2 w formie schematu blokowego.

3.2.3 Poddecyzja - karta wagonów

Wykorzystany w pracy algorytm wyboru pobieranych kart wagonów przedstawiono na Rysunku 3.3 w formie schematu blokowego.



Rysunek 3.2: Schemat blokowy algorytmu podejmującego decyzje o wyborze najlepszego podzbioru biletów

3.2.4 Decyzja - rezerwacja połączenia

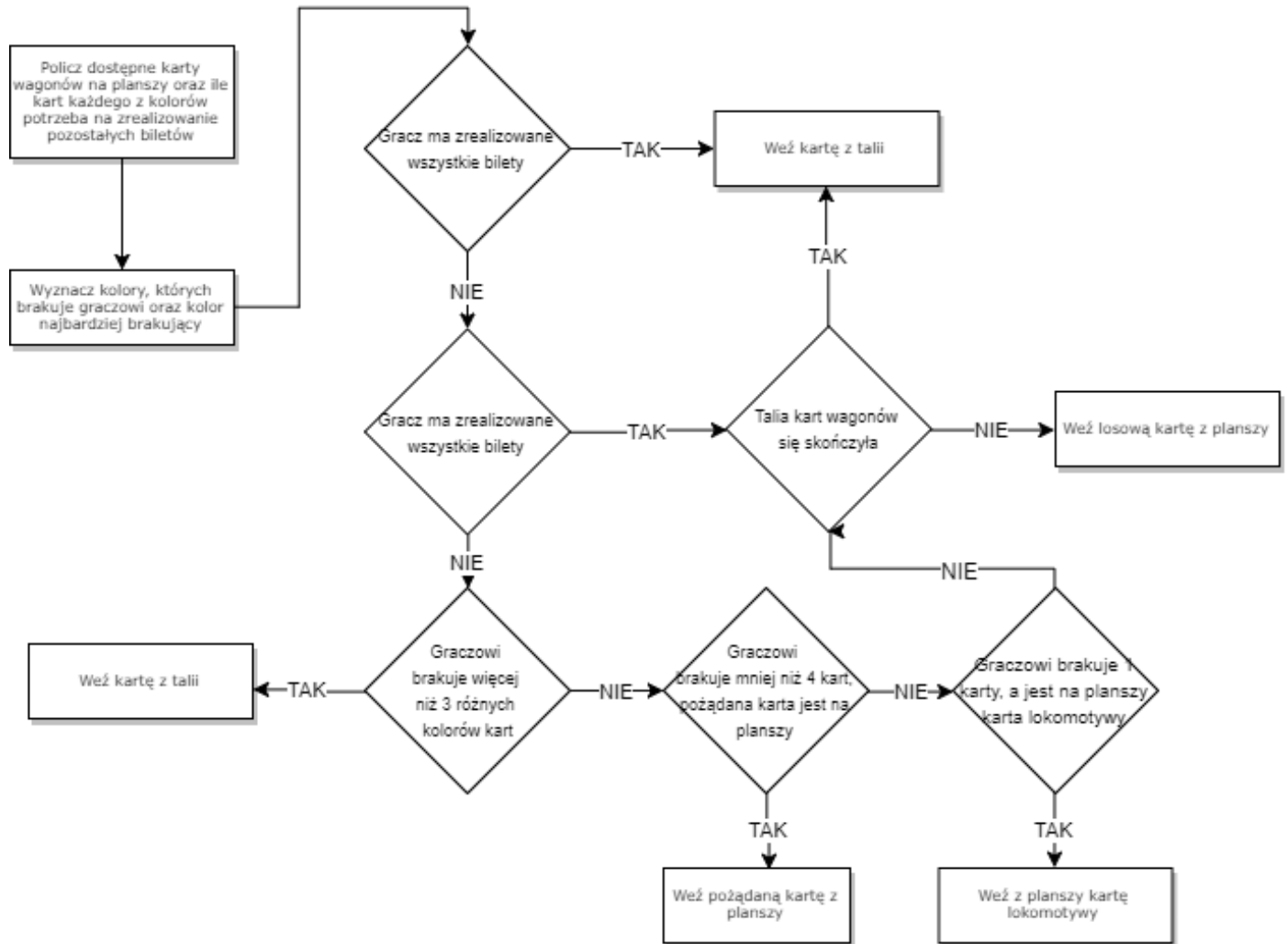
Wykorzystany w pracy algorytm rezerwacji najlepszego połączenia przedstawiono na Rysunku 3.4 w formie schematu blokowego.

3.2.5 Przygotowanie tury

1. Określ pulę wszystkich połączeń potrzebnych do realizacji biletów gracza (określane jako *cel*)
2. Z wszystkich połączeń na mapie wyznacz te które gracz może zrealizować w danej turze (określane jako *możliwości*) - pod warunkiem:

Dodaj połączenia zawierające się w *celu*

Dodaj wszystkie połączenia o długości co najmniej 5



Rysunek 3.3: Schemat blokowy algorytmu podejmującego decyzje o wyborze karty wagonu do pobrania

Dodaj wszystkie połączenia gdy długość zbioru *cel* jest równa 0

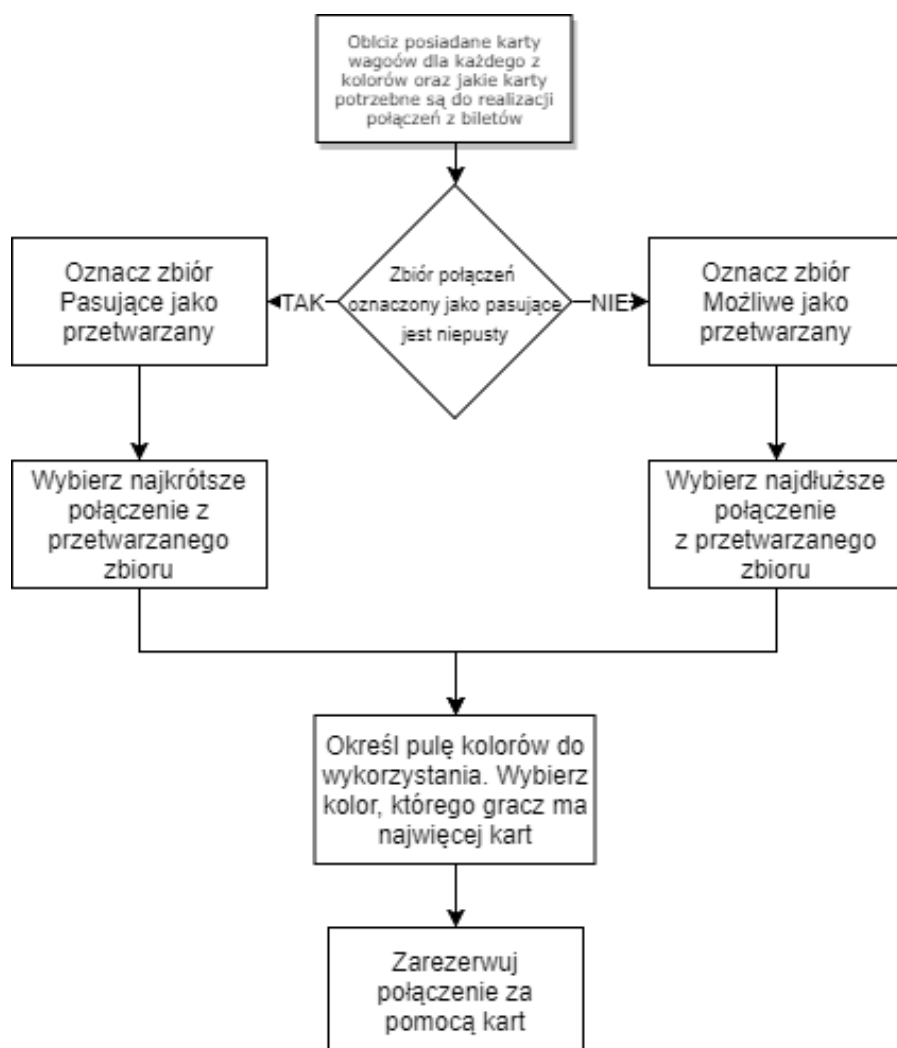
3. Dla wszystkich połączeń ze zbioru *cel* (które gracz może w ogóle zarezerwować)

Zawierające się w zbiorze *mozliwosci* dodaj do zbioru *pasujace*

W przeciwnym przypadku dodaj do zbioru *Brakujace*

3.2.6 Wyznaczanie tras do realizacji biletów

Planszę rozgrywki można przedstawić w formie modelu grafu, gdzie połączeniami są krawędzie a wierzchołkami miasta. Do wyznaczenia najbardziej korzystanego połączenia tras wykorzystano Algorytm Bellmana-Forda (wg. *Wprowadzenia do Algorytmów* - Bibliografia 5.). Algorytm ten pozwala nam na określenie całej najkrótszej ścieżki rozpoczynając od wierzchołka początkowego. Jako metodę porównywalczą dla kosztu ścieżki uznano:



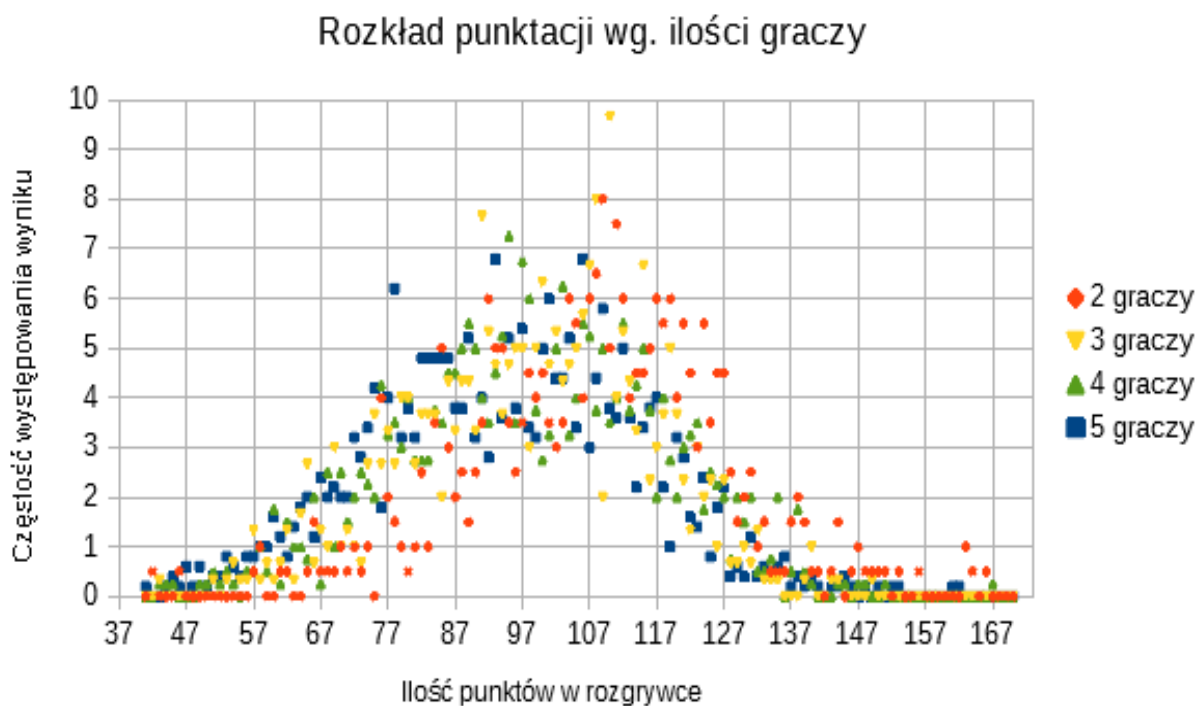
Rysunek 3.4: Schemat blokowy algorytmu rezerwującego najlepsze połączenie

1. Dla różnych kosztów (w wagonach) od węzła początkowego wybieramy ten który jest mniejszy
2. Dla równego kosztu (w wagonach) od węzła początkowego wybieramy tą ścieżkę dla której ścieżka gwarantuje większą ilość punktów

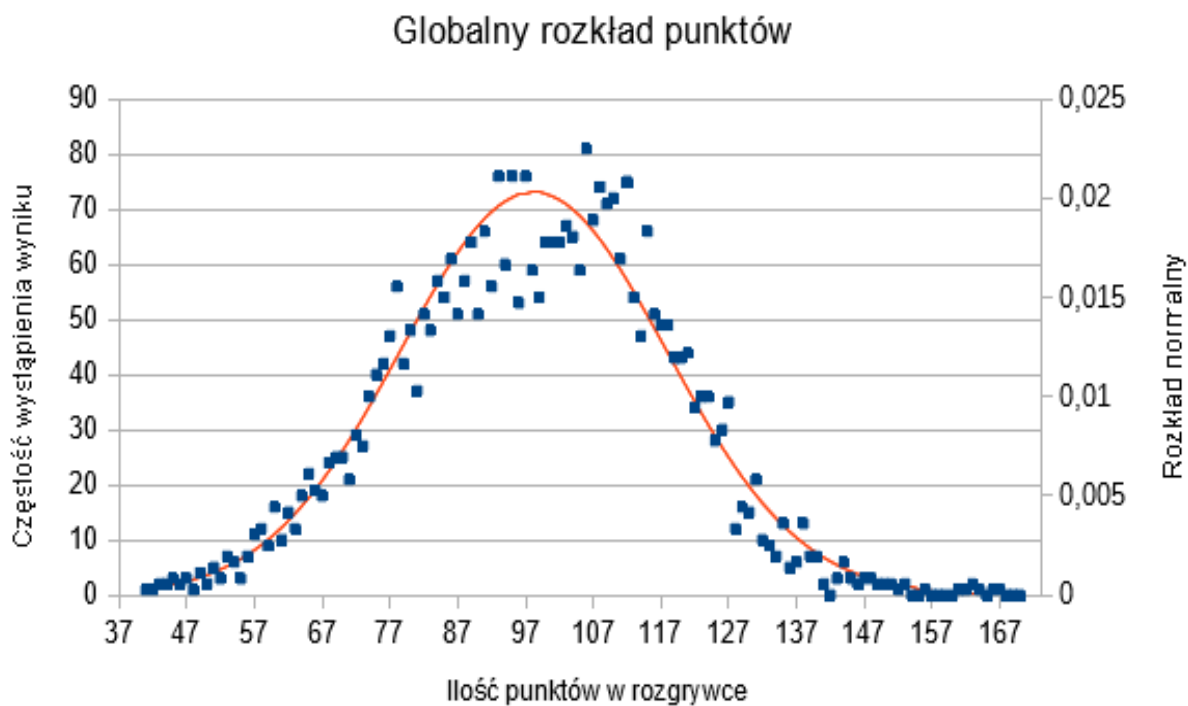
Przy osiągalnym koszcie realizacji biletów tj. koszt jest mniejszy od liczby wagoników gracza ścieżka o najniższym koszcie jest oznaczona jako cel realizacji. Wskazanie jest realizowane każdorazowo na początku tury gracza.

3.3 Wyniki modelu algorytmicznego

W trakcie doświadczenia uruchomiono serie gier według konfiguracji opisanej w Tablicy 2.2. Wyniki uzyskane w doświadczeniu przedstawiono w Tablicy 3.1. W ostatniej kolumnie umieszczono wyniki podsumowujące wszystkie rozgrywki (niezależnie od liczby graczy).



Rysunek 3.5: Rozkład punktacji wg. ilości graczy dla algorytmu klasycznego



Rysunek 3.6: Globalny rozkład punktów dla algorytmu klasycznego

Cecha	2 graczy	3 graczy	4 graczy	5 graczy	Globalnie
Minimum	37	14	43	34	14
Maksimum	173	149	182	162	182
Srednia	107,02	98,02	98,03	94,59	98,16
σ	19,70	18,00	19,38	19,71	19,66
Mediana	108	95	98	96	99
(%) > 110	44,60	38,00	28,00	21,60	27,51
(%) > 120	23,40	8,80	13,20	8,80	12,14
(%) > 140	3,80	0,40	0,70	0,96	1,17
Zrealizowane bilety	3,63	3,01	3,21	2,8	3,08
Niezrealizowane bilety	0,46	0,58	0,56	0,62	0,57

Tablica 3.1: Wyniki uzależnione od ilości graczy dla algorytmu klasycznego

3.4 Podsumowanie

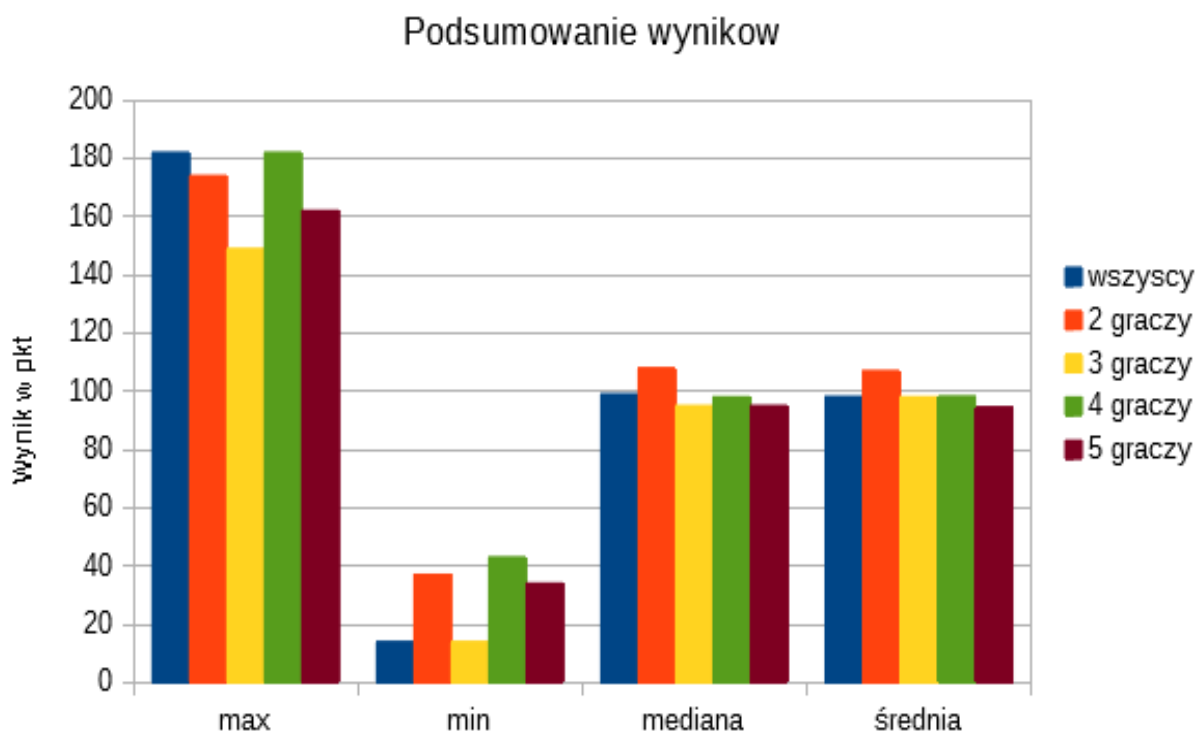
Podsumowując otrzymane dane możemy dojść do kilku wniosków. Przede wszystkim decyzje podejmowane przez gracza komputerowego przypominają te dokonywane przez rzeczywistego gracza. Gracz algorytmiczny ma swoje słabe oraz mocne strony co powoduje, że wykres częstości wartości punktów na koniec rozgrywki przypomina rozkład Gaussa co możemy zaobserwować na Rysunku 3.6, gdzie wykres rozkładu normalnego o otrzymanych parametrach zaznaczono linią.

Zgodnie z wynikami wskazanymi w Tablicy 3.1 możemy zauważyć, że przewidywany wynik gracza powinien wynieść ok. 98 punktów (wynosi tyle średni wynik punktowy oraz mediana). Obserwację potwierdza przebieg wykresu z Rysunku 3.6. Z zapisów rozgrywek wyczytać można również najczęściej pojawiającą się zdobycz punktową wynoszącą 106. Można ponadto zauważyć, że spora część rozgrywek kończy się wynikiem powyżej 87 punktów. Powyżej wyniku 140 punktów znajduje się znikoma część graczy - według tablicy zaledwie 1,17%.

Wnioski możemy również wynieść z Rysunku 3.5, który wskazuje rozkład punktów w zależności od ilości graczy. Najlepsze wyniki osiągają gracze uczestniczący w rozgrywce dwuosobowej. Jest to spowodowane powolnym zapełnianiem planszy przez co zdobywają mniejszą ilość punktów karnych za niezrealizowane bilety, jednocześnie gracz jest mniej karany za spóźniony ruch. W przypadku takiej konfiguracji można oczekiwać wyniku ok. 107 punktów.

Bardziej zbliżone do siebie są rozgrywki trój- oraz cztero-osobowe, dla których średni wynik wynosi 98 pkt. Można zauważyć, że dla rozgrywek trójosobowych po osiągnięciu maksimum częstości, częstość występowania wyników o danej ilości punktów maleje szybciej od rozgrywek czteroosobowych. Skutkiem tego jest znacznie wyższy procent graczy z wynikiem końcowym ponad 120 oraz ponad 140 punktów, co przedstawiono w Tablicy 3.1.

Kolejnym zauważalnym wnioskiem są lepsze wyniki dla rozgrywek 2 oraz 4 osobowych niż 3 oraz 5 graczowych. Można to zauważyć na Rysunku 3.7, gdzie w przypadku średniej oraz



Rysunek 3.7: Wyniki wg. ilości graczy dla algorytmu klasycznego

mediana jest zauważalny trend malejący wyniku względem ilości graczy. Jako przyczynę tej sytuacji można wskazać szybsze zapełnianiem się planszy. Można zaobserwować również niską średnią ilość niezrealizowanych biletów - można wnioskować, że co najmniej co drugi gracz zrealizował wszystkie swoje bilety. W kwestii zrealizowanych biletów ogólna średnia powyżej 3 biletów świadczy, że gracze zazwyczaj dociągali karty biletów.

Wysoka amplituda wyników jest spowodowana dużym spektrum możliwych stanów planszy (rezerwacji połączeń, biletów oraz celów graczy), przez co możliwe są wyniki bardzo dobre (powyżej 140 punktów) jak i złe (poniżej 60).

Rozdział 4

Model sieci neuronowej

4.1 Wprowadzenie do eksperymentu

W przeciwieństwie do jawnego działania algorytmu klasycznego funkcjonowanie sieci neuronowej jest bardziej niezależne od projektanta. Efektywność sieci neuronowych w dużym stopniu zależy od odpowiedniego przygotowania modelu danych. Dodatkowo, jakość procesu uczenia sieci zależy od dostarczonych do procesu uczenia danych treningowych. Motywacją tej części pracy jest sprawdzenie czy przygotowany zbiór danych może być podstawą do stworzenia wymagającego przeciwnika sterowanego siecią neuronową.

Przygotowanie sztucznej inteligencji dla tego typu zagadnień jest bardzo ciekawym problemem. W większości gier uczestnik rozgrywki ma określony lub ograniczony zakres dozwolonych w danym momencie typów decyzji. Jest to idealny przypadek na wykorzystanie klasyfikatora przykładowo DNNClassifier z biblioteki *Tensorflow*. Przygotowanie algorytmu, który podejmie odpowiednią decyzję niesie za sobą ryzyko w postaci znacznej rozbudowy algorytmu o dodatkowe rozgałęzienia i ujęcie licznych przypadków szczególnych lub ograniczenie się do dobrego działania w ograniczonym zakresie przypadków. Przygotowanie sieci neuronowej może pozwolić na obsługę zdarzeń szczególnych - zwłaszcza w przypadku, gdy takie zdarzenie znajdzie się w zbiorze danych uczących (ale wtedy algorytm klasyczny musi je generować).

W przypadku gier planszowych, przygotowanie modelu danych nie jest niewykonalnym zadaniem, gdyż sama rozgrywka odbywa się w postaci unormowanej - większość zdarzeń oraz stanów możemy przedstawić w matematycznej formie (jak wskazano w Rozdziale 1.2) - jako wartości zmienne. W pierwszym kroku pracy nad siecią neuronową należało zbudować taki model danych w oparciu o *Stan gry*, na podstawie którego sieć będzie mogła podjąć jednoznaczную decyzję. Model danych powinien być pozbawiony zbędnych typów danych. Kluczowe jednak dla zastosowania sieci głębokich jest posiadanie/wygenerowanie odpowiednio dużego zbioru trenującego.

Po przygotowaniu modelu przystąpiono do przygotowania danych uczących w oparciu o model algorytmiczny. Ostatnim krokiem przed rozpoczęciem procesu uczenia sieci było przygotowanie konfiguracji sieci, która będzie osiągać najlepsze wyniki w symulowanych rozgrywkach.

4.2 Model danych wejściowych

Model danych wejściowych przedstawiono w Tablicy 4.1. W trakcie przygotowania kierowałem się danymi jakie są wykorzystywane w trakcie podejmowania decyzji w algorytmie *klasycznym*

Nazwa	Typ	Zakres	Opis
Tura	Int	0-100	Aktualna tura gracza
Karty wagonów	Int	0-120	Suma kart wagonów gracza (wszystkich kolorów)
Wagony na planszy	Int	0-5	Ilość wagonów na planszy
Wagony w talii	Int	0-120	Ilość wagonów w talii
Wagony odrzucone	Int	0-120	Ilość odrzuconych wagonów
Bilety gracza	Int	0-30	Ilość biletów posiadanych przez gracza
Bilety w talii	Int	0-30	Ilość biletów pozostających w talii <i>biletów</i>
Wagony gracza	Int	0-45	Ilość wagoników posiadanych przez gracza
Punkty gracza	Int	-50 - 200	Ilość punktów posiadanych przez gracza
Pozostałe karty wagonów	Int	0 - 120	Ilość kart wagonów w talii oraz na planszy
Liczba połączeń do zrealizowania	Int	0 - 40	Ilość połączeń na planszy, które gracz może zarezerwować
Liczba połączeń pasujących	Int	0 - 40	Ilość połączeń, które gracz może zrealizować z trasy biletów.
Liczba połączeń celowych	Int	0 - 40	Ilość połączeń na trasie biletów.
Liczba połączeń brakujących	Int	0 - 40	Ilość połączeń jakie brakuje graczowi do zrealizowania wszystkich biletów
Liczba graczy	Int	2-5	Ilość graczy w rozgrywce
Min wagonów	Int	0 - 45	Najmniejsza liczba wagonów wśród pozostałych graczy
Max wagonów	Int	0 - 45	Największa liczba wagonów wśród pozostałych graczy
Średnia wagonów	Int	0 - 45	Średnia liczba wagonów wśród pozostałych graczy
Mediana Wagonów	Int	0 - 45	Mediana liczby wagonów wśród pozostałych graczy
Min biletów	Int	0 - 45	Najmniejsza posiadanych liczba biletów wśród pozostałych graczy
Max biletów	Int	0 - 45	Największa posiadanych liczba biletów wśród pozostałych graczy

Średnia biletów	Int	0 - 45	Średnia liczba posiadanych biletów wśród pozostałych graczy
Mediana biletów	Int	0 - 45	Mediana liczba posiadanych biletów wśród pozostałych graczy
Niezrealizowane bilety	Int	0 - 30	Liczba biletów <i>niezrealizowanych</i> przez aktywnego gracza
Zrealizowane bilety	Int	0 - 30	Liczba biletów <i>zrealizowanych</i> przez aktywnego gracza
Aktualny wynik gracza	Int	0 - 30	Liczba punktów bez liczenia punktów za bilety

Tablica 4.1: Dane wejściowe sieci neuronowej

4.3 Model danych wyjściowych

W eksperymencie wykorzystano klasyfikator, który operuje na czterech możliwych klasach wyjściowych (do podglądu w Tablicy 4.2):

#	Nazwa	Opis decyzji
0	Opuszczenie tury	Gracz nie może podjąć żadnej decyzji
1	Pobranie karty wagonów	Gracz powinien pobrać karty tego typu (akumulacja zasobów)
2	Pobranie karty biletów	Gracz powinien pobrać karty tego typu (dobranie celów)
3	Zarezerwowanie połączenia	Gracz powinien zabezpieczyć punkty

Tablica 4.2: Klasy określone przez klasyfikator

4.4 Struktura sieci oraz zbiór danych szkolących

W pracy wykorzystano klasyfikator oparty o głęboką sieć neuronową typu feed forward. Warstwa wejściowa składała się z 26 cech(features). Następnie były dwie warstwy ukryte, które składały się odpowiednio z 180 oraz 160 neuronów. Na wyjściu klasyfikacja opierała się na 4 neuronach. Strukturę warstw ukrytych wykorzystanej sieci wyznaczono empirycznie wielokrotnie uruchamiając proces uczenia dla różnych konfiguracji neuronów na warstwach). Ostatnia warstwa wyznaczała prawdopodobieństwo, że przetwarzany stan gry można sklasyfikować jako powiązaną z neuronem klasą. Wartość z najwyższym prawdopodobieństwem sieć uznawała jako zwracany wynik.

Jako źródło danych uczących wykorzystano plik zawierający 5700 stanów rozgrywki i odpowiadające im decyzje podjęte z wykorzystaniem modelu algorytmicznego. Z zapisu wszystkich rozgrywek wybrano 30 dla każdej konfiguracji liczby graczy z minimalnym wynikiem końcowym wynoszącym 120 punktów. Każda linia zawierała zapis jednego stanu rozgrywki.

Ostatnia wartość oznacza identyfikator klasy z Tablicy 4.2. Plik służący za źródło danych jest dostępny pod ścieżką `src/dataset/learn_all.csv` w repozytorium projektu.

Listing 4.1: Pojedynczy format danych uczących

```
3 2 , 1 2 , 5 , 2 5 , 5 1 , ( ... ) , 5 , 6 , 5 . 5 , 5 . 5 , 0 , 5 , 3 2 , 3 , 7 , 1
```

W procesie uczenia sieci wykorzystano próbki danych (batch size) wynoszące 250 zapisów (pojedynczych stanów rozgrywki). Jako funkcję aktywacyjną wykorzystano `tf.nn.relu`, którą można przedstawić jako

$$f(x) = \max(0, x) \quad (4.1)$$

Jako optymalizator użyto AdaGrad czyli algorytm adaptacyjnego gradientu. Jako funkcję redukującą stratę (loss) dla próbki danych wykorzystano funkcję SUM z biblioteki `tf.losses.Reduction`. Funkcja ta zwraca skalarną sumę ważonych strat. Sumarycznie w całej strukturze sieci znajdowało się 34120 połączeń pomiędzy węzłami.

4.5 Proces weryfikacji uczenia

W trakcie zbierania pomiarów kilkakrotnie zmieniano model danych. Sieć podejmowała często zabronione decyzje spowodowane wpływem niepotrzebnych *charakterystyk*. W związku z tym usunięto dane o poczynaniach innych graczy (np. liczbie kart wagonów), a dodano informację o ilości różnych kolorów kart wagonów posiadanych przez gracza oraz największą ilość kart tego samego koloru.

4.6 Wykorzystanie sieci neuronowej

Repozytorium z całym kodem projektu znajduje się pod wpisem 4. w Bibliografii pracy. W trakcie mojej pracy korzystałem z biblioteki *Tensorflow*, która ułatwiła mi pracę z sieciami neuronowymi. W celu utworzenia klasyfikatora (*Deep Neural Network Classifier*) wykorzystałem kod z Listingu 4.2. Można zauważyć kształt warstw ukrytych oraz liczbę możliwych klas.

Listing 4.2: Konfiguracja klasyfikatora

```
self.predictor = tf.estimator.DNNClassifier(
    feature_columns=self.my_feature_columns,
    hidden_units=[180, 160],
    n_classes=4,
    model_dir=model_dir,
    config=self.my_config
)
```

Najmniejszą spójną częścią mojej pracy był Eksperyment, który zawarty został w jednej klasie. Przebieg pojedynczego uruchomienia eksperymentu w formie kodu został zawarty w Listingu 4.3. Przed rozpoczęciem pracy musimy określić liczbę cykli. Każdy cykl składa się

z sprawdzenia czy w danej iteracji mamy przeprowadzić proces uczenia. Proces uczenia jest opcjonalny, gdyż możemy pracować z pomocą wyuczonej sieci. Następnie wyznaczamy parametry sieci w tym *accuracy* oraz *loss* dla próbki danych. Na samym końcu przeprowadzamy podaną w konfiguracji eksperymentu liczbę rozgrywek.

Listing 4.3: Funkcja przeprowadzająca pojedynczy eksperyment

```
def run_experiment(self, learn_it, learn_batch, with_learn,
    start_it=1, epochs = 1, playerStart = 2, playerEnd = 5,
    skipFirst = False):
    # przygotowanie plikow do zapisu eksperymentu
    experiment_start = datetime.utcnow()
    for li in range(start_it, learn_it+1):
        if with_learn and
            not (skipFirst and li == start_it):
            for i in range(epochs):
                self.make_learning(learn_batch)
            quality = self.make_evaluate(learn_batch)
            iteration_quality = self.make_iteration(li,
                playerStart, playerEnd+1)
    # zapis logow calego eksperymentu
```

Ostatnim fragmentem kodu, który warty jest omówienia jest fragment kodu gracza wykorzystującego sieć neuronową w którym sieć wskazuje jaką decyzję gracz powinien podjąć. W poniższym kodzie *predict_x* zawiera informacje o stanie planszy - dokładnie takim samym, na którego podstawie decyzję podejmuje gracz algorytmiczny. Dzięki poniższemu fragmentowi kodu ostatecznie gracz wykorzystujący sieć neuronowa może uczestniczyć w rozgrywkach.

Listing 4.4: Wyznaczenie klasy decyzji

```
predictions = self.predictor.predict(
    input_fn=lambda: game_data.pred_input_fn(predict_x,
    labels=None,
    batch_size=1))
result = DecisionType.DecisionType.PASS
for pred in zip(predictions):
    classId= pred[0][ 'class_ids' ][0]
    result = DecisionType.DecisionType(classId)
```

4.7 Wyniki i podsumowanie

Wyniki otrzymane w trakcie eksperymentu zostały zawarte w Tablicy 4.3, oraz zilustrowane na Rysunkach 4.1, 4.2 oraz 4.3. W ostatniej kolumnie umieszczono wyniki podsumowujące wszystkie rozgrywki (niezależnie od liczby graczy).

Wyniki otrzymane w części dotyczącej sieci neuronowych wskazują, że podobnie jak dla algorytmu klasycznego sieć radzi sobie w grze lepiej lub gorzej w zależności od trybu gry.

Cecha	2 graczy	3 graczy	4 graczy	5 graczy	Globalnie
Minimum	55	37	27	15	15
Maksimum	163	178	159	180	180
Srednia	106,41	100,01	100,13	96,23	99,69
σ	15,51	16,83	17,28	18,65	17,72
Mediana	107	100	101	98	101
(%) > 110	38,07	38,27	24,79	21,12	25,71
(%) > 120	14,81	10,24	10,12	7,54	9,92
(%) > 140	2,26	0,55	1,14	0,79	1,05
Zrealizowane bilety	3,91	3,55	3,70	3,31	3,56
Niezrealizowane bilety	0,28	0,43	0,38	0,37	0,37
Ruchy zabronione	7	9	8	22	46

Tablica 4.3: Wyniki uzależnione od ilości graczy dla sieci neuronowej

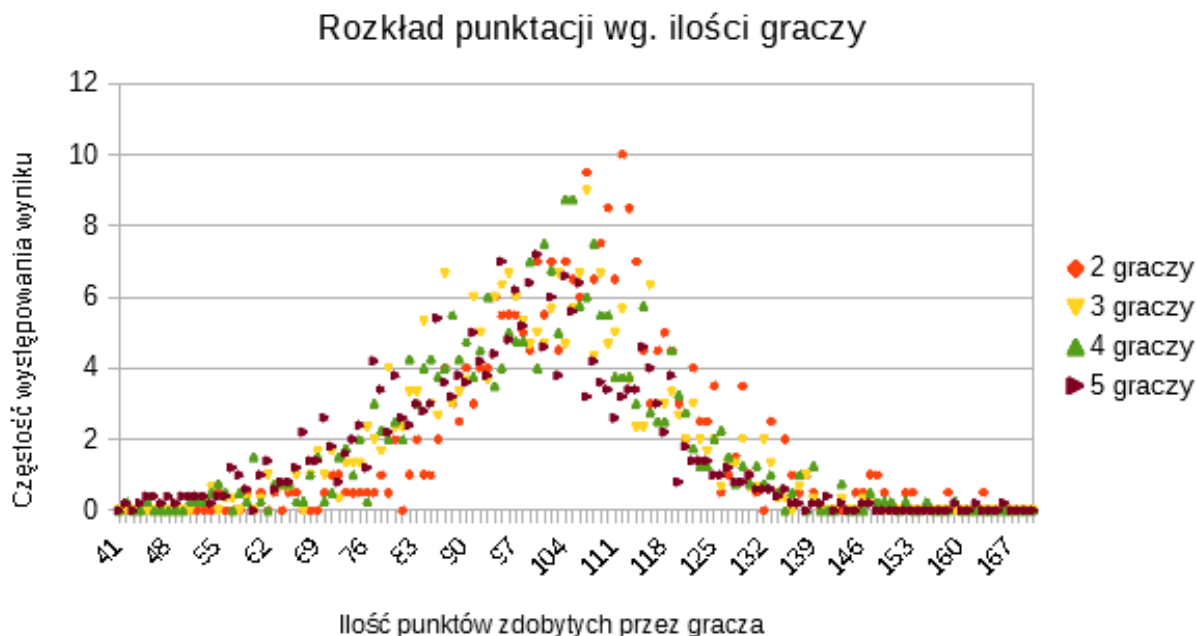
Analogicznie do modelu algorytmicznego wraz z wzrostem ilości graczy model radził sobie coraz gorzej. Widać to zwłaszcza w metryce *Ruchy zabronione*, który oznacza, że sieć wskazała decyzję, która w danej turze była niemożliwa do zrealizowania. Najwięcej takich sytuacji zaszło w przypadku rozgrywek dla 5 graczy.

W porównaniu do algorytmu, sieć radzi sobie porównywalnie, lub lepiej w rozgrywkach od trzech do pięciu graczy jednocześnie nieznacznie tracąc w rozgrywkach dwuosobowych. Można zaznaczyć, że średnia jest wyższa o 1,5 punktu zwycięstwa podczas gdy mediana o 2 punkty, czyli w typowej sytuacji 2% lepiej. Lepsza średnia jednocześnie została okupiona kosztem rozgrywek o bardzo dobrym wyniku. Porównując Rysunki 4.2 oraz 3.6 można zauważyć, że w przypadku rezultatów rozgrywek sieci neuronowej rozkład jest węższy, bardzo szybko rośnie częstość danego wyniku do maximum ok. 107 punktów. Następnie znacznie w porównaniu do modelu algorytmicznego szybciej spada.

Kolejną zauważalną analogią są bardzo zbliżone do siebie wyniki rozgrywek trój- oraz czterosobowych. W analizie konfiguracji trend malejącej średniej oraz mediany względem ilości graczy podobnie jak dla algorytmu jest malejący. Może zaskoczyć natomiast, że największe wyniki otrzymywane były w rozgrywkach 3 oraz 5 osobowych co możemy zaobserwować na Rysunku 4.3.

Na Rysunku 4.1 możemy natomiast zauważyć przebieg rozkładu punktów w zależności ilości graczy. Dla rozgrywek pięciograczowych wykres zaczyna rosnać najwcześniej, w okolicach 62 punktów zwycięstwa. Dla rozgrywek dla dwóch graczy wyniki zaczynają się częściej pojawiać dopiero od poziomu 83 punktów, i jednocześnie najbardziej się wybijają. Z dwóch pozostałych konfiguracji, ta dla czterech graczy ma bardziej ostry przebieg.

Ostatnią cechą rozgrywki, którą możemy ocenić na podstawie otrzymanych wyników jest średnia liczba zrealizowanych biletów - wyższy wynik oznacza lepsze poprowadzenie rozgrywki, oraz liczbę biletów, których nie udało się zrealizować. Pierwsza z tych metryk jest



Rysunek 4.1: Rozkład punktacji wg. ilości graczy dla sieci neuronowej

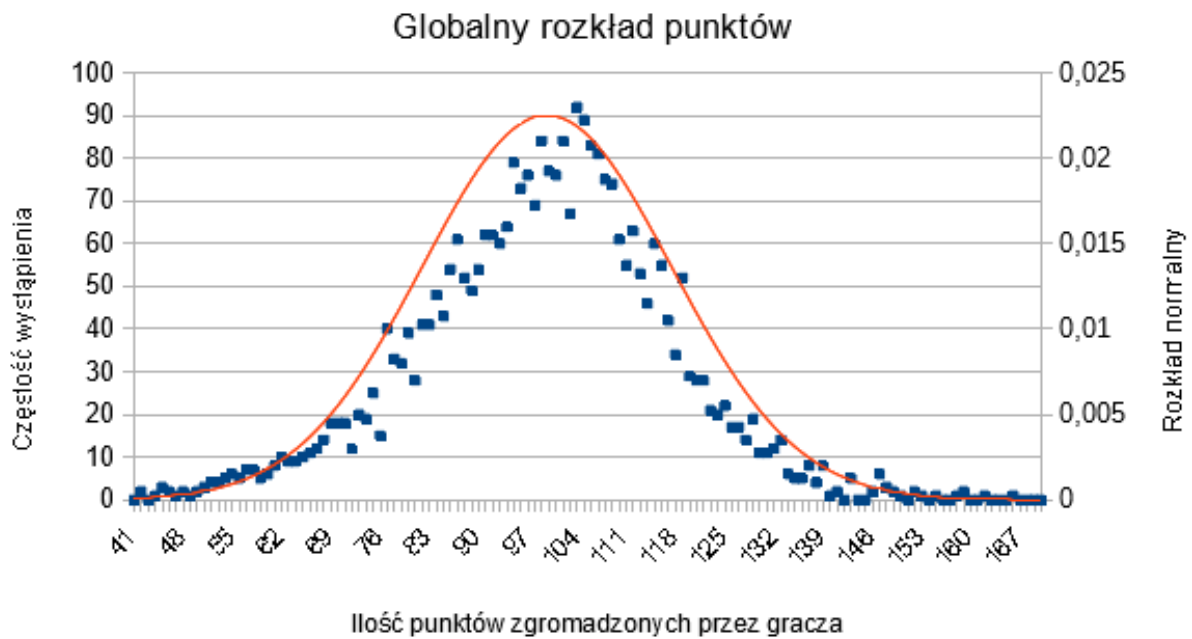
wyższa w przypadku modelu wykorzystującego sieci neuronowe aż o 0,5 biletu na gracza, co stanowi wzrost o 16% . Oznacza to, że przeciętnie gracz grał bardziej odważnie. Druga z metryk dotyczących biletów jest natomiast niższa w tym modelu o 0,2 bileta.

4.8 Wnioski na temat uczenia

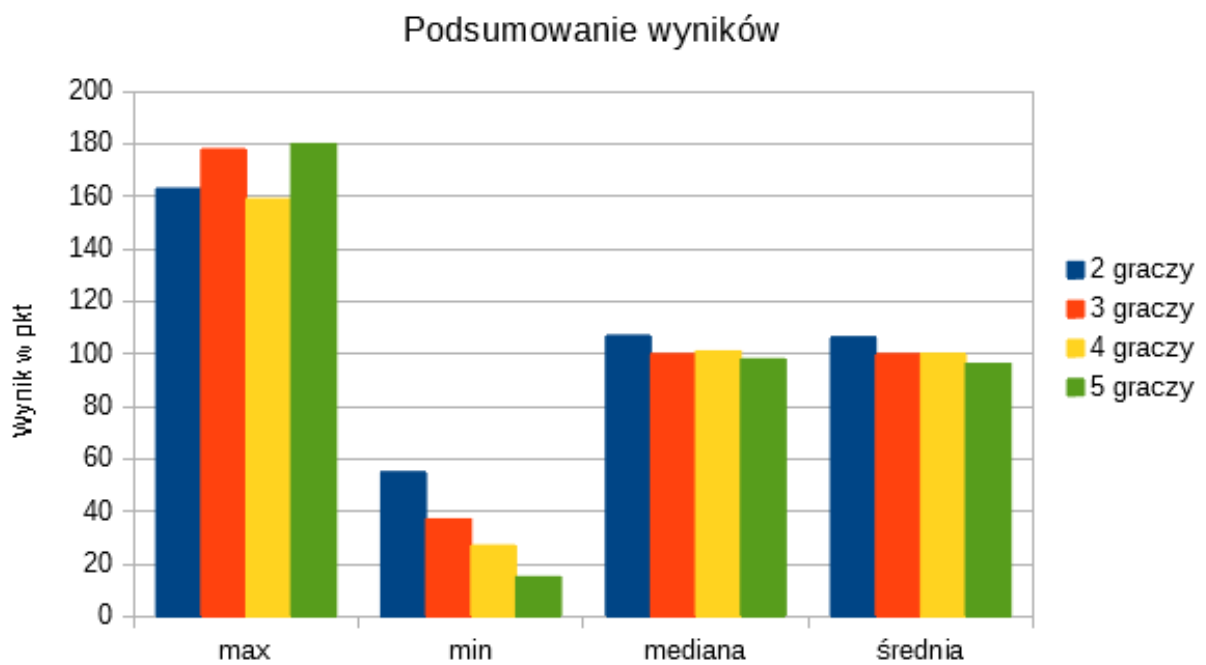
Zgodnie z podstawami uczenia maszynowego - dużo zależy od tego jaki jest model danych uczących oraz jakie jakościowo dane dostarczone są do sieci, którą chcemy nauczyć podejmować decyzje. Ograniczenia, jakie pojawiają się w przypadku zbioru danych uczących oraz słabości samych danych odbijają się na działaniu sztucznej sieci neuronowej i widać to na wspomnianym powyżej trendzie.

Lepsze średnie wyniki w symulacjach z wykorzystaniem sieci neuronowych są spowodowane tym, że jako źródło danych do zbioru uczącego wykorzystano zapisy partii z najwyższą zdobyczą punktową. Starano się by wyniku tych partii osiągnięto wynik końcowy co najmniej 130 punktów zwycięstwa. Można stwierdzić, że sieć osiągnęła poziom, który dla przeciętnego gracza byłby wyzwaniem (rekord gracza dla analizowanej konfiguracji gry 115 pkt).

Pomimo uzyskania wyższego średniego wyniku punkowego względem modelu algorytmicznego proces uczenia mógł odnieść lepsze rezultaty. Przykładowo, nie udało się sprawić, by w trakcie całej symulacji nie zostały podjęte decyzje klasyfikowane jako zabronione.



Rysunek 4.2: Globalny rozkład punktów dla sieci neuronowej



Rysunek 4.3: Wyniki wg. ilości graczy dla sieci neuronowej

Rozdział 5

Optymalizacja wyników i generalizacja sieci

5.1 Wprowadzenie do eksperymentu

Sztuczna inteligencja powinna w swoim działaniu dążyć do jak najlepszych wyników. W przypadku sztucznej sieci neuronowej najlepsze wyniki uzyskujemy poprzez dobór odpowiednich modeli do zbioru danych uczących. Sam dobór oraz wykorzystanie danych może się różnić w zależności od rozważanej sytuacji. W przypadku sztucznej inteligencji dla rozważanej gry możemy zastosować następujące podejścia:

1. Przygotować specjalny model dla każdej konfiguracji liczby graczy, każdy model ma swój zbiór danych uczących. Model zbudowany za pomocą danego zbioru rozgrywek zostanie wykorzystany jedynie w przypadku gier o takiej samej liczbie graczy. W dalszej części pracy nazwany **model specyficzny**
2. Przygotować wspólny zbiór danych uczących obejmujący wszystkie konfiguracje liczby graczy. W trakcie symulacji model wykorzystujemy w każdej rozgrywce - niezależnie od ilości uczestniczących graczy. W dalszej części pracy nazwany **model uniwersalny**

Motywacją dla porównania tych dwóch podejść jest sukces systemu tłumaczeń Google, który radził sobie lepiej, gdy szkolono go na wielu parach języków zamiast na jednej parze. W trakcie tej części swojej pracy chcę zbadać, który z wspomnianych modeli zapewni lepszy wynik punktowy w trakcie symulacji.

5.2 Przebieg doświadczenia

Na początku przygotowałem zbiory danych uczących dla każdej konfiguracji liczby graczy, jako klucz wybierając rozgrywki według najlepszego wyniku końcowego. Zbiór danych dla każdej konfiguracji liczby graczy składał się z zapisu decyzji trzydziestu rozgrywek. Dodatkowo na końcu wszystkie zbiory danych uczących zapisane zostały do dodatkowego pliku, który następnie zostanie wykorzystany do budowy modelu *uniwersalnego*.

Po przygotowaniu zbiorów danych uczących uruchomione zostało doświadczenie. Na początku przeprowadzenie uczenie sieci - 40 epok z rozmiarem batch'a wynoszącym 250 decyzji gracza. Dla każdej konfiguracji przeprowadzono po 60 rozgrywek - oddzielnie dla modelu uniwersalnego oraz modelu specyficznego. W sumie dla tej części przeprowadzono 480 gier testowych, których wyniki zamieszczono poniżej.

5.3 Omówienie wyników

Wyniki otrzymane w doświadczeniu zostały zawarte w Tablicach 5.1. W ostatniej kolumnie umieszczono wyniki podsumowujące wszystkie rozgrywki (niezależnie od liczby graczy).

Cecha	2 graczy	3 graczy	4 graczy	5 graczy	globalnie
Minimum	41	48	46	41	41
Maksimum	157	140	148	142	157
Srednia	101,66	99,4	97,36	92,05	96,57
Mediana	99,5	101	100	91	97
Ukończonych rozgrywek	60	55	60	56	231
(%) > 110	28,33	24,24	25,00	18,57	23,11
(%) > 120	18,33	10,30	8,75	8,21	10,31
(%) > 140	4,17	0,00	0,42	0,36	0,87
Zrealizowane bilety	467	575	873	951	2866
Niezrealizowane bilety	63	66	102	135	366

Tablica 5.1: Podsumowanie - modele operujące na specyficznych danych uczących

Cecha	2 graczy	3 graczy	4 graczy	5 graczy	globalnie
Minimum	45	49	54	37	37
Maksimum	154	133	144	143	154
Srednia	108,12	96,51	97,69	94,88	98,00
Mediana	108	97	99,5	96,5	99
Ukończonych rozgrywek	60	60	59	56	235
(%) > 110	43,33	24,44	23,73	21,07	25,86
(%) > 120	25,00	7,78	8,05	7,14	10,17
(%) > 140	3,33	0,00	1,27	0,71	1,10
Zrealizowane bilety	515	603	855	939	2912
Niezrealizowane bilety	41	90	83	111	325

Tablica 5.2: Podsumowanie - modele operujące na uniwersalnych danych uczących

Na podstawie osiągniętych wyników można wskazać, że lepsze wyniki gwarantuje model uniwersalny. Za kluczowe wskaźniki można uznać średni wynik oraz medianę punktów, które są wyższe w przypadku modelu uniwersalnego. Dodatkowo bardziej korzystna jest ilość dokonanych ruchów zabronionych, mniej ruchów w przypadku modelu uniwersalnego. Na korzyść

tego modelu jest również w przypadku każdej liczby graczy uczestniczących w rozgrywce procentowa ilość graczy, którzy przekroczyli wskazaną ilość punktów. Z drugiej strony model specyficzny ma wyższy najniższy wynik oraz wyższy najwyższy wynik(choć może być to fluktuacja).

Do ciekawszych wniosków prowadzą wyniki poszczególnych konfiguracji liczby graczy. Dla dwóch graczy uniwersalny model danych uczących (również wspólny model sieci) w trakcie symulacji osiąga znacząco lepsze rezultaty. Oczekiwana wartość punktowa jest wyższa w tym przypadku aż o 7 punktów, a w dodatku procentowy udział graczy z uzyskaną liczbą punktów wyższą od 110 punktów jest wyższy o 15 punktów procentowych. W tym modelu aż 43,33% wszystkich graczy osiąga co najmniej 110 punktów, a 25% zdobywa co najmniej 10 punktów więcej. W przypadku tej konfiguracji, wykorzystanie modelu uniwersalnego zapewni graczowi znacznie bardziej wymagającego przeciwnika.

Rekomendacja: Model uniwersalny

Sytuację, w której model precyzyjny osiąga lepsze rezultaty można zauważyć już w konfiguracji dla trzech graczy. Średnia zdobycz punktowa jest w tym przypadku lepsza o 3 punkty, a mediana o 4 punkty. Na korzyść modelu uniwersalnego działa natomiast ilość rozgrywek rozegranych bez wykonania ruchu zabronionego. Kluczowa dla wyboru rekomendacji jest średni oraz najwyższy wynik punktowy, mając na uwadze niebezpieczeństwo, że gracz zagra niezgodnie z zasadami gry.

Rekomendacja: Model precyzyjny

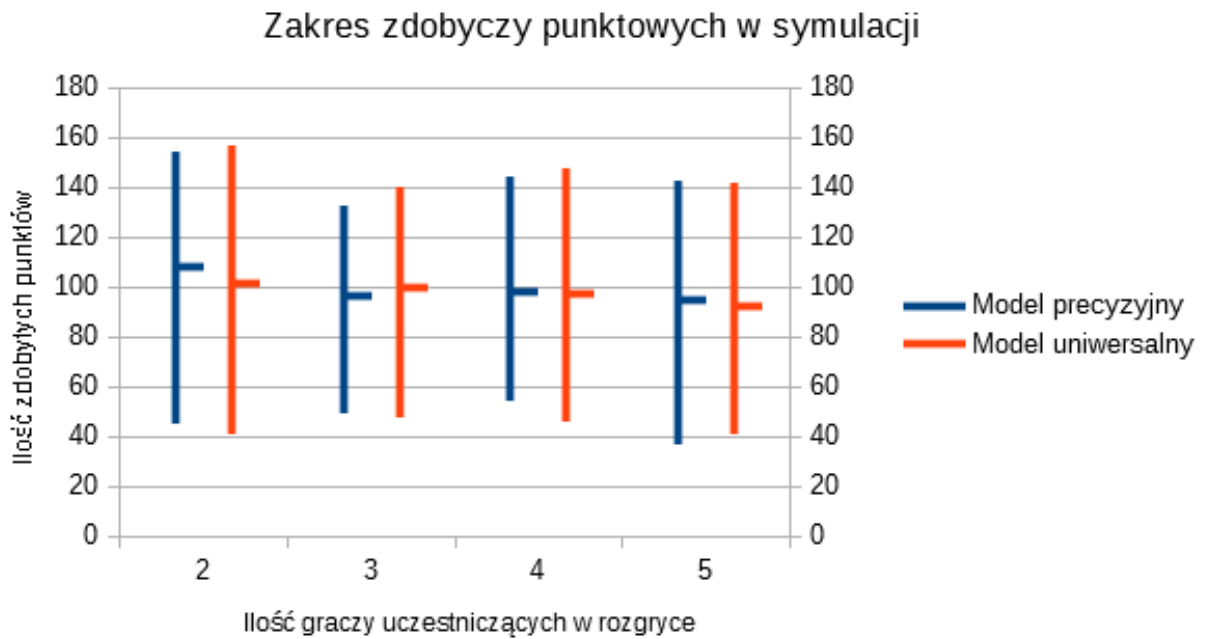
Dla kolejnej z konfiguracji możemy zauważyć zbliżone do siebie wyniki. Zarówno średnia jak i mediana wyników są bardzo zbliżone do siebie - 97,36 oraz 97,69. Procentowy udział wyników rozgrywek z wynikiem powyżej 140 punktów jest lepszy dla uniwersalnych danych uczących. Dla zwiększenia prawdopodobieństwa uzyskania wyniku powyżej 110 oraz 120 punktów należało wykorzystać model precyzyjny. Model uniwersalny w przypadku tej serii symulacji nie ukończył wszystkich rozgrywek. Ważnym czynnikiem w przypadku decyzji, które z podejść jest lepsze w tym przypadku jest liczba gier z zadowalającym wynikiem. Na korzyść przemawia model precyzyjny, dla którego jeden na czterech graczy zakończy rozgrywkę z 110 punktami lub więcej.

Rekomendacja: model precyzyjny

Ostatnią konfiguracją jest rozgrywka dla pięciu graczy. W tym przypadku lepsze wyniki osiągane są w przypadku modelu wyuczonego na uniwersalnych danych. Oczekiwana oraz mediana wyniku są wyższe o odpowiednio 2,83 oraz 5,5 punktów. Procentowy udział rozgrywek z wynikiem 110 oraz 140 punktów również działa na korzyść omawianego modelu. Dodatkowo, znacznie niższa jest liczba niezrealizowanych biletów.

Rekomendacja: model uniwersalny

5.4 Podsumowanie



Rysunek 5.1: Zakres punktów dla badanych modeli

Podsumowując aktualny rozdział można jednoznacznie określić, z którego z tych dwóch modeli lepiej będzie skorzystać. Model uniwersalny osiąga lepsze wyniki w kontekście wszystkich rozgrywek. Wyniki osiągane przy pomocy modelu uniwersalnego zawierają się w szerszym zakresie zdobytych punktowej co jest widoczne na Rysunku 5.1.

Rozdział 6

Dowód postępów działania sieci w procesie uczenia

6.1 Wprowadzenie do eksperymentu

Poza wskazaniem, że wyniki osiągane przez sieć neuronową są zbliżone do tych osiąganych przez algorytm należy udowodnić, że przygotowany model oraz założenia, jakimi się kierowałem przy projektowaniu sieci neuronowej są słuszne. W tym celu określono metryki, za pomocą których można opisać wyniki pracy sieci. Należy wskazać przede wszystkim, że wyniki osiągane w rozgrywkach sterowanych przez losowo zainicjalizowaną zainicja sieć neuronową są średnio gorsze od osiąganych przez *wyuczoną* sieć neuronową.

6.2 Model eksperymentu

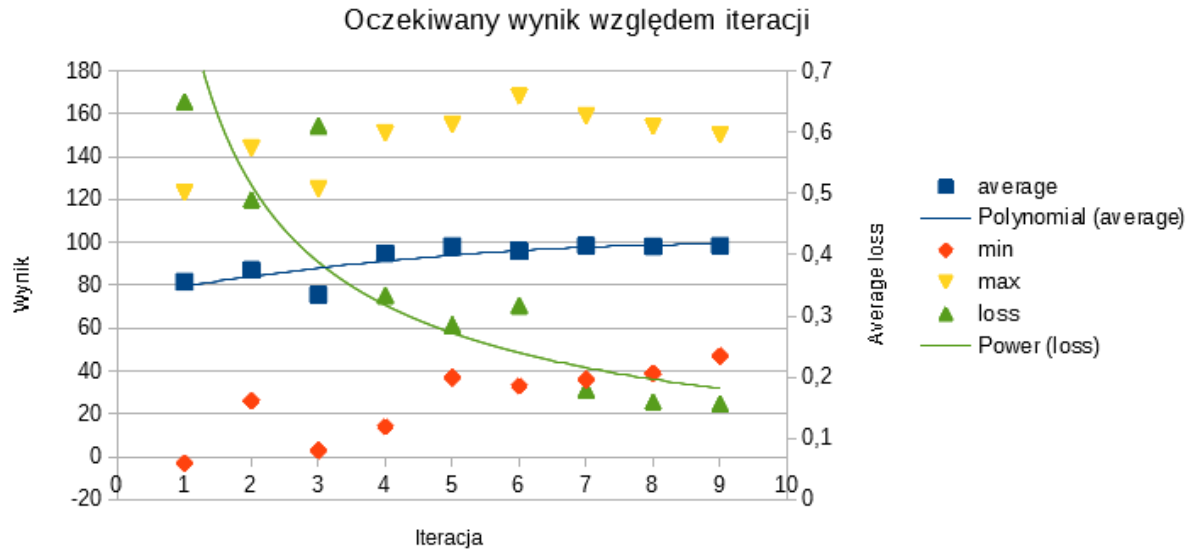
W celu udowodnienia działania modelu sieci neuronowej przeprowadziłem eksperyment w formie ewaluacji modelu w trakcie procesu uczenia sieci. Dla każdej epoki składającej się z uczenia za pomocą próbki zawierającej pięćset *zapisów decyzji* przeprowadziłem ewaluację modelu dla losowych 100 wpisów ze zbioru uczącego a następnie rozgrywałem sześćdziesiąt partii w każdej konfiguracji (od dwóch do pięciu graczy) liczby graczy co w sumie daje wynik 240 rozgrywek na iterację.

6.3 Wyniki i podsumowanie

Wyniki otrzymane w trakcie doświadczenia umieszczono w Tablicy 6.1.

W dowodzie pojawia się wyzwanie w ocenie postępów uczenia, gdyż rozważany problem nie jest trywialny i realna możliwość oceny sztucznej inteligencji pojawia się dopiero po rozegraniu całej partii trwającej średnio ok. 40 tur.

Najprostszym wyznacznikiem jest średnia zdobycz punktowa, która dla losowej konfiguracji nie skutkowałą żadną rozegraną do końca rozgrywką. W każdej z 240 gier, jeden z graczy



Rysunek 6.1: Rozkład punktacji w procesie uczenia

dokonał ruchu zabronionego. Natomiast po zaledwie jednej epoce gracz komputerowy wykorzystujący uczoną sieć neuronową i za jej pomocą wyznaczający swoje posunięcia był w stanie rozegrać partię do końca. Na 240 gier zaledwie jedna zakończyła się podjęciem decyzji zabronionej. Niestety wynik uzyskiwany przez tę sieć był względnie niski. Dla modelu algorytmicznego oczekiwana wynosiła 98,16 (W nawiązaniu do Tablicy 3.1), w tym przypadku wynik był aż o ok. 17 punktów niższy i wynosił 81,63 punkta. Niskie wartości miały również najniższy osiągnięty wynik (wartość ujemna spowodowana nieukończonymi biletami) jak i najwyższy wynik w całej serii rozgrywek.

Nie można wskazać dominującego trendu w dalszym procesie uczenia. Do szóstej iteracji procesu uczenia (próbka danych uczących - 600) średnia wartość wyników gracza reguranie wzrastała. Od piątej epoki oczekiwany wynik wynosił ok. 95 punktów, co można uznać za akceptowalny wynik w kontekście zbioru uczącego oraz dobry moment na zatrzymanie uczenia.

W trakcie procesu uczenia wzrastał natomiast najniższy osiągany wynik przez wszystkich graczy w wszystkich seriach testu. Natomiast wynik najwyższy utrzymywał się w granicach 150-160 punktów, który to wynik sprawiłby trudność doświadczonemu graczowi.

Dodatkowo, na podstawie rysunku 6.1 można zauważyć trend - rosnąca średnia wartość zdobyczy punktowej wraz z malejącą wartością średniej straty (parametr average loss). Na wykresie można dostrzec również powolną stabilizację najwyższego oraz najniższego wyniku, jaki osiągnęła sieć w trakcie symulacji gier. Są to oczywiście efekty oczekiwane w procesie uczenia, świadczące o coraz większej sprawności sieci.

Aktualna iteracja	0	1	2	3	4
Sumaryczna ilość danych uczących	0	100	200	300	400
Srednie odchylenie od próbki	7.614	0.649	0.490	0.610	0.333
Srednia pewnosc klasyfikacji	0.107	0.689	0.800	0.692	0.885
Liczba gier zakończonych przerwaniem (ruchem zabronionym)	240	1	32	4	47
Maksymalna liczba punktów	-	123	144	125	151
Minimalna liczba punktów	-	-3	26	4	47
Liczba graczy która ukończyła rozgrywki	0	835	702	824	654
Srednia zdobycz punktowa	-	81.63	87.23	75.69	94.80
Srednia liczba tur potrzebnych do zakonczenia rozgrywki	1.70	40.75	42.23	39.44	44.25
Srednia liczba ukonczonych biletów	-	2,21	2,39	1,9	3,08
Srednia liczba nieukonczonych biletów	-	0,43	0,33	0,54	0,32

Tablica 6.1: Wyniki pierwszych pięciu epok uczenia

Rozdział 7

Podsumowanie pracy

Przygotowanie sztucznej inteligencji dla gry *Wsiąść do pociągu* było pouczającym wyzwaniem. Do skutecznego działania konieczne było opracowanie algorytmów rozwiązujących grę w sposób klasyczny, a następnie w sposób bazujący na uczeniu maszynowym. Jako cel założono uzyskanie jak najwyższego oczekiwanego wyniku z możliwością wybicia się przy odpowiednim ułożeniu kart.

Dla redukcji złożoności problemów zrezygnowano z następujących elementów:

1. Dodatkowe dziesięć punktów dla posiadacza najdłuższej ścieżki
2. Interakcje polegające na blokowaniu sobie połączeń

oraz założono, że gracz zawsze stara się grać jak najlepiej dla swojego wyniku.

Opisywana w pracy gra jest skomplikowana z dużą ilością danych, które częściowo uznać można za losowe. Mamy do dyspozycji dwie różne talie - zasobów oraz biletów, problem grafowy oraz interakcje z innymi graczami. Trudnym zadaniem jest przygotować sztuczną inteligencję, która podejmie dobrą decyzję dla każdej możliwej sytuacji, nawet tej najbardziej niszowej oraz nieprzewidywalnej.

Otrzymane wyniki dowiodły, że zaproponowany algorytm może być wykorzystywany w trakcie rozgrywek oraz może posłużyć jako schemat działania, który można wykorzystać dalszych modelach gracza sterowanego siecią neuronową. Szczególnie ciekawym nie zbadanym modelem jest sieć rekurencyjna LSTM, która mogłaby być szkolona na sekwencjach rozgrywek i proponować kolejne posunięcia. Podejście sekwencyjne odnosi spore sukcesy w generowaniu tekstu i predykcji szeregów czasowych, można zatem mieć nadzieję, że sprawdziłyby się w przypadku gry planszowej.

Podsumowując można podkreślić, że mimo niejawnego sposobu reprezentacji decyzji sieć neuronowa radzi sobie nie gorzej niż algorytm rozgrywający grę klasycznie.

Rozdział 8

Słownik

Gracz Uczestnik rozgrywki - sterowany przez sztuczną inteligencję. Przygotowano poniższych uczestników

Model algorytmiczny Sztuczna inteligencja działająca w oparciu o algorytm. Wykorzystany do uzyskania danych uczących

Sieć neuronowa Sztuczna sieć neuronowa - klasyfikuje rodzaj decyzji jaka ma zostać podjęta przez gracza

Decyzja Podejmowana przez uczestnika rozgrywki decyzja (rezerwacja połączenia, pobranie kart wagonów lub pobranie kart biletów)

Decyzja zabroniona Podjęta przez gracza decyzja niemożliwa w danym momencie z punktu widzenia gry. *Przykładowo brakujące zasoby gry*

Bilet Losowane przez uczestnika rozgrywki zadanie do zrealizowania. Określony przez dwa miasta, które gracz musi ze sobą połączyć (za pomocą zarezerwowanych połączeń). W przypadku udanego zrealizowania zadania gracz zdobywa określoną liczbę punktów. W przeciwnym przypadku punkty są odejmowane od zdobytej puli.

Połączenie Połączenie dwóch sąsiadujących ze sobą miast. Połączenia charakteryzują się ilością nitek (jedna lub dwie), kolorystyką (jeden z ośmiu kolorów, lub połączenie bezbarwne) oraz długością. Można zbudować graf wykorzystując miasta jako węzły oraz połączenia jako krawędzie.

Sciezka/nitka/tor Jest to najmniejszy element połączenia zajmowany wyłącznie przez jednego z graczy. Jest opisywany przez indywidualny kolor. Jest przypisany do jednego połączenia, a jedno połączenie może mieć jeden (zwykłe połączenie) lub dwa (połączenie podwójne) tory.

Wagon Element (w fizycznej wersji gry) służący do znakowania zarezerwowanych połączeń. Jeden wagonik odpowiada jednej karcie wagonów służącej do zarezerwowania połączenia. Gracz na początku posiada 45 wagonów. W momencie, gdy jeden z graczy ma 2 wagony lub mniej gracze wykonują jeszcze po jednym ruchu po czym przechodzą do fazy podliczenia punktów na koniec rozgrywki.

Karta wagonu Element rozgrywki. W grze występuje jako jeden z 8 zasobów (kolorów) - po 10 sztuk, oraz karta joker (zastępuje dowolną kartę zasobów) - w ilości 12 sztuk. Gracz wykorzystuje karty zasobów do zrealizowania połączenia.

Stan gry Stan rozgrywki składający się z informacji o planszy (w tym kolejność ułożenia niewidocznych kart) oraz informacji o graczach. Stan gry jest przedstawiony w sposób pozbawiony nadmiarowych informacji - np. wagony przedstawione są jako licznik, a kolor gracza nie ma znaczenia - jedynie jego identyfikator.

Punkt (zwycięstwa) Jest to wynik gracza. Punkt zwycięstwa zdobywa się za realizację biletów oraz rezerwacje połączeń między miastami.

Rozdział 9

Bibliografia i odniesienia

1. A Few Useful Things to Know about Machine Learning
Dostęp: 6 września 2018
Pedro Domingos
Department of Computer Science and Engineering
University of Washington
<https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf>
2. Instrukcja do gry
Dostęp: 6 września 2018
https://www.wydawnictworebel.pl/repository/files/instrukcje/WdP_USA.pdf
3. Omówienie botów do gry - forum Board Game Geek
Dostęp: 6 września 2018
<https://boardgamegeek.com/thread/1523665/ai-project-solo-multiplayer-games>
4. Repozytorium projektu
Dostęp: 6 września 2018
<https://github.com/paqaos/msc-ticket-to-ride-nn-ai>
5. Wprowadzenie do algorytmów (Introduction to Algorithms)
Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L.
1997, Wydawnictwa Naukowo-Techniczne, Warszawa
6. Strona projektu Tensorflow
Dostęp: 12 września 2018
<https://www.tensorflow.org>

Spis rysunków

1.1	Plansza wykorzystana w rozgrywce	5
3.1	Schemat blokowy algorytmu podejmującego decyzje o typie wykonywanego ruchu	11
3.2	Schemat blokowy algorytmu podejmującego decyzje o wyborze najlepszego podzbioru biletów	12
3.3	Schemat blokowy algorytmu podejmującego decyzje o wyborze karty wagonu do pobrania	13
3.4	Schemat blokowy algorytmu rezerwującego najlepsze połączenie	14
3.5	Rozkład punktacji wg. ilości graczy dla algorytmu klasycznego	15
3.6	Globalny rozkład punktów dla algorytmu klasycznego	15
3.7	Wyniki wg. ilości graczy dla algorytmu klasycznego	17
4.1	Rozkład punktacji wg. ilości graczy dla sieci neuronowej	24
4.2	Globalny rozkład punktów dla sieci neuronowej	25
4.3	Wyniki wg. ilości graczy dla sieci neuronowej	25
5.1	Zakres punktów dla badanych modeli	29
6.1	Rozkład punktacji w procesie uczenia	31

Spis tablic

1.1	Ilość punktów otrzymywanych za zrealizowanie połączenia	5
2.1	Parametry rozgrywki	9
2.2	Konfiguracja danych testowych	9
3.1	Wyniki uzależnione od ilości graczy dla algorytmu klasycznego	16
4.1	Dane wejściowe sieci neuronowej	20
4.2	Klasy określone przez klasyfikator	20
4.3	Wyniki uzależnione od ilości graczy dla sieci neuronowej	23
5.1	Podsumowanie - modele operujące na specyficznych danych uczących	27
5.2	Podsumowanie - modele operujące na uniwersalnych danych uczących . . .	27
6.1	Wyniki pierwszych pięciu epok uczenia	32