26.05.2023, Gdańsk

***Sprawozdanie***

**Przewidywanie stanu pandemii przy pomocy sieci neuronowej typu feed-forward**

**Oskar Wilda**

**Jakub Wojtalewicz**

**Piotr Wesołowski**

***1. Wstęp teoretyczny***

**Modelowanie cząsteczkowe:**

Modelowanie cząsteczkowe, zwane również symulacją cząsteczkową, to technika, która pozwala na badanie zachowań i właściwości układów cząsteczek poprzez symulację ich interakcji na poziomie indywidualnych cząsteczek. Może być wykorzystywana do modelowania różnych systemów, w tym również do symulacji epidemii. Symulacja epidemii polega na modelowaniu rozprzestrzeniania się choroby w populacji, biorąc pod uwagę interakcje między ludźmi oraz ich ruchy a także współczynniki zaraźliwości, śmiertelności i wytrzymałości populacji.

**Zastosowanie modelowania cząsteczkowego do naszej symulacji epidemii:**

**Definicja cząstek:** Każda jednostka w populacji, na przykład osoba, może być reprezentowana jako pojedyncza cząstka w symulacji. Każda cząstka ma swoje właściwości, takie jak położenie, wektor prędkości, stan zdrowia (np. zdrowa, zakażona, wyleczona, martwa) oraz inne parametry.

**Śmiertelność cząstek:** Cząstka na każdym etapie życia może umrzeć a szansa na to zwiększa się wraz z wiekiem czy w wyniku zarażenia. Dodatkowo domyślnie osobnik zostaje zarażony bezobjawowo a ewentualne objawy mogą wystąpić z prawdopodobieństwem zależnym min. od wieku. W naszej symulacji prawdopodobieństwo śmierci zdrowego osobnika i średni czas życia cząstek jest wzorowany na autentycznym prawdopodobieństwie śmierci zdrowego obywatela Polski1.

**Ruch cząstek:** Cząsteczki mogą poruszać się w symulacji, co odzwierciedla ruch jednostek w rzeczywistości. W naszej symulacji zaimplementowaliśmy losowy sposób poruszania się cząsteczek oparty na zmianie wartości kąta obrotu w czasie, tak aby sam ruch wyglądał jak najbardziej naturalnie. Jednostki poruszają się ze zmienną prędkością, zależną od ich stanu zdrowia. Osobniki respektują kolizję pomiędzy sobą oraz brzegami mapy.

*1 Korzystamy z autentycznych danych dotyczących prawdopodobieństwa śmierci obywatela Polski w danym wieku udostępnionych przez stronę Głównego Urzędu Statystycznego z roku 2021.*

**Rozprzestrzenianie się choroby**: Na podstawie interakcji między cząstkami oraz ich ruchu, można symulować rozprzestrzenianie się choroby w populacji. Na przykład, jeśli zakażona osoba ma kontakt z inną osobą w określonym zasięgu, istnieje szansa na przekazanie wirusa. Nasza symulacja uwzględnia czynniki takie jak zdolność zarażenia się przez osobę zdrową oraz przez osobę, która już przebyła tę chorobę.

**Analiza wyników:** Po przeprowadzeniu symulacji można analizować jej wyniki, takie jak liczba zakażonych, niezakażonych, wyleczonych czy martwych. Analiza wyników może dostarczyć informacji na temat tempa rozwoju epidemii i pomóc w podejmowaniu decyzji mających na celu zminimalizowania tragicznych skutków epidemii.

**Sieć neuronowa typu feedforward:**

Sieć neuronowa typu feedforward to jedna z podstawowych architektur sztucznych sieci neuronowych, w której informacja przepływa tylko w jednym kierunku, od wejścia do wyjścia, nie ma w nich sprzężenia zwrotnego między neuronami w ramach tej samej warstwy ani między kolejnymi warstwami.

Przepływ danych odbywa się przez serię warstw, składających się z neuronów a każdy z nich w danej warstwie otrzymuje dane wejściowe, wykonuje pewne obliczenia i przekazuje wynik jako wyjście do neuronów kolejnej warstwy. Takie przekazywanie informacji odbywa się aż do ostatniej warstwy, która zwraca ostateczny wynik.

W naszej sieci mamy warstwę wejściową składającą się łącznie z M x N neuronów. Gdzie M to ilość danych, a N to ilość cech. W naszym przypadku mamy 9 cech. 5 z nich to współczynniki prawdopodobieństwa oraz 4 które są liczbą stanu zdrowia ludzi:

*Wszystkie poniższe współczynniki są z zakresu (0,1) i są wymnażane przez odpowiednie wartości prawdopodobieństw dla konkretnego wieku osobnika.*

***Warstwa inputu:***

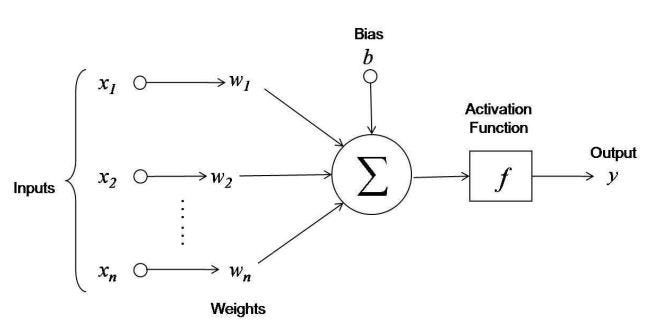
* Współczynnik prawdopodobieństwa śmierci dla osoby zarażonej,
* Współczynnik prawdopodobieństwa zarażenia się przez osobę zdrową,
* Współczynnik prawdopodobieństwa zarażenia się przez osobę wyzdrowiałą,
* Współczynnik prawdopodobieństwa wyleczenia się przez osobę zarażoną z objawami,
* Współczynnik prawdopodobieństwa wystąpienia objawów u osoby zarażonej,
* Liczba niezarażonych,
* Liczba zarażonych,
* Liczba wyleczonych,
* Liczba martwych,

Porównując różne możliwości zauważyliśmy, że nasza sieć działa najlepiej w zadowalającym czasie dla 2 warstw ukrytych, każda składająca się z 64 neuronów. Obliczane są w nich kombinacje liniowe wejść z odpowiednimi wagami i biasami. Następnie na wynikach tych kombinacji stosujemy funkcję aktywacji, wykonaliśmy próby dla funkcji **RELU**, **sigmoid** oraz **Tanh**. W przypadku ostatnich dwóch wartość losu nie schodziła poniżej niemalże 3000 co spowodowane było tym, że wartości wejściowe były tak duże lub tak małe, że po przekształceniu sigmoidalnym dążą one do wartości granicznych (0 lub 1), a gradient na tym obszarze jest bardzo bliski zeru. Z tego powodu oraz dla przyspieszenia treningu używaliśmy funkcji RELU. Warstwy ukryte pomagają sieci neuronowej wykryć bardziej skomplikowane wzorce i relacje w danych.

Ostateczne wyniki obliczeń w warstwie wyjściowej są interpretowane jako prognozy liczby stanu zdrowia ludzi które wykorzystywane są jako warstwa wejściowa w kolejnych latach symulacji.

**Propagacja wsteczna:**

Podczas procesu uczenia, wagi i biasy w sieci neuronowej są aktualizowane przy użyciu propagacji wstecznej, czyli algorytmu optymalizacji gradientowej, aby minimalizować funkcję kosztu, która mierzy różnicę między prognozami sieci a oczekiwanymi wartościami. W celu przyspieszenia czasu treningu modelu zaimplementowaliśmy algorytm Adam który pomaga aktualizować w trakcie iteracji wartość współczynnika learning rate. Dzięki wielokrotnemu (20000) iteracyjnemu dostosowywaniu wag i biasów sieć neuronowa jest w stanie dostosować się do danych treningowych i stworzyć model zdolny do dokonywania prognoz na nowych danych wejściowych które dzielimy w proporcjach 20:80.



gdzie:  
 - dane wejściowe i-tego wiersza - waga i-tego wiersza  
b - bias (obciążenie) z - waga danego neuronu.

∑ - wynik

f – funkcja aktywacji przyjmująca ∑ \* b a zwracająca wartość z zakresu (0,1) czyli y

**Rysunek poglądowy:**

Obraz zawierający diagram, krąg, linia

Opis wygenerowany automatycznie

**Powody wybrania sieci typu feedforward:**

* Sieci neuronowe feedforward są zdolne do modelowania złożonych zależności między różnymi czynnikami, co jest niezwykle przydatne w przypadku analizy stanu epidemii,
* Sieć feedforward jest zdolna do uczenia się na podstawie dużej ilości danych, a przewidywanie pandemii wymaga sporej ilości danych,
* Sieci neuronowe typu feedforward są elastyczne i mogą przyjmować wiele wejść jednocześnie, co w naszym przypadku pozwala na wejściu uwzględnić wiele parametrów, które posłużą do przewidywania rezultatów epidemii,
* Stan każdego następnego roku symulacji jest konsekwencją stanu roku poprzedniego. Do przewidywania takiej zależności idealnie nadaje się sieć feedforward

***2. Pozyskiwanie danych***

Do pozyskania danych posłużyliśmy się narzędziem Unity. Napisaliśmy własną symulację przebiegu epidemii, która na wejściu przyjmowała kilka różnych parametrów mających rozróżnić sposób przebiegu epidemii.

**Nasze parametry:**

* Prawdopodobieństwo zarażenia
* Prawdopodobieństwo śmierci zarażonej osoby
* Prawdopodobieństwo pojawienia się objawów u osoby chorej
* Prawdopodobieństwo na wyleczenie osoby z objawami
* Prawdopodobieństwo ponownego zarażenia po wyzdrowieniu

*Powyższe parametry mnożyliśmy przez pewną skalę (różną dla każdego parametru), żeby przy każdej symulacji uzyskać różniącą się od wszystkich innych epidemię.*

Każdej cząsteczce został przypisany wiek (w przedziale 0-40), który zwiększa się z czasem. Prawdopodobieństwo poszczególnych zdarzeń jest zależne od wieku cząsteczki (np. czym cząsteczka starsza tym większa szansa, że się zarazi).

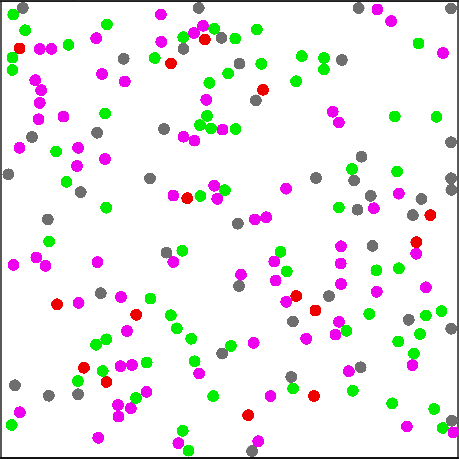
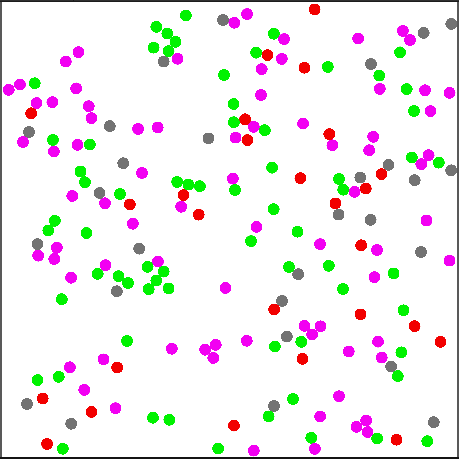
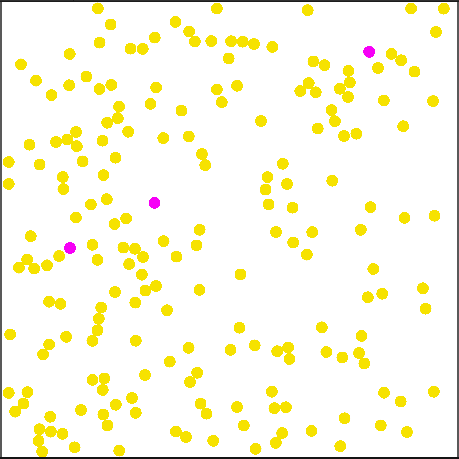
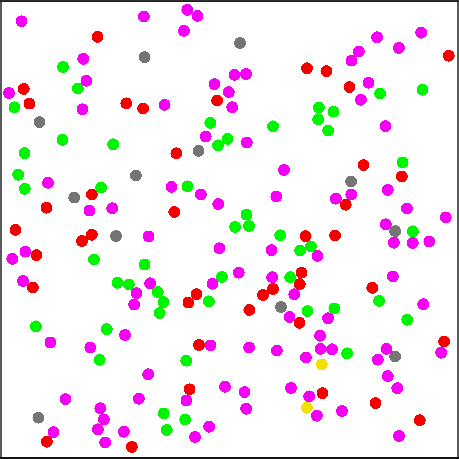
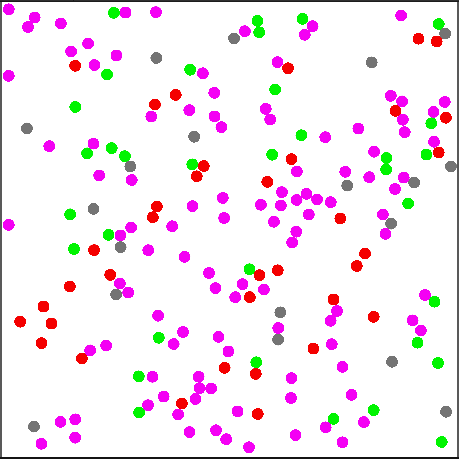
**Działanie symulacji:**

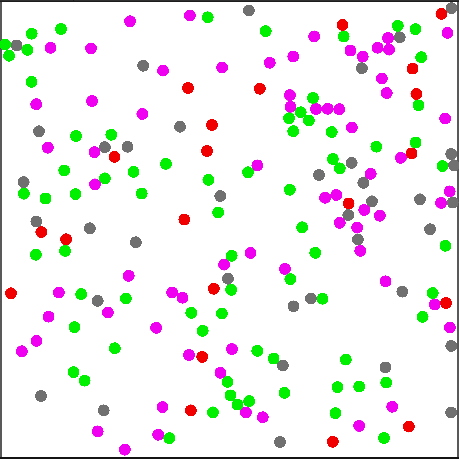
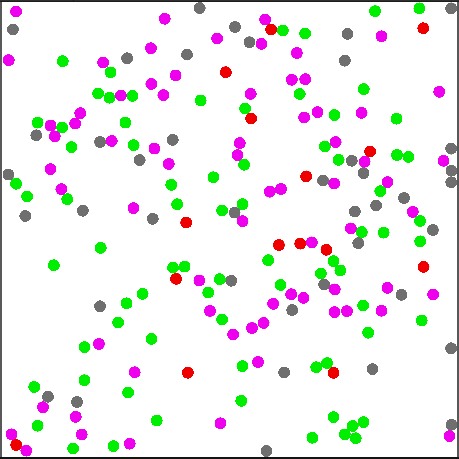
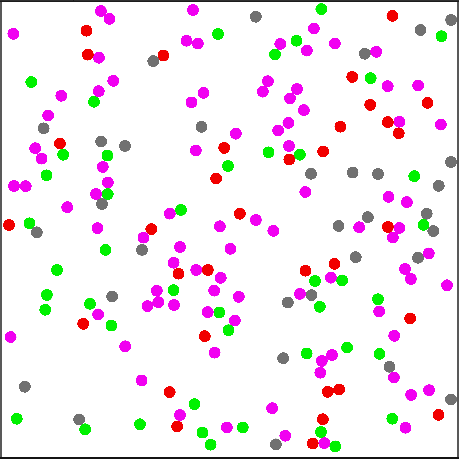
Do zwizualizowania przebiegu epidemii skorzystaliśmy z metody **modelowania cząsteczkowego** opisanej w pierwszej części sprawozdania**.** Każdą symulację przeprowadzaliśmy na 200 cząsteczkach z których na początku 3 były chore a pozostałe 197 zdrowe. Cząsteczki poruszały się w czasie zderzając się ze sobą. Symulacja przedstawiała przebieg epidemii w okresie 10 lat od jej rozpoczęcia. Podczas jednego „roku” cząsteczka 4 razy podejmowała każdą z prób opisanych w parametrach (jeżeli mogła taką czynność wykonać; jak miała objawy to próbowała się leczyć, zarazić inne cząsteczki lub umrzeć).

Przygotowaliśmy 3 różne zbiory danych. Dwa liczące około 170 000 przypadków i jeden liczący około 80 000.

**Przykład symulacji dla parametrów:**

Skala śmierci chorego: 0,05 | Skala zarażenia: 0,30 | Skala ponownego zachorowania: 0,10

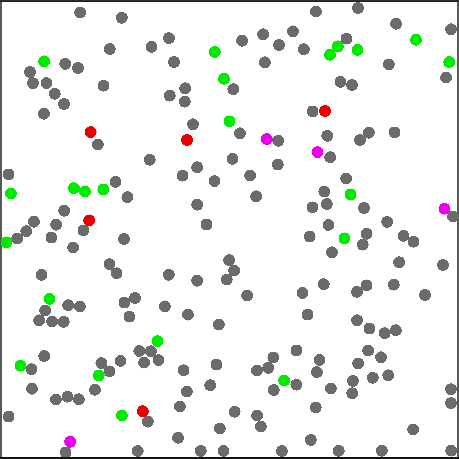
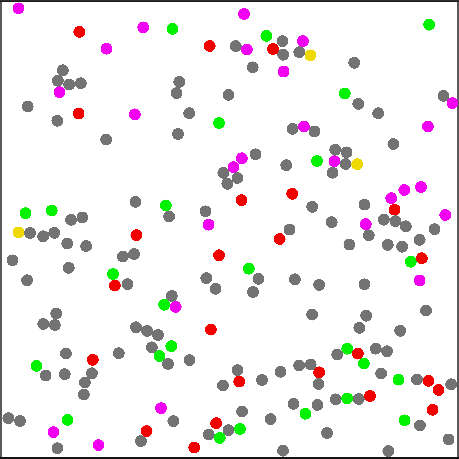
Skala wyzdrowienia: 0,70 | Skala pojawienia się objawów: 0,75

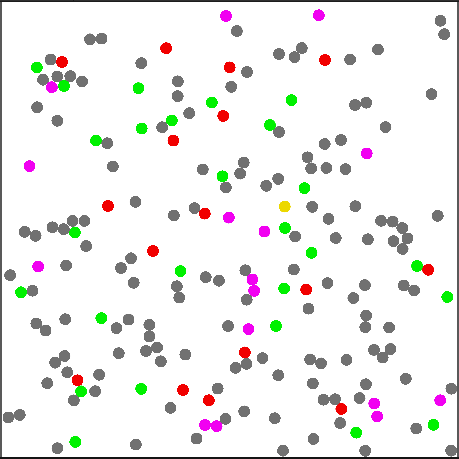
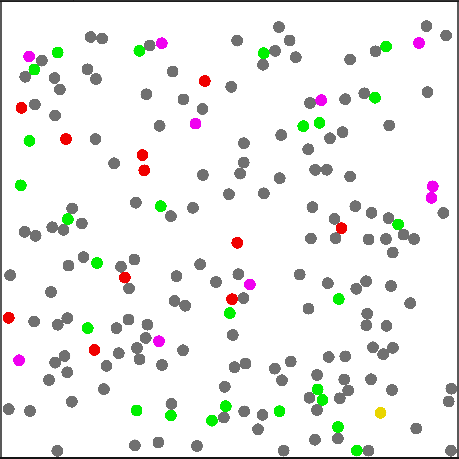
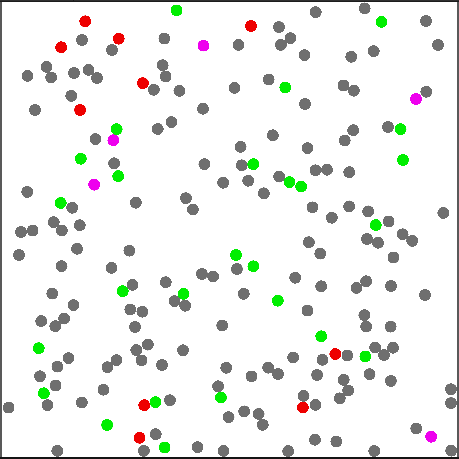
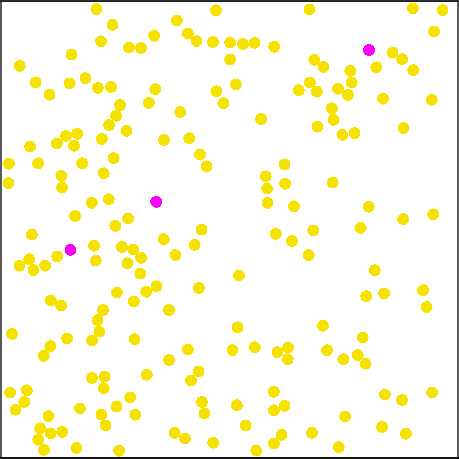
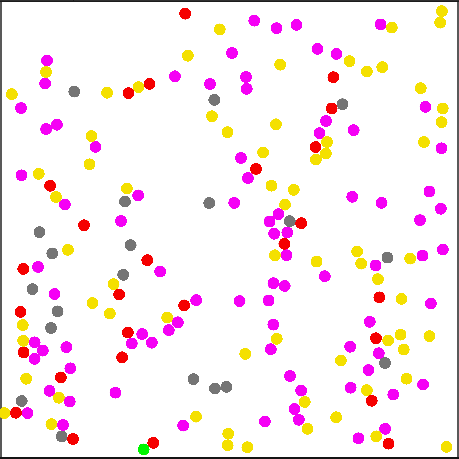
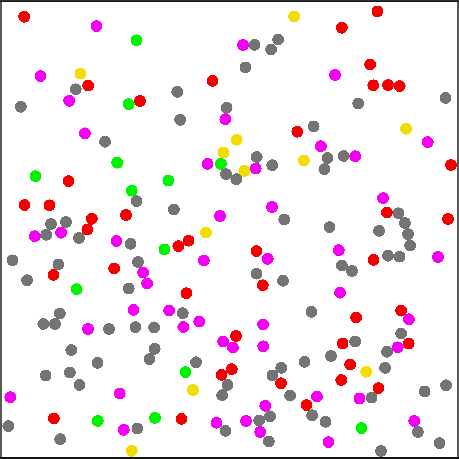
****

Parametry zostały w taki sposób dobrane, żeby przeważała liczba częsteczek które wyzdrowiały. I tak jak można zauważyć, od około 4 roku liczba cząsteczek po chorobie przeważa.

**Natępny przykład (bardzo śmiertelna choroba):**

Skala śmierci chorego: 0,5 | Skala zarażenia: 0,20 | Skala ponownego zachorowania: 0,20

******Skala wyzdrowienia: 0,20 | Skala pojawienia się objawów: 0,75

******

***3. Analiza problemu***

**Sposoby analizy epidemii:**

W naszym projekcie testowaliśmy 3 typy danych różniące się inputem i outputem oraz losowością symulowania epidemii. W naszej analizie staraliśmy się uwzględnić jak bardzo czynnik losowy jest w stanie zaburzyć przewidywanie stanu końcowego przez stworzoną przez nas sieć typu feedforward.

**Pierwszy typ(każda symulacja ma inny seed):**

* Cząsteczki zaczynają symulację zawsze w innym (losowym) miejscu,
* Tor ruchu cząsteczki jest inny w każdej symulacji,
* Input uwzględnia parametry choroby, a także stan symulacji na wejściu (ilość zdrowych, chorych itp.),
* Output zwraca stan symulacji po 1 roku (ilość zdrowych, chorych itp.),

**Drugi typ(każda symulacja ma taki sam seed):**

* Cząsteczki zaczynają symulację zawsze w tym samym miejscu,
* Tor ruchu cząsteczki jest taki sam dla każdej symulacji,
* Input uwzględnia parametry choroby, a także stan symulacji na wejściu (ilość zdrowych, chorych itp.),
* Output zwraca stan symulacji po 1 roku (ilość zdrowych, chorych itp.),

**Trzeci typ(każda symulacja ma inny seed + inny input oraz output):**

* Cząsteczki zaczynają symulację zawsze w innym (losowym) miejscu,
* Tor ruchu cząsteczki jest inny w każdej symulacji,
* Input uwzględnia parametry choroby, a także położenie, kierunek ruchu oraz stan w jakim znajduje się każda z cząsteczek,
* Output zwraca położenie, kierunek ruchu oraz stan w jakim znajduje się każda z cząsteczek po jednym roku symulacji,

*W dalszej części będziemy się głównie skupiać na typie drugim*

**Struktura naszej sieci:**

Testowaliśmy działanie sieci dla różnych wartości neuronów w warstwach ukrytych oraz różnych wartości learning rate. Model, który dał nam najlepsze wyniki wygląda w następujący sposób:

* 9 neuronów warstwy wejściowej,
* 2 warstwy ukryte, każda po 64 neurony,
* 4 neurony warstwy wyjściowej,
* ReLU jako funkcja aktywacyjna dla warstw ukrytych,
* początkowy learning rate wynoszący 0.05 który był modyfikowany przez algorytm Adam,

**Nauka sieci:**

Zbiór danych podzieliliśmy na dane treningowe oraz dane testowe w proporcji 80:20. Sieć trenowaliśmy dla różnych parametrów ostatecznie wyznaczając te dające najlepszy rezultat.

**4. Wyniki**

*Zastanawialiśmy się jaki wskaźnik błęd€ będzie najlepiej oddawał dokładność naszej symulacji. Z jednej strony preferowaliśmy ocenę błędów w skali oryginalnych danych do czego idealnie nadaje się Absolute Mean Error (MAE), jednakże w celu lepszego odwzorowania większego błędu i ich wariancji, dodatkowo zaimplementowaliśmy Root Mean Square Error (RMSE). Oba wskaźniki są wartością oczekiwaną „błędu”, czyli różnicy między estymatorem a wartością estymowaną.*

Poniższe dane pokazują różnice pomiędzy błędem w zależności od przyjętego wskaźnika. Wynika z nich, że średnio największe błędy wychodzą w estymacji liczby zarażonych lub niezarażonych co bardzo często przyjmuje duże wartości oraz dodatkowo każde odchylenie w szacowanej ilości zarażonych wpływa na błąd w ilości niezarażonych i vice versa. Dlatego więc RMSE będzie zawyżany z racji na duże wartości tych odchyleń co z kolei niweluje używanie wskaźnika MAE.

Obraz zawierający tekst, zrzut ekranu, Czcionka, menu

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznieZ racji na fakt, że zdecydowaliśmy się na korzystanie z MAE do oceny jakości dopiero po treningu naszej sieci a dokonywaliśmy obliczeń dla 20 000 iteracji, które trwały wiele długich godzin zdecydowaliśmy się na pewne oszacowanie. Na podstawie 1000 iteracji dla naszych danych doszliśmy do wniosku, że MAE który wykorzystamy do oceny dokładności naszych przewidywań to około 64% RMSE.

*W dalszej części sprawozdania do porównywania jakości naszych modeli przez pryzmat otrzymywanego błędu korzystamy z* **mean square error** (MSE). Jednakże procentowa wartość dokładności naszego modelu obliczana jest wg wzoru 1 – (4\*MAE/N) \*100% gdzie N = 200 z racji, że każda wartość wynikowa jest ograniczona maksymalną ilością jednostek w symulacji.

**Pierwszy model (Każda symulacja ma inny seed):**

**Zbiór danych:**

Nasz pierwszy zbiór danych zawierał 168 070 danych. Podzieliliśmy go w stosunku 20:80 na zbiór testowy i zbiór treningowy. Dane zostały zebrane z symulacji wykonanej w Unity. Symulacje zostały wykonane dla 5 parametrów, których wartości były kombinacją liczb, 0.11, 0.22, 0.33, 0.44, 0.55 i 0.66.

**Przykład:**

Jedna z symulacji: 0.44, 0.55, 0.11, 0, 0.22

Następna symulacja: 0.55, 0.55, 0.11, 0, 0.22

*Każda kolejna symulacja miała zwiększany pierwszy parametr aż do osiągnięcia limitu, po jego osiągnięciu dany parametr był zerowany i zwiększany był następny parametr.*

**Nauka modelu:**

Model był uczony przez 20 000 epok. Wykres błędu uzyskiwanego przez nas podczas nauki sieci prezentuje się następująco:

Obraz zawierający tekst, zrzut ekranu, Czcionka, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, menu

Opis wygenerowany automatycznie

Otrzymany loss MSE ≈ 40 oznacza, że średni błąd dla danej wartości wynosi . Dokładność przewidywań z roku na rok wynosi około:

1 – ((0,64 \* 6,3 \* 4) / 200) \* 100 % = 92,44%

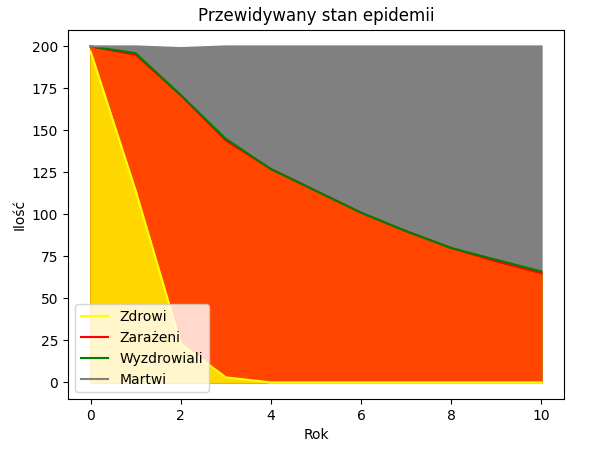
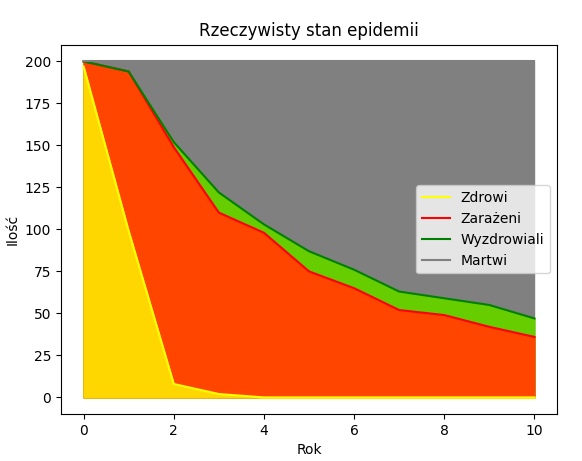
Wychodzi na to, że nasz model jest w stanie w pewnym stopniu przewidzieć stan epidemii dla następnego roku, ale naszym celem było przewidzenie stanu na 10 lat mając tylko do dyspozycji parametry wejściowe.

**Przewidywanie na 10 lat:**

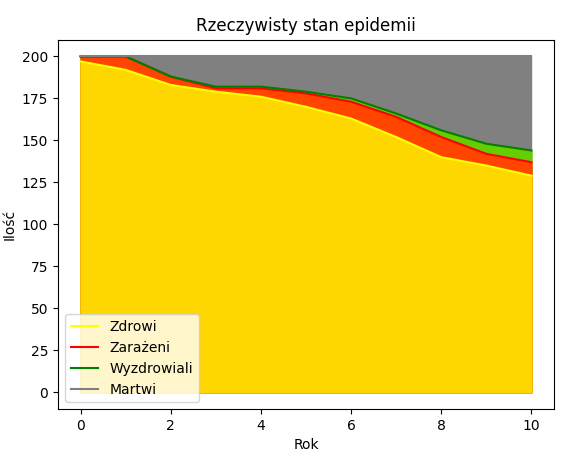
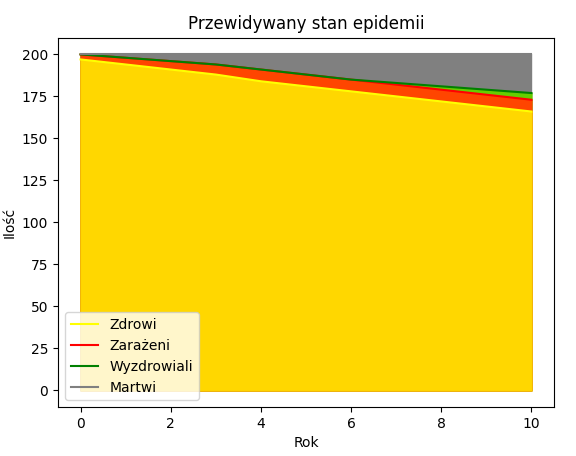
Sytuację na 10 lat można przewidzieć przewidując stan na drugi rok i te dane, które otrzymamy wstawić jako dane dla warstwy wejściowej dla dalszych przewidywań. Taki sposób nie jest pozbawiony problemów, które mogą wynikać z narastającego błędu. Przewidywania z każdym rokiem mogą odbiegać coraz bardziej od wartości prawdziwej. Jeżeli sieć w pierwszym roku pomyli się o nawet niewielką wartość to w kolejnych latach błąd rośnie w sposób wykładniczy, a po 10 latach może być już nie do zaakceptowania.

*Poniżej przedstawiamy kilka przewidzianych przez sieć stanów rozwoju epidemii. Dla różnych danych wygenerowanych oddzielnie.*

**Przykłady przewidywań:**

*Śmiertelność: 0,21 | Zaraźliwość: 0,73 | Ponowne zarażenie: 0,73 | Wyzdrowienie: 0,21 | Objawy: 0,21*

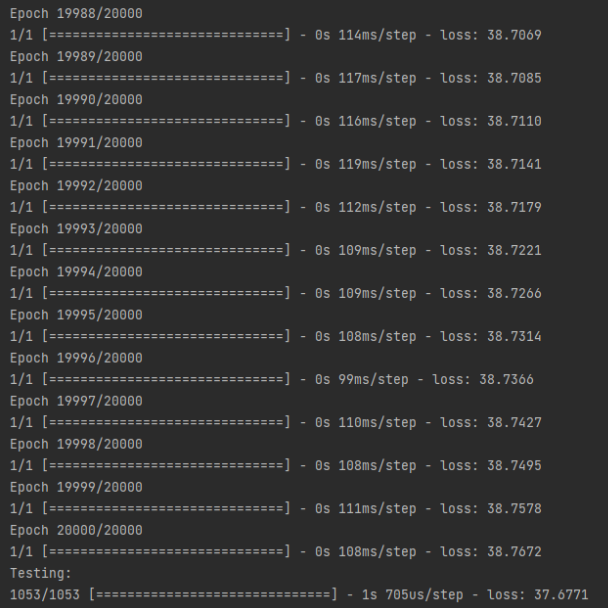
Jak można zauważyć sieć dobrze poradziła sobie z przewidzeniem stanu epidemii. Widząc, że zaraźliwość jest duża sieć wiedziała, że liczba zarażonych gwałtownie wzrośnie. Wysoki poziom szansy na ponowne zarażenie spowodował, że osoby, które wyzdrowiały szybko ponownie chorowały, przez co liczba wyzdrowiałych pozostawała niewielka.

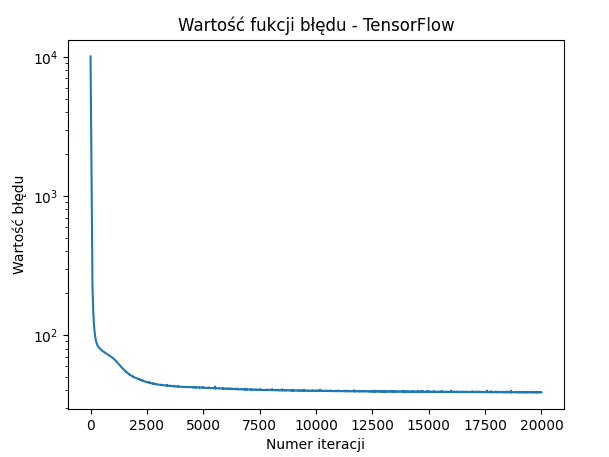
*Śmiertelność: 0,73 | Zaraźliwość: 0,21 | Ponowne zarażenie: 0,21 | Wyzdrowienie: 0,47 | Objawy: 0,21*

Z racji bardzo wysokiej śmiertelności i porównywalnie niskiej zaraźliwości choroba nie zdążyła się rozprzestrzenić. Co roku ilość chorych nie przekraczała 15 osób. Sieć znając parametry była w stanie to przewidzieć.

**Porównanie z tensorflow :**

Dla porównania skuteczności naszego modelu stworzyliśmy tak samo działający model za pomocą tensorflow. Wszystkie parametry sieci pozostawiliśmy takie same: ilość warstw ukrytych, ilość neuronów, algorytm Adama oraz funkcje aktywacyjne.

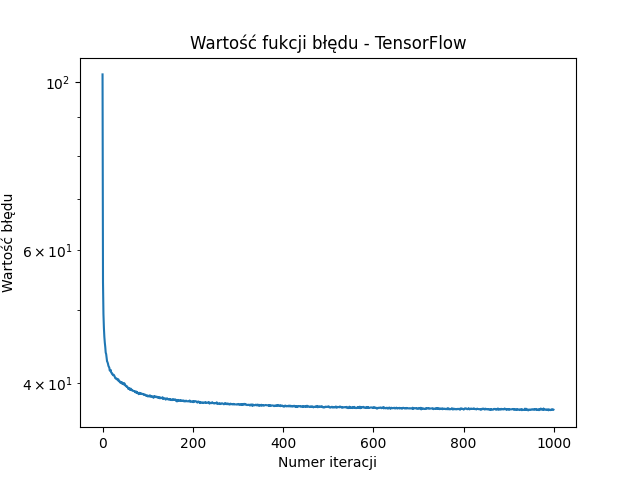
**Oto wyniki, które udało mu się uzyskać dla 20 000 epok:

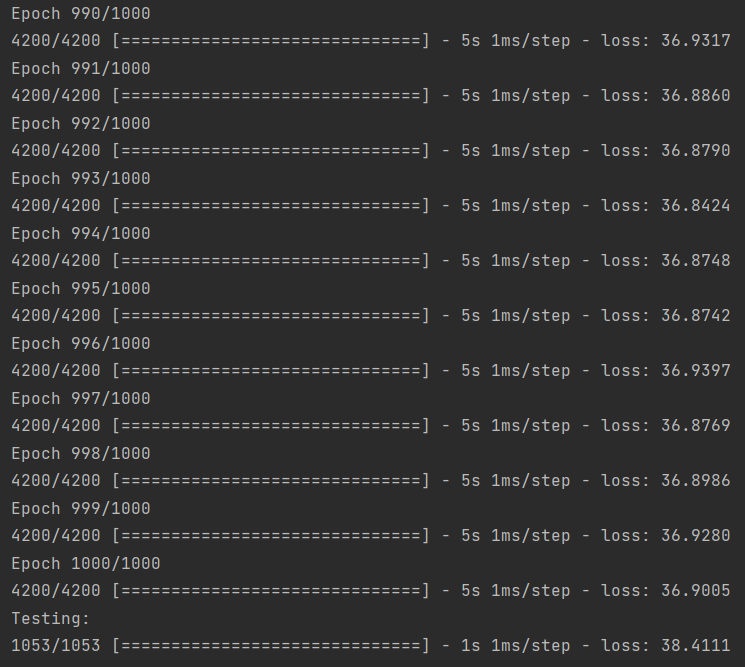


Wartość błędu uzyskanego przez tensorflow różni się tylko nieznacznie od uzyskanej w naszym modelu (nasz model: 39, tesnsorflow: 37 co daje dokładność na poziomie około 92,7% z roku na rok). Ta wiadomość była do przewidzenia, ponieważ tak zaimplementowany tensorflow działa na bardzo podobnej zasadzie jak stworzony przez nas model. Jedynymi różnicami w działaniu tych dwóch kodów jest czas wykonywania oraz stabilność wartości błędu.

Tensorflow wykonuje się ponad 2 razy szybciej i przy każdej iteracji zwraca błąd mniejszy lub nieznacznie większy od wyznaczonego dla poprzedniej iteracji. Model stworzony przez nas czasami po iteracji zwraca kilkukrotnie większy błąd MSE niż iterację wcześniej. Może to być spowodowane tym, że tensorflow korzysta z wielu różnych technik regularyzacji, które pomagają w uniknięciu przeuczenia modelu. W naszym modelu nie zaimplementowaliśmy żadnej z takich technik.

**Zmodyfikowany tensorflow:**

Porównaliśmy także działanie tensorflow z domyślnym **batch size = 32**. Wyniki, które uzyskaliśmy prezentują się następująco:



Wynikiem takiej zmiany było wydłużenie czasu trwania każdej iteracji, ale błąd już po kilku pierwszych był porównywalny do błędu dla kilkuset iteracji z poprzedniego przypadku.

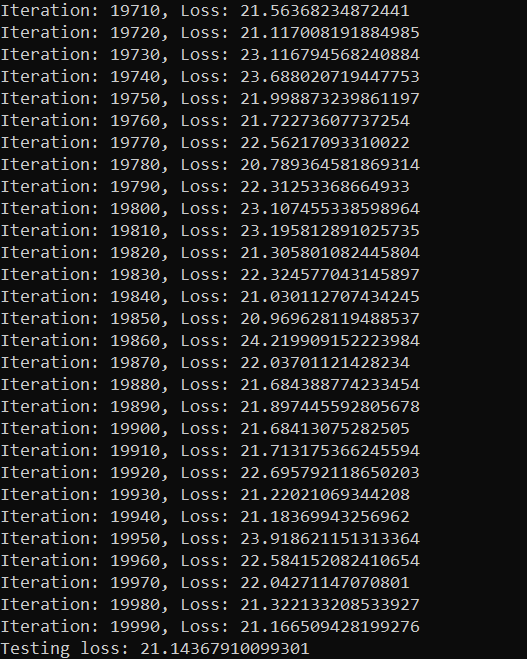
**Końcowy błąd po 1000 iteracjach jest porównywalny do błędu dla poprzedniego modelu tensorflow.**

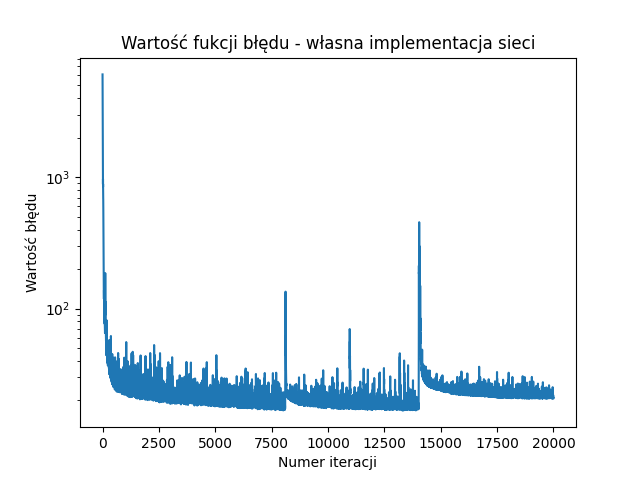
**Drugi model (Każda symulacja ma taki sam seed):**

**Zbiór danych:**

Nasz drugi zbiór danych zawierał 168 070 danych. Podzieliliśmy go tak samo jak zbiór pierwszy na zbiór testowy i zbiór treningowy 20:80. Symulacje w Unity zostały wykonane dla 5 parametrów, których wartości były kombinacją liczb 0.05, 0.16, 0.27, 0.38, 0.49, 0.6 i 0.71. Tym razem dane były pobierane dla przypadku, w którym ruch cząsteczek we wszystkich symulacjach był taki sam.

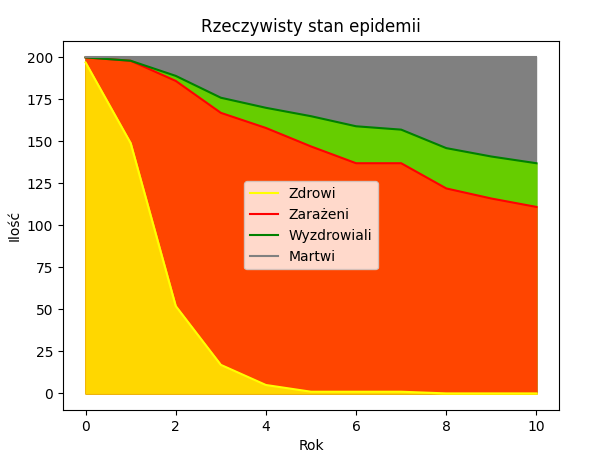
**Nauka modelu:**

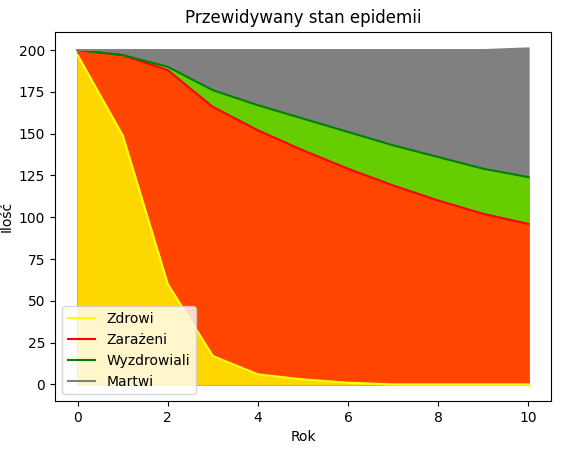
Tak jak w poprzednim przypadku model był uczony przez 20 000 epok. Wykres błędu uzyskiwanego przez nas podczas nauki sieci prezentuje się następująco:



Jak można zauważyć błąd dla tego przypadku wyszedł prawie dwukrotnie mniejszy. Oznacza to, że sieć lepiej była w stanie sobie poradzić z przewidywaniem stanu epidemii w takiej sytuacji. , a więc na podstawie estymacji można ocenić, że MAE ≈ 2,7, a więc dokładność naszej sieci wynosi 94,6%.

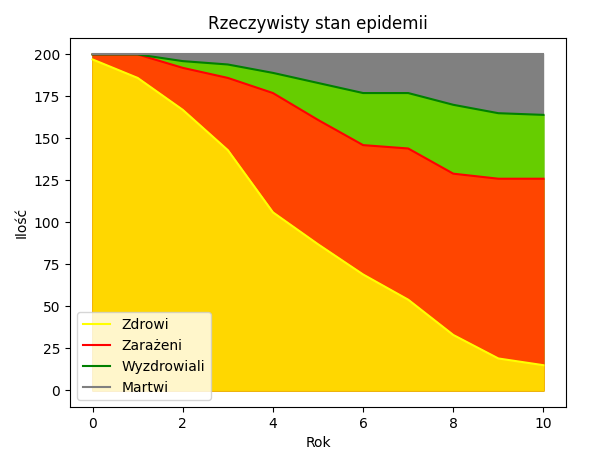
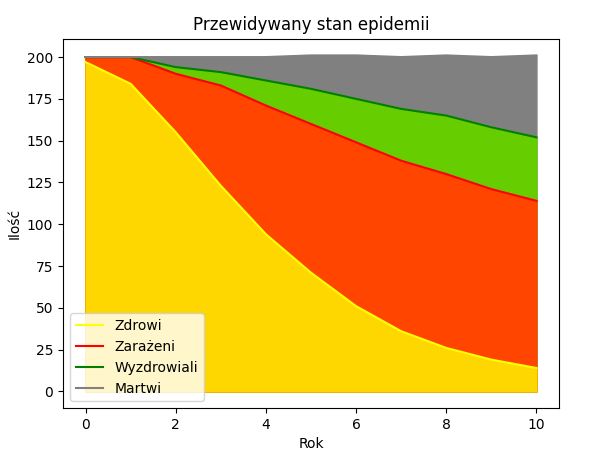
**Dla tego przypadku także spróbujemy przewidzieć stan na 10 lat:**

*Śmiertelność: 0,73 | Zaraźliwość: 0,21 | Ponowne zarażenie: 0,21 | Wyzdrowienie: 0,47 | Objawy: 0,21*



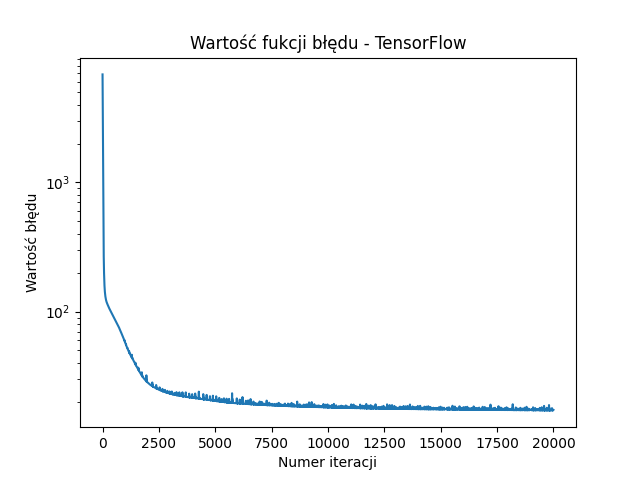
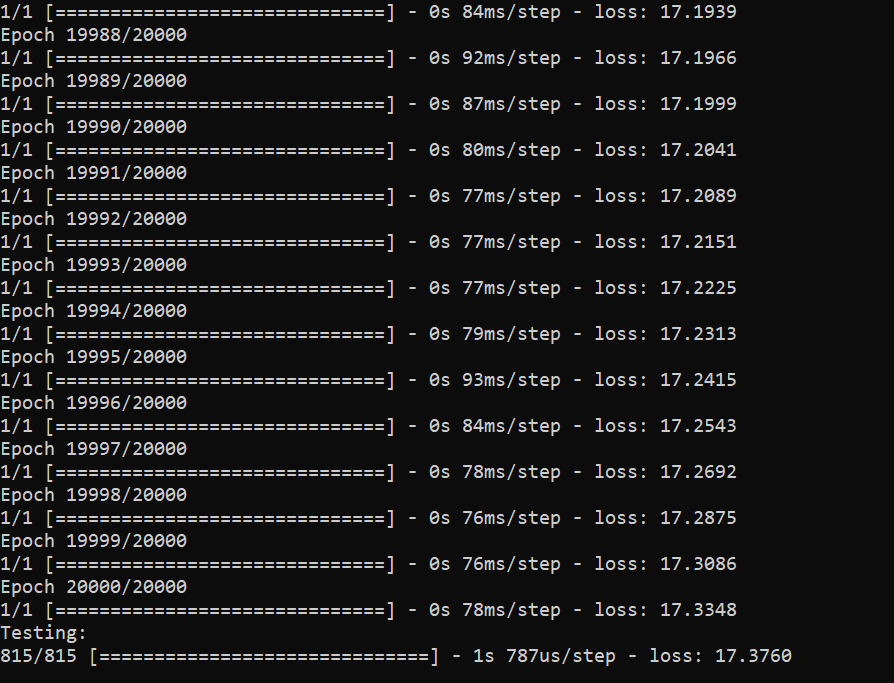
Przewidziany stan epidemii wygląda bardzo podobnie jak rzeczywisty stan epidemii. Loss na poziomie 20 pozwala na dokładniejsze przewidywanie stanu na każdy rok.

*Śmiertelność: 0,73 | Zaraźliwość: 0,21 | Ponowne zarażenie: 0,21 | Wyzdrowienie: 0,47 |*

*******Objawy: 0,21*

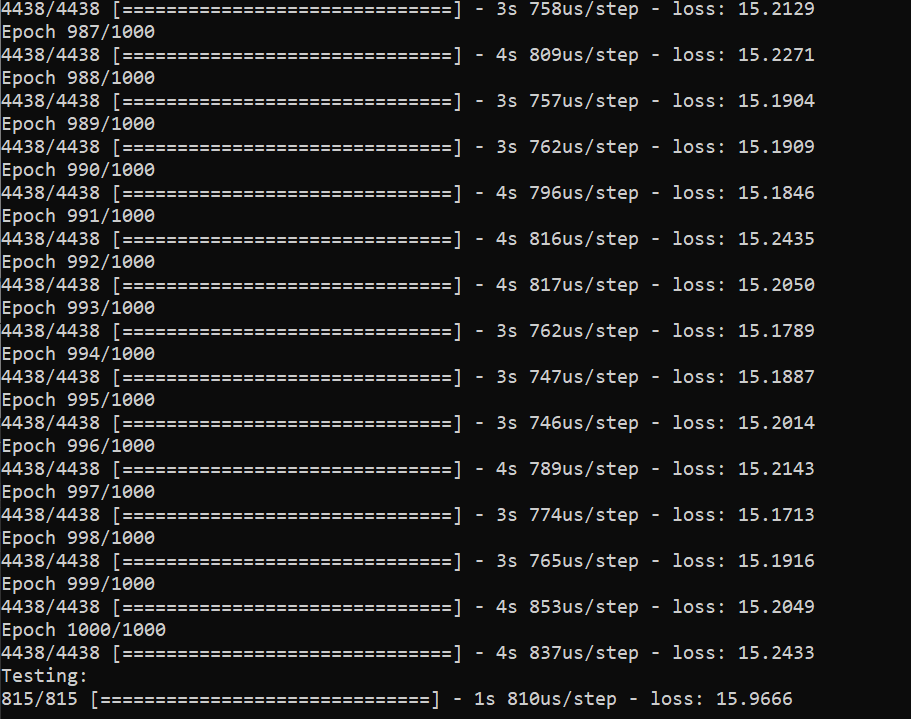
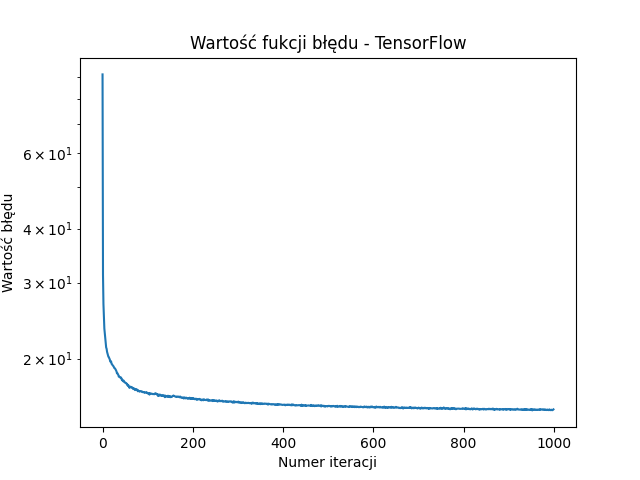
Dla tego przypadku także zrobiliśmy porównanie z modelem stworzonym za pomocą biblioteki tensorflow.

**Wyniki przez niego osiągnięte prezentują się w następujący sposób:**



Błąd uzyskany w tym przypadku jest niewiele mniejszy od błędu uzyskanego używając naszego własnego modelu. Loss wyniósł 17,37 co daje dokładność wynoszącą **95,0%** rdr.

**Zmodyfikowany tensorflow:**

Tak samo jak poprzednio dla **batch size = 32**.

W tym przypadku dla 1000 iteracji udało się uzyskać rekordowo niską wartość lossu wynoszącą 15,96 co przekłada się na **95,2%** rdr.

**Trzeci model:**

**Zbiór danych:**

Nasz drugi zbiór danych zawierał około 80 000 danych. Podzieliliśmy go tak samo jak zbiór pierwszy na zbiór testowy i zbiór treningowy 20:80. Odbiegał on w znaczny sposób od poprzednich 2 modeli.

Jako input zostały podane parametry choroby (5 takich jak poprzednio) oraz **macierz 200x7,** w której każdy rząd zawierał parametry innej cząsteczki. Kolumny zawierały odpowiednio: pozycję w X, pozycję w Y, obrót w osi Z, cztery pola stanu (chory, zdrowy, wyzdrowiały, martwy). Jeżeli dana cząsteczka była w którymś z tych stanów to w inpucie była podawana w tym miejscu 1 a w pozostałych 0.

Ilość warstw ukrytych była taka sama, ale ilość neuronów w nich się znajdujących była równa 700 dla każdej warstwy.

Jako output zwracaliśmy **macierz 200x7** z wartościami przedstawiającymi to samo co w macierzy inputu, ale wyznaczone po roku trwania symulacji.

**Nauka sieci:**

Z racji bardzo dużego inputu, outputu oraz dużej ilości neuronów w warstwach ukrytych nauka sieci trwała bardzo długo, dlatego musieliśmy wykonać ją na mniejszej ilości iteracji.

Pomimo pozornie niskiej wartości lossu błąd jest mocno zauważalny, a ostateczne wyniki odbiegają całkowicie od rzeczywistych. Niska wartość lossu jest spowodowana między innymi małym zakresem wartości danych.

Niestety ten sposób nie przyniósł pożądanych rezultatów. Jest to najpewniej spowodowane złym podejściem do analizy tego problemu. Analizowaliśmy położenie oraz wektor zwrotu każdej z cząsteczek, a w każdej z symulacji cząsteczki zaczynały z innego miejsca. Taki sposób jest zbyt skomplikowany dla sieci neuronowej. Lepszym pomysłem mogłoby być podzielenie mapy na mniejsze sektory i analizowanie ilości chorych i zdrowych w każdym sektorze.

**5. Dyskusja**

**Model pierwszy:**

Model pierwszy był najbardziej realistycznym podejściem. Wyniki przez niego uzyskiwane były w pewnym stopniu zadowalające. Dokładność rdr na poziomie 92% można uznać za satysfakcjonującą. Niestety czynnik losowości czasami utrudniał przewidywanie, ponieważ wyniki jednej symulacji mogą się różnić od wyników drugiej wykonanej dokładnie dla takich samych parametrów czego powodem jest losowość w ruchu każdej z cząsteczek.

*Wiek cząsteczek, który wpływał na szanse śmierci, zarażenia oraz wyzdrowienia także był zależny od czynnika losowości, co dodatkowo utrudniało przewidywanie.*

**Model drugi:**

Dzięki wyeliminowaniu czynnika losowości model drugi uzyskiwał bardziej dokładne wyniki od modelu pierwszego. Dokładność rdr wynosząca **95%** jest wysoką dokładnością, ale otrzymywane prognozy nie miałyby zastosowania dla przewidywania prawdziwej epidemii.

**Model trzeci:**

Mimo dobrych przewidywań nie spełnił on swojej roli. Dokładniejsze przewidywanie epidemii przy pomocy przemieszczenia, szybkości oraz stanu każdej z cząsteczek jest zbyt skomplikowaną rzeczą. Gdybyśmy chcieli w tak dokładny sposób przewidzieć stan świata to trzeba by było uwzględnić gęstość zaludnienia, bliskość chorych w stosunku do zdrowych lub inne podobne parametry, użyć do tego innej architektury sieci bądź dać jej o wiele więcej czasu i mocy obliczeniowej na trening.

**Ogólne spostrzeżenia:**

Nauczona przez nas sieć pomimo tego, że nie zawsze wskazuje dokładny stan zdrowia badanej grupy osób, jest w stanie stwierdzić jak i czy choroba, z którą mamy styczność rozwinie się w czasie kilku najbliższych lat. Dzięki takiej sieci możemy na przykład przewidzieć, czy choroba będzie w stanie zarazić wystarczającą ilość osób przed wygaśnięciem spowodowanym śmiercią wszystkich zarażonych.

Zadowalająca na pierwszy rzut oka dokładność na poziomie 90-kilku procent już po 5 latach spada do 75% a po 13 latach wynosi **około 50%.** Określenie szansy na wyleczenie, lub zachorowanie jest rzeczą ciężką, ale jest to możliwe przy dużym zbiorze danych. A wykorzystanie tych informacji może mieć bardzo duży wpływ na podjęcie przełomowych decyzji w tak ważnym aspekcie jakim jest zdrowie i życie ludzkie. Czasami oczywiście nie da się przewidzieć wszystkiego i nawet najlepsza sztuczna inteligencja nie będzie w stanie dać dokładnych prognoz, jednakże każdy wysiłek włożony w zapobieganie skutkom epidemii jest rzeczą godną podziwu i zasługującą na szacunek.