

Sztuczna Inteligencja - Projekt. Wizualizacja rozprzestrzeniającej się epidemii

maj 2023

1 Wprowadzenie

Celem projektu jest stworzenie modelu wizualnej reprezentacji rozprzestrzeniania się epidemii. Symulacje są przeprowadzane dla różnych wartości parametrów wejściowych.

Symulacja jest zaimplementowana w języku Python. Do manipulacji nad danymi została wykorzystana biblioteka numpy, a wizualizacje są realizowane za pomocą pakietów matplotlib.pyplot i matplotlib.animation. Wynikiem symulacji jest generowana reprezentacja graficzna położenia i stanu osobników w ograniczonej przestrzeni przedstawiająca ich poruszanie się i interakcje oraz wykresy liczby poszczególnych kategorii osobników z biegiem czasu.

Ostatecznie, zebrane dane są przekazywane do sieci neuronowej w celu predykcji liczebności poszczególnych kategorii osób.

2 Opis modelu

Na potrzeby symulacji stworzyliśmy następujący model: Populacja jest podzielona na trzy kategorie: osoby zainfekowane(I - infected), osoby podatne na zakażenie(S - susceptible) i osoby, które wyzdrowiały(R - recovered). Zdefiniowaliśmy parametry wejściowe:

- n - liczba populacji początkowej
- p_infected - procent osób zainfekowanych na początku symulacji
- r_transmission - promień transmisji
- p_transmission - prawdopodobieństwo transmisji
- t_recover - czas potrzebny na wyleczenie i przejście do stanu R - recovered
- p_isolated - procent osobników przebywających w kwarantannie

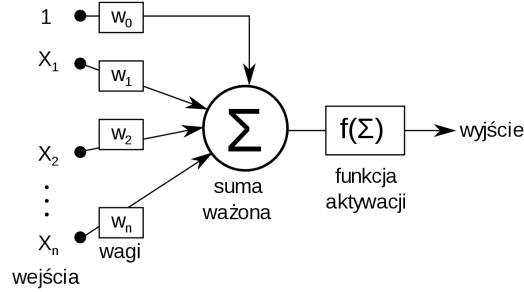
Na początku inicjalizujemy populację startową: tworzymy n osobników reprezentowanych przez klasę Person i ustawiamy je w losowej pozycji na planszy. W zależności od parametru p_infected w losowy sposób tworzymy podzbiory osobników zainfekowanych i obecnie objętych kwarantanną. Podczas

przebiegu symulacji dane z każdej iteracji są zapisywane do pliku CSV(prawdopodobieństwo transmisji, dzień, liczba zainfekowanych, podatnych i osobników, które wyzdrowiały). Do symulacji ruchu i interakcji osobników w każdej iteracji(frame) wywoływana jest funkcja update, która wykonuje następujące działania:

- Aktualizuje czas(frame) i liczy aktualnie zainfekowanych i osobników które wyzdrowiały
- Aktualizuje pozycje osobników
- Sprawdza, czy osobnik zainfekowany jest w stanie przenosić chorobę na pobliskie podatne osoby w promieniu transmisji, biorąc pod uwagę prawdopodobieństwo transmisji
- Aktualizuje kolor osobników na podstawie ich statusu
- Dołącza liczbę podatnych, zainfekowanych i wyzdrowiałych osób do odpowiednich list do realizacji wykresów
- Aktualizuje wykresy

3 Sztuczne sieci neuronowe

3.1 Model neuronu



Rysunek 1: Model matematyczny sztucznego neuronu

Model matematyczny sztucznego neuronu opisuje się równaniem:

$$y = f_a(z) = f_a \left(\sum_i^n x_i w_i + b \right) = f_a(xw + b), \quad (1)$$

gdzie x to n -wymiarowe wejście neuronu:

$$x = [x_1 x_2 \dots x_n] \in R^{[1 \times n]}, \quad (2)$$

w to wagi poszczególnych wejść neuronu:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix} \in R^{[n \times 1]}, \quad (3)$$

b to tzw. bias(obciążenie) neuronu. Jest to wartość liczbową, która wpływa na aktywację neuronu niezależnie od wejść. Bias pozwala na przesunięcie progu aktywacji neuronu. Wektor wag i bias ustawiane są w trakcie uczenia w celu optymalizacji działania modelu w zadanym problemie.

Obliczenie wyjścia neuronu dla paczki(batch) danych:

$$y = f_a(Xw + \text{broadcast}(b)), \quad (4)$$

gdzie X to macierz paczki danych wejściowych o rozmiarze BS x n (BS - batch size):

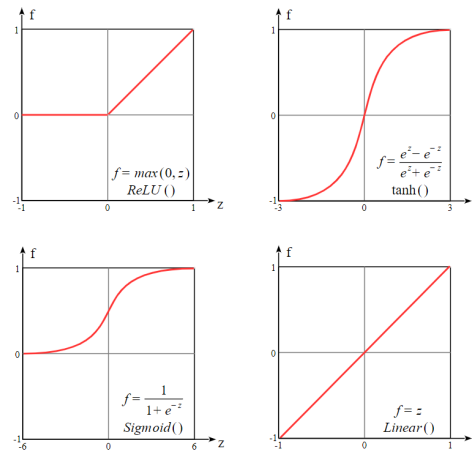
$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \dots \\ x^{(BS)} \end{bmatrix} \in R^{[BS \times n]} \quad (5)$$

Uzyskany wektor:

$$y \in R^{[BS \times 1]} \quad (6)$$

na i-tej pozycji zawiera odpowiedź neuronu na wejście $x^{(i)}$.

3.2 Funkcje aktywacji



Rysunek 2: Podstawowe rodzaje funkcji aktywacji

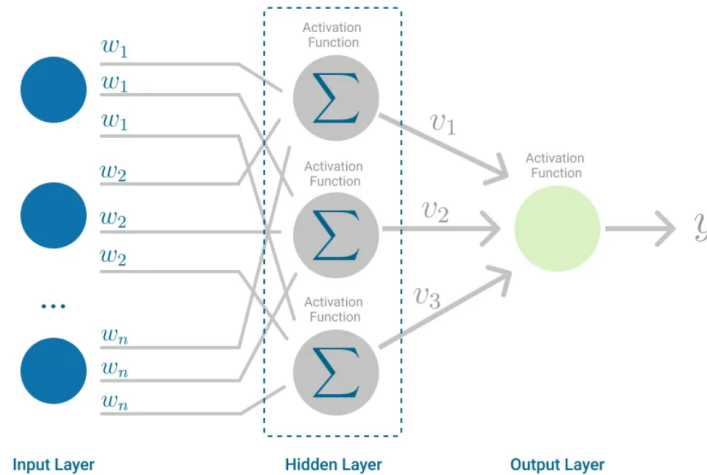
Funkcje aktywacji wprowadzają nieliniowość do działania neuronu, umożliwiając modelowi neuronu rozwiązywanie bardziej skomplikowanych problemów.

- ReLU - funkcja liniowa, która wyprowadzi wejście bezpośrednio, jeśli jest dodatnie, w przeciwnym razie wyprowadzi zero
- tanh - funkcja tanh jest rozciągniętą i przesuniętą wersją funkcji Sigmoid
- Sigmoid - ogranicza wyjście neuronu do przedziału (0, 1), dzięki czemu takie wyjście można interpretować jako znormalizowaną reakcję neuronu na wejście, gdzie 1 oznacza maksymalne "pobudzenie"

- Linear - oznacza, że wyjście neuronu jest równe logitowi, neuron oblicza jedynie kombinację liniową wejść i nie wykonuje żadnego nieliniowego przekształcenia, neuron równoważny modelowi regresji liniowej

3.3 MLP - Multilayer Perceptron

MLP to wielowarstwowa sieć neuronowa utworzona poprzez kaskadowe połączenie warstw gęstych, tak, że wyjście warstwy poprzedniej jest wejściem warstwy następnej. Wielowarstwowy Perceptron



Rysunek 3: Multilayer Perceptron

ma warstwy wejściowe i wyjściowe oraz jedną lub więcej ukrytych warstw z wieloma neuronami. I podczas gdy w Perceptronie neuron musi mieć funkcję aktywacji, która narzuca próg, jak ReLU lub Sigmoid, neurony w Perceptronie wielowarstwowym mogą używać dowolnej funkcji aktywacji.

Powodem, dla którego przyjmujemy dla MLP funkcję ReLU, jest to, że umożliwia ona lepszą optymalizację przy użyciu metody stochastycznego spadku po gradiencie, bardziej wydajne obliczenia i jest niezmienna w skali, co oznacza, że na jej charakterystykę nie ma wpływu skala danych wejściowych.

Perceptron wielowarstwowy należy do kategorii algorytmów z propagacją wsteczną, ponieważ dane wejściowe są łączone z początkowymi wagami w sumie ważonej i poddawane funkcji aktywacji, tak jak w Perceptronie. Różnica polega jednak na tym, że każda kombinacja liniowa jest propagowana do następnej warstwy.

Każda warstwa zasila następną wynikiem obliczeń, wewnętrzną reprezentacją danych. To przechodzi przez ukryte warstwy do warstwy wyjściowej.

Wsteczna propagacja to mechanizm uczenia się, który umożliwia wielowarstwowemu Perceptronowi iteracyjne dostosowywanie wag w sieci w celu zminimalizowania funkcji kosztów. Wymagane jest żeby funkcja, która łączy wejścia i wagi w neuronie, na przykład suma ważona i funkcja progowa, na przykład ReLU, były różniczkowalne.

W każdej iteracji, po przekazaniu sum ważonych przez wszystkie warstwy, gradient MSE jest obliczany dla wszystkich par wejściowych i wyjściowych. Następnie, aby go propagować wstecz, wagi pierwszej ukrytej warstwy są aktualizowane o wartość gradientu. W ten sposób wagi są propagowane z powrotem do punktu początkowego sieci neuronowej.

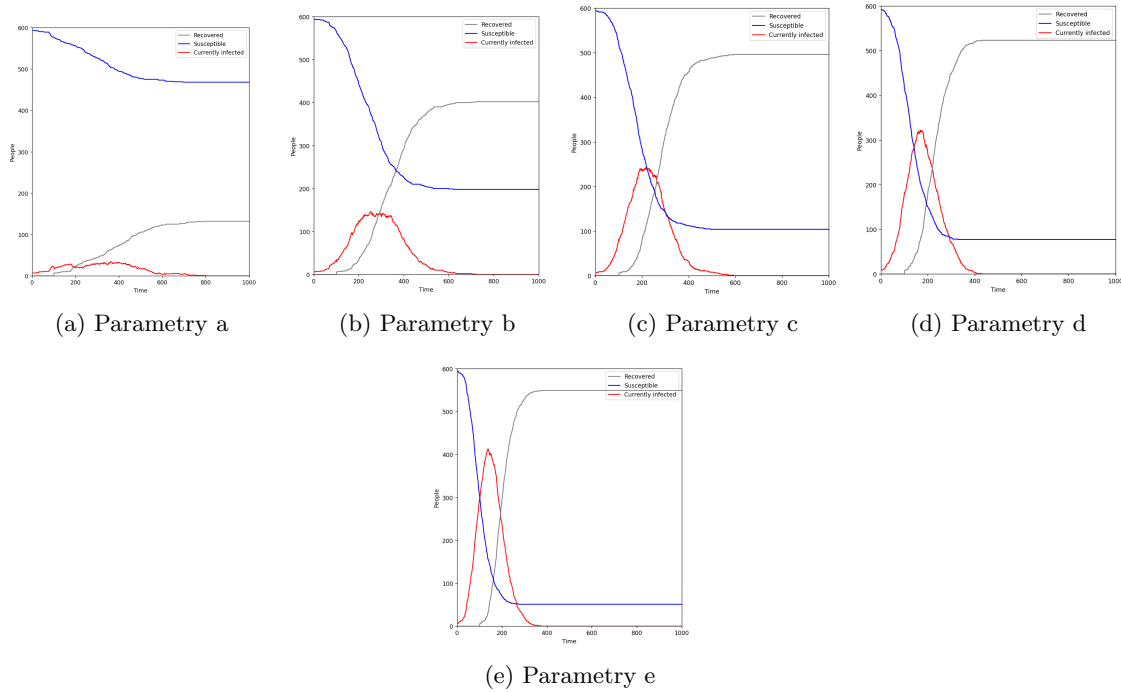
$$\Delta_w(t) = -\varepsilon \frac{dE}{dw(t)} + \alpha \Delta_w(t-1), \quad (7)$$

gdzie $\Delta_w(t)$ - gradient obecnej iteracji, ε - bias, dE - błąd, $dw(t)$ - wektor wag, α - współczynnik uczenia, $\alpha \Delta_w(t-1)$ - gradient poprzedniej iteracji

Ten proces trwa do momentu zbieżności gradientu dla każdej pary wejście-wyjście, co oznacza, że nowo obliczony gradient nie zmienił się bardziej niż określony próg zbieżności w porównaniu z poprzednią iteracją.

4 Wyniki i wnioski

4.1 Wyniki symulacji



Rysunek 4: Zestaw wykresów wynikowych dla różnych parametrów wejściowych

Powyższe wykresy przedstawiają wyniki czterech różnych symulacji z następującymi parametrami wejściowymi:

- a) $n = 600$, $r_transmission = 2$, $p_transmission = 2$, $p_isolated = 40$, $t_recover = 100$,
 $p_infected = 1$
- b) $n = 600$, $r_transmission = 2$, $p_transmission = 2$, $p_isolated = 40$, $t_recover = 100$,
 $p_infected = 2$
- c) $n = 600$, $r_transmission = 2$, $p_transmission = 2$, $p_isolated = 40$, $t_recover = 100$,
 $p_infected = 3$
- d) $n = 600$, $r_transmission = 2$, $p_transmission = 2$, $p_isolated = 40$, $t_recover = 100$,
 $p_infected = 4$
- e) $n = 600$, $r_transmission = 2$, $p_transmission = 2$, $p_isolated = 40$, $t_recover = 100$,
 $p_infected = 5$

4.2 Przebieg procesu predykcji

Na początku wygenerowaliśmy plik CSV z danymi wejściowymi do sieci neuronowej. W celu dokonania predykcji wykorzystaliśmy sieć neuronową MLP.

Danymi wejściowymi są dwie cechy oraz trzy wektory z wynikami symulacji. Cechami są prawdopodobieństwo transmisji oraz numer dnia. Pierwszy wektor zawierał liczbę osób zdrowych danego dnia, drugi wektor liczbę zarażonych danego dnia, a trzeci liczbę ozdowieńców.

Rolę funkcji aktywującej pełni funkcja ReLU. Funkcję optymalizatora przyjął Adam (Adaptive Moment Estimation). Jako funkcji straty użyliśmy pierwiastkowego błędu średniokwadratowego (mse).

Funkcja $\text{relu}(x)$ przyjmuje wartość x jako argument wejściowy i zwraca maksimum z przedziału od 0 do x . Oznacza to, że wartości ujemne są zastępowane przez zero, a wartości dodatnie pozostają niezmienione.

Adam to adaptacyjny optymalizator szeroko stosowany w uczeniu głębokim do aktualizacji wag modeli podczas uczenia. Łączy w sobie dwie techniki: Adaptive Gradient Descent i Stochastic Gradient Descent.

Adam dostosowuje się do różnych szybkości aktualizacji wag dla każdego parametru, wykorzystując pierwszy i drugi moment gradientu. Przechowuje wykładniczo malejące średnie wyniki gradientu i kwadratu gradientu dla każdego parametru. Pozwala to na dostosowanie szybkości uczenia się dla każdego parametru na podstawie historii gradientu.

Optymalizator Adam wykorzystujemy do aktualizacji wag modelu podczas treningu.

Sieć neuronowa była trenowana trzykrotnie, oddzielnie dla każdej grupy wyników modeli SIR, co oznacza, że cechy (prawdopodobieństwo zarażenia i dzień) pozostawały takie same, zamieniany był jedynie wektor wejściowy, czyli każdy z trzech wektorów wyjściowych symulacji.

4.3 Wyniki oraz wnioski dotyczące przebiegu predykcji

Wynikami predykcji są wektory S, I, R, gdzie:

- S - wektor zawierający liczbę osób zdrowych / podatnych na zarażenie na dzień
- I - wektor zawierający liczbę osób zainfekowanych na dzień
- R - wektor zawierający liczbę ozdrowieńców na dzień

Poniżej widoczne są wykresy prognozowanego wyniku dla prawdopodobieństw zarażenia równych:

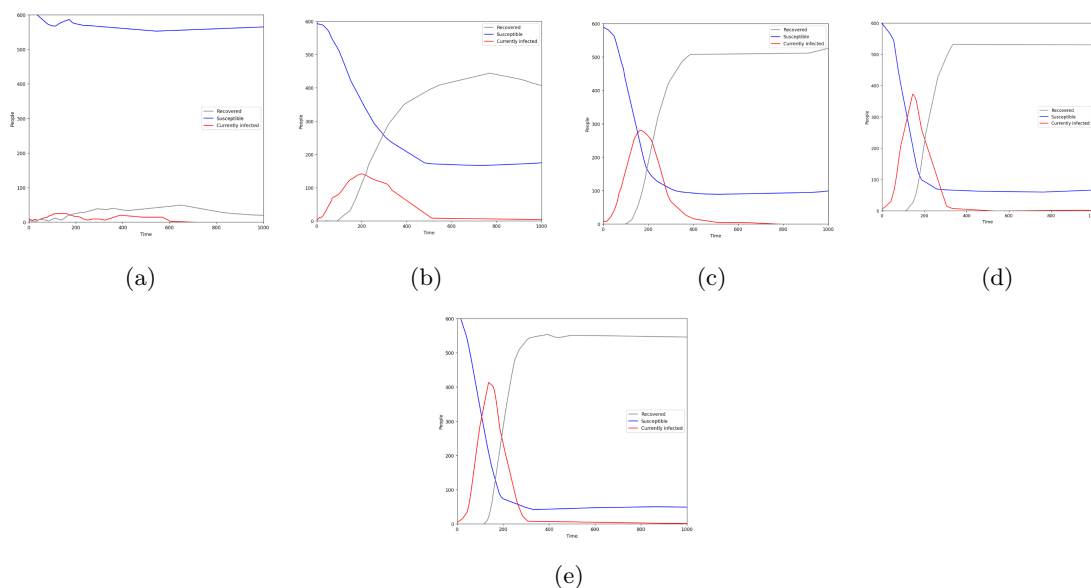
a) $p_{\text{infected}} = 1$

b) $p_{\text{infected}} = 2$

c) $p_{\text{infected}} = 3$

d) $p_{\text{infected}} = 4$

e) $p_{\text{infected}} = 5$



Rysunek 5: Zestaw wykresów prognozowanych wyników dla różnych prawdopodobieństw zarażenia

4.4 Interpretacja wyników

Sieć neuronowa przewidywała sytuację na następne 1000 dni.

a) Prawdopodobieństwo zarażenia 1%:

Przy następującym prawdopodobieństwie, sieć przewiduje relatywnie stałą liczbę osób podatnych na zakażenie, która wynosi około 570 osób. Niemniej jednak przy takiej sytuacji sieć przewiduje od dnia 0 do 700 ilość zarażeń, oscylującą między kilkoma przypadkami do około 30 na dzień. Po 700 dniach ilość infekcji dąży do 0. Ilość ozdowieńców po około 150 dniach zaczęłaby rosnąć, przerastając ilość zarażonych aż do punktu kulminacyjnego, jakim jest okres dni 600-700, w którym ilość osób zakażonych zaczyna dążyć do zera, a ilość ozdowieńców zaczyna maleć, ze względu na brak nowych zakażeń.

b) Prawdopodobieństwo zarażenia 2%:

Przy następującym prawdopodobieństwie, sieć dla pierwszych dwustu dni przewiduje drastyczny wzrost ilości osób zainfekowanych, która w dniu 200 wyniesie prawie 150 osób. Po tym dniu ilość zakażeń zacznie maleć. Po 500 dniach ilość infekcji spadnie do kilku przypadków dziennie, jednak nigdy nie spadnie do zera. Pierwsze wyzdrowienia przewidywane są na dzień około 30 i od tego momentu ilość ozdowieńców będzie rosnąć aż do okolic dnia 800, po którym ta ilość zacznie powoli maleć. Prognoza ta jest nieprzychylna dla ludzi.

c) Prawdopodobieństwo zarażenia 3%:

Ponieważ prawdopodobieństwo zarażenia jest coraz większe, ilość osób zakażonych także jest coraz większa. Podobnie jednak do poprzedniego przypadku, ilość zarażeń będzie szybko rosnąć aż do okolic dnia dwusetnego, jednak w sytuacji takiego prawdopodobieństwa ilość osób zainfekowanych dojdzie do prawie 300 osób na dzień. Po dniu maksimum nowych przypadków infekcji będzie coraz mniej aż do dnia około osiemsetnego, gdzie ilość nowych przypadków spadnie do zera. Jest to jednocześnie prognoza bardziej negatywna oraz bardziej pozytywna od poprzedniej.

d) Prawdopodobieństwo zarażenia 4%:

Podobnie jak dla przypadku prawdopodobieństwa zarażenia 2% jak i 3% ilość osób zarażonych będzie rosła aż do okolic dnia dwusetnego, gdzie epidemia osiągnie swoje apogeum, zarażając tego dnia prawie 400 osób. Następnie sieć neuronowa przewiduje coraz mniejszą ilość zarażeń, gdzie po dniu pięćsetnym liczba nowych przypadków wyniesie 0. Jednakże w okolicach dnia 700 infekcja powróci i aż do tysięcznego dnia będzie codziennie atakować po kilka osób. Jest to prognoza zdecydowanie najbardziej negatywna dotychczas. Pierwsze przypadki ozdowieńców przewidywane są na dzień około stu dwudziesty, gdzie ilość ozdowieńców będzie rosnąć aż do dnia około trzysetnego, i pozostanie relatywnie stała aż do dnia tysięcznego.

e) Prawdopodobieństwo zarażenia 5%:

Jak można zauważyć z wykresu, sytuacja nie różni się szczególnie od poprzedniego przypadku. Maksymalna ilość osób zarażonych danego dnia przekroczy 400, jednak nie jest to tak drastyczna różnica, jak w porównaniach poprzednich prognoz. Faktyczną różnicą jednak jest fakt, że w przeciwieństwie do poprzedniej prognozy, ilość zarażeń nigdy nie spadnie zera, a będzie oscylować w okolicach kilku nowych przypadków infekcji na dzień.

Analizując i porównując powyższe prognozy można dojść do poniższych wniosków:

- a) bez względu na prawdopodobieństwo zarażenia, apogeum epidemii przypada na dzień około dwusetny
- b) po wystąpieniu maksimum nowych przypadków infekcji, liczba nowych zakażeń maleje w tempie porównywalnym do tego w jakim ta liczba rosła