AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Prognozowanie wystąpienia udaru mózgu - klasyfikacja

Kierunek: Informatyka i Ekonometria

Opracowały: Klaudia Pajor, Anna Pietryka, Justyna Mastek, Karolina Urbaniak

Spis treści

1. Cel projektu

Celem projektu jest przeprowadzenie prognozy ex post wystąpienia udaru mózgu za pomocą uczenia sztucznych sieci neuronowych. Predykcja dotycząca klasyfikacji osoby jako narażonej na udar mózgu opiera się o czynniki takie jak: płeć, wiek, występowanie nadciśnienia tętniczego i chorób serca, stan cywilny, typ zatrudnienia, miejsce zamieszkania, średni poziom glukozy we krwi, wskaźnik BMI oraz status palacza. Zostały one wybrane, gdyż z logicznego punktu widzenia mogą mieć wpływ na występowanie analizowanego zjawiska. Wszelkie predykcje oraz analiza zostały w całości wykonane w języku Python.

2. Opis danych

Wykorzystane w projekcie dane to 5110 obserwacji pacjentów dokonanych przy przyjęciu do szpitala. Zmienną objaśnianą (Y) w modelu jest występowanie udaru mózgu (*wartość 1 gdy pacjent miał udar, 0 gdy nie*). Uwzględnione zostało 19 zmiennych objaśniających: wiek w latach (X_1), występowanie nadciśnienia tętniczego (*wartość 1 gdy występuje, 0 w innym wypadku* - X_2), występowanie choroby serca (*wartość 1 gdy występuje, 0 w innym wypadku* - X_3), stan cywilny *(wartość 1 gdy osoba kiedykolwiek była zamężna/żonata, wartość 0 w innym wypadku* - X_4), średni poziom glukozy we krwi (X_5), wartość wskaźnika BMI (X_6), typ zatrudnienia (*dziecko* - X_7, *sektor prywatny* - X_8, *samozatrudnienie* - X_9, *sektor publiczny* - X_{10}, *nigdy nie pracował* - X_{11}), płeć ($kobieta - X_{12}, mężczyzna - X_{13}, inne - X_{14}$), miejsce zamieszkania (*wieś* - X_{15}, *miasto* - X_{16}), status palacza (*były palacz* - X_{17}, *nigdy nie palił* - X_{18}, *palący* - X_{19}).

Poszczególne kolumny z wartościami nienumerycznymi zostały podzielone i skategoryzowane – zawarto je w osobnych kolumnach w zależności od szczegółowości czynnika i tak przez „1” oznaczone zostało występowanie określonej wartości czynnika, a przez „0” oznaczony został brak. Przykładowo kolumna zawierająca dane o płci osoby została podzielona na trzy (kobieta, mężczyzna, inna), a wartość odpowiadająca konkretnemu pacjentowi została oznaczona we właściwej kolumnie przez „1” (pozostałe wartości zostały oznaczone przez „0”). Zredukowana została również ilość danych wykorzystywana do przeprowadzenia badania – do 418 obserwacji (część w wyniku usuwania braków obserwacji,

pozostała w celu zrównoważenia ilości wystąpienia udarów, gdyż znacząco więcej było pacjentów bez niego). Dane przed przygotowaniem do przeprowadzenia badania można zobaczyć w pliku *dane.csv*.

3. Algorytm zastosowany do uczenia sztucznej sieci neuronowej

W celu uczenia sieci, w projekcie wykorzystany został algorytm wstecznej propagacji błędów (ang. *backpropagation algorithm*). Jest to najbardziej popularny algorytm wykorzystywany do uczenia sieci wielowarstwowych z nauczycielem. Schemat działania jest następujący: do sieci wprowadzane są przykładowe wartości poprawnych danych, a sama sieć ucząc się na ich podstawie, naśladuje wyniki. Dane wprowadzane to sygnały wejściowe oraz oczekiwane sygnały wyjściowe – tzw. ciąg uczący. Działanie algorytmu opiera się na regule delta – wyznaczana jest średniokwadratowa funkcja błędu dla sieci (*Q(w)*), a następnie dąży się do znalezienia jej minimalnej wartości poprzez odpowiednie dostosowanie wag.

Ogólny schemat procesu trenowania sieci prezentuje się następująco:

1) Określana jest liczba warstw oraz liczba neuronów w warstwach – topologia sieci.
2) Zainicjowane w sposób losowy zostają wagi (małe wartości).
3) Dla danego wektora uczącego, warstwa po warstwie, obliczana jest odpowiedź sieci.
4) Dla kolejnych neuronów wyjściowych obliczany jest błąd równy różnicy pomiędzy wartością obliczoną oraz poprawną.
5) Błędy propagowane są do poprzedzających warstw.
6) Każdy neuron modyfikuje wagi na podstawie wartości błędu i wielkości przetwarzanych w kolejnym kroku sygnałów.
7) Powtarzane są czynności od punktu 3) dla kolejnych wektorów uczących. Po wykorzystaniu wszystkich wektorów losowo zmieniana jest ich kolejność i wykorzystywane są one powtórnie.

8) Algorytm zatrzymuje swoje działanie, gdy osiągnięta zostanie określona ilość iteracji wykorzystywania zbioru uczącego lub gdy sieć osiągnie ustaloną granicę średniego błędu.

Wzory użyte na potrzeby zbudowania sieci:

- Funkcja aktywacji: funkcja sigmoidalna [12]

$$y(x) = \frac{1}{1 + e^{-\beta * x}}$$

- Pochodna funkcji sigmoidalnej:

$$y'(x) = y(x) * (1 - y(x))$$

- Uczenie sieci do przodu (Forward Propagation) [10]

Wzór dla pojedynczego neuronu:

$$y_i = f(w_{(x1)i} * x_1 + \cdots + w_{(x_n)i} * x_n)$$

y –> wyjście neuronu

f –> funkcja aktywacyjna

$x_1, \ldots, x_n$ –> kolejne wartości wyjść neuronów połączonych z tym, dla którego jest wyliczane wyjście

$w_{(x_n)i}$ –> waga danego połączenia

- Propagacja wsteczna (Backpropagation) [10]

Obliczany jest błąd wartości wyjściowego neuronu (porównanie wyjścia ze znanym wynikiem):

$$\delta = z - y$$

Obliczanie błędów kolejnych neuronów (od ostatniej warstwy z neuronem wyjściowym aż po pierwszą z danymi wejściowymi) i zmiana wag połączeń – wzór dla jednego neuronu:

$$gradient = \delta * y'(y_i)$$

y_i –> wartość wyjściowa neuronu

Następnie dla każdego połączenia neurona z poprzedniej warstwy, z którymi połączony jest neuron, na którym stosowana jest propagacja wsteczna, stosowany jest wzór:

$$\Delta wagi_i = \eta * gradient * y_i + \alpha * \Delta wagi_{i-1}$$

η –> learning rate

α –> momentum

$\Delta\, wagi_{i-1}$ –> poprzednia różnica wag

$$W = w + \Delta\, wagi_{i-1}$$

w –> poprzednia waga połączenia

W –> zaktualizowana waga połączenia

4. Wprowadzenie do badania

W celu stworzenia projektu wykorzystane zostały biblioteki takie jak „numpy", „pandas", „math", „matplotlib.pyplot" oraz „pylab". Dane podzielono na część uczącą (292 obserwacje) i testową (126 obserwacji).

Dane wejściowe zostały sprawdzone w kwestii ewentualnych braków - na tej podstawie usunięto 201 wierszy, w których one występowały (początkowa ilość obserwacji była na tyle duża, że nie wpłynęło to negatywnie na jakość przeprowadzenia badania). Następnie dokonana została kategoryzacja poszczególnych czynników w celu określenia występowania danej wartości. Każda z kolumn, w której dane nie były numeryczne została podzielona i dane w nich występujące zostały doprowadzone do postaci zero-jedynkowej (wartość „1" – dany czynnik występuje, wartość „0" - brak występowania). Wykorzystaną funkcją aktywacji jest funkcja sigmoidalna, która przyjmuje wartości ze zbioru od 0 do 1, dzięki czemu łatwiej jest interpretować wynik końcowy badania.

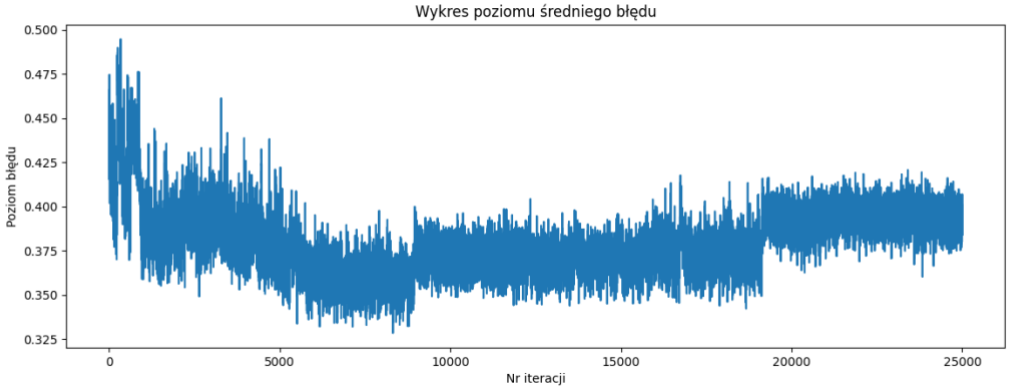
Zastosowane zostały współczynniki eta (Learning Rate - współczynnik uczenia, który określa, jak bardzo wagi są korygowane przy każdej aktualizacji) oraz alfa (Momentum – wpływa na aktualizację wag podczas uczenia).

5. Zależność poziomu średniego błędu od cech sieci

W niniejszym rozdziale przedstawione zostaną zależności błędu od ilości iteracji, występowania współczynnika alfa, ilości warstw ukrytych, a także ilości neuronów w warstwie. Pokazane zostaną wykresy zmiany poziomu średniego błędu, wykresy poziomu błędu dla kolejnych obserwacji zbioru testowego oraz procentowe wartości skuteczności sieci (umiejętności poprawnej klasyfikacji przypadków).

5.1. Przypadek pierwszy

Badany jest zbiór uczący wraz ze zbiorem testowym. Wartości współczynników (Learning Rate oraz Momentum) prezentują się następująco: eta = 0.01 oraz alfa = 0.6. Wykonanych zostało 25 000 iteracji. Sieć składała się z jednej warstwy ukrytej, w której znajdowało się 19 neuronów. Wagi wylosowane zostały z rozkładu normalnego. Poniżej przedstawiono wykresy poziomu błędów.

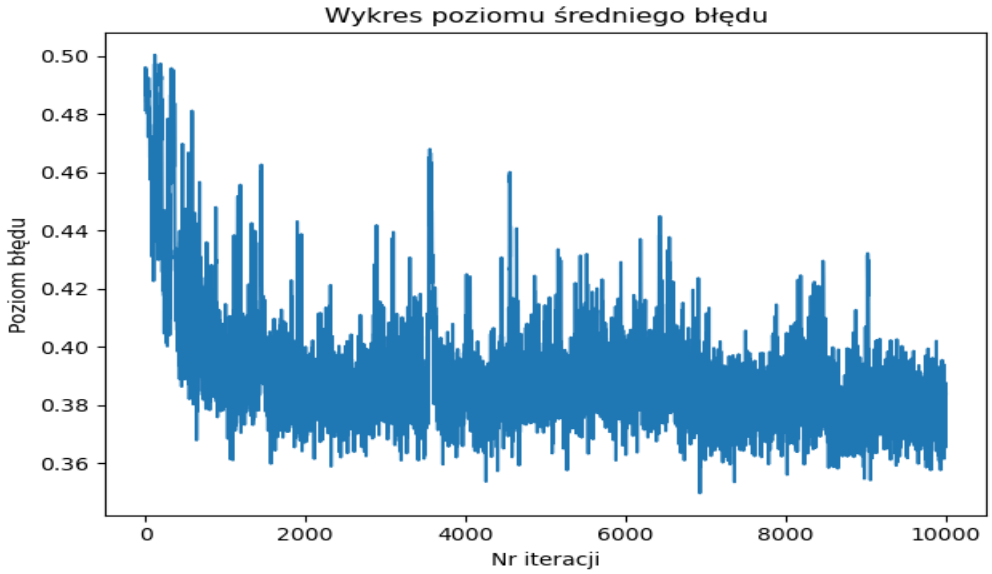


Wykres 1 - Wykres poziomu średniego błędu dla 25 tys. iteracji– przypadek 1

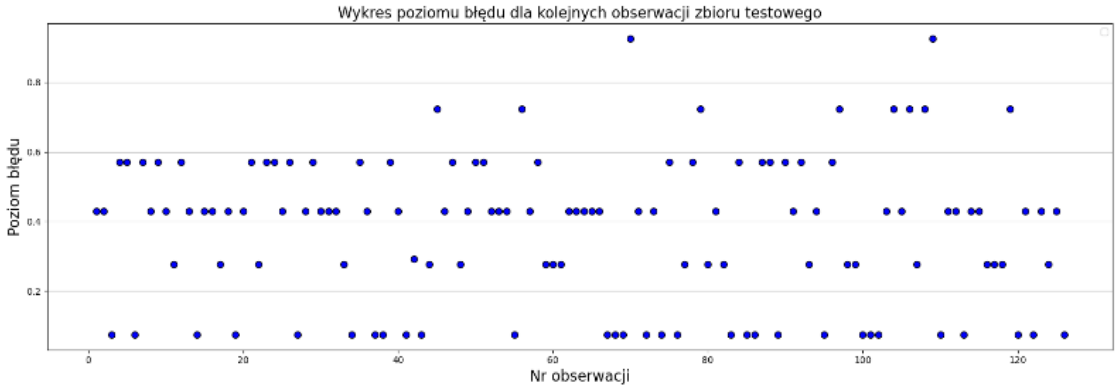
Wykres 2 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 25 tys. Iteracji – przypadek 1

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 68.15%, a na zbiorze testowym sieci wynosi: 67.46%.

Kolejnym krokiem była zmiana ilości iteracji na 10 000 w celu sprawdzenia jej wpływu na skuteczność badania. Poniżej przedstawione zostały analogiczne wykresy.



Wykres 3 - Wykres poziomu średniego błędu dla 10 tys. iteracji – przypadek 1



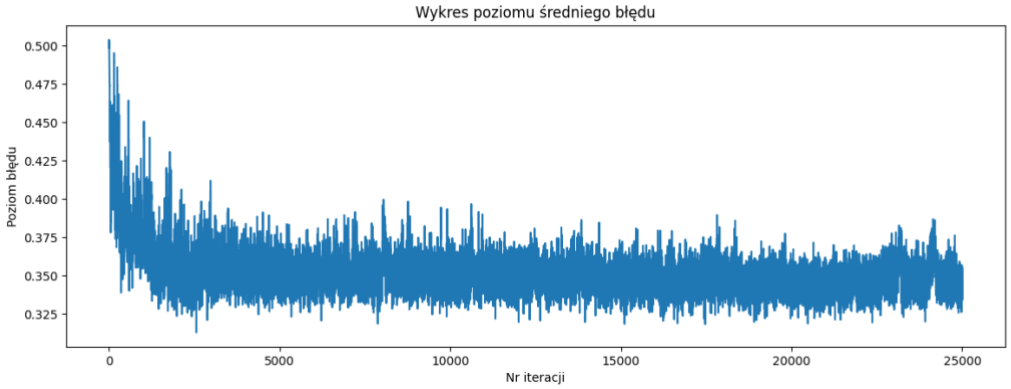
Wykres 4 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 10 tys. Iteracji – przypadek 1

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 69.86%, a na zbiorze testowym sieci wynosi: 73.02%.

Zmniejszenie ilości iteracji wpłynęło na zwiększenie skuteczności działania.

5.2. Przypadek drugi

Badany jest zbiór uczący wraz ze zbiorem testowym. Wartości współczynników (Learning Rate oraz Momentum) prezentują się następująco: eta = 0.01 oraz alfa = 0.6. Wykonanych zostało 25 000 iteracji. Sieć składała się z jednej warstwy ukrytej, w której znajdowało się 10 neuronów. Wagi wylosowane zostały z rozkładu normalnego. Poniżej przedstawiono wykresy poziomu błędów.

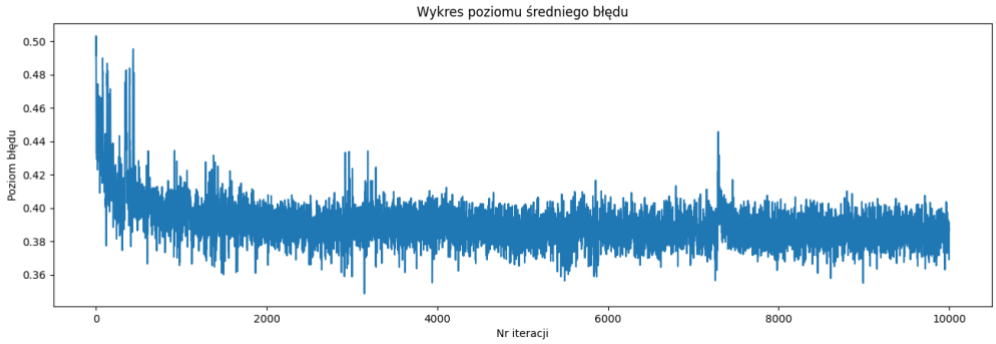


Wykres 5 - Wykres poziomu średniego błędu dla 25 tys. Iteracji – przypadek 2

Wykres 6 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 25 tys. Iteracji - przypadek 2

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 77.86%, a na zbiorze testowym sieci wynosi: 70.63%.

Kolejnym krokiem była zmiana ilości iteracji na 10 000 w celu sprawdzenia jej wpływu na skuteczność badania. Poniżej przedstawione zostały analogiczne wykresy.



Wykres 7 - Wykres poziomu średniego błędu dla 10 tys. Iteracji – przypadek 2

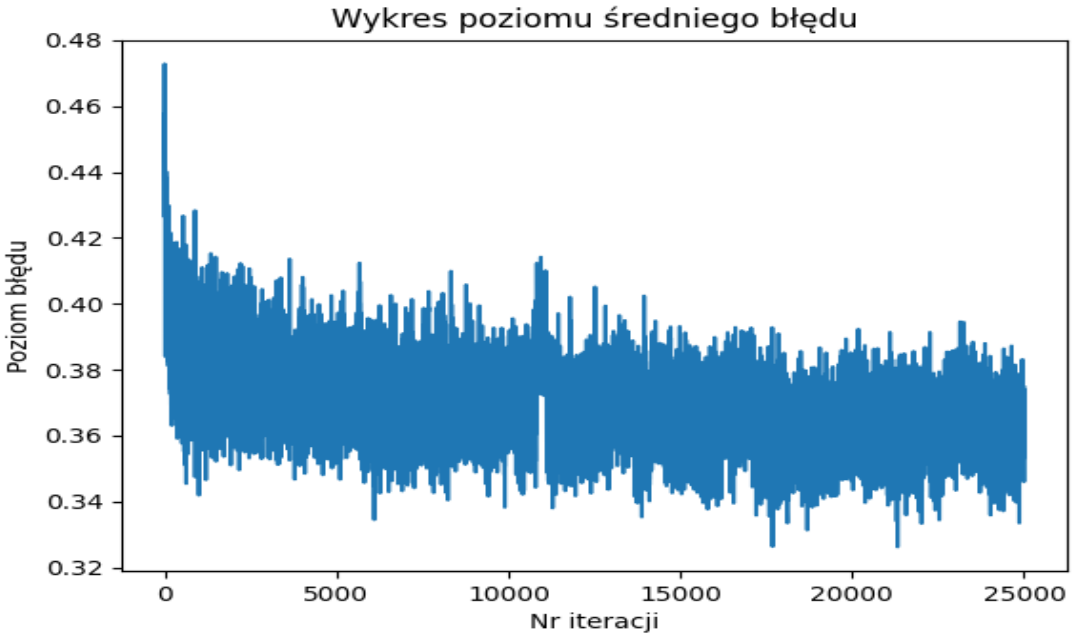
Wykres 8 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 10 tys. Iteracji - przypadek 2

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 65.23%, a na zbiorze testowym sieci wynosi: 60.76%.

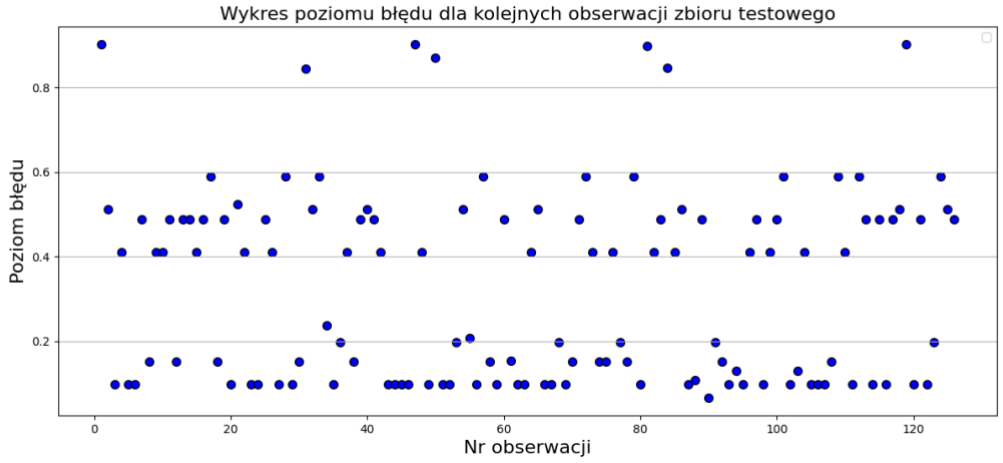
Zmniejszenie ilości iteracji wpłynęło na zmniejszenie skuteczności działania.

5.3. Przypadek trzeci

Badany jest zbiór uczący wraz ze zbiorem testowym. Wartości współczynników (Learning Rate oraz Momentum) prezentują się następująco: eta = 0.01, a współczynnik alfa został usunięty w celu sprawdzenia jego wpływu na jakość badania. Wykonanych zostało 25 000 iteracji. Sieć składała się z jednej warstwy ukrytej, w której znajdowało się 19 neuronów. Wagi wylosowane zostały z rozkładu normalnego. Poniżej przedstawiono wykresy poziomu błędów.



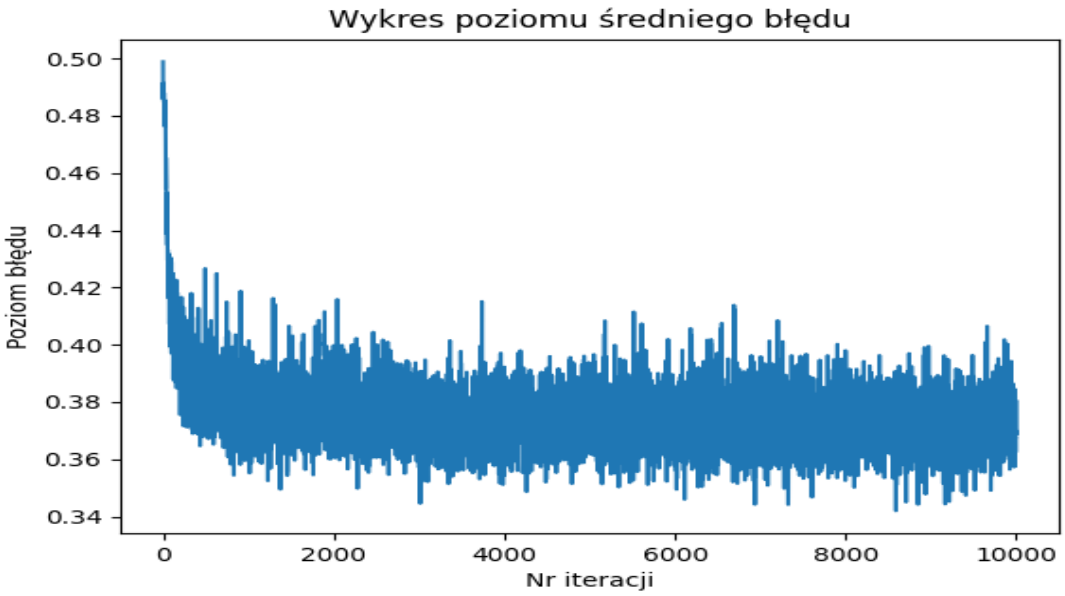
Wykres 9 - Wykres poziomu średniego błędu dla 25 tys. Iteracji – przypadek 3



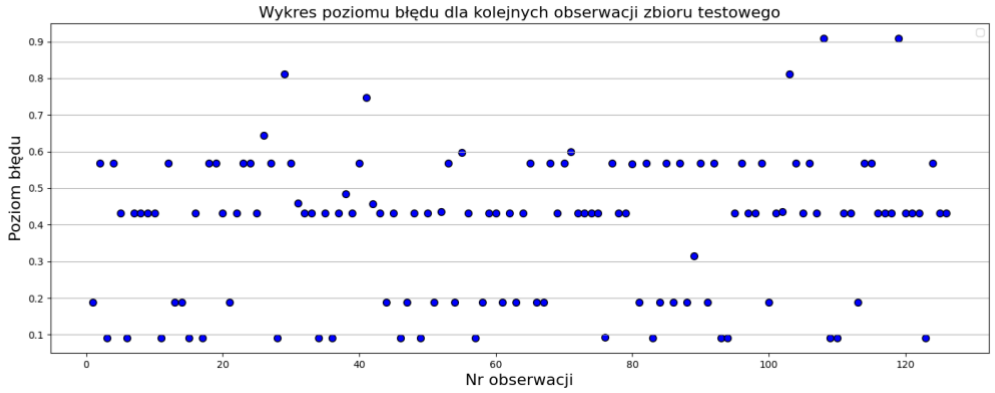
Wykres 10 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 25 tys. Iteracji - przypadek 3

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 68.84%, a na zbiorze testowym sieci wynosi: 79.37%.

Kolejnym krokiem była zmiana ilości iteracji na 10 000 w celu sprawdzenia jej wpływu na skuteczność badania. Poniżej przedstawione zostały analogiczne wykresy.



Wykres 11 - Wykres poziomu średniego błędu dla 10 tys. Iteracji – przypadek 3



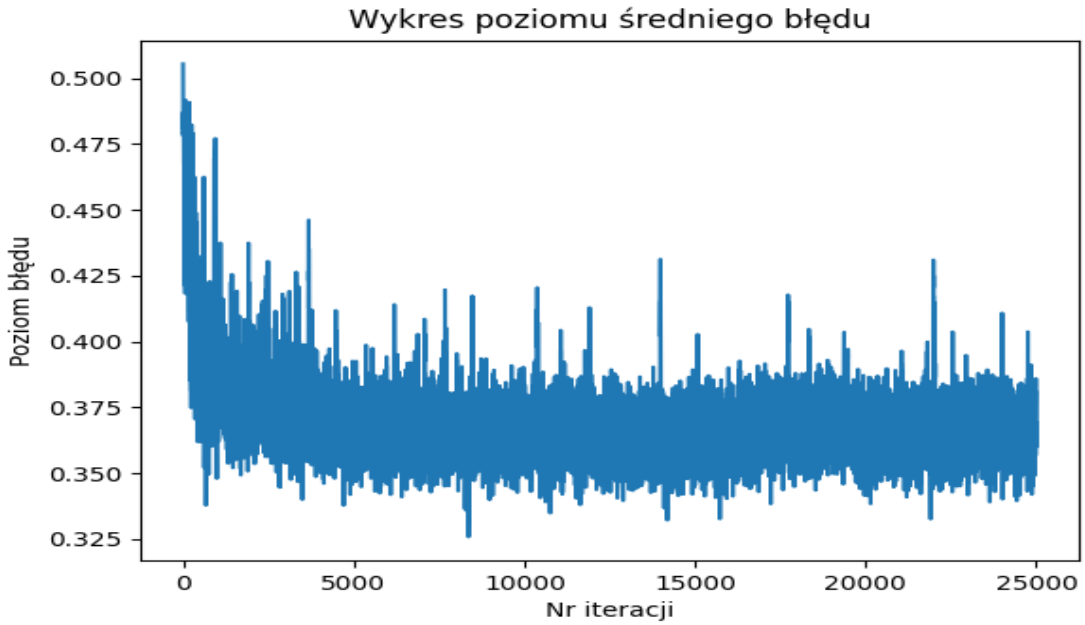
Wykres 12 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 10 tys. Iteracji - przypadek 3

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 70.89%, a na zbiorze testowym sieci wynosi: 71.43%.

Zmniejszenie ilości iteracji wpłynęło na zmniejszenie skuteczności działania na zbiorze uczącym, natomiast na zbiorze testowym zarejestrowano polepszenie skuteczności.

5.4. Przypadek czwarty

Badany jest zbiór uczący wraz ze zbiorem testowym. Wartości współczynników (Learning Rate oraz Momentum) prezentują się następująco: eta = 0.01 oraz alfa = 0.6. Wykonanych zostało 25 000 iteracji. Sieć składała się z dwóch warstw ukrytych, w których znajdowało się kolejno 19 oraz 10 neuronów. Wagi wylosowane zostały z rozkładu normalnego. Poniżej przedstawiono wykresy poziomu błędów.

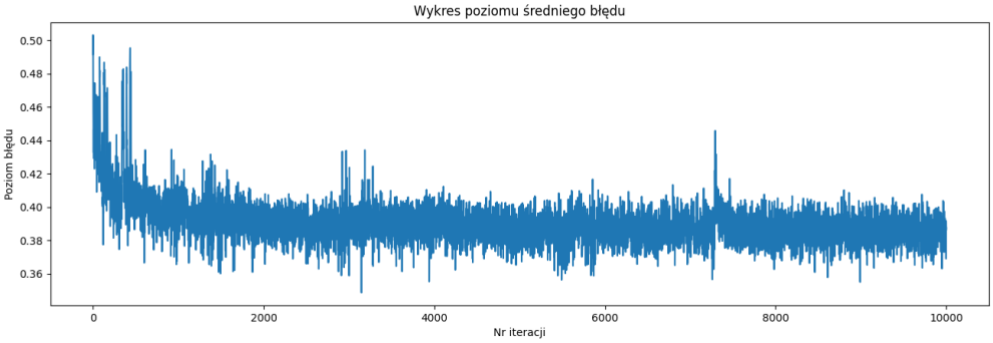


Wykres 13 - Wykres poziomu średniego błędu dla 25 tys. Iteracji – przypadek 4

Wykres 14 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 25 tys. Iteracji - przypadek 4

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 71.58%, a na zbiorze testowym sieci wynosi: 73.81%.

Kolejnym krokiem była zmiana ilości iteracji na 10 000 w celu sprawdzenia jej wpływu na skuteczność badania. Poniżej przedstawione zostały analogiczne wykresy.



Wykres 15 - Wykres poziomu średniego błędu dla 10 tys. Iteracji – przypadek 4

Wykres 16 - Wykres poziomu błędu dla kolejnych obserwacji zbioru testowego dla 10 tys. Iteracji - przypadek 4

Skuteczność na zbiorze uczącym sieci wynosi w przybliżeniu: 65.75%, a na zbiorze testowym sieci wynosi: 72.22%.

Zmniejszenie ilości iteracji wpłynęło na zmniejszenie skuteczności działania.

6. Podsumowanie

Poniżej zestawiona została poglądowa tabela wartości poszczególnych elementów kolejnych etapów przeprowadzanej analizy:

Przypadek	**Il. iteracji**	**Il. warstw**	**Il. neuronów w warstwie**	**alfa**	**eta**	**Skuteczność na zbiorze uczącym**	**Skuteczność na zbiorze testowym**
1	25 000	3	19, 19, 1	0.6	0.01	68.15%	67.46%
1	10 000	3	19, 19, 1	0.6	0.01	69.86 %	73.02%
2	25 000	3	19, 10, 1	0.6	0.01	77.86%	70.63%
2	10 000	3	19, 10, 1	0.6	0.01	65.23%	60.76%
3	10 000	3	19, 19, 1	Brak	0.01	70.89 %	71.43 %
3	25 000	3	19, 19, 1	Brak	0.01	68.84%	79.37%
4	25 000	4	19, 19, 10, 1	0.6	0.01	71.58%	73.81%
4	10 000	4	19, 19, 10, 1	0.6	0.01	65.75%	72.22%

Analizując wyniki badania można wysunąć wnioski, że pomimo modyfikowania ilości iteracji, liczby neuronów, występowania współczynnika alfa, a także ilości warstw ukrytych, ciężko jest przewidzieć udar mózgu na podstawie wybranych danych. Końcowe części wszystkich wykresów ukazują, że podczas nauki sieci błąd ulega stabilizacji, osiągając poziom, poniżej którego nie jest w stanie zmaleć, co oznacza, że w danej architekturze jest to jego minimalna wartość. Zmiana architektury sieci poprzez usunięcie alfy daje najlepszą skuteczność na zbiorze testowym na poziomie 79.37% przy 25 000 iteracji oraz zastosowaniu jednej warstwy ukrytej zawierającej 19 neuronów.

Biorąc pod uwagę fakt, że celem projektu jest przewidzenie wystąpienia udaru u pacjenta, skuteczność sieci na zbiorze testowym okazuje się być dość zaskakująco wysoka, ponieważ dane charakteryzują się multimodalnością (obejmują, wykorzystują, angażują czynniki na wiele sposobów) - nie są jednoznaczne. Analogiczne badania – choć z uwzględnieniem większej ilości medycznych danych – przeprowadzone zostały w pracy „The Probability of Ischaemic Stroke Prediction with a Multi-Neural-Network-Model” [4]. Osiągnięta w nich

skuteczność sieci wyniosła aż 98.53%. Prawie dwudziestoprocentowa rozbieżność pomiędzy skutecznością tejże sieci a sieci stworzonej na potrzeby niniejszej pracy wynika najprawdopodobniej z doboru odmiennych zmiennych objaśniających. Dla porównania, w pracy „Machine Learning for Outcome Prediction of Acute Ischemic Stroke Post Intra-Arterial Therapy" [5] osiągnięta skuteczność oscyluje w granicach 80%.

Podsumowując, skuteczność na poziomie około 60-80% jest satysfakcjonującym rezultatem przeprowadzonego badania.

7. Źródło danych

Do przeprowadzenia badania użyte zostały dane dotyczące udaru mózgu udostępnione na następującej stronie internetowej:

https://www.kaggle.com/fedesoriano/stroke-prediction-dataset?fbclid=IwAR1_8RQHKtDrMwi-x1A8sZS-KJrRmCsT8YtWK-mYDiNye-UffxdpjXVp70A

8. Bibliografia

Podczas tworzenia projektu wykorzystane zostały następujące źródła:

[1] Ryszard Tadeusiewicz, Bartosz Leper, Barbara Borowik, Tomasz Gąciarz, Odkrywanie właściwości sieci neuronowych: przy użyciu programów w języku C#, Wydawnictwa PAU, Kraków 2007

[2] Stanisław Osowski, Sieci neuronowe w ujęciu algorytmicznym, Wydawnictwa Naukowo-Techniczne, Warszawa 1996

[3] Ryszard Tadeusiewicz, Maciej Szaleniec, Leksykon sieci neuronowych, Wydawnictwo Fundacji „Projekt Nauka", Wrocław 2015

[4] Yan Liu, Bo Yin, Yanping Cong, The Probability of Ischaemic Stroke Prediction with a Multi-Neural-Network Model, 2020

[5] Hamed Asadi, Richard Dowling, Bernard Yan, Peter Mitchell, Machine Learning for Outcome Prediction of Acute Ischemic Stroke Post Intra-Arterial Therapy, 2014

[6] http://home.agh.edu.pl/~horzyk/lectures/biocyb/BIOCYB-SieciNeuronowe.pdf [na dzień 16.03.2021r.]
[7] https://www.jeremyjordan.me/nn-learning-rate/ [na dzień 24.03.2021r.]
[8] https://thecodacus.com/2017/08/14/neural-network-scratch-python-no-libraries/ [na dzień 24.03.2021r.]
[9] https://www.coursera.org/lecture/deep-neural-network/train-dev-test-sets-cxG1s [na dzień 22.03.2021r.]
[10] https://www.cri.agh.edu.pl/uczelnia/tad/inteligencja_obliczeniowa/08%20-%20Uczenie%20-%20pogl%C4%85dowe.pdf?fbclid=IwAR1igjkvBzDCL-7hgeY7zA2iFdtzayALyorfuZkRB5MPk7bX1_oNI3oYZ-A [na dzień 16.03.2021r.]
[11] https://developer.ibm.com/technologies/artificial-intelligence/articles/l-neural/?fbclid=IwAR24g-hTnLey8glH83HLjge1SeAoyg1UbiuilgzKClzX4703o0merVO_NZo [na dzień 17.03.2021r.]
[12] https://www-users.mat.umk.pl/~piersaj/www/contents/teaching/wsn2013/wsn-notatki.pdf

9. Spis wykresów