

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Informatyka (INF)

Przedmiot: Podstawy obliczeń neuronowych

## Sprawozdanie

TEMAT:

Komputerowe wspomaganie diagnozowania  
stanów ostrego brzucha z wykorzystaniem  
sztucznych sieci neuronowych

AUTORZY:

Tomasz Szewczuk 226163

Adam Węglowski 226175

# 1. Streszczenie

W pracy przedstawione zostanie działanie głębokich sieci neuronowych na podstawie poprawności diagnozowania stanów ostrego brzucha, w tym przypadku wykorzystana zostanie baza danych składająca się ze symptomów ostrego zapalenia brzucha. Dodatkowo wykorzystany zostanie „One hot encoding”, który pozwala na szybsze wyuczenie sieci neuronowych. Eksperyment zostanie przeprowadzony, a jego wyniki przedstawione i omówione.

**Słowa kluczowe:** sieci neuronowe, sieci głębokie, One hot encoding”, walidacja krzyżowa

## 2. Informacje Ogólne

Ostre zapalenia brzucha, nazywane też często „ostрым brzuchem”, to grupa różnorodnych chorób jamy brzusznej objawiających się nagle, przebiegających w gwałtowny sposób i wymagających intensywnej diagnostyki i szybkiego leczenia, często obejmującego leczenie operacyjne przeprowadzane w trybie pilnym. Leczeniem ostrych chorób jamy brzusznej zajmują się chirurdzy ogólni, ponieważ jednym z podstawowych elementów leczenia jest podjęcie decyzji czy szybka operacja jest konieczna i następnie przeprowadzenie jej jak najszybciej u tych chorych, którzy tego wymagają. Ze względu na potrzebę szybkiego podejmowania decyzji zostanie przeprowadzony eksperyment sprawdzający jakość i skuteczność rozpoznawania stanów ostrego zapalenia brzucha przez sieci neuronowe. W ostatnich latach nastąpił dynamiczny rozwój sztucznej inteligencji, a wraz z nią głębokich sieci neuronowych. Algorytmy uzyskują bardzo wysokie wyniki w wielu zadaniach między innymi w rozpoznawaniu obrazu, rozpoznawaniu mowy lub tekstu pisanego. Z tego względu podczas badań przeanalizowane zostanie działanie głębokich sieci neuronowych do diagnozowania ostrego zapalenia brzucha.

## 3. Cel projektu

Celem projektu było sprawdzenie i ocena jakości działania sieci neuronowych – sieć jednokierunkowa z 1 warstwą ukrytą dla 3 różnych liczby neuronów w warstwie ukrytej oraz dla uczenia metodą propagacji wstecznej z momentum i bez momentum. W eksperymencie zostanie również sprawdzona zależność ilości epok do dokładności sieci. Dodatkowym celem pracy jest przybliżenie czytelnikom głębokich sieci neuronowych na przykładzie diagnozowania choroby.

Na potrzeby zadania wykorzystano zbiór: Stany ostrego brzucha (zbiór zawiera 476 osobników).

## 4. Opis problemu medycznego jako zadania klasyfikacji

Problem można podzielić na 8 klas diagnozy, które zostały przedstawione w poniższej tabeli.

*Tabela 1: Ilość osobników przypisanych do danej klasy*

Nr	Klasy (diagnozy)	Ilość osobników danej klasy
1	Niecharakterystyczny ból brzucha	157
2	Ostre zapalenie wyrostka robaczkowego	141
3	Zapalenie woreczka żółciowego	55
4	Ostre zapalenie trzustki	32

5	Niedrożność mechaniczna jelit	29
6	Perforowany wrzód trawienny	28
7	Zapalenie uchyłków jelit	17
8	Inne przyczyny ostrego bólu brzucha	17

Każdy osobnik posiada 31 cech dyskretnych takich jak:

Płeć, Wiek, Intensywność bólu, Czynniki nasilające ból, Czynniki przynoszące ulgę, Progresja bólu, Czas trwania bólu, Charakter bólu na początku zachorowania, Charakter bólu obecnie, Lokalizacja bólu na początku zachorowania, Lokalizacja bólu obecnie, Tętno, Nudności i wymioty, Apetyt, Wypróżnienia, Opory patologiczne, Objaw Murphy'ego, Objaw Blumberga, Oddawanie moczu, Poprzednie niestrawności, Żółtaczka w przeszłości, Poprzednie operacje brzuszne, Leki, Stan psychiczny, Skóra, Temperatura (pacha), Ruchy oddechowe powłok brzusznych, Wzdęcia, Umieszczenie bolesności uciskowej, Obrona mięśniowa, Wzmoczone napięcie powłok brzusznych.

## 5. Opis technologii – sieci neuronowe

Sieci neuronowe zyskały swoją popularność już w poprzednich dekadach ubiegłego wieku, jednak ich prawdziwy rozwój miał miejsce dopiero w XXI wieku. Są one jednym z wielu zagadnień, szeroko pojętej, sztucznej inteligencji. Zyskują one coraz większe zainteresowanie wśród naukowców i obserwatorów interesujących się nowymi technologiami.

Podstawowym elementem budowy sieci neuronowych jest pojedynczy neuron, który łączy się z innymi neuronami, tworząc poszczególne warstwy sieci. Budowa oraz sposób działania ma naśladować działanie ludzkiego mózgu. Sieć neuronowa składa się najczęściej z trzech połączonych ze sobą warstw. Pierwsza warstwa nazywana jest warstwą wejścia – gromadzi ona dane, a następnie przesyła dalej (każdy neuron pierwszej warstwy dostarcza dane, do każdego neuronu drugiej warstwy, czyli warstwy ukrytej). Druga warstwa to już wspomniana wyżej warstwa ukryta – to w niej ma miejsce proces uczenia. Trzecia warstwa to warstwa wyjściowa – dostarcza wyniki przeprowadzonego eksperymentu.

Na pierwszy rzut oka układ wydaje się prosty, jednak nie ujmuje to złożoności, ponieważ sieć może opierać się na nieskończonej liczbie warstw neuronowych.

## 6. Opis stosowanych algorytmów

### 6.1 Oczyszczenie i przygotowanie danych

Z otrzymanego pliku „Stany\_ostrego\_brzucha-dane.xls” z danymi zostały usunięte puste kolumny, a nazwy kolumn zostały przesunięte do nagłówków kolumn

### 6.2 Ranking cech

Do stworzenia rankingu cech został wykorzystany współczynnik korelacji Pearsona. Pozwala on zmierzyć liniowe korelacje między dwoma zmiennymi losowymi.

Współczynnik korelacji Pearsona dla próby statystycznej definiowany jest następująco:

$$r = r_{xy} = \frac{\sum x_i y_i - \bar{x} \bar{y}}{\sqrt{\sum x_i^2 - n \bar{x}^2} \sqrt{\sum y_i^2 - n \bar{y}^2}}$$

Gdzie:

$\bar{x}$  - średnia arytmetyczna z  $x$

$\bar{y}$  - średnia arytmetyczna z  $y$

Wynikiem powyższej formuły jest liczba  $-1 \leq r \leq 1$ .

Współczynnik określa tzw. korelację ujemną ( $-1 \leq r < 0$ ) oraz dodatnią ( $0 \leq r < 1$ )

Wartość  $r = 0$  oznacza brak korelacji.

Tabela 2: Ranking cech, ustanowiony przy pomocy współczynnika korelacji Pearsona

Cecha	Współczynnik korelacji Pearsona
Leki	0,38147
Intensywność bólu	0,343642
Czas trwania bólu	0,342621
Stan psychiczny	0,220931
Tętno	0,211945
Objaw Murphy'ego	0,202348
Umiejscowienie bolesności uciskowej	0,199496
Wypróżnienia	0,195283
Lokalizacja bólu obecnie	0,13194
Wzmoczone napięcie powłok brzusznych	0,063734
Poprzednie operacje brzuszne	0,058093
Charakter bólu na początku zachorowania	0,044857
Lokalizacja bólu na początku zachorowania	0,044759
Wiek	0,037162
Płeć	0,037053
Temperatura (pacha)	0,011734
Opory patologiczne	-0,01553
Żółtaczka w przeszłości	-0,03483
Poprzednie niestrawności	-0,04121
Apetyt	-0,04488
Wzdęcia	-0,05439
Progresja bólu	-0,11011
Obrona mięśniowa	-0,17172
Oddawanie moczu	-0,20076
Czynniki nasilające ból	-0,22156
Skóra	-0,30596
Charakter bólu obecnie	-0,40594
Ruchy oddechowe powłok brzusznych	-0,43451
Czynniki przynoszące ulgę	-0,46196
Objaw Blumberga	-0,47966
Nudności i wymioty	-0,54109

## 6.3 Budowa Sieci neuronowej

Przed samą budową sieci neuronowej zastosowaliśmy „One hot encoding”, czyli kodowanie etykiet jako tablicę numeryczną. Użyliśmy tej metody, ponieważ dane nie

są ze sobą skorelowane np. Lokalizacja bólu jest przedstawiana formie etykiet, które określają położenie. Z tego powodu sieć ze względu na swoją strukturę i ilość warstw ukrytych, mogłaby nie zdążyć się wyuczyć. Wszystkie kolumny, które zawierają więcej wartości niż 0 i 1 zostały poddane temu zabiegowi oprócz ostatniej kolumny z klasą diagnozy.

Label Encoding			One Hot Encoding			
Food Name	Categorical #	Calories				
Apple	1	95				
Chicken	2	231				
Broccoli	3	50				

→

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

*Rys.1: Przykład działania „One hot encoding”*

Kolejną czynnością wykonaną przed budową sieci, było przygotowanie danych do dwukrotnej walidacji krzyżowej. Zostało to wykonane, poprzez podzielenie danych na dwa równe pod względem ilości rekordów podzbiory.

Następnie została wykonana budowa sieci neuronowej. Zastosowana sieć składa się z 3 warstw:

- z warstwy wejściowej (liczba neuronów wejściowych = 106, liczba neuronów wyjściowych = 106, funkcja aktywacji = ‘relu’),
- warstwy ukrytej (dla warstwy ukrytej użytkownik deklaruje ilość neuronów, funkcja aktywacji = ‘relu’),
- warstwy wyjściowej (liczba neuronów wyjściowych = 8, funkcja aktywacji = ‘softmax’).

Parametrem metody uczenia BP jest Stochastic Gradient Descent (SGD) – metoda spadku gradientu, dla którego momentum jest ustawiane przez użytkownika w trakcie działania programu. Do nauki modelu zastosowano 5 epok.

Wyjaśnienie:

\* 106 neuronów wejściowych w warstwie wejściowej wynika z zastosowania One hot encodingu dla najlepszego uzyskanego wyniku

\* relu -  $f(x) = \max(0, x)$

\* softmax - Funkcja softmax lub inaczej znormalizowana funkcja wykładnicza, jest uogólnieniem funkcji sigmoidalnej, Transformuje ona K-wymiarowy wektor wartości rzeczywistych do K-wymiarowego wektora wartości w zakresie (0; 1), którego wartości sumują się do 1. Funkcja ta jest często implementowana w ostatniej warstwie sieci wykorzystywanej do zadań klasyfikacji, gdy klasy wykluczają się wzajemnie. Wartość  $f(x)_j$  reprezentuje prawdopodobieństwo przynależenia wartości wejściowej do klasy j.

## 7. Implementacja

### 7.1 Środowisko implementacyjne

Do projektu wykorzystano środowisko Anaconda wraz z narzędziem Jupyter Notebook. Narzędzie to, uruchamiane w przeglądarce za pomocą lokalnego serwera, jest notatnikiem, w którym można umieścić kilka typów komórek:

- Uruchamialny kod Pythona (obsługuje też inne języki)
- Dokumentację w postaci Markdown
- Czysty tekst

Narzędzie Jupyter zostało wybrane, ponieważ:

- Podczas pracy w zespole inni programiści mogą korzystać z naszych wyników i wykresów
- Można podłączyć różne źródła danych
- Szybko zbadać dane – wykonać zapytanie, wyfiltrować, obliczyć statystyki (mediana, średnia i inne)
- Łatwo go przechować w systemie kontroli wersji – notatniki są plikami json

### 7.2 Język programowania i biblioteki

Jako język programowania został wybrany Python. Jest to język programowania wysokiego poziomu ogólnego przeznaczenia, o rozbudowanym pakiecie bibliotek standardowych, którego ideą przewodnią jest czytelność i klarowność kodu źródłowego. Jego składnia cechuje się przejrzystością i zwięzłością.

Python rozwijany jest jako projekt Open Source zarządzany przez Python Software Foundation, która jest organizacją non-profit.

#### Wykorzystane biblioteki:

**Keras** jest to ramka projektowa uczenia głębokiego języka Python, która umożliwia wygodne definiowanie i trenowanie dowolnych modeli uczenia głębokiego.

**Pandas** to biblioteka open source, licencjonowana przez BSD, zapewniająca wydajne, łatwe w użyciu struktury danych i narzędzia do analizy danych dla języka programowania Python.

**NumPy** to podstawowy pakiet do obliczeń naukowych z Pythonem.

**Itertools** to zbiór narzędzi do obsługi iteratorów.

**Scikit-learn** to biblioteka do uczenia maszynowego dla języka programowania Python. Zawiera różne algorytmy klasyfikacji, regresji i grupowania, w tym zwiększanie gradientu.

## 8. Plan eksperymentu i wyniki

Plan:

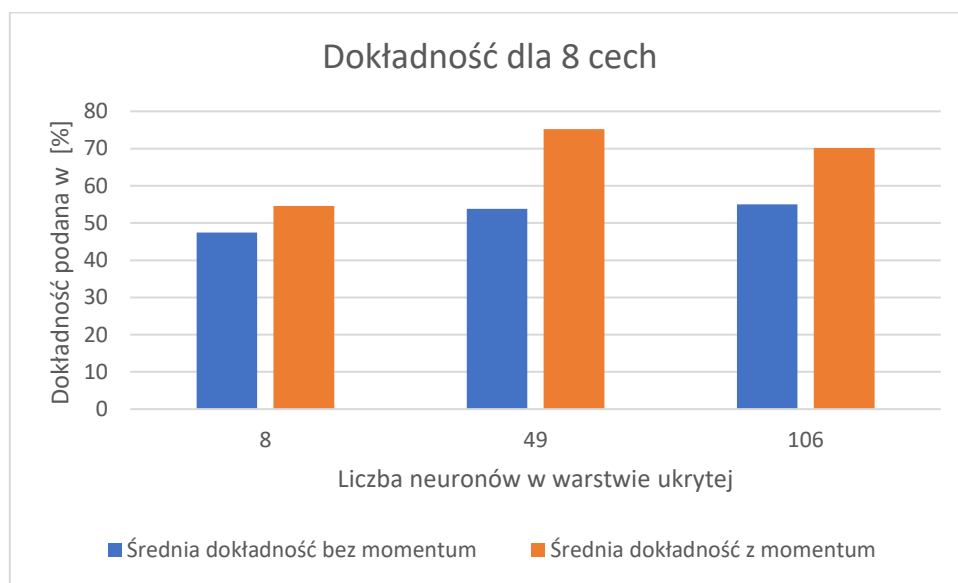
- Uruchomić sieć dla 8, 49, 106 neuronów warstwie ukrytej bez momentum
- Uruchomić sieć dla 8, 49, 106 neuronów warstwie ukrytej z momentum
- Powtórzyć powyższe kroki po 5 razy dla różnych ilości cech

- Przeprowadzić analizę porównawczą uzyskanych wyników
- Dla najlepszej uzyskanej metody przeprowadzenie badań sprawdzających zależność ilości epok do uzyskanej dokładności.
- Analiza porównawcza uzyskanych wyników

Wyniki:

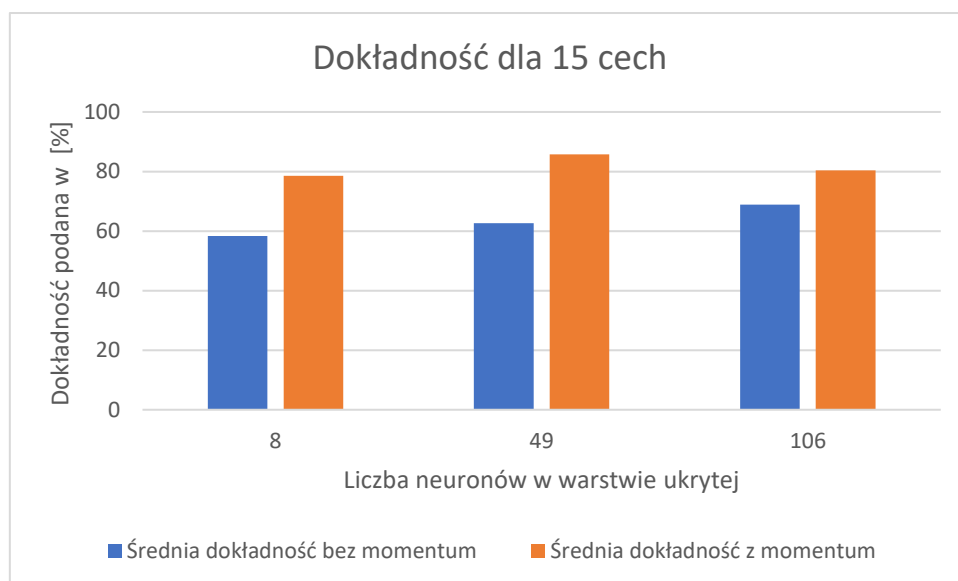
*Tabela 3: Wyniki dla pierwszych 8 wyselekcjonowanych cech wg. rankingu*

liczba neuronów w warstwie ukrytej	Średnia dokładność bez momentum	Średnia dokładność z momentum
8	47.48%	54.62%
49	53.78%	75.21%
106	55.04%	70.17%



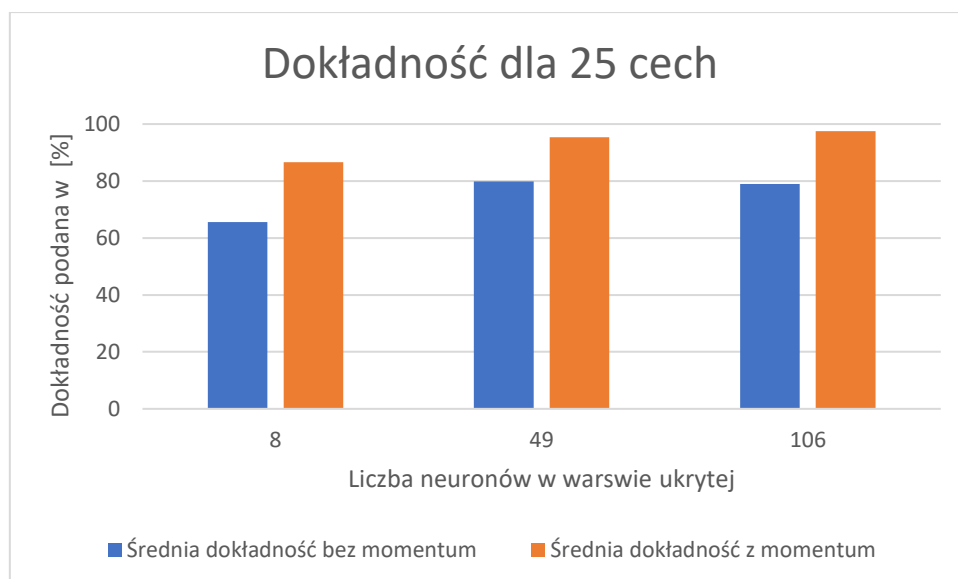
*Tabela 4: Wyniki dla pierwszych 15 wyselekcjonowanych cech wg. rankingu*

liczba neuronów w warstwie ukrytej	Średnia dokładność bez momentum	Średnia dokładność z momentum
8	58.38%	78.57%
49	62.61%	85.72%
106	68.93%	80.41%



*Tabela 5: Wyniki dla pierwszych 25 wyselekcjonowanych cech wg. rankingu*

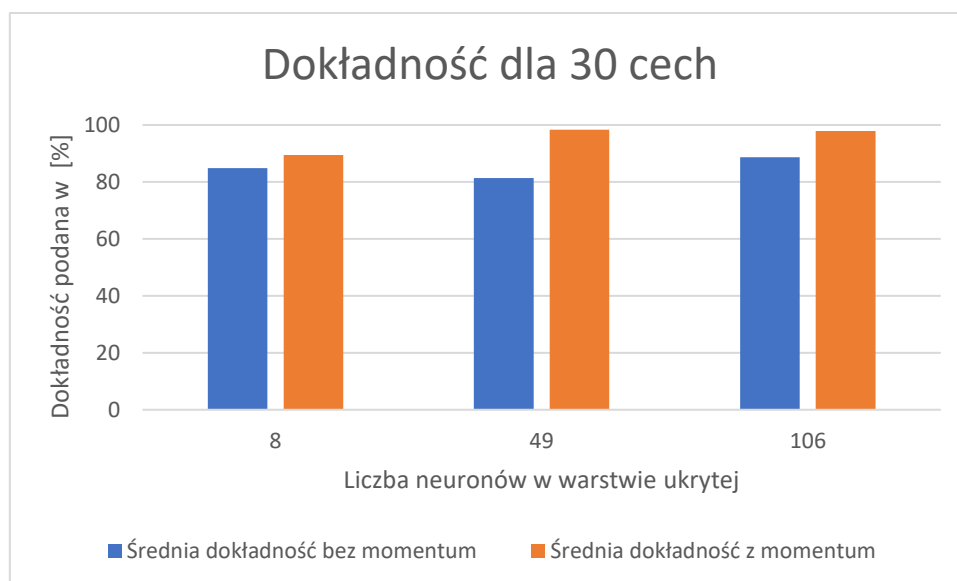
liczba neuronów w warstwie ukrytej	Średnia dokładność bez momentum	Średnia dokładność z momentum
8	65.55%	86.55%
49	79.83%	95.38%
106	78.99%	97.48%



*Tabela 6: Wyniki dla pierwszych 30 wyselekcjonowanych cech wg. rankingu*

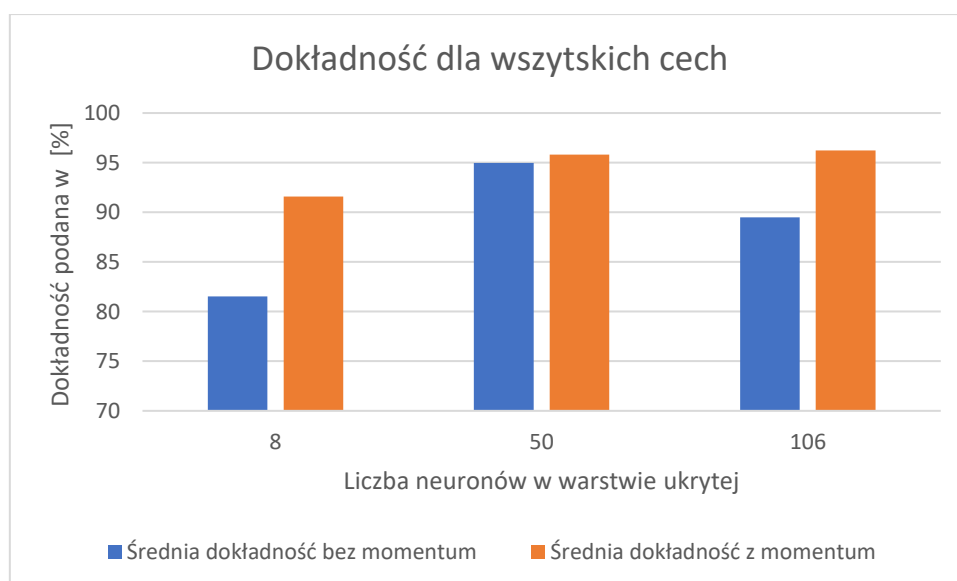
liczba neuronów w warstwie ukrytej	Średnia dokładność bez momentum	Średnia dokładność z momentum
8	84.87%	89.50%
49	81.39%	98.32%
106	88.66%	97.90%





*Tabela 7: Wyniki dla wszystkich wyselekcjonowanych cech wg. rankingu*

liczba neuronów w warstwie ukrytej	Średnia dokładność bez momentum	Średnia dokładność z momentum
8	81.51%	91.60%
49	94.96%	95.8%
106	89.50%	96.22%

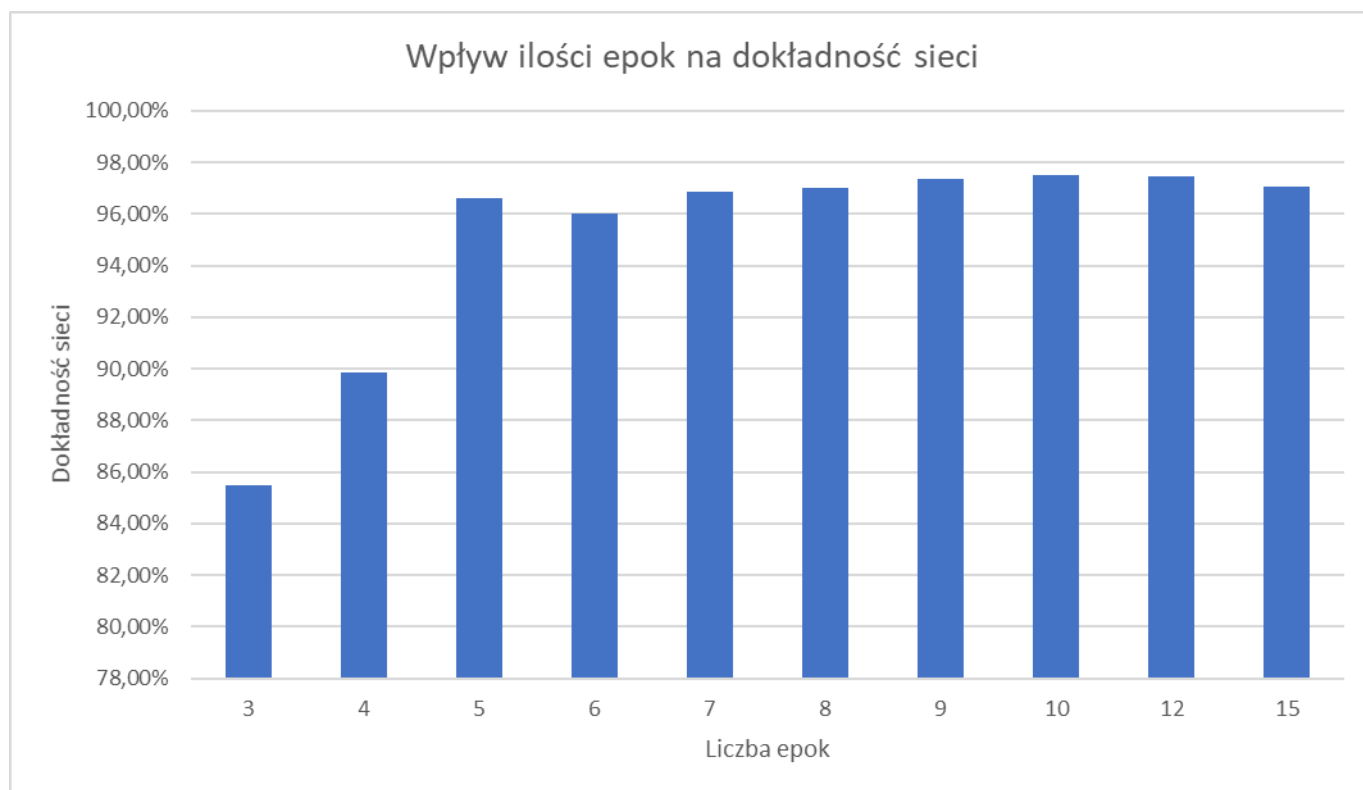


W wyniku eksperymentu zależności liczby neuronów oraz propagacji wstecznej z momentum i bez momentum do dokładności sieci możemy stwierdzić, że ilość neuronów i cech ma wpływ na dokładność. Im więcej cech i neuronów tym lepiej, dlatego też najlepsze wyniki otrzymano dla wszystkich cech. Średnia dokładność bez momentum jest gorsza niż z momentum. W konsekwencji otrzymanych rezultatów do zbadania wpływu ilości liczby epok do dokładności sieci zostanie użyty model z momentum wykorzystujący 106 neuronów oraz wszystkie cechy.

Wyniki:

Tabela 8: Wpływ ilości epok na dokładność sieci

liczba epok	3	4	5	6	7	8	9	10	12	15
Dokładność w (%)	85,50	89,83	96,60	96,01	96,87	97,02	97,37	97,53	97,48	97,06



## 9. Podsumowanie

Z eksperymentu wynika, że:

- Sieci neuronowe uczą się bardzo szybko, dlatego większa liczba epok w przypadku badania dla wszystkich dostępnych cech mogłaby spowodować przeuczenie.
- Dla mniejszej liczby cech, w celu zwiększenia dokładności, należałoby zwiększyć liczbę epok.
- Ilość neuronów i cech ma wpływ na dokładność, z przeprowadzonych badań wynika, że im więcej cech i neuronów tym lepiej, dlatego też najlepsze wyniki otrzymano dla wszystkich cech.
- Średnia dokładność bez momentum jest gorsza niż z momentum. Problem zwłaszcza jest widoczny, w przypadku mniejszej ilości cech.
- Dokładność sieci zależy od liczby epok.
- W badanym przykładzie najodpowiedniejszą liczbą epok do badania sieci jest pięć. Zwiększenie ich liczby prowadzi do przetrenowania sieci, czyli do zbytowego dopasowania sieci neuronowej do danych testowych.
- Sieci neuronowe bardzo dobrze sprawdzają się przy zadaniach klasyfikacji, w tym do diagnozowania chorób medycznych.

## Literatura

1. Dokumentacja Python [www.python.org](http://www.python.org)
2. Dokumentacja Keras - <https://keras.io/>
3. One hot encoding - <https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>
4. Francois Chollet - Deep Larning Praca z językiem Python i biblioteką Keras.
5. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning", Nature, vol. 521, no. 7553, p. 436, 2015.