Spis treści

Rozdział 1

Wstęp

1.1 Cel projektu

1.2 Motywacja

Rozdział 2

Problem medyczny

Wybrany przeze mnie problem medyczny dotyczy klasyfikacji stanów ostrego brzucha. Za ten stan odpowiedzialne mogą być różne choroby, które zawsze wymagają interwejcji lekarza.

2.1 Opis chorób

Do klasyfikacji jest 8 chorób, zatem sieć neuronowa będzie miała za zadanie przypisać 1 z 8 klas. Są to:

1. Ostre zapalenie wyrostka robaczkowego
2. Zapalenie uchyłków jelit
3. Niedrożność mechaniczna jelit
4. Perforowany wrzód trawienny
5. Zapalenie woreczka żółciowego
6. Ostre zapalenie trzustki
7. Niecharakterystyczny ból brzucha
8. Inne przyczyny ostrego bólu brzucha

Histogram pokazuje, że rozkład klas jest nierównomierny. Na 476 obiektów aż 157 to 'Niecharakterystyczny ból brzucha' i 141 ma etykietę 'Ostre zapalenie wyrostka robaczkowego'. Czyli do 2 klas należy ponad 60% obiektów. Może to mieć negatywny wpływ na jakość klasyfikacji.

2.2 Opis cech

Dane do tego problemu zawierają 31 cech. Są to odpowiedzi z wywiadu medycznego i wyniki przeprowadzonych badań. Możliwe wartości parametrów przedstawione są poniżej. Jak widać wszystkie liczby są naturalne mniejsze niż 11, także normalizacja czy skalowanie danych nie jest konieczne.

Tablica 2.1: Wszystkie cechy z odpowiedziami

L.p.	**Pytanie**	**Możliwe odpowiedzi**
Ogólne		
1	Płeć	1) męska 2) żeńska

Kontynuacja na następnej stronie

Tablica 2.1 – *Wszystkie cechy z odpowiedziami - c.d.*

L.p.	Pytanie	Możliwe odpowiedzi
2	Wiek	1) poniżej 20 lat 2) 20 - 30 lat 3) 21 - 30 lat 4) 31 - 40 lat 5) 41 - 50 lat 6) powyżej 50 lat
	Ból	
3	Lokalizacja bólu na początku zachorowania	1) prawa górna ćwiartka 2) lewa górna ćwiartka 3) górna połowa 4) prawa połowa 5) lewa połowa 6) centralny kwadrat 7) cały brzuch 8) prawa dolna ćwiartka 9) lewa dolna ćwiartka 10) dolna połowa
4	Lokalizacja bólu obecnie	0) brak bólu 1) prawa górna ćwiartka 2) lewa górna ćwiartka 3) górna połowa 4) prawa połowa 5) lewa połowa 6) centralny kwadrat 7) cały brzuch 8) prawa dolna ćwiartka 9) lewa dolna ćwiartka 10) dolna połowa
5	Intensywność bólu	0) łagodny/brak 1) umiarkowany 2) silny
6	Czynniki nasilające ból	0) brak czynników 1) oddychanie 2) kaszel 3) ruchy ciała
7	Czynniki przynoszące ulgę	0) brak czynników 1) wymioty 2) pozycja ciała
8	Progresja bólu	1) ustepujący 2) bez zmian 3) nasilający się
9	Czas trwania bólu	1) mniej niż 12 godzin 2) 12 - 24 godzin 3) 24 - 48 godzin 4) powyżej 48 godzin
10	Charakter bólu na początku zachorowania	1) przerywany 2) stały 3) kolkowy

Kontynuacja na następnej stronie

Tablica 2.1 – *Wszystkie cechy z odpowiedziami - c.d.*

L.p.	Pytanie	Możliwe odpowiedzi
11	Charakter bólu obecnie	0) brak bólu 1) przerywany 2) stały 3) kolkowy
	Inne objawy	
12	Nudności i wymioty	0) brak 1) nudności bez wymiotów 2) nudności z wymiotami
13	Apetyt	1) zmniejszony 2) normalny 3) zwiększony
14	Wypróżnienia	1) biegunki 2) prawidłowe 3) zaparcia
15	Oddawanie moczu	1) normalne 2) dysuria
	Historia	
16	Poprzednie niestrawności	0) nie 1) tak
17	Żółtaczka w przeszłości	0) nie 1) tak
18	Poprzednie operacje brzuszne	0) nie 1) tak
19	Leki	0) nie 1) tak
	Ogólne badanie	
20	Stan psychiczny	1) pobudzony/cierpiący 2) prawidłowy 3) apatyczny
21	Skóra	1) blada 2) prawidłowa 3) zaczerwieniona (twarz)
22	Temperatura (pacha)	1) poniżej 36.5 stC 2) 36.5 - 37 stC 3) 37 - 37.5 stC 4) 37.5 - 38 stC 5) 38 - 39 stC 6) powyżej 39 stC
23	Tętno	1) poniżej 60 /min 2) 60 - 70 /min 3) 70 - 80 /min 4) 80 - 90 /min 5) 90 - 100 /min 6) 100 - 110 /min 7) 110 - 120 /min 8) 120 - 130 /min 9) powyżej 130 /min
	Oglądanie brzucha	
24	Ruchy oddechowe powłok brzusznych	1) normalne 2) zniesione

Kontynuacja na następnej stronie

Tablica 2.1 – *Wszystkie cechy z odpowiedziami - c.d.*

L.p.	Pytanie	Możliwe odpowiedzi
25	Wzdęcia	0) nie 1) tak
Badania palpacyjne		
26	Umiejscowienie bolesności uciskowej	0) brak bólu 1) prawa górna ćwiartka 2) lewa górna ćwiartka 3) górna połowa 4) prawa połowa 5) lewa połowa 6) centralny kwadrat 7) cały brzuch 8) prawa dolna ćwiartka 9) lewa dolna ćwiartka 10) dolna połowa
27	Objaw Blumberga	0) negatywny 1) pozytywny
28	Obrona mięśniowa	0) nie 1) tak
29	Wzmożone napięcie powłok brzusznych	0) nie 1) tak
30	Opory patologiczne	0) nie 1) tak
31	Objaw Murphy'ego	0) negatywny 1) pozytywny

2.3 Selekcja cech

Selekcja cech jest procesem wymaganym, gdy dane nie są dobrej jakości w wielu algorytmach uczenia maszynowego. Polega ona na wyborze podzbioru najlepszych cech według ustalonego kryterium. Analitycy danych przeprowadzają selekcję z następujących powodów:

- uproszczenie modelu, w celu ułatwienia interpretacji przez badaczy,
- skrócenie czasu treningu,
- zmniejszenie wymiarowości modelu,
- zwiększenie generalizacji poprzez uniknięcie zjawiska przeuczenia.

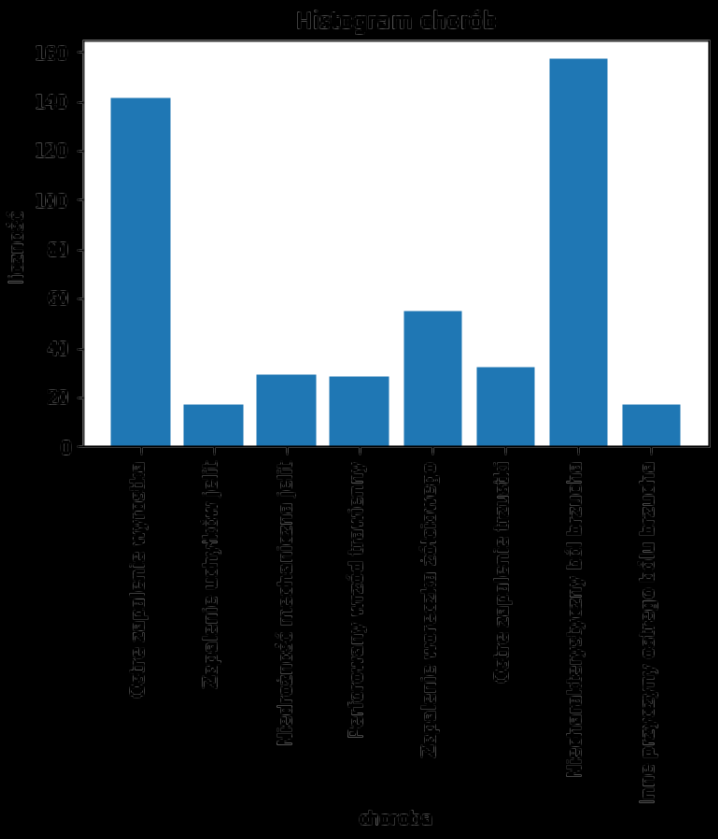
2.3.1 Test chi2

Metoda, którą wybrałem to test chi2. Jest to jedna z technik nieparametrycznych. Nadaje się bardzo dobrze do oceny istotności statystycznej cechy. Test ten polega na obliczeniu podanego poniżej wyrażenia dla każdej z cech i wybraniu takich, dla których wartość jest największa.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Gdzie:

- O_i - wartość mierzona,
- E_i - wartość oczekiwana,



Rysunek 2.1: Histogram występowania chorób

- n - liczba obiektów.

Wartości testu dla wszystkich cech mają następujące wartości:

Tablica 2.2: Wartości chi2 dla wszystkich cech

L.p.	**Cecha**	**Wartość chi2**
1	Charakter bólu obecnie	127.811
2	Czynniki przynoszące ulgę	87.453
3	Nudności i wymioty	84.633
4	Czas trwania bólu	84.273
5	Umiejscowienie bolesności uciskowej	77.456
6	Lokalizacja bólu obecnie	70.865
7	Czynniki nasilające ból	59.357
8	Tętno	58.152
9	Apetyt	54.489
10	Wypróżnienia	42.184
11	Charakter bólu na początku zachorowania	32.127
12	Lokalizacja bólu na początku zachorowania	31.430
13	Ruchy oddechowe powłok brzusznych	31.192
14	Progresja bólu	30.502
15	Objaw Blumberga	21.387

Kontynuacja na następnej stronie

Tablica 2.2 – *Wszystkie cechy z odpowiedziami - c.d.*

L.p.	**Pytanie**	**Możliwe odpowiedzi**
16	Wiek	21.228
17	Skóra	20.202
18	Intensywność bólu	18.438
19	Temperatura (pacha)	17.708
20	Stan psychiczny	15.930
21	Leki	15.554
22	Objaw Murphy'ego	13.666
23	Obrona mięśniowa	13.062
24	Oddawanie moczu	12.322
25	Wzmożone napięcie powłok brzusznych	11.406
26	Wzdęcia	8.771
27	Opory patologiczne	8.504
28	Poprzednie operacje brzuszne	7.007
29	Płeć	6.195
30	Poprzednie niestrawności	4.470
31	Żółtaczka w przeszłości	0.590

Najlepszymi cechami są te, które mają wysoką wartość chi2. Zatem ograniczając liczbę cech, do klasyfikacji brane będą te z góry tabeli. Cechy o niskiej wartości, jak na przykład 'Żółtaczka w przeszłości', nie polepszą klasyfikacji, a mogą ją nawet pogorszyć.

Rozdział 3

Techologie

3.1 Python

Rysunek 3.1: Logo języka Python

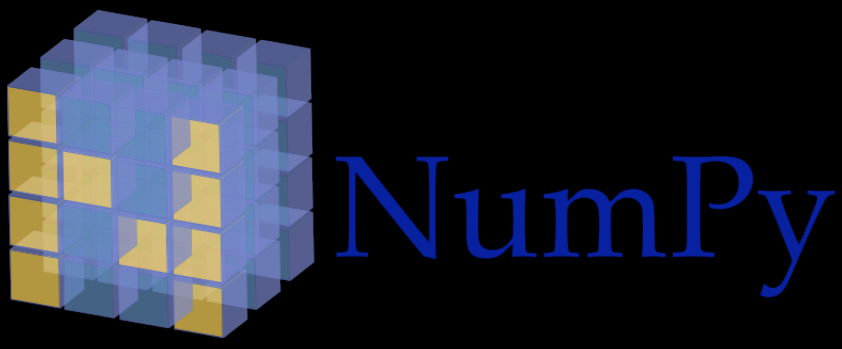
Python to otwarto-źródłowy, wysoko poziomowy język programowania ogólnego przeznaczenia. Stworzony został 26 lat temu przez holenderskiego programistę Guido van Rossuma. Najpolularniejszy interpreter Pythona napisany jest w języku C. Jednak w odróżnieniu od C, C++ i Javy, Python jest interpretowalny i nie używa się w nim nawiasów klamrowych do oddzielenia bloków kodu. Jest przez to bardziej czytelny i nie odstrasza ludzi aspirujących do bycia programistami. Zamias klamr stosuje się wcięcia w kodzie, które powinny wynosić 4 spacje na każdy poziom. Od wersji 3.5 w Pythonie można jawnie stosować typowanie, czyli na przykład twórca funkcji może umieścić informację w kodzie, jakiego typu powinny być argumenty i jaki typ funkcja zwraca. Dzięki temu czas potrzebny na zrozumienie cudzego kodu staje się krótszy.

Python ma wiele zastosowań:

- nauka programowania,
- web development,
- aplikacje konsolowe,
- aplikacje okienkowe,
- gry komputerowe,
- naukowe,
- analiza danych.

Od kilku lat Python zyskuje duże zainteresowanie naukowców z różnych dziedzin nauki z racji swojej prostoty i wszechstronności. Powstało również wiele gotowych modułów do zastosowania w uczeniu maszynowym, ale nie będę używał ich w tym projekcie.

3.2 NumPy



Rysunek 3.2: Logo NumPy

NumPy to otwarto-źródłowa biblioteka do Pythona służąca do obliczeń naukowych. Umożliwia przechowywanie danych w wielowymiarowych tablicach i macierzach (tensorach) oraz wykonywanie skomplikowanych funkcji na nich. Napisana została w większości w języku C, co sprawia, że kod wykonywany jest szybciej niż w samym Pythonie. Tablice z NumPy są wykorzystywane w wielu bibliotekach, jako podstawowa struktura danych. W tym projekcie używam jej do przechowywania wag w każdej warstwie sieci.

3.3 matplotlib

Rysunek 3.3: Logo matplotlib

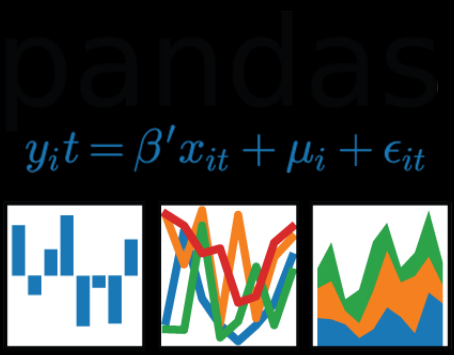
Matplotlib to najpopularniejsza biblioteka do tworzenia wykresów w Pythonie. Wraz z biblioteką NumPy bardzo często wykorzystywana jest do analizy i wizualizacji danych. Jest bardzo prosta w obsłudze. W kilka linii jesteśmy w stanie stworzyć prosty wykres i wyeksportować go do pliku graficznego. Wspiera takie typy wykresów jak:

- liniowy,
- histogram,
- punktowy,
- 3D,
- biegunowy.

Pozwala również wyświetlać obrazy w oknach z poziomu skryptu w Pythonie.

3.4 pandas

Pandas to biblioteka napisana w Pythonie służąca do manipulacji i analizy danych. Oferuje struktury danych, które ułatwiają operowanie na plikach csv, json i xlsx. Umożliwia operacje podobne do znanych z języka SQL. Są to: grupowanie danych, sortowanie po indeksie lub po innej kolumnie, łączenie tabel i usuwanie duplikatów.



Rysunek 3.4: Logo pandas

3.5 Git

Rysunek 3.5: Logo Gita

Git to rozproszony system kontroli wersji, czyli narzędzie do śledzenia zmian w plikach źródłowych. Jest to oprogramowanie używane głównie do zarządzania kodem. ale może być używane również do trzymania historii innych plików. Git ma na celu szybkość, spójność danych i wspieranie pracy rozproszonej wśród zespołów. Nie wymaga ciągłego dostępu do Internetu. Jest wykorzystywany w prawie wszystkich nowoczesnych projektach.

Git został napisany przez Linusa Torvaldsa w 2005 roku, jako narzędzie do tworzenia jądra Linuksa, gdyż żaden inny system kontroli wersji nie spełniał jego wymagań.

Główną strukturą w Gicie jest repozytorium. Każde repozytorium przypisane jest do jednego projektu. Posiada ono historię w formie grafu skierowanego, który jest drzewem. Git umożliwia poruszanie się po tym drzewie pozwalając przeglądać repozytorium w danym stanie.

Praca z Gitem rozpoczyna się od sklonowania istniejącego repozytorium lub stworzenia nowego, pustego. Użytkownik po zmianie jakiegoś pliku śledzonego przez Gita może zrobić 'commit', czyli zapisać obecny stan projektu. Każdy 'commit' ma przypisaną wiadomość, w której twórca 'commita' informuje, co zmienił. Po 'scommitowaniu' można zsynchronizować stan repozytorium z głównym serwerem. Dopiero wtedy inni użytkownicy mogą zobaczyć, jakie zaszły zmiany i pobrać do siebie najnowszą wersję.

Git to potężne narzędzie, każdy programista powinien potrafić z niego korzystać. Łatwo jest poznać podstawy Gita i nie wymaga dużo czasu opanowanie ich. Zaawansowana znajomość Gita pozwala na robienie niesamowitych rzeczy w repozytorium.

W projekcie inżynierskim korzystam z Gita do zapisywania postępów w tworzeniu aplikacji. Kod jest przechowywany na serwerze firmy GitHub.

Rysunek 3.6: Logo Dockera

3.6 Docker

Docker to otwarto-źródłowe narzędzie służące do konteneryzacji aplikacji. Zapewnia dodatkową warstę abstrakcji nad systemem operacyjnym. Działa zarówno na Linuksie, jak i na Windowsie. Pierwsze wersje Dockera od 2013 roku napisane były w Pythonie, a kolejne w języku Go.

W wielu aplikacjach Docker używany jest w celu ułatwienia wdrożenie aplikacji na serwery produkcyjne. Jest przydatny również w czasie wytwarzania dla deweloperów, gdyż idealnie nadaje się na środowisko testowe.

Docker udostępnia na swojej stronie internetowej wiele predefiniowanych obrazów z zainstalowanymi aplikacjami. które są gotowe do użycia. Zalogowani użytkownicy mogą również publikować swoje własne obrazy zbudowane przez nich. Pozwala to dzielić się swoją pracą z całą społecznością.

Praca z Dockerem polega na uruchomieniu kontenera z wybranego obrazu. Obraz dockerowy to tak jakby zapisany stan maszyny wirtualnej. Kontener jest konkretną uruchomioną instancją obrazu.

Poza oficjalnymi obrazami, istnieje również możliwość tworzenia własnych obrazów do poszczególnych aplikacji. Polega to na stworzeniu pliku domyślnie o nazwie Dockerfile, gdzie podany jest obraz bazowy oraz lista komend do wykonania. Po zbudowaniu obrazu jedną komendą, można uruchomiać kontenery.

W swoim projekcie inżynierskim korzystałem z Dockera, badania przeprowadzałem na serwerze, gdzie nie ma zainstalowanych wymaganych przeze mnie zależności. Dlatego to było jedynym rozwiązaniem.

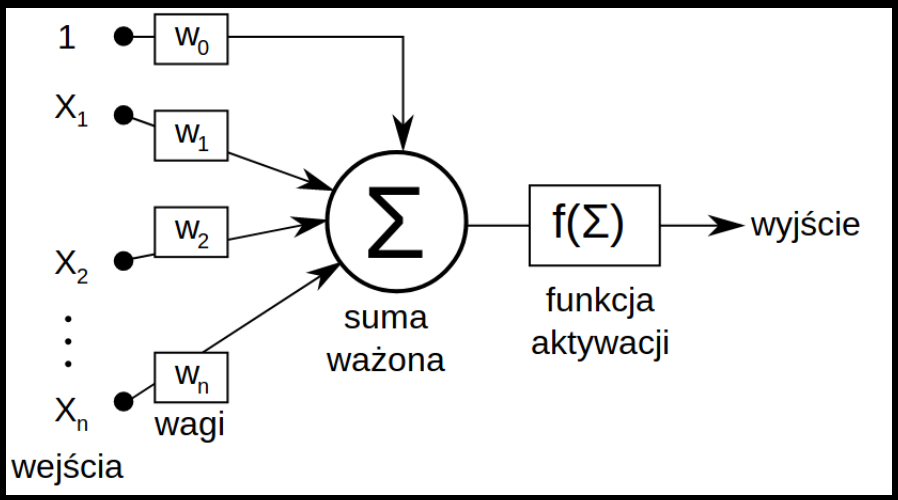
Rozdział 4

Sieć neuronowa

4.1 Wprowadzenie

Sztuczna sieć neuronowa to pewna struktura matematyczna, która może być zaimplementowana programowo lub sprzętowo. Początkowo taki twórcy takich modeli inspirowali się zwierzęcym mózgiem, w którym połączone ze sobą neurony tworzą sieć. Taka sieć przetwarza sygnały wejściowe wykonując na nich pewne operacje. Sieci wykorzystywane są często do rozwiązywania problemów klasyfikacji, z racji na ich zdolność uczenia. Np. potrafią przetwarzać zdjęcia i opisywać, co się na nich znajduje. Przed skorzystaniem z sieci należy ją nauczyć, co sprowadza się do przekazywania na wejście sieci danych uczących razem z poprawną klasą, do której dane obiekty należą.

4.2 Neuron



Rysunek 4.1: Schemat neuronu

Neuron stanowi podstawowy budulec sztucznej sieci neuronowej. Składa się z ustalonej liczby wejść, wraz z odpowiadającymi im wagami. Ponadto neuron zawiera nieliniową funkcję aktywacji oraz jedno wyjście. Jego zadanie to obliczenie iloczynu skalarnego wektora wejść z wektorem wag. Dodatkowo możemy przyjąć, że bias jest dodatkowym wejściem neuronu o wartości 1. Następnie obliczona suma ważona poddawana jest funkcji aktywacji i przekazywana na wyjście neuronu. W procesie uczenia wagi w neuronie zmieniają się tak, by wyliczona wartość funkcji błędu była jak najmniejsza.

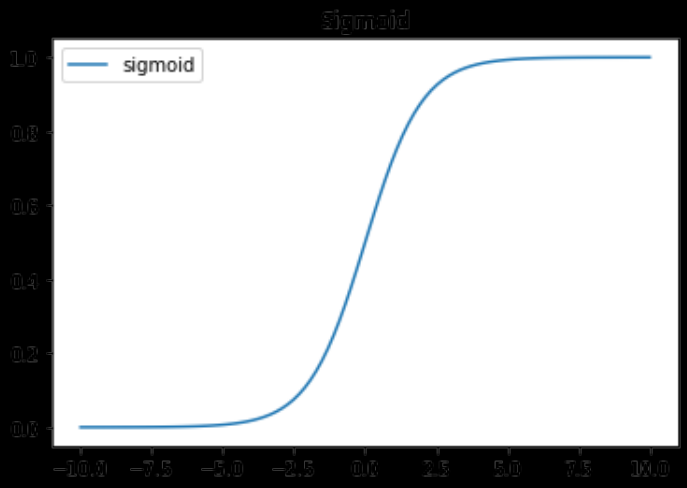
4.2.1 Funkcja aktywacji

Funkcja aktywacji to funkcja, która wykorzystywana jest w sztucznych sieciach neuronowych, a dokładniej w samym neuronie do zmiany wartości wyjścia. Ma to na celu sprawienie, że sieć jest w stanie lepiej się uczyć nawet przy małej liczby neuronów. W uczeniu maszynowym znanych jest wiele rodzajów takich funkcji. W tej pracy opiszę tylko dwie, które użyłem do budowy sieci neuronowej.

Sigmoid

Pierwszą opisywaną funkcją jest sigmoid, zwana też 'sigmoidalną funkcją unipolarną'. Bardzo dobrze nadaje się jako funkcja aktywacji, gdyż jej dziedzina to cały zbiór liczb rzeczywistych. Ma tę cechę, że zbiór warości mieści się w zakresie (0, 1). Jest to również wada, że szybko się 'nasyca'. Kolejnym minusem tej funkcji jest to, że wartości nie zcentralizowane wokół zera. Ponadto wykorzystuje funkcję ekpotencjalną, która jest kosztowna obliczeniowo. Jej wzór to:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Rysunek 4.2: Funkcja aktywacji - sigmoid

Tangens hiperboliczny

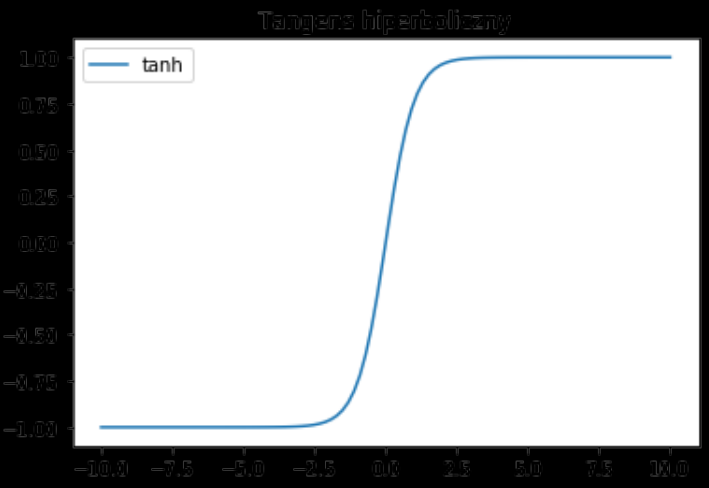
Kolejna funkcja, która wykorzystywana jest w sieciach neuronowych to tangens hiperboliczny (tanh). W tym przypadku również dziedziną jest zbiór liczb rzeczywistych. Spłaszcza wyjście w zakresie (-1, 1). Podobnie jak sigmoid szybko się 'nasyca'. W przeciwieństwie do poprzedniej funkcji jest scentralizowany wokół zera. Tanh również korzysta z funkcji ekpotencjalnej. Jej wzór to:

$$tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

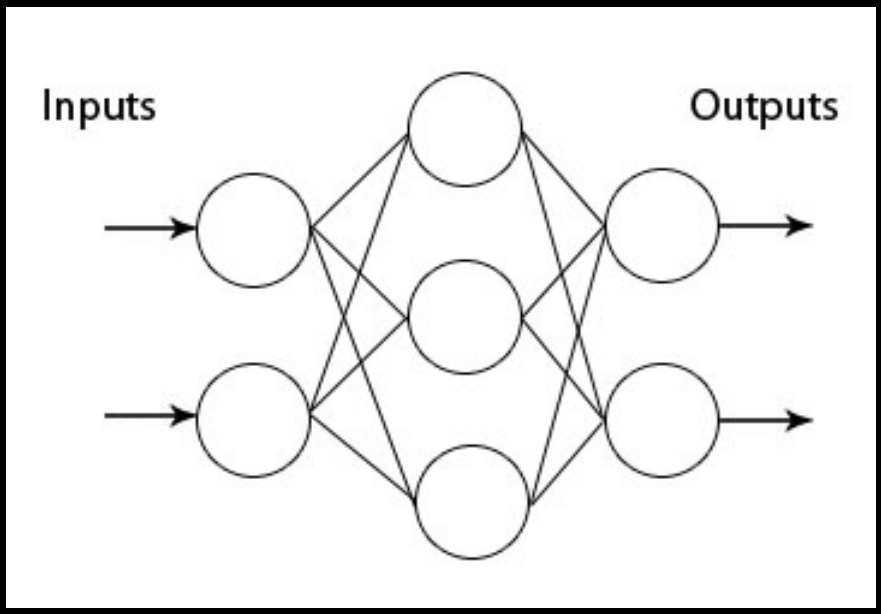
4.3 Model wielowarstwowy

Kiedy pojedyncze neurony mają te same sygnały wejściowe, tworzą wówczas tak zwaną warstwę w sieci neuronowej. Dlatego sieć neuronowa ma budowę warstwową. Zawsze występuje warstwa wejściowa i wyjściowa. Ponadto mogą występować warstwy ukryte. Ich ilość zależy od rozwiązywanego problemu.

W warstwie wejściowej znajduje się tyle neuronów, ile jest badanych cech. W moim przypadku będzie mniej niż 31. Neurony w tej warstwie nie mają wag, lecz przekazują dalej dokładnie to, co otrzymały.



Rysunek 4.3: Funkcja aktywacji - tanh



Rysunek 4.4: Schemat neuronu

Liczba neuronów w warstwie wyjściowej również nie jest przypadkowa. Warstwa ta składa się z takiej samej liczby neuronów, co liczba klas w zadanym problemie. Problem medyczny, na którym pracuję dotyczy klasyfikacji ośmio klasowej, dlatego będzie osiem neuronów wyjściowych. Każdy z nich będzie zwracał wartość przynależności do danej klasy.

Rozdział 5

Opis architektury aplikacji

5.1 Schemat warstwy

```
class Layer:
def __init__(self, shape, activation='sigmoid'):
...

def feedforward(self, x: np.ndarray) -> np.ndarray:
...

def calc_delta(self, d: np.ndarray = None):
...

def calc_gradient(self):
...

def update_weights(self, learning_rate=.2):
...
```

Powyższy fragment kodu przedstawia schemat klasy Layer. Jest to implementacja jednej warstwy w sieci neuronowej. Przypomina schemat struktury danych zwanej listą dwukierunkową, gdyż zawiera referencje do poprzedniej i następnej warstwy. Klasa zawiera w sobie tablicę, która jest składa się z wag połączeń do poprzedniej warstwy.

Przy tworzeniu instancji należy podać krotkę liczba oznaczającą kształt warstwy. Dodatkowo można przekazać nazwę funkcji aktywacji, którą domyślnie jest to sigmoid.

Zadanie funkcji 'feedforward' to przyjęcie tablicy liczb z poprzedniej warstwy, obliczenie iloczynu skalarnego z aktualnymi wagami i poddanie wyjścia funkcją aktywacji. Następnie funkcja powinna rekurencyjnie wywołać samą siebie na następnej warstwie jeśli nie jest ostatnia w sieci. W przeciwnym przypadku zwraca wyliczone wyjście całej sieci.

Funkcja 'calc_delta' wywoływana jest rekurencyjne, ale w przeciwnym kierunku. Oblicza ona różnicę pomiędzy spodziewanym wyjściem warstwy a aktualnym. Pozwoli to później skorygować wagi każdej warstwy.

Następna funkcja to 'calc_gradient'. Jest równnież wywoływana rekurencyjnie zaczynając od końca sieci. Oblicza wartość gradientu na podstawie wyjścia warstwy oraz wartości delty.

Ostatnią funkcją jest 'update_weights'. Jak jej nazwa wskazuje, to właśnie ona zmienia wartości wag w warstwach odejmując iloczyn obliczonego gradientu ze współczynnikiem uczenia. W miarę uczenia współczynnik uczenia może się zmieniać, dlatego uznałem, że to dobre miejsce na dostarczenie tego współczynnika warstwie sieci neuronowej.

5.2 Schemat modelu

5.2.1 Proces uczenia

Rozdział 6

Przeprowadzone badania

Rozdział 7

Podsumowanie

7.1 Dalsze możliwości rozwoju

7.2 Co mogłem zrobić lepiej

Tekst podsumowania

Spis rysunków

Spis tablic