

Politechnika Warszawska  
Wydział Elektryczny  
Kierunek Informatyka

# SPECYFIKACJA IMPLEMENTACYJNA

Wykonał: Aliaksandr Karolik

Warszawa, 23.03.2019

---

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Informację o środowisku programistycznym</b>	<b>2</b>
<b>3</b>	<b>Diagram modułów</b>	<b>3</b>
<b>4</b>	<b>Opis modułów</b>	<b>4</b>
4.1	Moduł argumentsParser . . . . .	4
4.2	Moduł fileParser . . . . .	4
4.3	Moduł fileWriter . . . . .	5
4.4	Moduł training . . . . .	6
4.5	Moduł maths . . . . .	6
4.6	Moduł predict . . . . .	6
<b>5</b>	<b>Testy jednostkowe</b>	<b>7</b>
5.1	Moduł fileParser . . . . .	7
5.2	Moduł training . . . . .	8
5.3	Moduł predict . . . . .	8
5.4	Moduł argumentsParser . . . . .	9

---

## 1 Wprowadzenie

Program będzie umożliwiał rozpoznawanie języka tekstu za pomocą sztucznej sieci neuronowej. Uczenie sieci będzie za pomocą jedną z najczęściej stosowanych technik uczenia się w sieciach neuronowych. Technika ta ma nazwę *Metoda propagacji wstecznej*.

Metoda ta polega na przesyłaniu błędów od warstwy wyjściowej sieci do jej warstwy wejściowej. Wartości tych błędów zwalają na drodze iteracyjnych zmian wag połączeń w sieci poszukiwać minimum błędu pomiędzy Pożądanym wynikiem a aktualnym wyjściem sieci.

## 2 Informację o środowisku programistycznym

Program zostanie napisany w języku Python wersji trzeciej. Program będzie pisany w zintegrowanym środowisku programistycznym PyCharm wersji 2018.1.3.

Następujące biblioteki języka Python zostaną użyte w projekcie:

- Biblioteka **NumPy**. Biblioteka jest podstawowym zestawem narzędzi dla języka Python umożliwiającym zaawansowane obliczenia matematyczne, w szczególności do zastosowań naukowych (tzw. obliczenia numeryczne, jak mnożenie i dodawanie macierzy, diagonalizacja czy odwrócenie, całkowanie, rozwiązywanie równań, itd.).
- Biblioteka **Matplotlib**. Matplotlib – biblioteka do tworzenia wykresów dla języka programowania Python i jego rozszerzenia numerycznego NumPy.

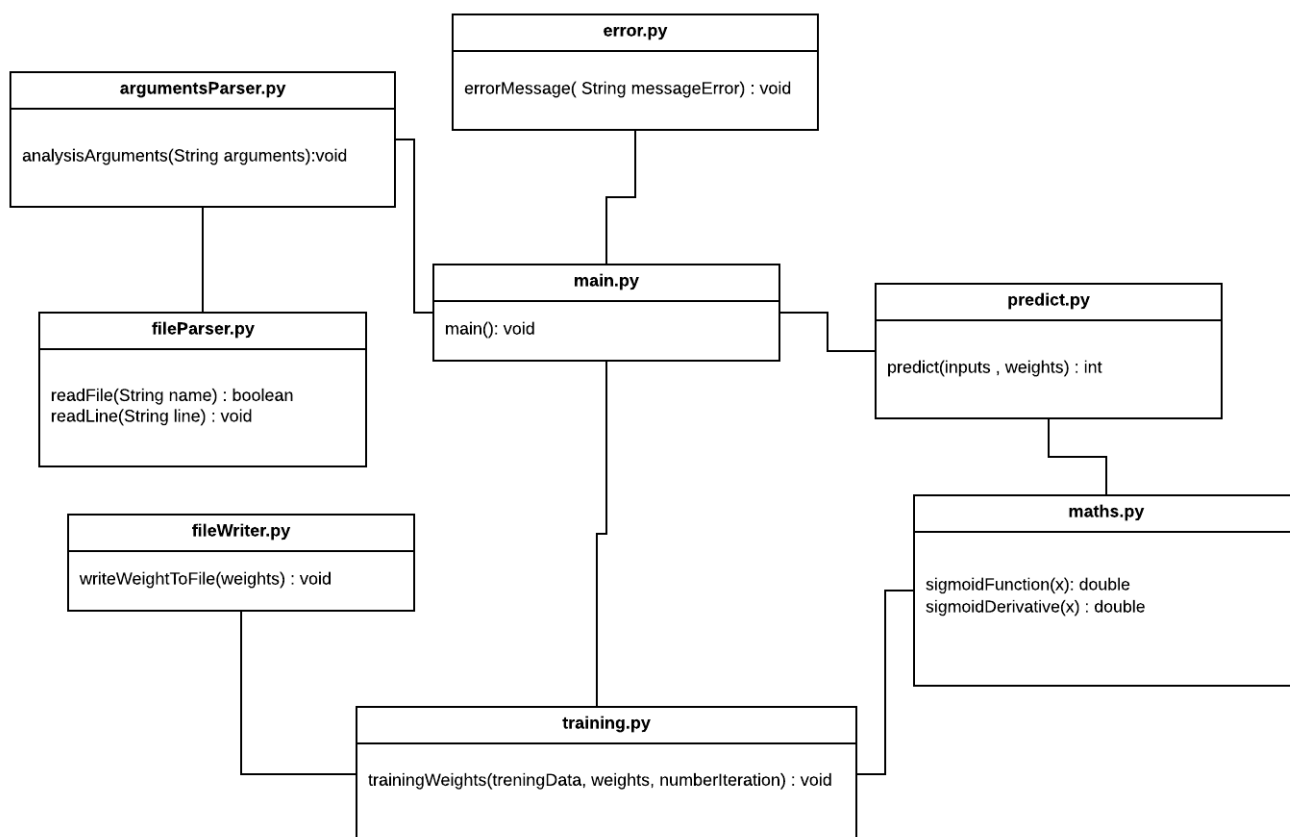
Sprzęt, na którym program będzie testowany:

- Intel Core i5-7300HQ 2,5 GHz
- Pamięć RAM o pojemności 8 GB

- 
- System Ubuntu 18.04.1

### 3 Diagram modułów

Poniższy diagram prezentuje podstawową architekturę programu. Diagram pozwala zaobserwować najważniejsze moduły programu oraz podstawowe funkcje modułów. Diagram jest początkową wizją programu to znaczy, że podczas implementacji mogą wejść zmiany.



Rysunek 1: Podstawowy diagram modułów

---

## 4 Opis modułów

### 4.1 Moduł `argumentsParser`

Moduł jest odpowiedzialny za przetwarzanie argumentów z którymi program został uruchamiony. Główną funkcją tego modułu jest funkcja o nazwie **`analysisArguments`**.

Zadaniem funkcji **`analysisArguments`** jest przetwarzanie argumentów z którymi program był uruchamiony. Funkcja musi rozpoznać w którym trybie program musi działać. W zależności od wybranego trybu działania programu funkcja musi rozpoznawać dodatkowe parametry podane przez użytkownika. Argumenty które musi rozpoznawać funkcja są następujące:

- **`-t`** argument mówiący o tym że program ma działać w trybie uczenia sieci neuronowej;
- **`-p`** argument mówiący o tym że program ma działać w trybie rozpoznawania języka tekstu;
- **`-f`** argument mówiący że następny argument jest plikiem wejściowym;
- **`-i`** argument mówiący o tym że następnie będzie podana ilość iteracji w ciągu których sieć będzie się uczyć;

### 4.2 Moduł `fileParser`

Moduł jest odpowiedzialny za poprawne wczytanie pliku wejściowego podanego przez użytkownika. Główną metodą tego modułu jest funkcja o nazwie **`readLine`**.

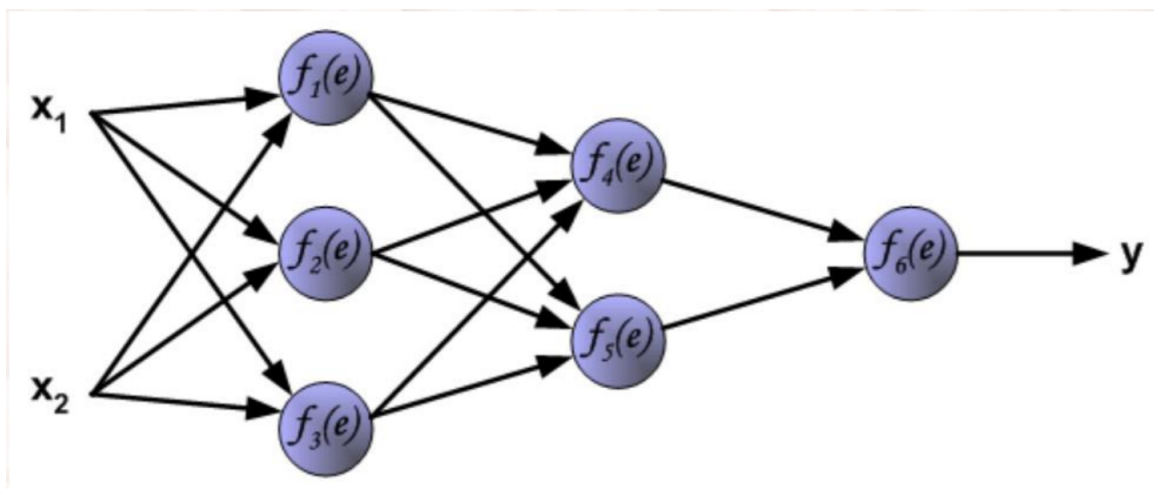
Zadaniem funkcji jest przeczytanie linii tekstu. Podczas czytania linii znak po znaku funkcja ma zliczać wystąpienia poszczególnych liter w tekście i zapisywać ilość wystąpień do macierzy. Następnie ta macierz będzie przetwarzana przez sieć neuronową dla rozpoznania języka.

### 4.3 Moduł fileWriter

Moduł jest odpowiedzialny za odpowiednie zapisanie wag które charakteryzują połączenia między neuronami.

Dla implementacji struktury sieci będzie urzyta lista list w taki sposób, iż główna lista reprezentuje warstwy sieci, a wewnętrzne listy reprezentują neurony w tych warstwach.

Przykład zapisania wag w pamięci dla sieci następującej postaci:



Rysunek 2: Przykład sieci

Główna lista będzie zawierać poniższe listy

$x_1$	$x_2$		$y_1$	$y_2$	$y_3$		$y_4$	$y_5$	
F1	w11	w12	F4	w41	w42	w43	F6	w64	w65
F2	w21	w22	F5	w51	w52	w53			
F3	w31	w32							

Funkcja z modułu **fileWrite** o nazwie **writeWieghtToFile** ma odpowiednio zapisać wagi do pliku aby przy użyciu programu dla rozpoznania języka tekstu program mógł z powrotem wczytać wagi.

---

#### 4.4 Moduł training

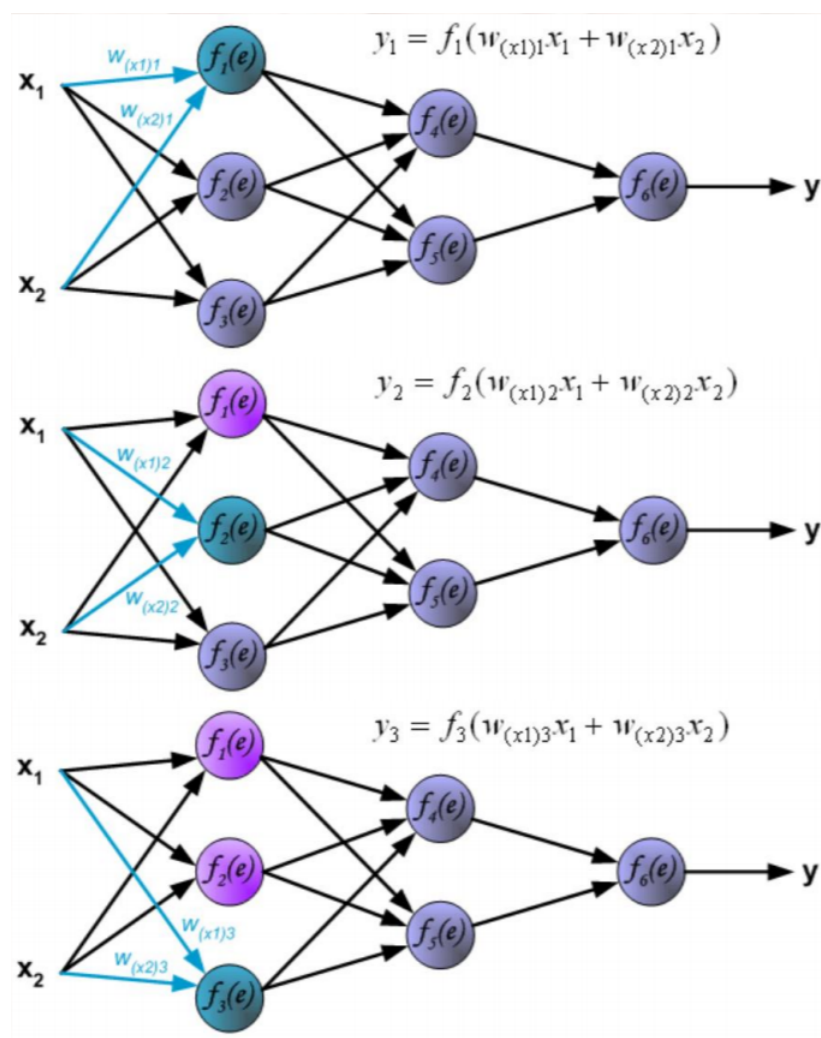
Jeden z najważniejszych modułów programu odpowiadający za uczenie sieci neuronowej czyli za zmianę wag połączeń pomiędzy neuronami. Główną funkcją modułu jest funkcja o nazwie **trainingWeights**. W funkcji **trainingWeights** będzie zaimplementowany algorytm wstecznej propagacji błędów (*ang. backpropagation*). Dla obliczenia wartości sigmoidalnej funkcji oraz jej pochodnej będą użyte funkcje z modułu **maths**.

#### 4.5 Moduł maths

Moduł odpowiadający za matematyczne obliczenia. Główną rolą tego modułu jest wyodrębnienie matematycznych działań od logiki algorytmu. Moduł będzie odpowiadał za obliczenia wartości sigmoidalnej funkcji oraz jej pochodnej. Również w tym modułu będą zaimplementowane funkcje działające na macierzach.

#### 4.6 Moduł predict

Moduł będzie odpowiadał za rozpoznanie języka. Najważniejszą funkcją tego modułu jest funkcja o nazwie **predict**. Funkcja używając wagi oraz dane wejściowe z pliku podanego przez użytkownika będzie przeprowadzać obliczenia poszczególnych wartości funkcji aktywacji dla neuronów wszystkich warstw. W wyniku obliczeń uda się stwierdzić czy jest to język na przykład angielski. Poniżej jest pokazany mechanizm obliczenia wartości funkcji aktywacji dla neuronów:



Rysunek 3: Przykład obliczenia funkcji aktywacji

## 5 Testy jednostkowe

### 5.1 Moduł fileParser

Test będzie polegać na wczytaniu przez program wcześniej przygotowanego pliku w którym już wiadoma ilość wystąpień poszczególnych liter. Przykładowa zawartość pliku:



---

aaaaaabbbbb cc dd mm

Test zostanie zaakceptowany gdy program poprawnie przetworzy podany plik wejściowy. Dla przykładowej zawartości pliku musi powstać odpowiednia lista reprezentująca wystąpienia poszczególnych liter w tekście. Lista będzie zawierać następujące wartości:

6  
4  
2  
0  
0  
.  
.  
.  
.  
2

## 5.2 Moduł training

Testowanie tego modułu będzie polegać na tym że zostanie przygotowany zestaw danych w taki sposób że można będzie przewidzieć jakie wartości wag połączeń pomiędzy neuronami ustawią się po jednej iteracji.

Test zostanie zaakceptowany jeżeli po jednej iteracji dla przygotowanych danych wagi połączeń będą poprawne. W innym przypadku test zostanie odrzucony.

## 5.3 Moduł predict

Dla testowania tego modułu będzie wygenerowany plik zawierający wagi połączeń pomiędzy neuronami. Test będzie polegał na sprawdzaniu czy moduł poprawnie rozróżnia język.

---

Test zostanie zaakceptowany gdy na wejście będzie podany tekst w języku angielskim oraz funkcja rozpoznająca język zwróci 1 oraz gdy zostanie podany tekst w języku niemieckim funkcja zwróci 0.

#### **5.4 Moduł argumentsParser**

Moduł zostanie przetestowany podczas implementacji całego programu.