przedmiot:

Programowanie komputerów

Temat:

**Uniwersalna biblioteka sieci neuronowych.**

**Autor:**

**Dawid Małkiewicz**

POLITECHNIKA ŚLĄSKA

WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

Gliwice. 2013

[1. Wstęp 3](#_Toc367281354)

[1.1. Cel projektu 3](#_Toc367281355)

[1.2. Przewodnik po treści pracy 3](#_Toc367281356)

[2. Analiza problemu 3](#_Toc367281357)

[2.1. Definicja oraz opis neuronu 3](#_Toc367281358)

[2.2. Definicja oraz opis sieci neuronowej 3](#_Toc367281359)

[3. Specyfikacja zewnętrzna 4](#_Toc367281360)

[3.1. Konfiguracja platformy programistyczno – testowej 4](#_Toc367281361)

[4. Specyfikacja wewnętrzna 4](#_Toc367281362)

[4.1. Architektura biblioteki 4](#_Toc367281363)

[5. Narzędzia programistyczne 5](#_Toc367281364)

[5.1. Testy jednostkowe 5](#_Toc367281365)

[5.2. Statyczna analiza kodu źródłowego 6](#_Toc367281366)

[5.3. Analiza stylu kodu źródłowego 7](#_Toc367281367)

[6. Uwagi końcowe 7](#_Toc367281368)

[6.1. Wnioski 7](#_Toc367281369)

# Wstęp

## Cel projektu

Celem projektu było stworzenie uniwersalnej implementacji jednokierunkowej sieci neuronowej, która pozwoli na jej użycie w wielu innych projektach informatycznych.

## Przewodnik po treści pracy

Rozdział 1 zawiera krótki wstęp, zarys problemu oraz opis celu opracowywanego projektu.

Rozdział 2 poświęcono na analizę tematu dotyczącego projektowanej biblioteki, w którym opisano wymagania funkcjonalne i nie funkcjonalne oraz przyjęte założenia.

Rozdział 3 zawiera opis specyfikacji zewnętrznej, w której opisano konfigurację platformy programistyczno - testowej.

W rozdziale 4 skupiono się na specyfikacji wewnętrznej, w której opisano architekturę biblioteki.

Rozdział 5 zawiera opis narzędzi użytych podczas implementacji.

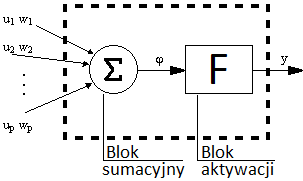
Rozdział 6 zawiera uwagi końcowe oraz wnioski ze stworzenia implementowanej biblioteki.

# Analiza problemu

## Definicja oraz opis neuronu

Neuron jest to prosty obiekt pozwalający przetwarzać wartości sygnałów na jego wejściu w pojedynczą wartość wystawianą na jego wyjściu. Sposób przetwarzania sygnałów wejściowych jest określany przez funkcje aktywacji (np. poprzez funkcję sigmoidalną).

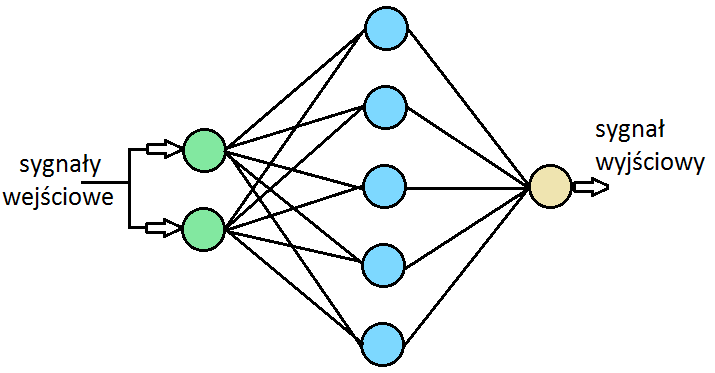
Poniżej została przedstawiona graficzna reprezentacja modelu neuronu.



## Definicja oraz opis sieci neuronowej

Sieć neuronowa (sztuczna sieć neuronowa) jest to struktura matematyczna, która jest implementowana w sposób sprzętowy lub (częściej) w sposób programowy. Przetwarza ona sygnały wejściowe za pomocą neuronów ułożonych w warstwy.

Poniżej została przedstawiona graficzna reprezentacja modelu sieci neuronowej, która posiada warstwę meuronów wejściowych, jedną warstwę ukrytą oraz jeden neuron wyjściowy.



# Specyfikacja zewnętrzna

## Konfiguracja platformy programistyczno – testowej

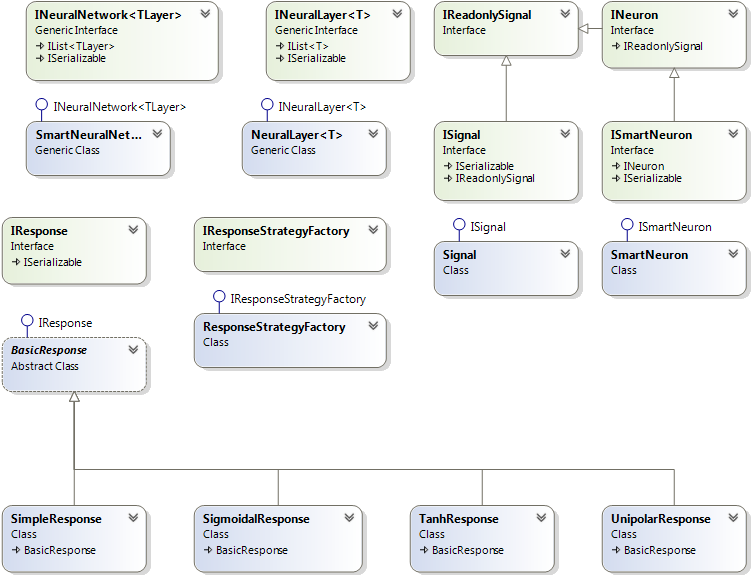
Fizyczną maszyną, na której było przeprowadzane programowanie oraz testowanie systemu, jest 2 rdzeniowa platforma Intel z 4GB pamięci RAM, na której pracuje system Microsoft Windows 7. Na maszynie tej zostało zainstalowane:

* zintegrowane środowisko programistyczne Microsoft Visual Studio 2012
* NUnit
* Microsoft StyleCop
* Microsoft FxCop

# Specyfikacja wewnętrzna

## Architektura biblioteki

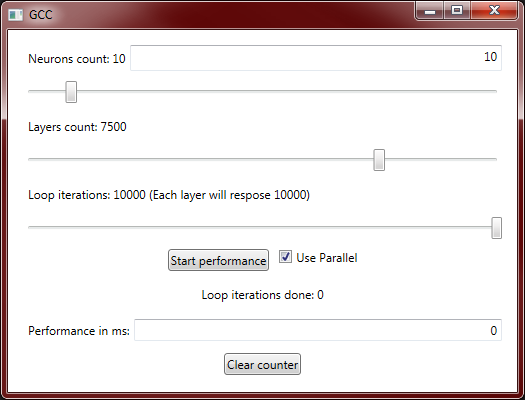
Przyjęto następujący diagram klas zaimplementowanej biblioteki.



# Narzędzia programistyczne

## Testy jednostkowe

Niniejszy projekt zakłada pokrycie kodu źródłowego testami jednostkowymi, aby mieć pewność, że zaimplementowane rozwiązanie jest wolne od błędów - gdyż projekt nie zawiera GUI, które pozwalałoby zaprezentować działanie zaimplementowanej biblioteki. W projektcie istnieje GUI (wykonane w technologii WPF), jednak służo ono jedynie pomocniczo, aby w bardzo dużym przybliżeniu badać czasy wykonania operacji poszczególnych warstw sieci neuronowej, obserwować w jaki sposób wzrasta zużycie pamięci RAM w zależności o wybranych parametrów, a także aby mieć możliwość badania, które metody poszczególnych klas zajmują najwięcej czasu – co pozwala po napisaniu testu dokonać pewnych optymalizacji (czasowej lub pamięciowej).



Projekt powinien utrzymać pokrycie kodu na poziomie minimum 75% - co powinno przyczynić się do skrócenia czasu późniejszej optymalizacji kodu.

## Statyczna analiza kodu źródłowego

Podczas procesu implementacji biblioteki projekt był wspomagany statyczną analizą kodu źródłowego za pomocą narzędzia Microsoft FxCop. Statyczna analiza kodu źródłowego pozwoliła dość wcześnie wykryć błędy, co przełożyło się na skrócenie czasu potrzebnego na implementację. Kolejnym ważnym aspektem statycznej analizy jest możliwość zwiększenia jakości kodu poprzez jego analizę pod kątem wielu prostych lub złożonych reguł. W ten sposób wykryto dość wcześnie niektóre błędy lub zaniżenie jakości kodu. Poniżej przedstawiono listę najczęściej pojawiających się błędów podczas procesu implementacji projektu:

* dyskretne wartości wewnątrz metod („magic number”), które powinny widnieć jako stałe z opisowymi nazwami;
* czy plik z kodem źródłowym nie przekracza wcześniej ustalonego limitu linii kodu – jeżeli plik przekracza tą wartość jest to znak, że klasa znajdująca się w tym pliku została niepoprawnie zaprojektowana. Zgodnie z dobrymi praktykami programistycznymi klasy powinny reprezentować jak najbardziej atomową część implementowanego systemu. Zdecydowanie powinno unikać się pisania klas z kilkoma tysiącami linii („god class”), gdyż taki kod jest trudny w zarządzaniu i utrzymaniu;
* jeśli metody klasy nie operują na wartościach pól klasy należy oznaczy metody te słowem kluczowym „static”;
* jeżeli wszystkie metody w klasie są statyczne to również klasę należy oznaczyć jako statyczną;
* jeżeli metoda zwraca obiekt, należy najpierw sprawdzić czy nie jest on stworzony zanim użyta zostanie jego nie statyczna metoda lub wartość.

## Analiza stylu kodu źródłowego

Podczas procesu implementacji systemu projekt był wspomagany analizą stylu kodu źródłowego za pomocą narzędzia Microsoft StyleCop. Analiza ta pozwoliła na zwiększenie jakości jak i czytelności kodu źródłowego. Podczas początkowej fazy implementacyjnej wspomniane narzędzie zgłaszało naruszenie wielu reguł, co pozwoliło już na samym początku implementacji projektowanego systemu wyeliminowanie niepożądanego stylu kodowania. Z czasem jednak coraz rzadziej narzędzie było wykorzystywane w projekcie ze względu na ciągłe przestrzeganie reguł stylu kodu źródłowego w projektowanej bibliotece.

# Uwagi końcowe

## Wnioski

Wynikiem niniejszego projektu jest programowa biblioteka umożliwiająca stworzenie jednokierunkowej sieci neuronowej.

Projekt został zaimplementowany zgodnie ze ogólnymi standardami programistycznymi (np. wykorzystanie wzorców projektowych, zasad SOLID) oraz z myślą o tym aby możliwa była bezinwazyjna modyfikacja poszczególnych części niewpływająca na resztę biblioteki.

Proponowanym dalszym kierunkiem prac jest rozszerzenie projektu o sieci rekurencyjne oraz o samoorganizujące się mapy, zwane również sieciami Kohonena.