Sprawozdanie nr 2

Temat ćwiczenia: Budowa i działanie sieci jednowarstwowej

1. Cel ćwiczenia.

Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz nauczenie ich rozpoznawania wielkości liter.

1. Wykonanie zadania.
2. Wygenerowanie danych uczących i testujących, zawierających 10 dużych i 10 małych liter dowolnie wybranego alfabetu w postaci dwuwymiarowej tablicy.
3. Implementacja dwóch jednowarstwowych sieci – każda według innego algorytmu.
4. Uczenie sieci przy różnych współczynnikach uczenia.
5. Testowanie sieci.

Ćwiczenie zostało wykonane w języku C++. Wykorzystano model sieci korzystającej z funkcji unipolarnej oraz sigmoidalnej.

Duże litery wykorzystane w ćwiczeniu: ABCDEFGHKL

Małe litery wykorzystane w ćwiczeniu: abcdefghij

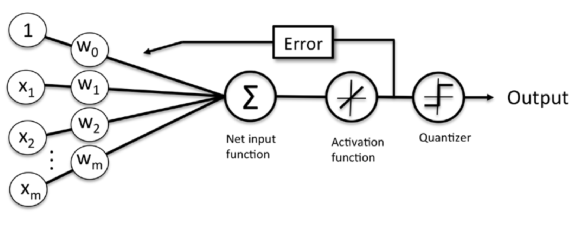
1. Syntetyczny opis budowy oraz wykorzystanego algorytmu uczenia.

Rodzaj sieci neuronowej zależy od sposobu połączenia neuronów tej sieci oraz od kierunku przepływu sygnałów w sieci. Każdy typ sieci ma własne metody doboru wag, czyli uczenia. Istnieje bardzo wiele rodzajów sieci neuronowych jednak najbardziej podstawowe, obrazujące budowę i sposób działania to:

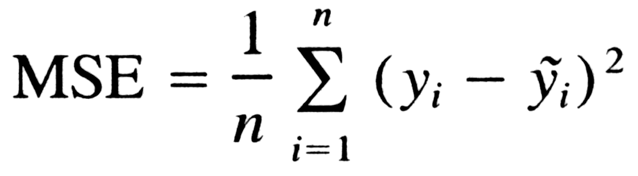
* **sieci jednokierunkowe**
  + **jednowarstwowe**
  + wielowarstwowe
* sieci rekurencyjne
* sieci komórkowe

**Sieci jednokierunkowe**których typowym przykładem jest perceptron jednowarstwowy (rys. perceptronu) składają się z neuronów ułożonych w warstwach o jednym kierunku przepływu sygnałów i połączeniach między warstwowych jedynie między kolejnymi warstwami. Sieć tego typu posiada warstwę wejściową, wyjściową i warstwy ukryte. Z funkcjonalnego punktu widzenia układ taki można traktować jako układ aproksymacji funkcji nieliniowej wielu zmiennych *y = f(u)* .

W wykorzystanej sieci jednowarstwowej neuronem jest ADALINE - adaptacyjny neuron liniowy. Miarą błędów jest MSE i MAPE.



MSE – błąd średniokwadratowy obliczany ze wzoru:

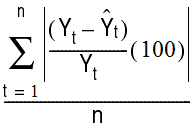


y - wartość oczekiwana,

y tylda - wartość otrzymana,

n - liczba wejść.

MAPE to średni błąd bezwzględny, wyrażany procentowo, obliczany ze wzoru:



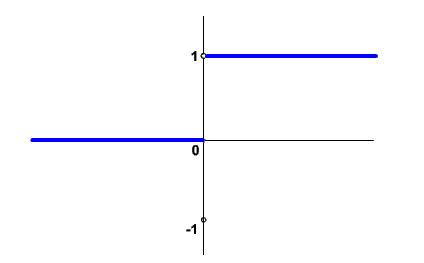
y - wartość oczekiwana,

y tylda - wartość otrzymana,

n - liczba wejść.

* Opis funkcji progowej unipolarnej.

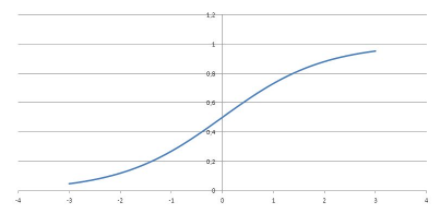




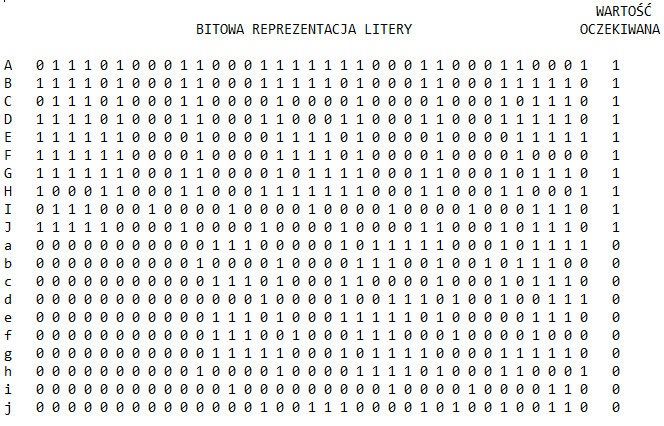
* Opis funkcji sigmoidalnej unipolarnej.

Sam wzór jest bardzo skomplikowany jednak jego pochodna jest bardzo prosta do wyznaczenia. Dzięki temu proces nauki jest bardzo prosty.  
We wzorze pojawia się współczynnik beta. Jego wartość określa 'ściśnięcie' wykresu. Im beta mniejsze tym wykres bardziej ściśnięty, przy beta rosnącym do nieskończoności wykres pokrywa się z wykresem funkcji progowej.



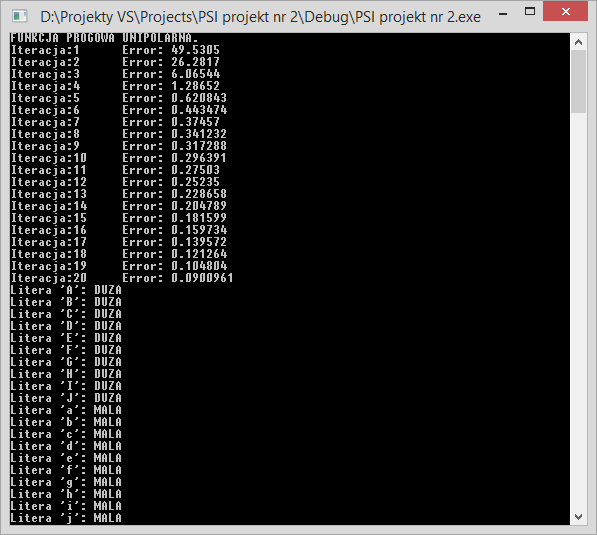


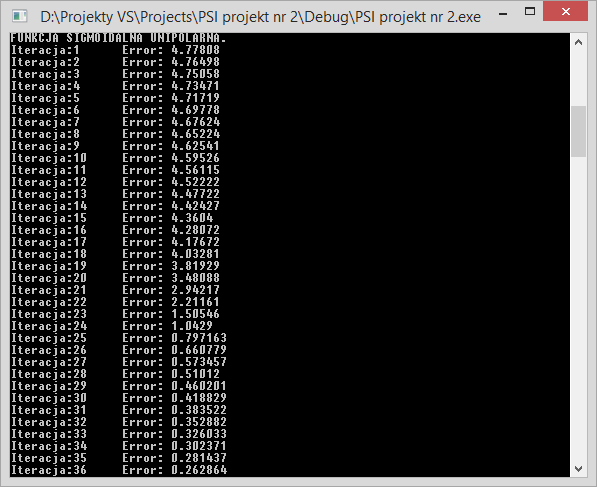
1. Zastosowane dane uczące.

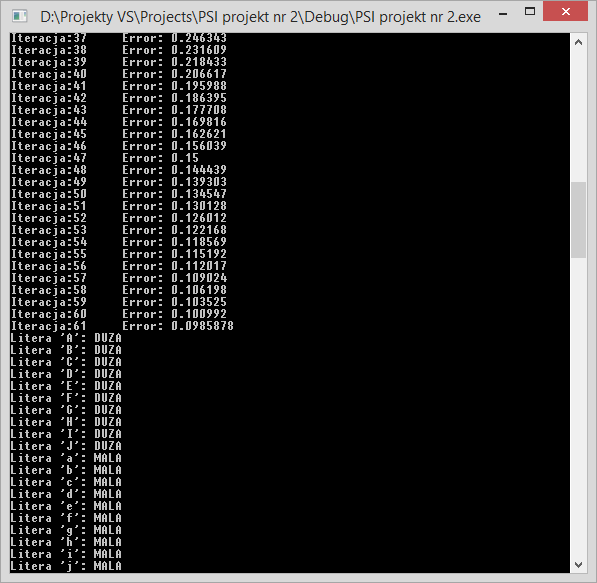


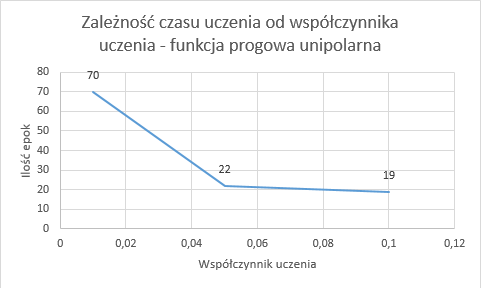
1. Otrzymane wyniki.

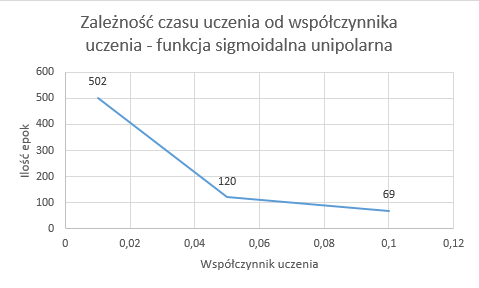
LR = 0.1



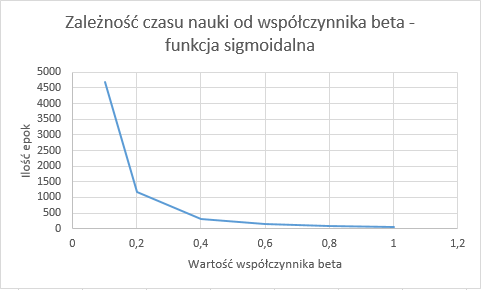








Analizując powyższe wykresy można zauważyć, że wartość współczynnika uczenia ma znaczny wpływ na ilość epok potrzebnych do nauki. Im mniejszy tym więcej czasu potrzeba (analogicznie jak podczas uczenia perceptronu). Sieć Wykorzystująca funkcje aktywacji sigmoidalną potrzebuje więcej czasu na naukę rozpoznawania wielkości liter.



Analizując powyższy wykres można zauważyć, że na efektywność uczenia sieci ma wpływ funkcja aktywacji. Im mniejsza wartość współczynnika β, tym potrzeba więcej epok nauki.

1. Wnioski.

Przeprowadzony proces uczenia jest procesem jednowarstwowym tzn. wyznaczana jest „granica” pomiędzy odpowiedziami poprawnymi a błędnymi.

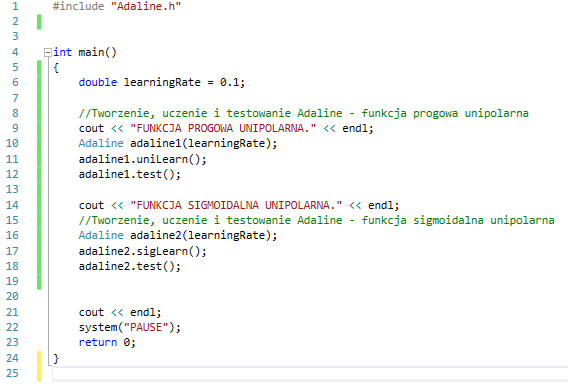
Skuteczność procesu uczenia zależy od współczynnika uczenia. Wraz z jego wzrostem proces uczenia przebiega szybciej.

Uczenie trwa do tej chwili aż dal wszystkich danych wejściowych błąd będzie się poniżej założonego progu uczenia.

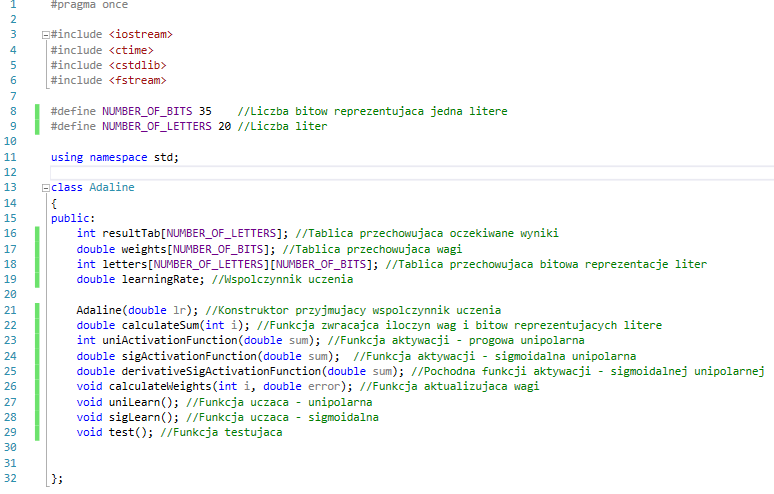
W samym modelu sieci wykorzystującej funkcję sigmoidalną wpływ na efektywność uczenia ma współczynnik β (akceptowalną wartością współczynnika jest ten >=0.5, który powoduje, że sieć zachowuje się stabilnie i jej uczenie jest w miarę akceptowalne.

1. Listing kodu.

Źródło.cpp



Adaline.h



Adaline.cpp

