Jakub Bąk  
Scenariusz nr3

**Temat​​ ćwiczenia:**

Budowa i działanie sieci wielowarstwowej.

**​​Cel ​​ćwiczenia:**

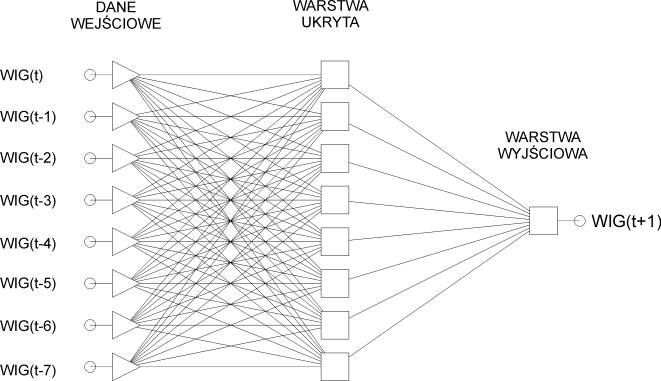
Celem ćwiczenia jest poznanie budowy i działania wielowarstwowych sieci neuronowych

poprzez uczenie z użyciem algorytmu wstecznej propagacji błędu rozpoznawania

konkretnych liter alfabetu.

Sieć neuronowa to określona liczba Perceptronów współpracująca ze sobą .Perceptron ma kilka wejść, do których przypisane są wagi, określające, jak duży wpływ ma wielkość z danego wejścia na wynik. Zestaw danych ze wszystkich wejść jest podstawiany do pewnego wzoru określonego przez programistę – w oryginalnym perceptronie było to dodawanie. Jeśli suma wszystkich wejść (z uwzględnieniem wag) przekroczy określony z góry próg, perceptron da odpowiedź pozytywną (logiczne 1), a jeśli nie – negatywną (logiczne 0). Łatwo zauważyć, że taki układ jest niewiele lepszy od zwykłych bramek logicznych. Dziś stosuje się inny rodzaj sztucznych neuronów, które dają odpowiedź w formie liczby rzeczywistej, czyli nie tylko 0 lub 1, ale dowolną liczbę pomiędzy.

Sieć neuronowa uczy się na przykładach: trzeba jej przedstawić jakąś liczbę już rozwiązanych przykładów. Tok uczenia sieci można zacząć od przypisania równych lub losowych wag każdemu wejściu. Sprawdzamy, czy odpowiedź sieci jest zgodna z pożądanym wynikiem, a następnie zmieniamy wagi tak, żeby wynik zbliżał się do pożądanej odpowiedzi. Mechanizm regulacji wag jest też jednak częścią sieci neuronowej, tak zwaną regułą uczenia, i działa automatycznie. Perceptron jako neuron jest trudny w uczeniu, bo zmiany wielkości wejściowych lub wag mają skutek albo całkowity, albo zerowy – nie można obserwować, jak drobne zmiany przybliżają odpowiedź sieci do pożądanego wyniku.

Jeden sztuczny neuron ma mizerne możliwości rozwiązywania problemów. Wiele neuronów można połączyć w warstwy, gdzie jedna warstwa przekazuje wyniki kolejnej, aż do najwyższej warstwy, dającej odpowiedź.

Najprostsze sieci neuronowe podają informacje tylko w jednym kierunku – każdy neuron reaguje na jeden zestaw danych wejściowych tylko raz. Bardziej złożone sieci są rekurencyjne, co znaczy, że dane wychodzące z którejś z ukrytych warstw są podawane do jednej z poprzednich warstw. Sprzężenie zwrotne powoduje, że sieć po jakimś czasie osiąga stan równowagi i „decyduje się” na konkretną odpowiedź.

**Dla przypadku opisanego w sprawozdaniu :**

**Warstwa Wejściowa : 36**

**Warstwa Ukryta : 15**

**Warstwa Wyjściowa 20**

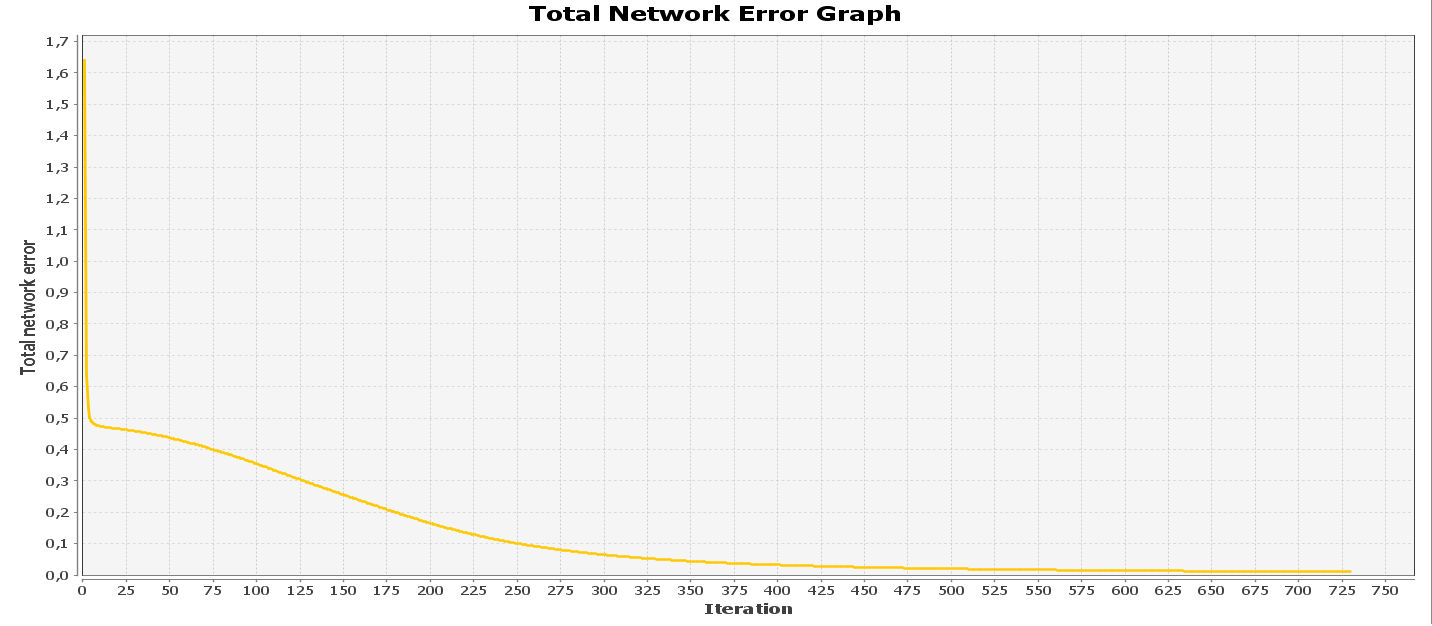
**Zasada uczenia :** **Momentum Background Propagation**

**Typ sieci : MultiLayerPerceptron**

Do testów stworzyłem 6 sieci różniących się wagami oraz współczynnikiem nauczania poniżej prezentuje wyniki uczenia jak i testów dla każdej z nich :

**SIEC # 1**   
*Wagi : 0,5 , Współczynnik uczenia : 0,2*

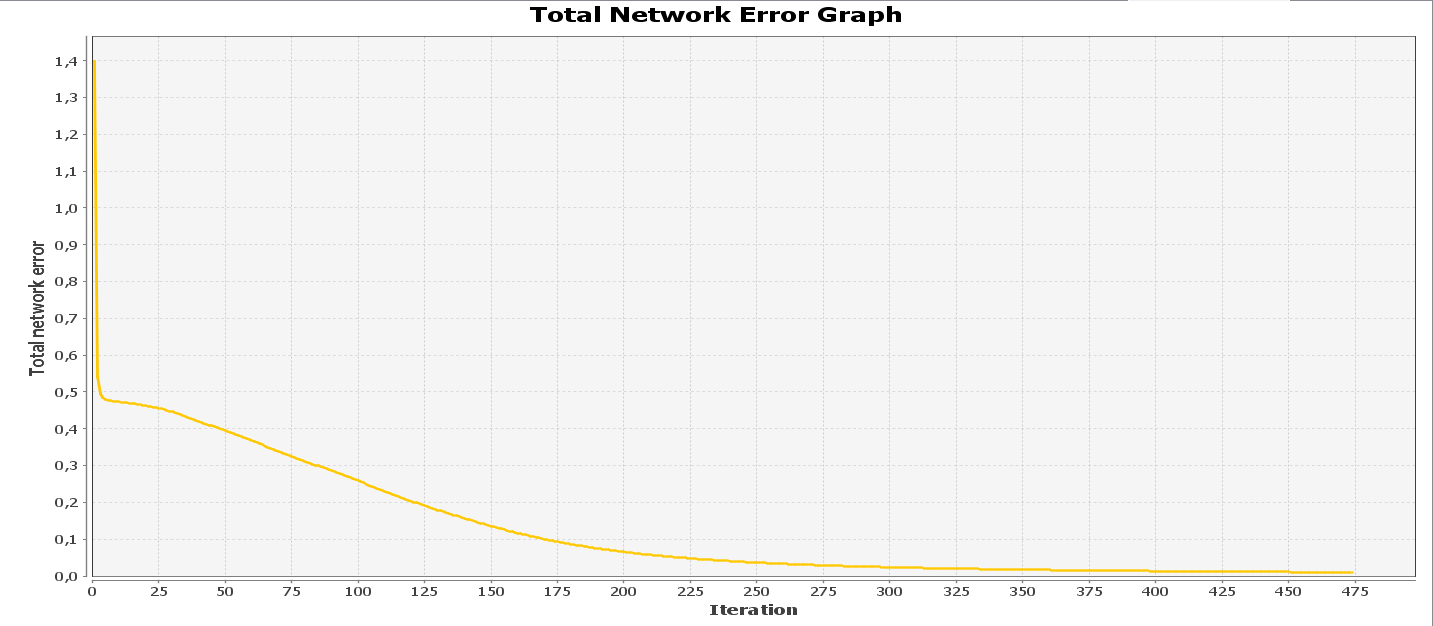
*Wyniki Treningu :*



*Wyniki Testowania :*

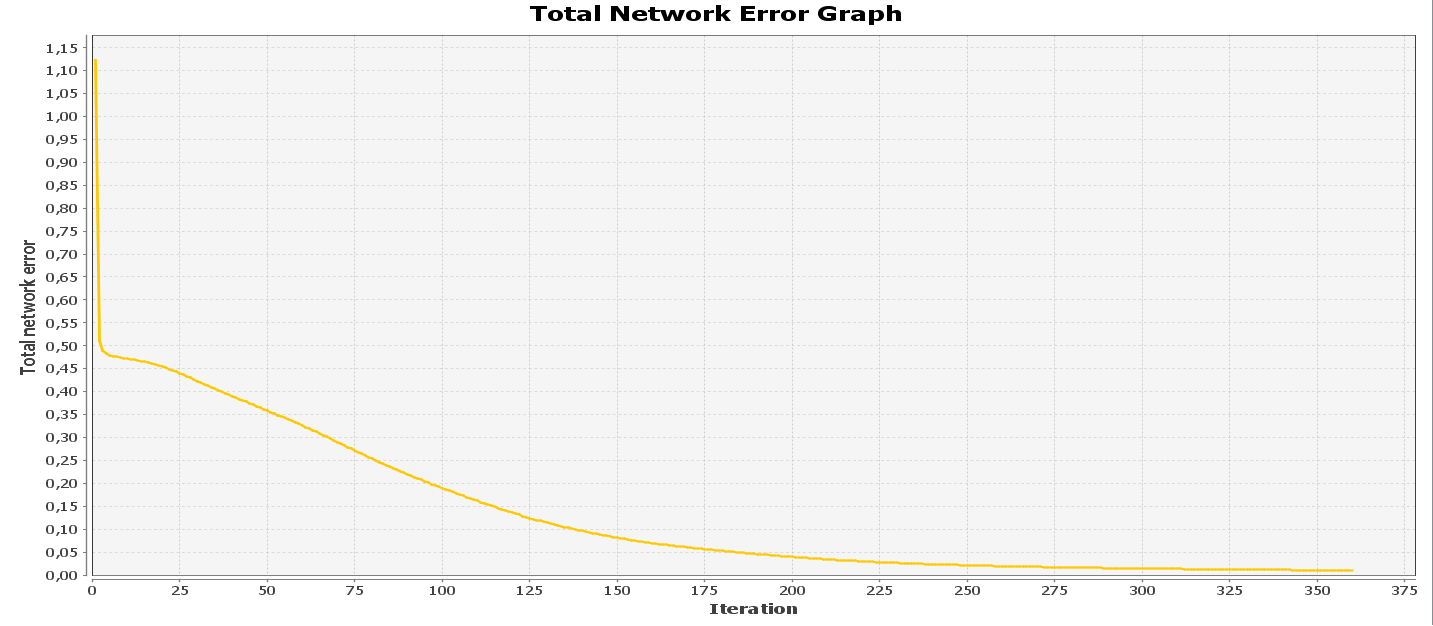
**SIEC # 2**  
*Wagi : 0,5 , Współczynnik uczenia : 0,3*

*Wyniki Treningu :*



*Wyniki Testowania:*

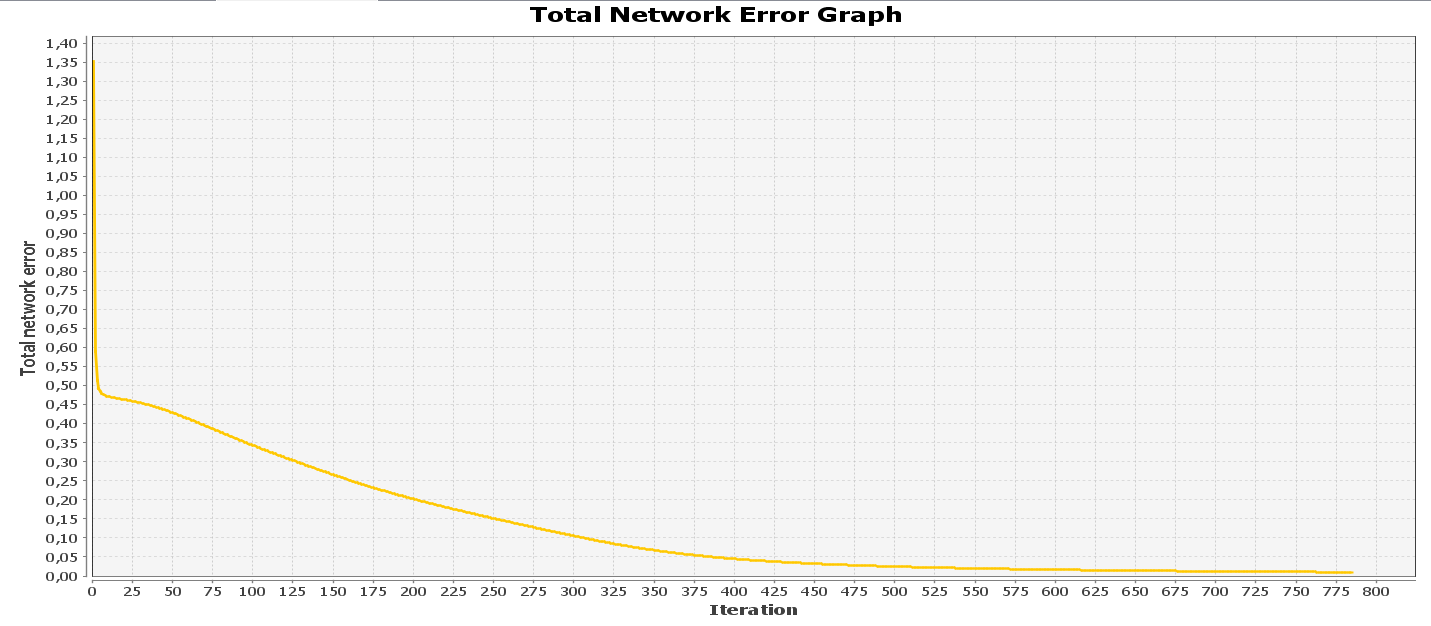
**SIEC # 3**  
*Wagi : 0,5 , Współczynnik uczenia : 0,5*

*Wyniki Treningu :*

*Wyniki Testowania:*

**SIEC # 4**  
*Wagi : 0,7 , Współczynnik uczenia : 0,2*

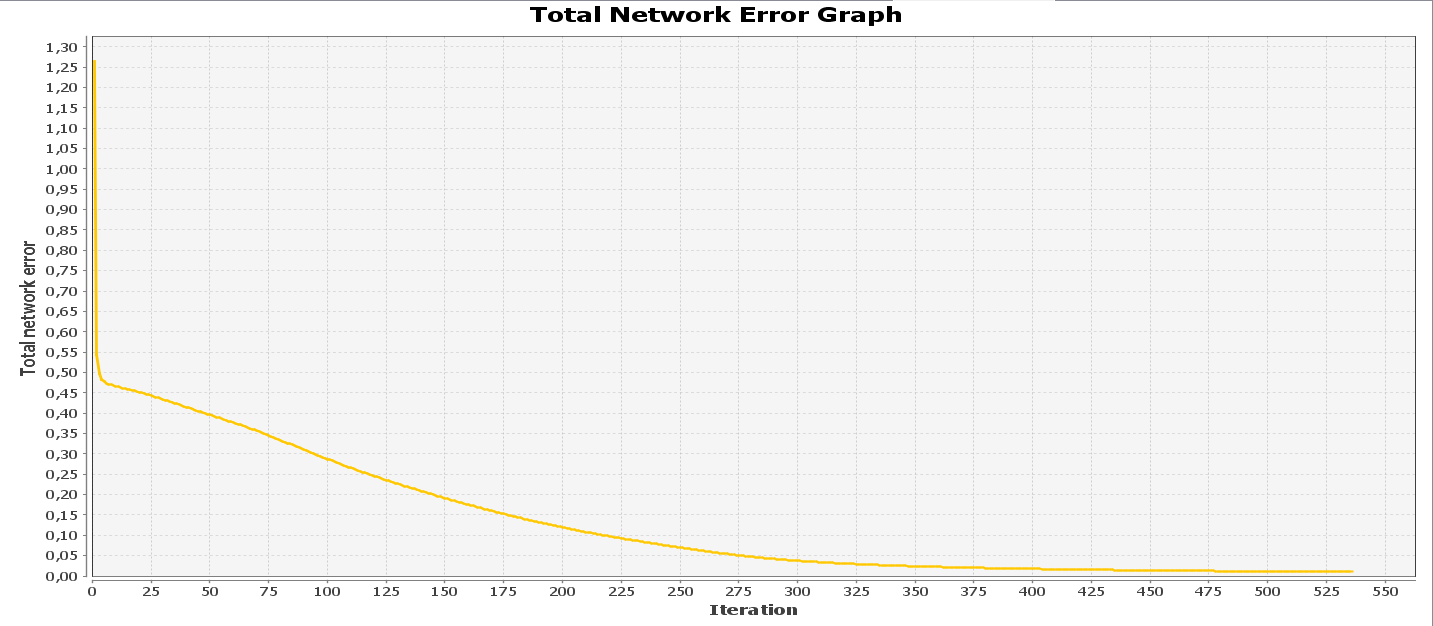
*Wyniki Treningu :*



*Wyniki Testowania:*

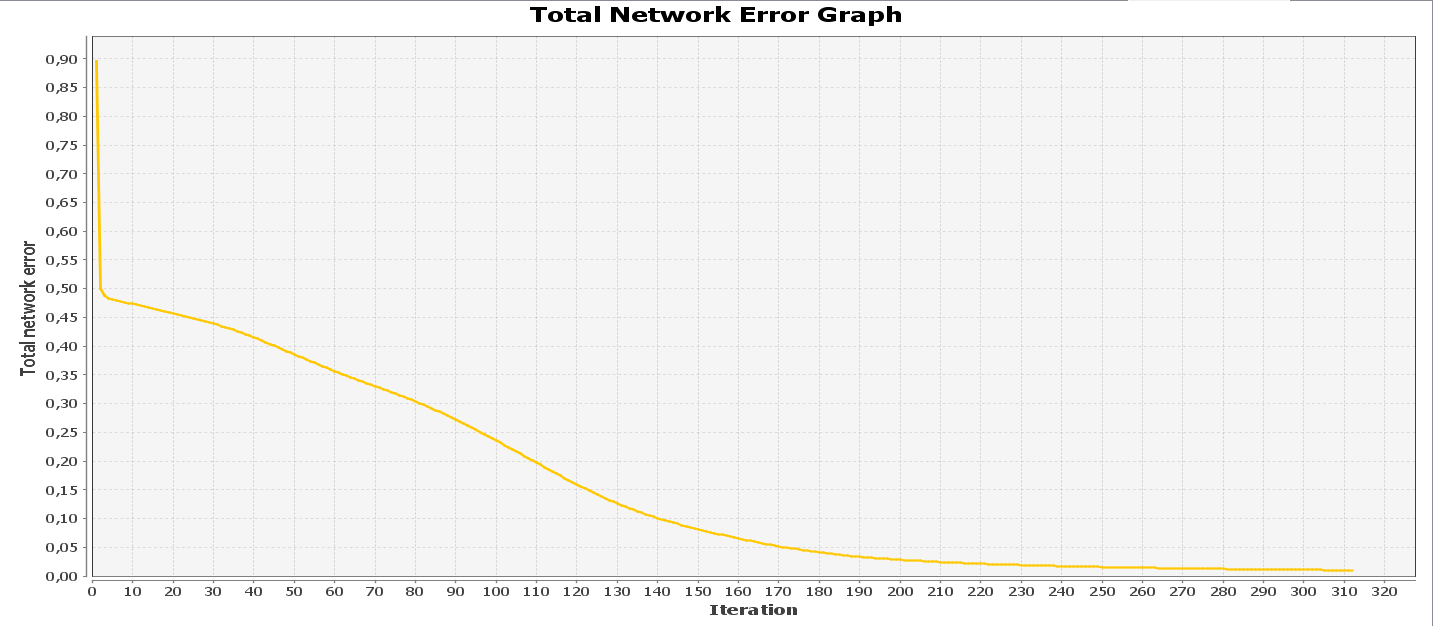
**SIEC # 5**  
*Wagi : 0,7 , Współczynnik uczenia : 0,3*

*Wyniki Treningu :*



*Wyniki Testowania:*

**SIEC # 6**  
*Wagi : 0,7 , Współczynnik uczenia : 0,5*



*Wyniki Treningu :*

*Wyniki Testowania :*

**Wnioski :**

Im większa ilość neuronów w warstwie ukrytej, tym szybciej sieć może się nauczyć danego zagadnienia. Należy pamiętać że więcej neuronów powoduje większe zapotrzebowanie na zasoby systemowe. Dzięki algorytmowi wstecznej propagacji błędu można budować sieci wielowarstwowe, a co za tym idzie rozwiązywać bardziej złożone problemy. Współczynnik bezwładności miał wpływ na wydajność sieci przy zniekształconych literach. Źle dobrany wpływał także na proces uczenia sieci, która nie była w stanie rozpoznać liter zadanych wcześniej w zbiorze uczącym.