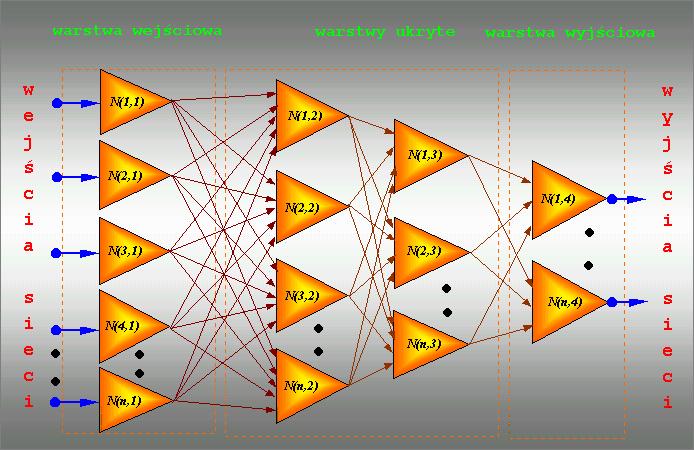
Dariusz Nowak, 278679

**Scenariusz nr 3**

**Budowa i działanie sieci wielowarstwowej**



Rysunek 1 schemat sieci wielowarstwowej

Sieci jednokierunkowe dzielą się na jednowarstwowe, dwuwarstwowe i wielowarstwowe. Sieci jednowarstwowe mogą rozwiązać jedynie wąską klasę problemów. Sieci dwu i wielowarstwowe mogą rozwiązać znacznie szerszą klasę i są pod tym względem równoważne, jednak stosuje się do nich inne algorytmy uczenia (dla wielowarstwowych są one prostsze).(wikipedia)

Sieci wielowarstwowe składają się z trzech bądź większej liczby warstw neuronów. Pierwsza warstwa, określana mianem warstwy wejściowej, zawiera tyle neuronów, ile jest zmiennych wejściowych. Te wejścia są zazwyczaj nazywane neuronami wejściowymi, które dają sygnały wejściowe. Warstwa ta otrzymuje dane z zewnątrz sieci. Każdy neuron z warstwy wejściowej łączy się ze wszystkimi neuronami z obszaru ukrytego. Obszar ukryty składa się niekiedy z więcej niż jednej warstwy; każdy neuron z pierwszej warstwy ukrytej powiązany jest wówczas ze wszystkimi neuronami warstwy drugiej. Jeżeli obszar ukryty składa się więcej niż dwóch warstw, neurony z drugiej warstwy łączą się ze wszystkimi neuronami z warstwy trzeciej. Najkrócej można powiedzieć, że warstwy te stanowią narzędzie, służące do takiego przetwarzania sygnałów wejściowych, by warstwa wyjściowa mogła łatwiej znaleźć potrzebną odpowiedź. Działanie neuronów warstw pośrednich nie jest bezpośrednio widoczne dla użytkownika sieci – w tym sensie można mówić, że są one ukryte. Neurony warstwy ukrytej pełnią rolę pośredników. Pośredniczą między wejściem i wyjściem oraz wypracowują zestawy wstępnie przetworzonych danych wejściowych, z których korzystać będą neurony dalszych warstw przy określeniu końcowego wyniku. Ostatnia warstwa wchodząca w skład obszaru ukrytego połączona jest z warstwą wyjściową. Istnieje także jeden lub więcej neuronów wyjściowych. Dzięki nim otrzymujemy wynik działania sieci. Każdy neuron wyjściowy przyjmuje sygnały wejściowe, przetwarza je i wytwarza sygnał wyjściowy. Sygnały te stanowią rozwiązania stawianych sieci zadań.(http://panda.bg.univ.gda.pl/~prezes/sn/MODELE.htm, brak daty)

Sieci o większej liczbie warstw ukrytych uczą się dłużej i proces uczenia jest słabiej zbieżny.

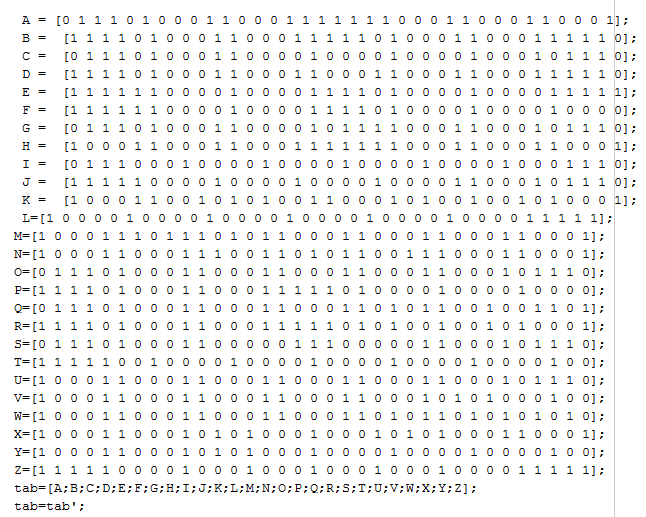
Jeżeli pomimo zwiększenia liczby neuronów sieć uczy się źle, wówczas należy podjąć próbę dodanie dodatkowej drugiej warstwy ukrytej i zmniejszyć jednocześnie liczbę neuronów ukrytych.

Zbyt duża liczba neuronów - wydłużenie czasu uczenia jak również może doprowadzić do tzw. nadmiernego dopasowania. Sieć uczy się wtedy "na pamięć" pewnych szczegółów i zatraca zdolność do uogólnień. Taka sieć uczy się wprost idealnie i wiernie odwzorowuje wartości zbioru uczącego. Jednakże wyniki testowania sieci na danych spoza zbioru uczącego są zazwyczaj bardzo słabe.(wykład prof. dr hab. inż. Jana Kusiaka)

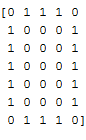
**Algorytm wstecznej propagacji błędów**

Jest to podstawowy algorytm uczenia nadzorowanego wielowarstwowych jednokierunkowych sieci neuronowych. Podaje on przepis na zmianę wag wij dowolnych połączeń elementów przetwarzających rozmieszczonych w sąsiednich warstwach sieci. Oparty jest on na minimalizacji sumy kwadratów błędów uczenia z wykorzystaniem optymalizacyjnej metody największego spadku. Dzięki zastosowaniu specyficznego sposobu propagowania błędów uczenia sieci powstałych na jej wyjściu, tj. przesyłania ich od warstwy wyjściowej do wejściowej, algorytm propagacji wstecznej stał się jednym z najskuteczniejszych algorytmów uczenia sieci.

**Wykonanie zadań ze scenariusza**



Do macierzy wprowadzonych zostało 26 wektorów 35 elementowych(w uproszczeniu, ponieważ z założenia jest to macierz 5x7), z których każdy jest zero-jedynkową reprezentacją danej drukowanej litery alfabetu.



Rysunek 2 reprezentacja litery 'O'

https://i.gyazo.com/f7d4e8327724ae948dc4e20db79ec739.png

Rysunek 3 struktura sieci wielowarstwowej

Struktura sieci wygląda w sposób następujący:

Newff – funkcja przyjmująca

Zakresy – jest to macierz 35x2 w której każdy wiersz wygląda w następujący sposób – [0 1]. Reprezentuje on górną i dolną wartość każdego wejścia.

35- ilość neuronów w warstwie ukrytej(w przypadku wielu warstw wektor wygląda następująco – [x1 x2 x3 xw]) gdzie kolejne indeksy x reprezentują kolejne warstwy.

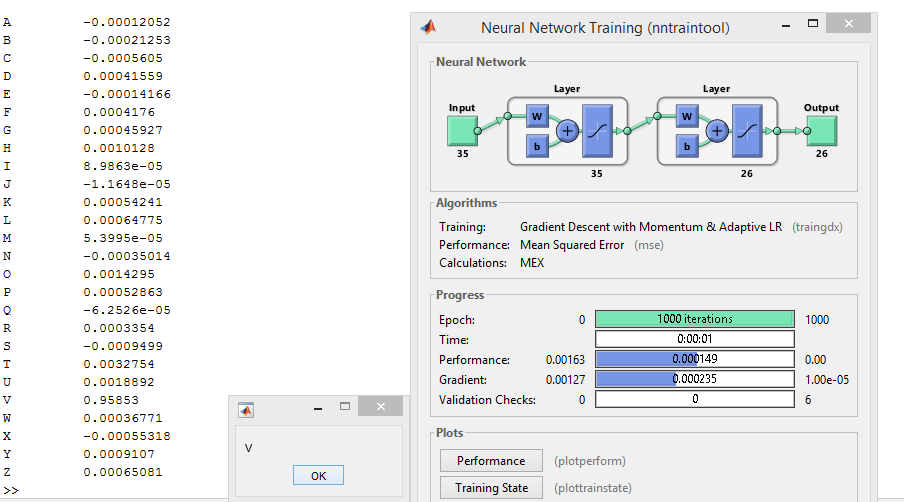
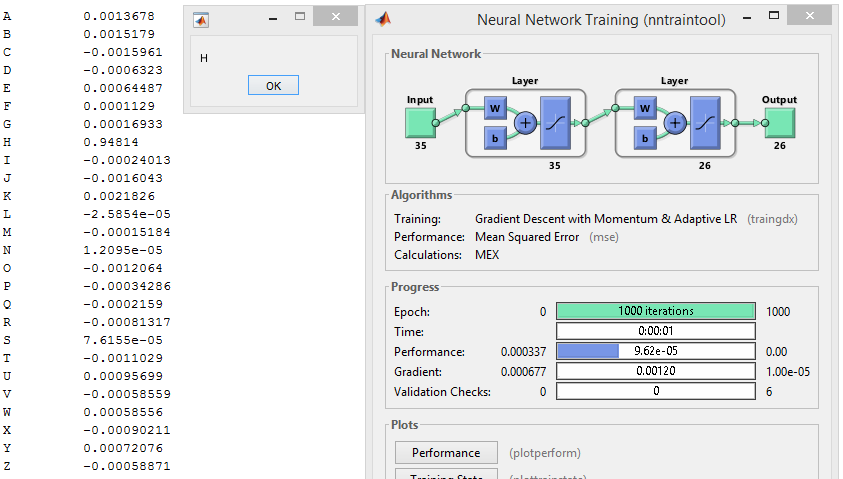
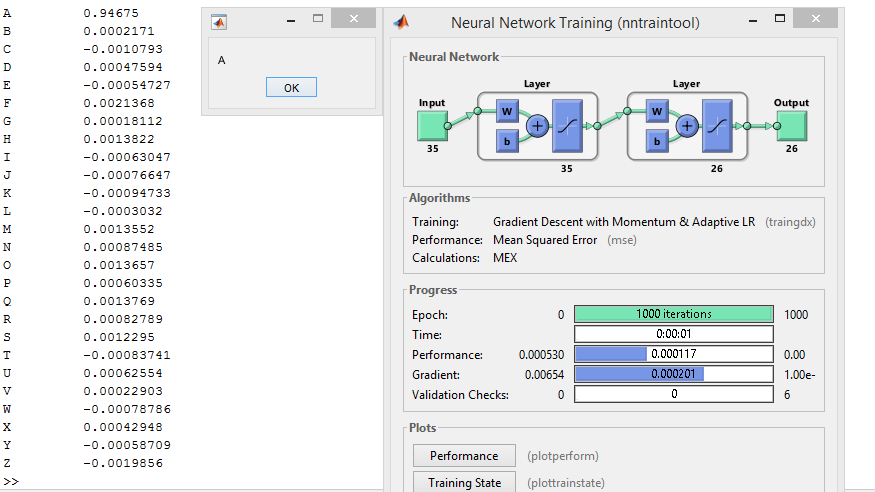
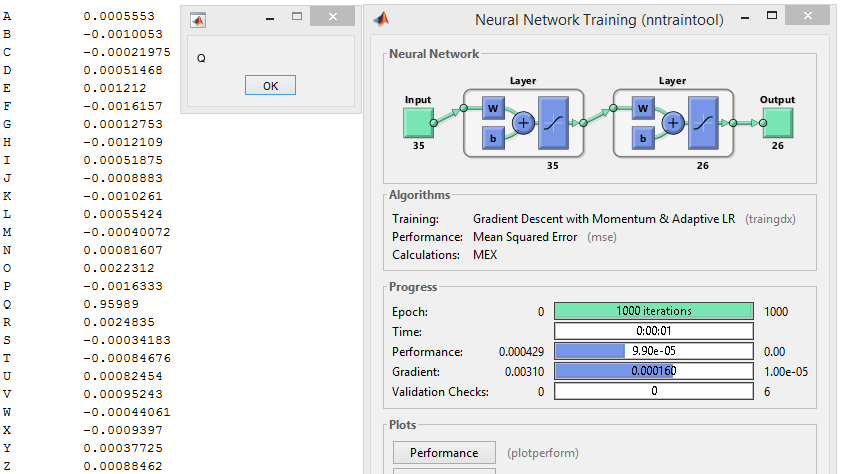
26/xw z przykładu. Jest to wyjście danej sieci, w tym przypadku 26 wyjść sieci.

https://i.gyazo.com/e85386efa0f71bb1024f2cf1d71f5a6b.png

Rysunek 4 trening sieci

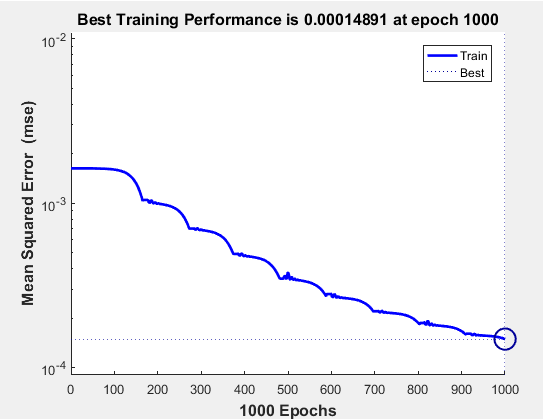
Do treningu sieci wykorzystany został algorytm propagacji wstecznej błędu z momentum i adaptacją współczynnika (Opisany wyżej).

Zwrot programu dla losowych kilku liter:



Spostrzeżenie 1. Program rozpoznaje litery poprawnie, jednak sieć nie zwraca wartości 1.

Wniosek 1. Cieć neuronowa rozpoznaje litery jako kształty z dozą niepewności. Doświadczalnie próbowałem ustalić najlepszą kombinacje wszystkich zmiennych mających bezpośredni wpływ na wynik i za każdym razem(przy przynajmniej 17 neuronach warstwy ukrytej) wynik jest bardzo zbliżony do 1.



Rysunek 5 proces treningu

Spostrzeżenie 2. Algorytm największy progres wykonuje do 200setnej epoki, zauważalny do 900 i stosunkowo mały pomiędzy 900 i 1000 epoką.

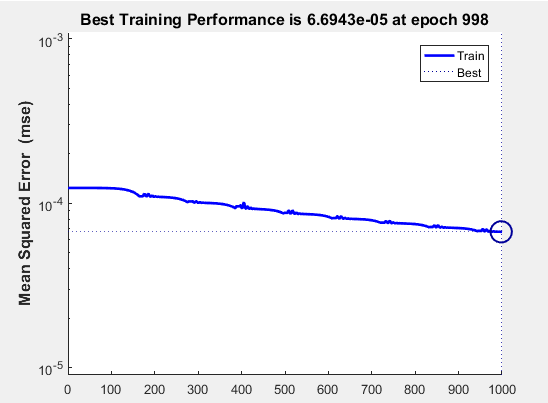
Wniosek 2. Dla 35 neuronów warstwy ukrytej program potrzebuje ok. 1000 epok, aby uzyskać satysfakcjonujące efekty na poziomie >0,9.

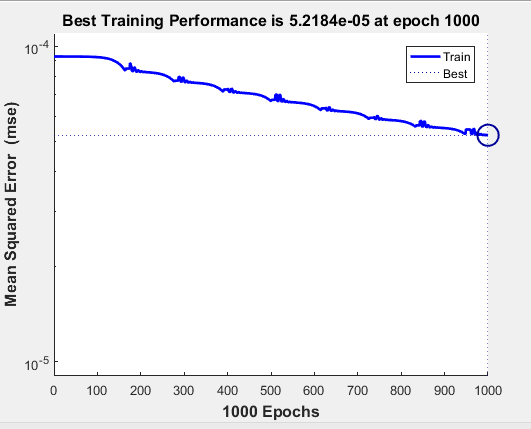
https://i.gyazo.com/02c8d26f7495858c54219782961db113.png

https://i.gyazo.com/3030a7b95be9d6228b8f394ab99ce0ac.png

https://i.gyazo.com/736df776ec8dec5c410c3974b404baac.png

Rysunek 6 wyniki dla różnych współczynników uczenia





Współczynnik uczenia na poziomie [0 10] nie wpływa w żaden sposób na zwrot programu oraz w istotny sposób na tempo treningu. Wartości są praktycznie niezauważalnie różne.