Rafał Lebioda 195639

Piotr Kaźmierczak 195611

Informatyka, studia dzienne, semestr VI, rok akademicki 2016/2017,

Laboratoria: czwartek, 8:15

Inteligentna Analiza Danych

Zadanie 1 – Sieci MLP

# Cel zadania

Zadanie polegało na utworzeniu oraz wykorzystaniu wielowarstwowej sieci dla trzech różnych przypadków. Dla każdej utworzonej sieci należało przeprowadzić proces nauki w celu uzyskania sieci, która będzie w stanie rozwiązać dany problem

# Wstęp

Głównym elementem sieci neuronowych są neurony przetwarzające. Każdy neuron posiada dowolną ilość wejść oraz jedno wyjście. Dodatkowo neuron może być wyposażony w tak zwany bias, czyli dodatkowe wejście na którym występuje stała wartość. Waga dla tego wejścia jest modyfikowana jak wszystkie pozostałe wagi. Neurony są pogrupowane w warstwy, które można podzielić na warstwy: danych, ukryte oraz wyjściowe. Każde dwie warstwy są ze sobą połączone to oznacza, że na wejściach neuronów warstwy znajdują się wyjścia z neuronów warstwy poprzedniej.

# Obliczanie wyjścia neuronu

# Do wejść doprowadzane są sygnały dochodzące z neuronów warstwy poprzedniej. Każdy sygnał mnożony jest przez odpowiadającą mu wartość liczbową zwaną wagą. Waga może być pobudzająca - dodatnia lub opóźniająca – ujemna.  Zsumowane iloczyny sygnałów i wag stanowią argument funkcjiaktywacji neuronu. Wartość funkcji aktywacji jest sygnałem wyjściowym neuronu i propagowana jest do neuronów warstwy następnej. Do naszego programu wykorzystaliśmy dwa rodzaje funkcji aktywacji :

* sigmoidalną funkcję aktywacji
* identycznościową funkcją aktywacji

W wypadku neuronów znajdujących się w warstwie danych ich wyjścia są wejściem całej sieci. W przypadku gdy neurony należą do warstwy wyjściowej, ich wyjścia są wyjściem całej sieci. Wyjście neuronu dane jest wzorem :

# Wsteczna propagacja błędu

Jest to metoda umożliwiająca modyfikację wag w sieci o architekturze warstwowej, we wszystkich jej warstwach. W wypadku wstecznej propagacji błędów proces nauki polega na policzeniu błędu każdego z neuronów rozpoczynając od neuronów wyjściowych. Dla neuronów w warstwie wyjściowej policzenie błędu jest dość proste i sprowadza się do wzoru :

gdzie:

s - suma wartości wejściowych pomnożonych przez wagi

y – wartość otrzymana

d – wartość oczekiwana

Natomiast obliczenie błędu dla neuronów w warstwie ukrytej polega na pobraniu z wszystkich neuronów z warstwy następnej błędów pomnożonych przez odpowiednią wagę, następnie należy zsumować wszystkie te wartości i pomnożyć przez pochodną funkcji aktywacji. Można to zapisać w postaci wzoru :

Z racji tego, że wagi są wykorzystywane w procesie obliczania błędu, należy obliczyć błędy dla wszystkich neuronów w sieci, a dopiero po tym procesie następuje zmiana wag jednocześnie we wszystkich warstwach. Modyfikowanie wag wyraża się wzorem:

gdzie:

µ - współczynnik nauki

- współczynnik momentum

- wartość z poprzedniej iteracji procesu nauki

# Opis implementacji

Implementacja programu została wykonana w języku C#. Program został napisany w taki sposób by można było utworzyć sieć o dowolnej ilości warstw ukrytych z dowolną ilością neuronów w warstwie. Zdecydowaliśmy się na implementacje typowo obiektową co zdecydowanie w efekcie końcowym zaważyło na wydajności programu w porównaniu gdybyśmy przeprowadzili implementacje przy pomocy macierzy. Główna klasą jest klasa Siec która posiada listę warstw oraz informację o poprzedniej oraz następnej warstwie. Każda warstwa posiada swoją listę neuronów. Interfejs użytkownika pozwala na ustalenie współczynników nauki oraz momentum, wykorzystanie biasu, ustawienie błędu jako warunek stopu nauki oraz ustawienie dowolnej ilości warstw i neuronów. Ponadto została dodana możliwość serializacji sieci do pliku .xml oraz wczytywanie sieci z takiego pliku co pozwala na wyuczenie sieci i zapisaniu jej. Umożliwia to w przyszłości wczytanie takiej sieci i kontynuowanie nauki lub wykorzystanie takiej wyuczonej już sieci do rozwiązania danego problemu. Program umożliwia przeprowadzanie nauki na danych pochodzących z różnych plików tekstowych oraz istnieje możliwość wczytania danych testowych w celu zweryfikowania czy sieć jest poprawnie nauczona.