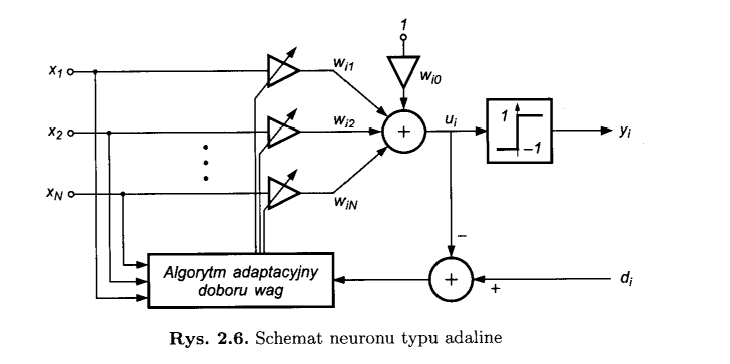
Małgorzata Bień

Sprawozdanie: Zagadnienie nr 2

​Cel​ ​ćwiczenia: poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter

1. Opis budowy wybranych algorytmów:

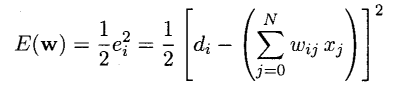
* Algorytm Adaline:



Na początku algorytmu ustawiamy losowe wagi z zakresu (0,1). Wejściem dla naszego neuronu będzie ciąg wartości 0 lub 1 tj. zakodowanej w ten sposób litery. Następnie wyliczamy sumę dla każdego połączenia wejście-waga. Kolejnym krokiem jest obliczenie zmiennej theta, czyli różnicy między wartością oczekiwaną a aktualną z wcześniej obliczonej sumy. Następnie uaktualniamy wagi:

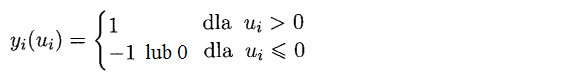


Na podstawie aktualnych wag, wartości na wejściu, zmiennej theta oraz współczynnika uczenia. W algorytmie Adaline proces uczenia polega na minimalizacji funkcji celu którą definiujemy wzorem:

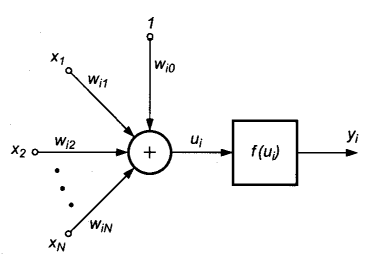


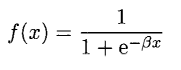
Oznacza to, że uaktualniamy wagi dopóki wartość wyjściowa jest odpowiednio małym przybliżeniem aktualnego wyniku. Wcześniej przyjmujemy liczbę będącą warunkiem zakończenia procesu uczenia.

Podczas testowania wykorzystujemy funkcję aktywacji:

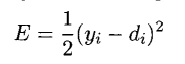


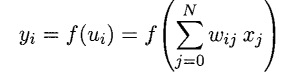
* Algorytm Neuronu Sigmoidalnego



W algorytmie tym w początkowej fazie postępujemy podobnie. Różnicą jest wykorzystywana funkcja aktywacji:

Jest to funkcja sigmoidalna unipolarna. Parametrem β jest dobierany przez użytkownika, najczęściej równy 1. Wartością x będzie suma połączenia na wejściu neuronu.

Uczenie neuronu sigmoidalnego odbywa się zwykle w trybie z nauczycielem, przez minimalizację funkcji celu, która w przypadku jednej pary uczącej (x,d) definiowana jest dla i-tego neuronu w postaci:



Gdzie

We wzorze tym ui jest to suma połączeń wejściowych, f(ui) jest funkcją aktywacji a i

di wartością zadaną na wyjściu.

1. Zestawienie wyników

* Algorytm Adaline

|  |  |  |  |
| --- | --- | --- | --- |
| Adaline | Learning rate | Epoch | MSE (\*10^(-10)) |
|  | 0.001 | 3458 | 9,98 |
|  | 0.0025 | 1049 | 9,99 |
|  | 0.005 | 702 | 9,93 |
|  | 0.0075 | 485 | 9,91 |
|  | 0.01 | 263 | 9,94 |
|  | 0.025 | 125 | 9,58 |
|  | 0.05 | 59 | 9,47 |
|  | 0.1 | 30 | 9,86 |
|  | 0.5 | 7 | 6,39 |

Przykładowe wyniki uzyskane podczas uczenia:

|  |  |  |
| --- | --- | --- |
| 0.001 | Epoch | MSE (\*10^(-10)) |
|  | 3903 | 9,98 |
|  | 3017 | 9,98 |
|  | 3761 | 9,97 |
|  | 3667 | 9,99 |
|  | 2943 | 9,99 |

|  |  |  |
| --- | --- | --- |
| 0.01 | Epoch | MSE (\*10^(-10)) |
|  | 286 | 9,92 |
|  | 248 | 9,94 |
|  | 257 | 9,91 |
|  | 285 | 9,99 |
|  | 243 | 9,95 |

|  |  |  |
| --- | --- | --- |
| 0.1 | Epoch | MSE (\*10^(-10)) |
|  | 28 | 9,56 |
|  | 25 | 9,87 |
|  | 39 | 9,98 |
|  | 36 | 9,96 |
|  | 24 | 9,95 |

|  |  |  |
| --- | --- | --- |
| 0.05 | Epoch | MSE (\*10^(-10)) |
|  | 44 | 9,56 |
|  | 55 | 9,05 |
|  | 50 | 9,37 |
|  | 73 | 9,5 |
|  | 76 | 9,88 |

* Algorytm neuronu sigmoidalnego

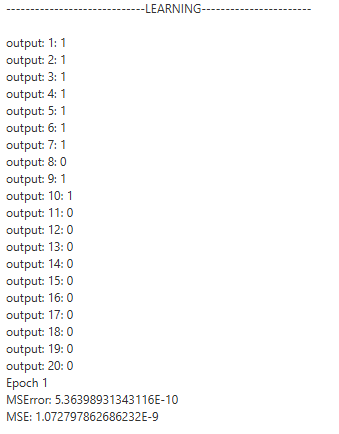
|  |  |  |  |
| --- | --- | --- | --- |
| Sigmoidal | Learning rate | Epoch | MSE (\*10^(-6)) |
|  | 0.001 | 2328 | 9,98 |
|  | 0.0025 | 923 | 9,99 |
|  | 0.005 | 471 | 9,99 |
|  | 0.0075 | 456 | 9,97 |
|  | 0.01 | 390 | 9,96 |
|  | 0.025 | 136 | 9,99 |
|  | 0.05 | 51 | 9,85 |
|  | 0.1 | 29 | 9,87 |
|  | 0.5 | 6 | 9,97 |

Przykładowe wyniki uzyskane podczas uczenia:

Przy minimalnym błędzie: 0,00001

|  |  |  |
| --- | --- | --- |
| 0.001 | Epoch | MSE (\*10^(-6)) |
|  | 1842 | 9,99 |
|  | 1769 | 9,99 |
|  | 3148 | 9,98 |
|  | 2553 | 9,99 |

|  |  |  |
| --- | --- | --- |
| 0.05 | Epoch | MSE (\*10^(-6)) |
|  | 67 | 9,86 |
|  | 62 | 9,93 |
|  | 42 | 9,99 |
|  | 41 | 9,68 |

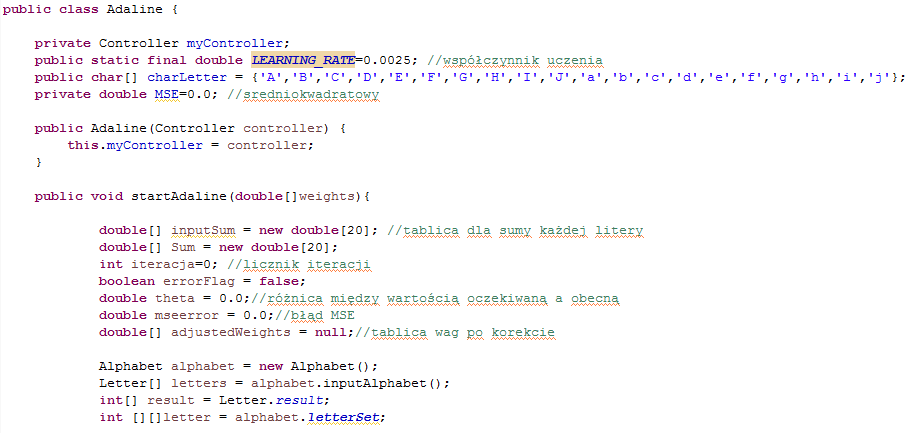


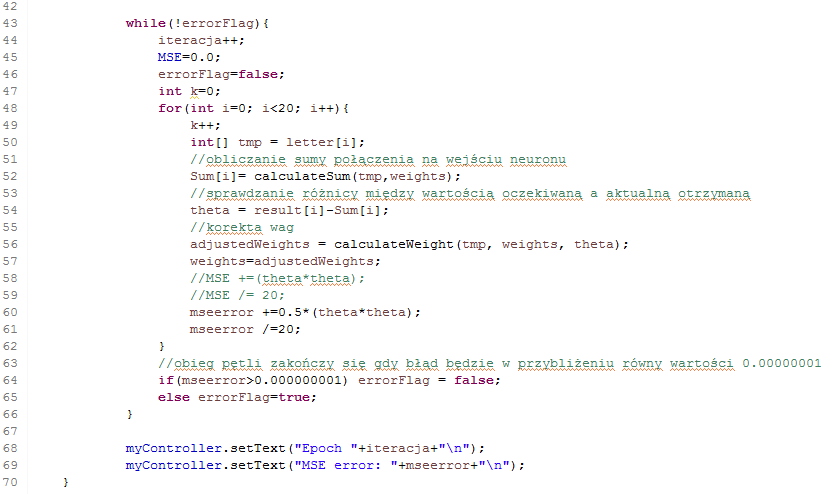
Przykładowy wynik działania algorytmu Adaline przy współczynniku 0,5.

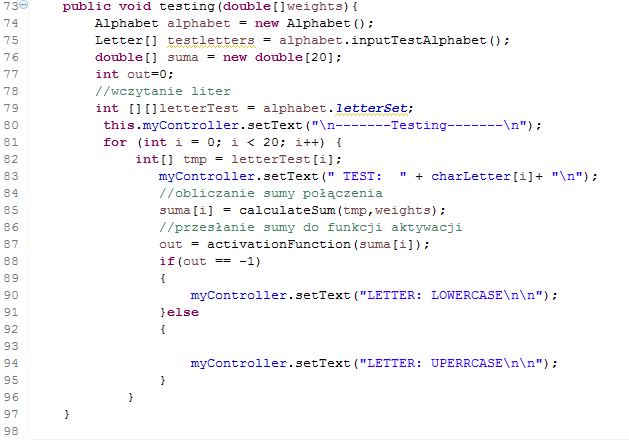
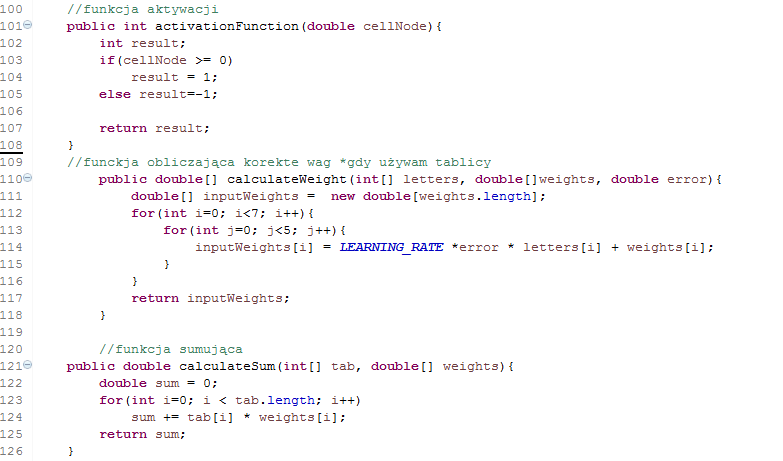
Wynik testowania przy użyciu algorytmu neuronu sigmoidalnego:

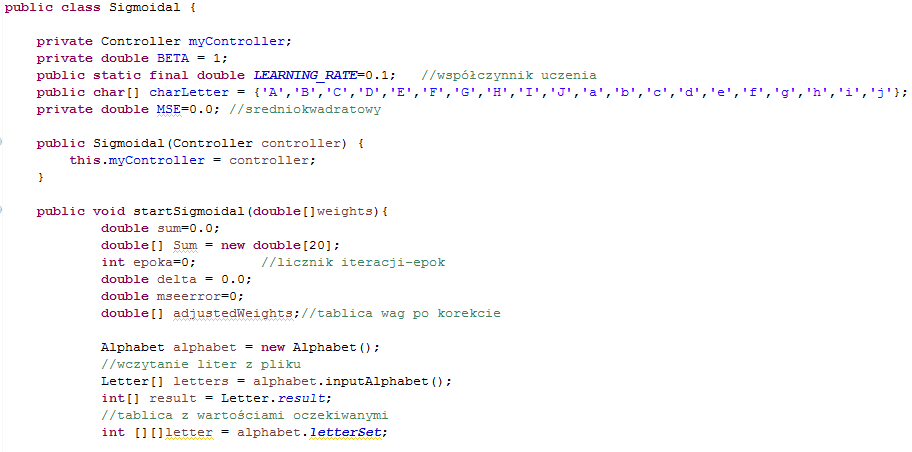


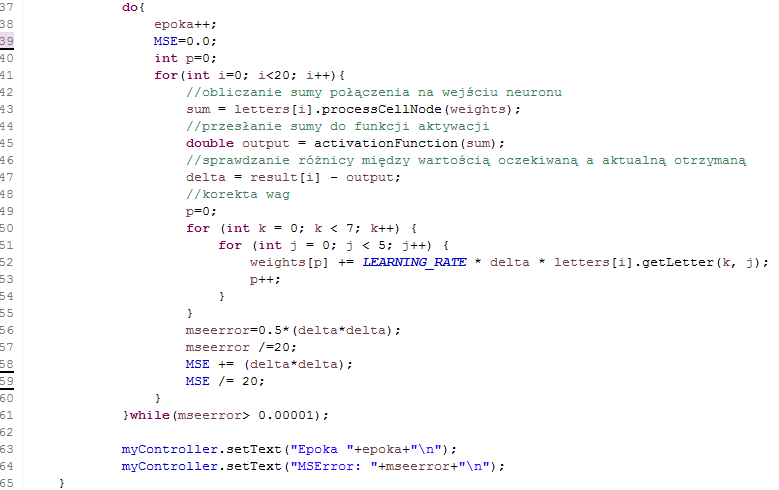
1. Listing kodu programu:

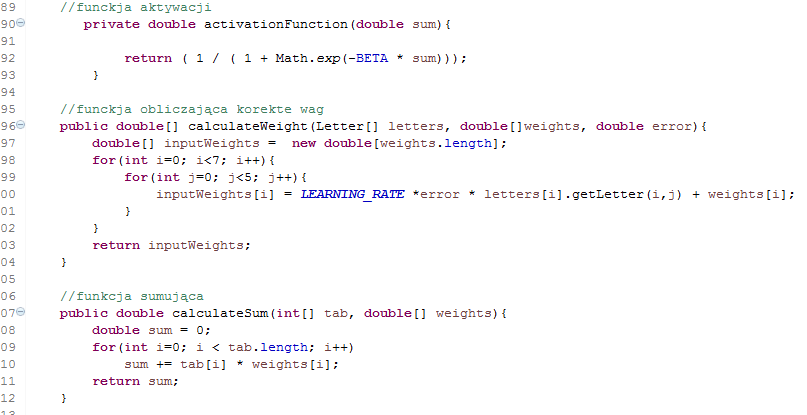




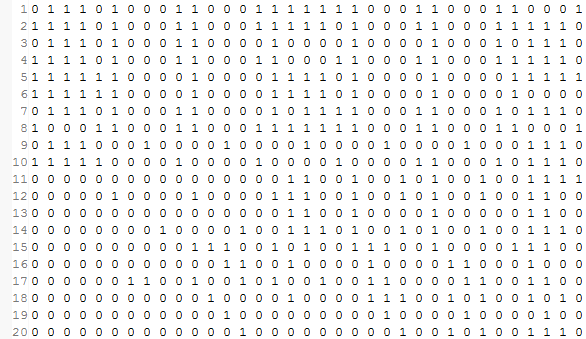








Dane uczące:



1. Analiza:

Uczenie dla obu algorytmów przeprowadziłam z różnymi współczynnikami uczenia. Za najniższą wartość przyjęłam 0.001. Z tabeli wyników można łatwo zauważyć dużą różnicę między liczbą epok dla neuronu sigmoidalnego i adaline przy tym współczynniku. Pierwszy potrzebował aż 299950 iteracji, drugi 3458. Widać również duży spadek liczby epok dla najwyższej ustawionej wartości współczynnika czyli 0.5. Dla adaline jest to już tylko 7 a dla neuronu sigmoidalnego 882 iteracji. Podczas testowania neuron napisany według algorytmu adaline bardzo często się mylił podczas gdy drugi zawsze dawał 100% poprawnych odpowiedzi.

1. Wnioski:

Na podstawie otrzymanych wyników można spostrzec że istnieje duża różnica pomiędzy liczbą epok dla tych wybranych algorytmów. Ilość iteracji neuronu Adaline znajduje się w przedziale od ponad 3 tys. do 7. Dla neuronu sigmoidalnego są to wartości z granicy kilku, kilkunastu i kilkudziesięciu tysięcy, najmniejszą jest 882 osiągana przy najwyższym współczynniku uczenia. Można zatem stwierdzić, że algorytm adaline uczy się szybciej. Kolejnym wnioskiem jest wpływ współczynnika uczenia na liczbę epok potrzebną do uczenia się neuronu. Im większa wartość tym sieć uczy się szybciej. Biorąc pod uwagę poprawność działania algorytmów można wnioskować że neuron typu sigmoidalnego jest bardziej poprawny. Podczas testowania zawsze dawał 100% poprawnych odpowiedzi a algorytm adaline bardzo często się mylił.