Sprawozdanie
Podstawy sztucznej inteligencji

Tymoteusz Przyłucki
gr 3

Wstęp teoretyczny:

Model neuronu Adaline (Adaptive Linear Neuron) został podany przed Widrowa w 1960 roku. Jego istota polega na sposobie uczenia się – korekta wag odbywa się w oparciu o porównywanie oczekiwanej odpowiedzi z potencjałem membranowym neuronu co sprowadza się do zmiany we wzorze opisującym popełniany błąd (po tej zmianie δ nie reprezentuje wprost błędu):

$$\delta_\mu(t) = d_\mu - u_\mu(t)$$

Dzięki takiemu zdefiniowaniu wartości δ, pomimo nieliniowego charakteru neuronu,możliwe jest zastosowanie gradientowego algorytmu uczenia bazującemu na minimalizacji funkcji średniokwadratowej.

Propozycja Widrowa wprowadziła neuron o nieliniowej (bipolarnej) funkcji przejścia.

$$y = sgn\left(\sum_{i=0}^{n} w_i x_i\right) = \begin{cases} 1 \;\; gdy \;\; \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \;\; gdy \;\; \sum_{i=0}^{n} w_i x_i \le 0 \end{cases}$$

Aktualnie w literaturze można spotkać również definicje neuronów Adaline w których jako funkcja aktywacji jest wykorzystywana funkcja liniowa, wówczas odpowiedź neuronu może być opisana wzorem:

$$y = k\sum_{i=0}^{n} w_i x_i$$

Wprowadzenie współczynnika proporcjonalności k jest jedynie zmianą ilościową, a nie jakościową. W przypadku włączenia elementu Adaline w strukturę sieci wielowarstwowej jego rolę mogą przejąć wagi połączeń wychodzących z elementu.

W praktyce z neuronów Adaline buduje się najczęściej sieci jednowarstwowe zwane Madaline (Many ADALINE). W sieciach tych, każdy neuron uczony jest zgodnie z regułą adaline.

Zadanie zostało wykonane w programie NeurophStudio. Dane zostały przygotowanie pod konkretną sieć neuronową – Adaline. Uczenie nie powiodło się na jednowarstwowej sieci neuronowej. W celu sprawdzenia poprawności danych wejściowych przeprowadziłem uczenie na sieci wielowarstwowej, próba okazała się sukcesem. Założyłem, że dla narzędzia w którym próbowałem zrealizować zadanie, problem rozpoznawania 20 znaków jest zbyt złożony.

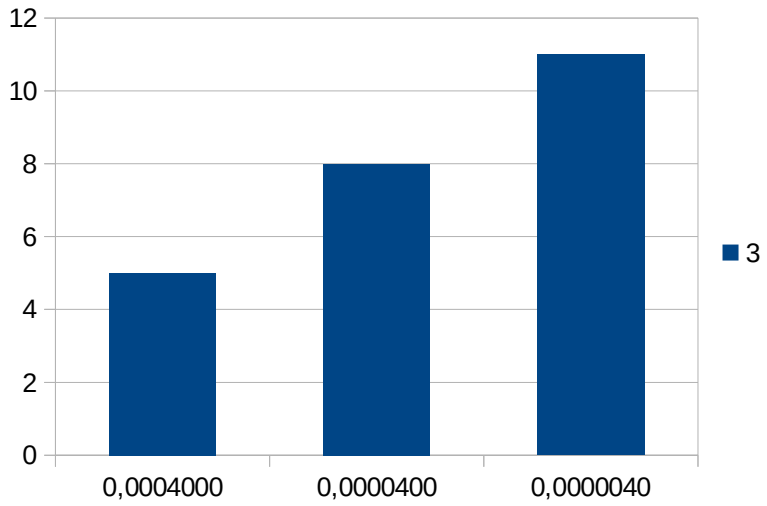
Poprawne zadanie zostało wykonane za pomocą języka C++, wykorzystując wcześniej zdobytą wiedzę na temat funkcjonowania sieci Adaline. Implementacja sieci jednowarstwowej polegała na rozszerzeniu kodu z Laboratorium 1.

Wyniki:

```
numer iteracji: 1921   blad sredniokwadratowy: 2.43133
numer iteracji: 1922   blad sredniokwadratowy: 2.43043
numer iteracji: 1923   blad sredniokwadratowy: 2.42953
numer iteracji: 1924   blad sredniokwadratowy: 2.42864
numer iteracji: 1925   blad sredniokwadratowy: 2.42774
numer iteracji: 1926   blad sredniokwadratowy: 2.42685
numer iteracji: 1927   blad sredniokwadratowy: 2.42595
numer iteracji: 1928   blad sredniokwadratowy: 2.42506
numer iteracji: 1929   blad sredniokwadratowy: 2.42417
numer iteracji: 1930   blad sredniokwadratowy: 2.42327
numer iteracji: 1931   blad sredniokwadratowy: 2.42238
numer iteracji: 1932   blad sredniokwadratowy: 2.42149
numer iteracji: 1933   blad sredniokwadratowy: 2.4206
numer iteracji: 1934   blad sredniokwadratowy: 2.41971
numer iteracji: 1935   blad sredniokwadratowy: 2.41882
numer iteracji: 1936   blad sredniokwadratowy: 2.41793
numer iteracji: 1937   blad sredniokwadratowy: 2.41705
numer iteracji: 1938   blad sredniokwadratowy: 2.41616
numer iteracji: 1939   blad sredniokwadratowy: 2.41528
numer iteracji: 1940   blad sredniokwadratowy: 2.41439
numer iteracji: 1941   blad sredniokwadratowy: 2.41351
numer iteracji: 1942   blad sredniokwadratowy: 2.41262
numer iteracji: 1943   blad sredniokwadratowy: 2.41174
numer iteracji: 1944   blad sredniokwadratowy: 2.41086
numer iteracji: 1945   blad sredniokwadratowy: 2.40998
numer iteracji: 1946   blad sredniokwadratowy: 2.4091
numer iteracji: 1947   blad sredniokwadratowy: 2.40822
numer iteracji: 1948   blad sredniokwadratowy: 2.40734
numer iteracji: 1949   blad sredniokwadratowy: 2.40646
numer iteracji: 1950   blad sredniokwadratowy: 2.40558
numer iteracji: 1951   blad sredniokwadratowy: 2.4047
numer iteracji: 1952   blad sredniokwadratowy: 2.40383
numer iteracji: 1953   blad sredniokwadratowy: 2.40295
numer iteracji: 1954   blad sredniokwadratowy: 2.40208
numer iteracji: 1955   blad sredniokwadratowy: 2.4012
numer iteracji: 1956   blad sredniokwadratowy: 2.40033
numer iteracji: 1957   blad sredniokwadratowy: 2.39946
numer iteracji: 1958   blad sredniokwadratowy: 2.39858
numer iteracji: 1959   blad sredniokwadratowy: 2.39771
numer iteracji: 1960   blad sredniokwadratowy: 2.39684
numer iteracji: 1961   blad sredniokwadratowy: 2.39597
numer iteracji: 1962   blad sredniokwadratowy: 2.3951
numer iteracji: 1963   blad sredniokwadratowy: 2.39424
numer iteracji: 1964   blad sredniokwadratowy: 2.39337
```

Wnioski:

Przeprowadziłem testy dla różnych funkcji aktywacji. Najbardziej trafną była funkcja a*x+1/x, gdzie a współczynnik (od 0.004 do 0,000004) a x dana wejściowa. Błąd średniokwadratowy dążył do 0 w każdym przypadku, z różną prędkością. Im mniejsza była funkcja aktywacji tym wolniej sieć się uczyła. Podczas zwiększania współczynnika a, możemy zauważyć znaczny spadek błędu średniokwadratowego, czyli wzrost szybkości uczenia się sieci neuronowej. Dla funkcji (1/x) , zauważono znaczny wzrost błędu kwadratowego. Najszybszy spadek zauważono dla funkcji (a*x*1/x).



Na wykresie przedstawiona jest zależność pomiędzy maksymalnym błędem średniokwadratowym (kolumna) a współczynnikiem a (wiersz). Im większy współczynnik a, tym mniejszy startowy błąd średniokwadratowy (dla iteracji = 0)