

PROGRAMOWANIE SIECIOWE

LABORATORIUM 1.

Zaprojektowanie sieci neuronowej do rozwiązywania prostego zadania klasyfikacji

Autor:

Rafał Wasicki
232622

Prowadzący:

Mgr inż. Aneta Górniak

20 marca 2019



Politechnika Wrocławska

1 Opis wykonanego ćwiczenia

1. Na zajęciach wykonano jednowarstwową sieć neuronową typu feed-forward. Za jej pomocą zaprojektowano i przeprowadzono proste zadanie klasyfikacji

2. Opis i fragmenty kodu źródłowego:

(a) Implementacja macierzy wag:

```
n = size(baza_ucz_we, 1); %liczba wejść
k = size(baza_ucz_wy, 1); %liczba neuronów
a = -0.1;                  %zakres górny przedziału
b = 0.1;                   %zakres dolny przedziału
W = (b-a)*rand(n,k)+a;
```

(b) Implementacja algorytmu sieci neuronowej

```
for ep=1 : epoki
    L =randperm(3,1);      %algorytm losujący
    x = baza_ucz_we(:,L);  %wektor pobudzenia
    u = W'*x;              %wektor wejścia
    y = 1./(1+exp(-beta*u)); %wektor wyjścia
    %sigmoidalna funkcja aktywacji w zadaniu pierwszym
    y = (1-exp(-beta*u))./(1+exp(-beta*u));
    %sigmoidalna bipolarna funkcja aktywacji w zadaniu drugim
    ty = baza_ucz_wy(:, L);
    d = ty - y;
    dW = eta*x*d'; %macierz poprawek
    W = W + dW;
end
```

2 Wnioski

1. Poprawność działających skryptów:

- Zadanie pierwsze:

W tym zadaniu wystąpił problem z klasyfikacją. Kategorie samochód i motocykl są do siebie podobne, więc dla podstawowego zestawu atrybutów rozwiązanie nie było zadowalające. Powiększenie zestawu atrybutów o odpowiedź na pytanie czy klasyfikowany obiekt posiada poduszkę powietrzną sprawiło, że identyfikacja dawała prawidłowe rezultaty

- Zadanie drugie:

W tym zadaniu dane sprawdzające były przypadkami szczególnymi swoich kategorii. Sieci neuronowe nie są w stanie poprawnie opracowywać takich przypadków szczególnych. Zwiększenie ilości atrybutów na takie, które charakteryzowałyby takie przypadki szczególne i odróżniały je od innych kategorii, a także zawarcie przypadków szczególnych w danych testowych mogłoby rozwiązać przedstawiony problem.

2. Rola neuronu i pozostałych elementów sieci:

- **Neuron** - przetwarza proste wartości sygnałów wejściowych na jeden sygnał wyjściowy.
- **Funkcja aktywacji** - funkcja, według której obliczana jest wartość wyjścia neuronu.
- **Epoki** - liczba przejść pętli uczącej sieci neuronowej.
- η - współczynnik uczenia się sieci neuronowej.

3. Wpływ parametrów na uzyskane rozwiązania:

- **Epoki** - większa ilość epok wpływa pozytywnie na jakość nauczania sieci neuronowej.
- η - Większy współczynnik wpływa na szybsze uczenie się sieci, ale może też uwzględniać błędy, nauczyć się niepoprawnego schematu.
- **Atrybuty** - Dobór atrybutów na kluczowe znaczenie przy projektowaniu klasyfikującej sieci neuronowej. Zbyt mała ilość atrybutów może powodować niejednoznaczność odpowiedzi, natomiast

zbyt duża ich ilość wprowadza niepotrzebne informacje do sieci. Wyniki klasyfikacji są zawsze niejednoznaczne, jednak dąży się do osiągnięcia wysokiego prawdopodobieństwa, że sieć działa poprawnie.

- **Rozmiar wektorów** - musi być dopasowany do wielkości badanego problemu.

4. Poprawę efektywności sieci neuronowej można uzyskać poprzez:

- dopasowanie ilość atrybutów sieci.
- Zwiększenie ilość danych uczących lub epok.
- Dobranie funkcję aktywacji.
- Dodanie kolejne warstwy sieci.
- Zwiększenie ilość neuronów w warstwie.