

Budowa i działanie sieci wielowarstwowej typu feedforward

1. Cel projektu

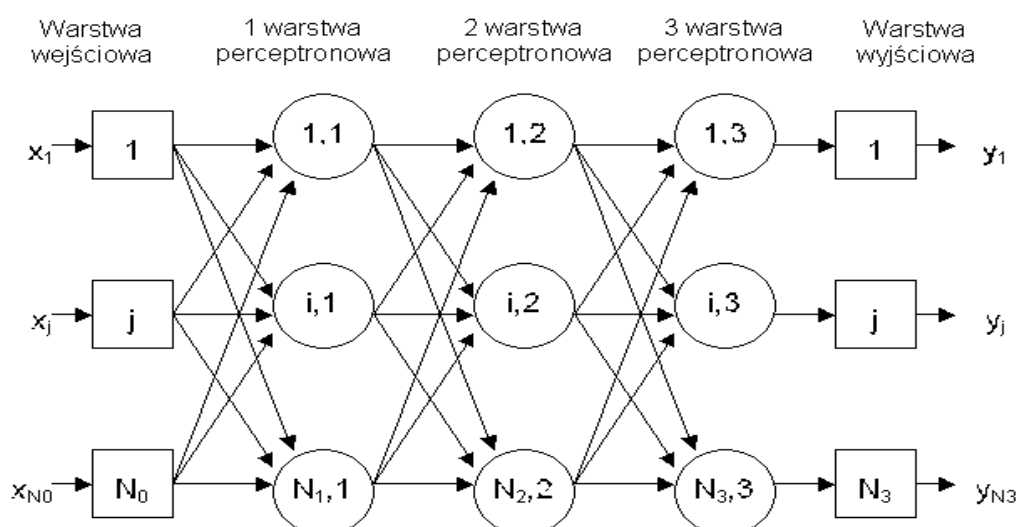
Celem ćwiczenia jest zapoznanie się z budową oraz działaniem wielowarstwowych sieci neuronowych poprzez uczenie kształtu funkcji matematycznej z użyciem algorytmu wstecznej propagacji błęd.

2. Podstawowe pojęcia

Sieć neuronowa (sztuczna **sieć neuronowa**) – ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu.

Jest to zbiór neuronów realizujących różne cele. W przypadku sztucznych sieci neuronowych jest to sztuczna struktura zaprojektowana i zbudowana w taki sposób, aby modelowała działanie naturalnego układu nerwowego, a w szczególności mózgu.

Sieć wielowarstwowa jest siecią połączonych ze sobą neuronów, w której neurony przesyłają sygnały wszystkim neuronom kolejnej warstwy (na zasadzie „każdy z każdym”). Pozwala ona na tworzenie sieci niemal o dowolnej charakterystyce. Działanie takiej sieci polega na liczeniu odpowiedzi neuronów w kolejnych warstwach – najpierw w pierwszej, do której trafiają sygnały z wejść sieci, potem korzystając z wyników poprzedniej warstwy, liczymy odpowiedzi kolejnych warstw neuronów. Odpowiedzi znajdujące się w ostatniej warstwie są wynikami sieci.



3. Opis działania

a) Wzór dla funkcji Rastrign 3D:
$$f(\mathbf{x}) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$$

Gdzie: $A = 10$,
 $x_i \in [-5.12; 5.12]$
 $f(0) = 0$

(źródło: https://en.wikipedia.org/wiki/Rastrigin_function)

- b) Dane wejściowe - tablica 11-elementowa, liczby z przedziału $[-2; 2]$ z krokiem 0.4.
- c) Dane wyjściowe - tablica 11- elementowa zawierającą wyniki działania funkcji Rastrign 3D dla danych wejściowych.
- d) Do zaimplementowania sieci wielowarstwowej polecenie `feedforwardnet(2)`, które tworzy sieć wielowarstwową zawierającą 2 warstwy ukryte.
- e) Testowanie działania sieci, z użyciem algorytmu wstecznej propagacji i bez niego – aby wprowadzić ten algorytm należy dodać parametr treningu `net.trainFcn = 'traingd'`.

4. Wykonanie zadania i kod

```
close all; clear all; clc;

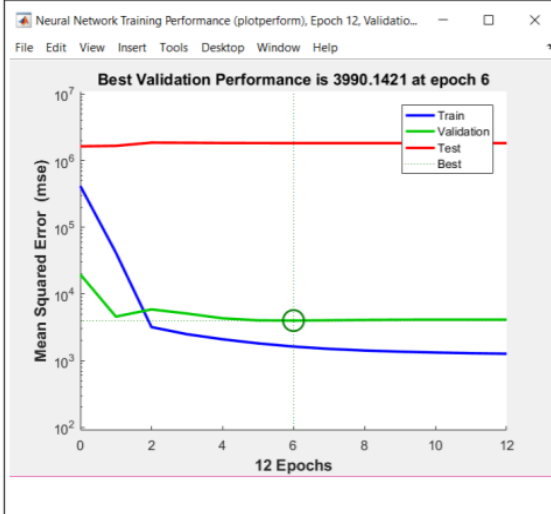
wej_in = [-2 -1.6 -1.2 -0.8 -0.4 0 0.4 0.8 1.2 1.6 2];
wej_out = [1.6633e+03 1.6880e+03 1.6867e+03 1.7764e+03
1.7800e+03 0.0 1.9495e+03 1.9825e+03 2.1477e+03 2.1791e+03
2.3262e+03];

testowe = zeros(1);

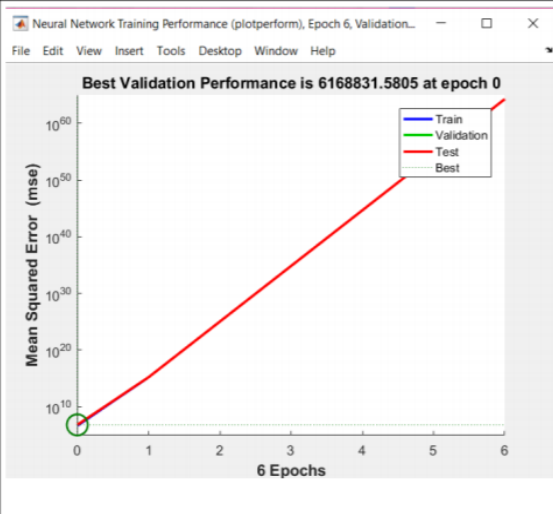
net = feedforwardnet(3); %tworzenie sieci z 2 warstwami
ukrytymi
net.trainFcn = 'traingd'; %algorytm wstecznej propagacji
net.trainParam.lr = 0.5; %wspolczynnik uczenia
net.trainParam.mc = 0.5; %bezwladnosc
net = train(net, wej_in, wej_out);
efekty = zeros(size(net));

for i = 1:11
    testowe(i) = RastrignTest3D(wej_in(i)); %wygenerowanie
prawidłowych wyników działania funkcji RastrignTest3D
    efekty(i) = sim(net, wej_in(i)); %test działania sieci
end
```

Sieć wielowarstwowa o wsp. uczenia i momentum 0.5



Sieć wielowarstwowa z alg. Propagacji błędu o wsp. uczenia 0.1 i momentum 0.5



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
 Training: Levenberg-Marquardt (trainlm)
 Performance: Mean Squared Error (mse)
 Calculations: MEX

Progress

| | | | |
|--------------------|----------|---------------|----------|
| Epoch: | 0 | 12 iterations | 1000 |
| Time: | | 0:00:01 | |
| Performance: | 4.16e+05 | 1.27e+03 | 0.00 |
| Gradient: | 1.83e+06 | 4.93e+03 | 1.00e-07 |
| Mu: | 0.500 | 500 | 1.00e+10 |
| Validation Checks: | 0 | 6 | 6 |

Plots

Performance (plotperform)
 Training State (plottrainstate)
 Error Histogram (ploterrhist)
 Regression (plotregression)

Plot Interval: 1 epochs

Opening Performance Plot

Stop Training Cancel

Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
 Training: Gradient Descent (traingd)
 Performance: Mean Squared Error (mse)
 Calculations: MEX

Progress

| | | | |
|--------------------|----------|--------------|----------|
| Epoch: | 0 | 6 iterations | 1000 |
| Time: | | 0:00:01 | |
| Performance: | 4.50e+06 | 4.50e+06 | 0.00 |
| Gradient: | 7.45e+06 | 5.63e+35 | 1.00e-05 |
| Validation Checks: | 0 | 6 | 6 |

Plots

Performance (plotperform)
 Training State (plottrainstate)
 Error Histogram (ploterrhist)
 Regression (plotregression)

Plot Interval: 1 epochs

Validation stop.

Stop Training Cancel



5. Wnioski

- Sieciami wielowarstwowymi, podobnie jak sieciami jednowarstwowymi, można sterować dopasowując poszczególne parametry uczenia, takie jak współczynnik uczenia, czy bezwładność (momentum).
- Sieci wielowarstwowe bez algorytmu propagacji błędów działają lepiej niż sieci posiadające ten algorytm
- Najgorsze wyniki osiągają te sieci, których współczynnik uczenia jest równy bezwładności.