

LABORATORIUM METODY SZTUCZNEJ INTELIGENCJI

Laboratorium nr 5

Temat: **Uczenie jednowarstwowych sieci neuronowych z zastosowaniem algorytmu wstecznej propagacji błędów**

Celem laboratorium jest zapoznanie się z metodami uczenia dwuwarstwowych sieci neuronowych (SN) z zastosowaniem algorytmu wstecznej propagacji błędów na podstawie przybornika Matlab Neural Network Toolbox. Do aproksymacji funkcji nieliniowych jednej zmiennej zastosowane zostaną SN z lokalnymi funkcjami bazowymi (*funkcje Gaussa*) oraz funkcjami bazowymi typu: *funkcje liniowe, funkcje sigmoidalne unipolarne i bipolarne*.

1. Uczenie SN z zastosowaniem algorytmu wstecznej propagacji błędów

Algorytm wstecznej propagacji błędów określa strategię doboru wag w sieci wielowarstwowej przy wykorzystaniu gradientowych metod optymalizacji.

Przykład 1. Realizacja SN dwuwarstwowej z zastosowaniem Toolboxa Neural Network

SN dwuwarstwowa z sigmoidalnymi bipolarnymi funkcjami aktywacji neuronów w warstwie ukrytej wraz z algorytmem uczenia wag sieci, została zrealizowana przy pomocy następującej procedury w Matlab-ie:

```
x=[...]; %Zadanie wektora wejściowego do SN
y=[...]; %Wartości zadanej funkcji y(x)

ogr=[0 2]; %Wartości ograniczeń wektora wejściowego x

siec_neur = newff(ogr,[5,1],{'tansig', 'purelin'}, 'traingdm', 'learnbd', 'sse');
%Inicjalizacja SN z sigmoidalnymi bipolarnymi funkcjami aktywacji neuronów (tansig) w warstwie ukrytej

NN_out(1,:)=sim(siec_neur,x); %Obliczanie wartości wyjścia dla zainicjalizowanej sieci

%purelin - funkcja aktywacji liniowa,
%tansig - sigmoidalna bipolarna funkcja aktywacji,
%logsig - sigmoidalna unipolarna funkcja aktywacji,
%radbas - radialna funkcja aktywacji

figure
subplot(3,1,1)
plot(x,y,'o',x,NN_out(1,:),'-') %Wizualizacja wyniku
subplot(3,1,2)

siec_neur.trainParam.epochs = 1500; %Liczba epok procesu uczenia
siec_neur.trainParam.show = 100; %Wyświetlanie błędów co ... epok
siec_neur.trainParam.goal = 0.01; %Zadany minimalny błąd odwzorowania
siec_neur.trainParam.min_grad = 1e-10;
net.trainParam.lr = 0.01; %Zadanie wartości współczynnika wzmocnienia uczenia

[siec_neur] = train(siec_neur,x,y); %Procedura uczenia wag SN
NN_out(2,:)=sim(siec_neur,x); %Obliczanie wartości wyjścia
z wyuczonej SN

subplot(3,1,3)
plot(x,y,'o',x,NN_out(2,:),'-') %Wizualizacja wyniku
```

Przykładowy algorytm sieci zbudowano dla następujących danych:

- estymowana funkcja: $f(x) = e^{-(x-2)(x-2)} \sin(3x) + 0.3x$,

- przedział zmienności wartości wejściowej: $x \in \langle 0, 2 \rangle$,

- przyjęto liczbę neuronów warstwy ukrytej $SN I_{neur} = 5$,

- zastosowanej funkcje aktywacji neuronów: sigmoidalne bipolarne, $S(x) = \frac{2}{1 + \exp(-\beta x)} - 1$,

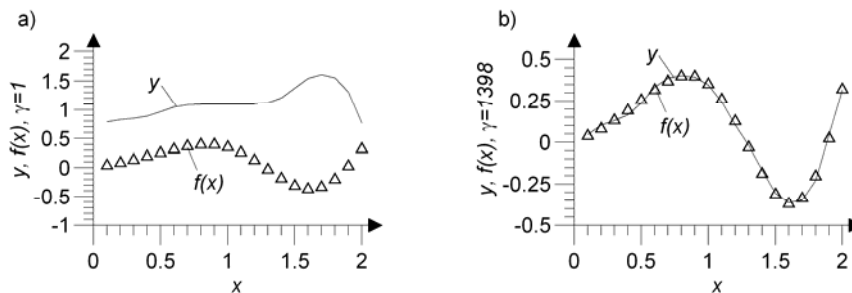
- przyjęto wartość współczynnika wzmocnienia uczenia $\alpha = 0.01$.

Algorytm uczenia SN z zastosowaniem gradientowej metody uczenia z momentem, zrealizowany przy użyciu funkcji *train*, podaje wartość błędów SSE po danej liczbie epok:

TRAININGDM, Epoch 0/1500, SSE 25.3514/0.005, Gradient 38.5982/1e-010
TRAININGDM, Epoch 100/1500, SSE 0.292753/0.005, Gradient 0.53014/1e-010

TRAININGDM, Epoch 1300/1500, SSE 0.00544918/0.005, Gradient 0.0158427/1e-010
TRAININGDM, Epoch 1398/1500, SSE 0.0049978/0.005, Gradient 0.0143844/1e-010
TRAININGDM, Performance goal met.

W wyniku symulacji SN z sigmoidalnymi bipolarnymi funkcjami aktywacji neuronów, uczonej metodą gradientową, w zadaniu estymacji zadanej nieliniowej funkcji $f(x)$ dla podanych wcześniej parametrów pracy, otrzymano następujące przebiegi wartości wyjścia z SN y (rys. 1.a) dla pierwszej epoki procesu uczenia ($\gamma=1$), oraz ostatniej epoki procesu uczenia $\gamma=1398$ (rys. 1.b)).



Rys. 1.a) Zadana nieliniowa funkcja $f(x)$ oraz wartości wyjścia z SN y dla $\gamma=1$, b) zadana nieliniowa funkcja $f(x)$ oraz wartości wyjścia z SN y dla $\gamma=1398$

Po $\gamma=1398$ epokach procesu uczenia SN osiąga zadany poziom dokładności odwzorowania.

2. Zadania do wykonania

Zbudować model SN z zastosowaniem procedury pokazanej w przykładzie 1, dla funkcji aktywacji neuronów w postaci:

- a) funkcji sigmoidalnych unipolarnych w warstwie ukrytej,
 - b) funkcji sigmoidalnych bipolarnych w warstwie ukrytej,
 - a) funkcji liniowych w warstwie ukrytej,
 - b) funkcji radialnych typu krzywa Gaussa w warstwie ukrytej,
- w postaci m -pliku w pakiecie Matlab. Zadaniem sieci jest aproksymacja nieliniowej funkcji jednej zmiennej $f(x)$:

a) $f(x) = 1 - e^{-x} - \sin(x)$,

b) $f(x) = \frac{1}{2} \sin(x)$,

c) $f(x) = \sin(2x) + 0.5 \cos(0.5x)$,

d) $f(x) = 5 - e^{-x} - x^2$,

e) $f(x) = 0.1 \cdot (2x^2 - 5x) - 3e^{-(x-2)(x-2)}$,

f) $f(x) = \sin(x) + \frac{1}{1 + e^{-3(x-3)}}$,

g) $f(x) = 3 \cdot e^{-2x} + 0.7 \cdot x$,

h) $f(x) = e^{-(x-2)(x-2)} \sin(3x) + 0.3x$,

dla $x \in \langle 0, 5 \rangle$. Przykład funkcji zadanej należy dobrać zgodnie z nr zespołu.

Parametry SN dla poszczególnych zespołów przedstawiono w tab.1.

Tab.1. Parametry SN dla poszczególnych zespołów

nr zespołu	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
$m=l_{neur}$	6	7	5	8	9	10	12	6	8	7	9	8
α	0.02	0.035	0.05	0.023	0.011	0.042	0.021	0.05	0.04	0.03	0.015	0.044

Należy przeprowadzić następujące badania:

Należy przeprowadzić badania symulacyjne dla algorytmu SN z sigmoidalnymi unipolarnymi funkcjami aktywacji neuronów zrealizowanego w postaci m -pliku *Matlab*-a, dla danych podanych w tabeli (pozostałe parametry przyjąć jak w przykładzie 1),

- zbadać wpływ liczby neuronów na proces uczenia SN,
- zbadać wpływ wartości współczynnika wzmocnienia uczenia α na proces uczenia SN,
- zbadać wpływ typu zastosowanych funkcji aktywacji neuronów w warstwie ukrytej na jakość aproksymacji i szybkość procesu uczenia (funkcje sigmoidalne unipolarne, sigmoidalne bipolarne, liniowe, Gaussa).

Sprawozdanie powinno zawierać:

- opis matematyczny rozwiązywanego problemu,
- dane przyjęte w symulacji,
- listingi programów,

w szczególności

1. Przebieg ćwiczenia

- przykładowy listing kodu *Matlab*-a służący do wygenerowania SN z sigmoidalnymi unipolarnymi funkcjami aktywacji neuronów oraz procedurę uczenia SN,
- przykładowy listing kodu *Matlab*-a służący do wygenerowania SN z sigmoidalnymi bipolarnymi funkcjami aktywacji neuronów oraz procedurę uczenia SN,
- przykładowy listing kodu *Matlab*-a służący do wygenerowania SN z liniowymi funkcjami aktywacji neuronów oraz procedurę uczenia SN,
- przykładowy listing kodu *Matlab*-a służący do wygenerowania SN z funkcjami aktywacji neuronów typu krzywa Gaussa oraz procedurę uczenia SN,

2. Wyniki symulacji

- 2.1. Wyniki dla SN z sigmoidalnymi unipolarnymi funkcjami aktywacji neuronów:
 - wykres (analogicznie jak w przykładzie 1):
 - a) funkcji zadanej $f(x)$ oraz wyjścia z SN dla $\gamma=1$,
 - b) funkcji zadanej $f(x)$ oraz wyjścia z SN dla $\gamma=n$, gdzie n to zadana liczba epok procesu uczenia wag SN lub liczba przeprowadzonych epok procesu uczenia (w przypadku, gdy uczenie zakończyło się sukcesem dla mniejszej liczby epok niż zadana ilość maksymalna),
 - wartości błędu SSE dla poszczególnych epok procesu uczenia (analogicznie jak w przykładzie 1).
 - 2.2. Wyniki dla SN z sigmoidalnymi bipolarnymi funkcjami aktywacji neuronów (analogicznie jak w pkt. 2.1).
 - 2.3. Wyniki dla SN z liniowymi funkcjami aktywacji neuronów (analogicznie jak w pkt. 2.1).
 - 2.4. Wyniki dla SN z radialnymi funkcjami aktywacji neuronów typu krzywe Gaussa (analogicznie jak w pkt. 2.1).
- **wnioski.**