**Katedra Mechaniki Stosowanej i Robotyki**
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

# METODY SZTUCZNEJ INTELIGENCJI

## Laboratorium nr 6

Temat: **Aproksymacja funkcji z zastosowaniem jednowarstwowych sieci neuronowych z rozszerzeniami funkcyjnymi**

# 1. Cel ćwiczenia

Celem ćwiczenia jest budowa sztucznych sieci neuronowych (SN) jednowarstwowych, liniowych ze względu na wagi oraz ich zastosowanie do aproksymacji funkcji nieliniowych jednej zmiennej. Zastosowane zostaną SN z lokalnymi funkcjami bazowymi (***funkcje Gaussa***) oraz nielokalnymi funkcjami bazowymi w warstwie ukrytej (***funkcje sigmoidalne bipolarne***).

# 2. Sieci neuronowe liniowe ze względu na wagi

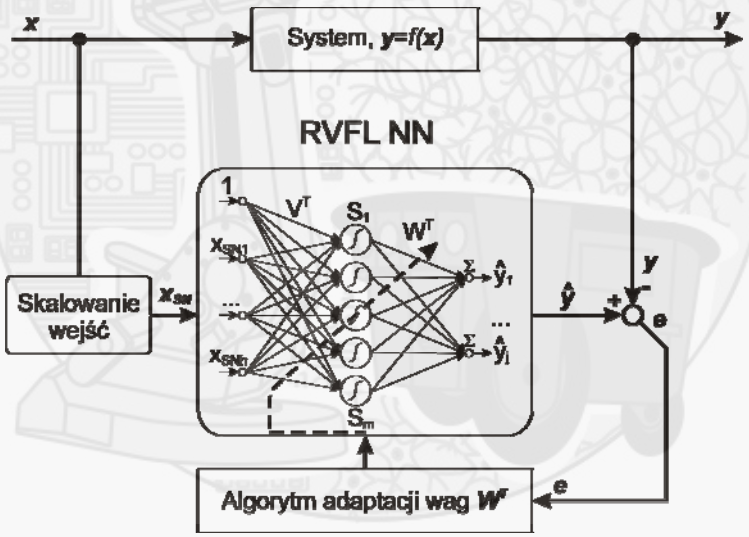
Estymatę dowolnej funkcji ***f(x)*** zapiszemy jako

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{W}^T \boldsymbol{S}(\boldsymbol{x}), \tag{1}$$

gdzie $\boldsymbol{x}$ to wektor wejściowy do sieci, $\boldsymbol{W}$ to wektor wag sieci neuronowej, ***S(.)*** to wektor funkcji bazowych. W zależności od zastosowanych funkcji aktywacji neuronów SN jednowarstwowe liniowe ze względu na wagi możemy podzielić na:

- SN z lokalnymi funkcjami bazowymi, np. SN z ***funkcjami Gaussa*** (ang. *Radial Basis Functions Neural Network - RBF NN*),
- SN z nielokalnymi funkcjami bazowymi, np. SN z ***funkcjami sigmoidlanymi*** (ang. *Random Vector Functional Link NN - RVFL NN*).

Schemat układu z pracującą równolegle SN estymującą dowolną nieliniową funkcję, z zastosowaniem do adaptacji wag metody uczenia z nauczycielem, przedstawiono na rys. 1.




Rys. 1. Schemat układu z SN estymującą funkcję ***y=f(x)***

### 2.1. Sieci neuronowe typu *RVFL*

Sieć neuronowa typu *RVFL* jest siecią z losowo wybranymi wagami warstwy wejściowej do sieci, z sigmoidalnymi funkcjami aktywacji neuronów. Jeżeli w miejsce wektora funkcji bazowych ***S(x)*** wybierzemy

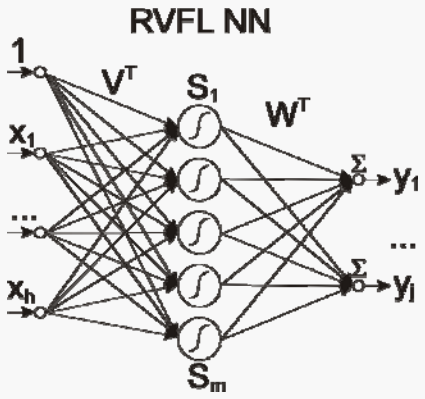
$$\boldsymbol{S}(\boldsymbol{x}) = \boldsymbol{S}\left(\boldsymbol{V}^T \boldsymbol{x}_V\right), \tag{2}$$

to estymata funkcji ***f(x)*** jest dana równaniem

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{W}^T \boldsymbol{S}\left(\boldsymbol{V}^T \boldsymbol{x}_V\right) \tag{3}$$

gdzie $\boldsymbol{V}$ to macierz stałych wag warstwy wejściowej, $\boldsymbol{x}_V = \left[1, \boldsymbol{x}^T\right]^T$ to wektor zmiennych wejściowych do sieci *RVFL*. Taka sieć jest liniowa ze względu na wagi i jest uniwersalnym aproksymatorem. W sieciach jednowarstwowych uczeniu podlegają jedynie wagi warstwy wyjściowej. Schemat sieci

neuronowej *MIMO* (ang. *Multi Input Multi Output*) *RVFL* o *h* wejściach, *j* wyjściach, oraz *m* neuronach w warstwie ukrytej przedstawiono na rys. 2.




Rys. 2. Schemat SN typu *RVFL*

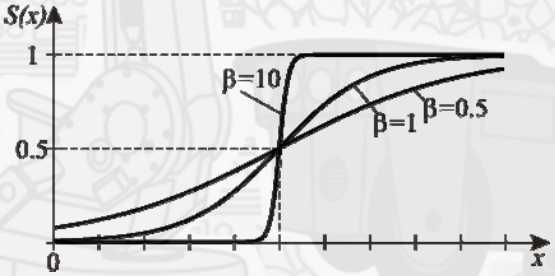
Funkcje aktywacji neuronów dla sieci *RVFL* są wybierane jako funkcje sigmoidalne:
- unipolarne:

$$\boldsymbol{S}(\boldsymbol{x}) = \frac{1}{1 + exp(-\beta \boldsymbol{V}^T \boldsymbol{x})}, \tag{4}$$

- bipolarne:

$$\boldsymbol{S}(\boldsymbol{x}) = \frac{2}{1 + exp(-\beta \boldsymbol{V}^T \boldsymbol{x})} - 1, \tag{5}$$

gdzie $\beta$ jest współczynnikiem.




Rys. 3. Przebieg funkcji sigmoidalnych unipolarnych dla różnych wartości współczynnika $\beta$

W celu uniknięcia pracy sieci w obszarze nasycenia funkcji sigmoidlanych zaleca się losowanie wag macierzy $\boldsymbol{V}$ z przedziału $V_{hm} \in \langle -0.5; 0.5 \rangle$.

Założenie ciągłej funkcji aktywacji umożliwia zastosowanie w uczeniu metody gradientowej. Zwykle przyjmuje się metodę największego spadku, zgodnie z którą aktualizacja wag odbywa się w kierunku ujemnego gradientu funkcji energetycznej.
Zdefiniujmy błąd odwzorowania nieliniowej funkcji *f(**x**)* przez SN *RVFL* opisaną równaniem (3) jako

$$e = \hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}), \tag{6}$$

oraz miarę błędu

$$E = \frac{1}{2} e^2. \tag{7}$$

Zastosowanie metody gradientowej (metody najszybszego spadku), powszechnie stosowanej w teorii optymalizacji, do adaptacji wag warstwy wyjściowej SN *RVFL*, pozwala na iteracyjne wyznaczanie kolejnych przybliżeń optymalnych wag sieci (neuronu) w kierunku ujemnego gradientu funkcji energetycznej (ze względu na wagi warstwy wyjściowej *W* SN). Przyrost wartości wag SN zapiszemy jako

$$\Delta \boldsymbol{W} = -\alpha \frac{\partial E}{\partial \boldsymbol{W}} = -\alpha e \boldsymbol{S}\left(\boldsymbol{V}^{\boldsymbol{T}} \boldsymbol{x}\right), \tag{8}$$

gdzie $\alpha$ to dodatni współczynnik uczenia o stałej wartości, $\alpha \in (0,1)$. Biorąc pod uwagę iteracyjny charakter obliczeń, zależność opisującą wartości wag warstwy wyjściowej SN *RVFL* w *k*-tym kroku procesu dyskretnego możemy zapisać jako:

$$\boldsymbol{W}_{\mathrm{k+1}} = \boldsymbol{W}_{\boldsymbol{k}} - \alpha e_k \boldsymbol{S}\left(\boldsymbol{V}^{\boldsymbol{T}} \boldsymbol{x}_{\mathrm{k}}\right). \tag{9}$$

Metoda gradientowa adaptacji wag SN (9) gwarantuje osiągnięcie jedynie minimum lokalnego, natomiast wyjście ze strefy przyciągania określonego minimum lokalnego nie jest możliwe do osiągnięcia przy zastosowaniu tej metody. Problem ten można w pewnym stopniu rozwiązać poprzez zastosowanie metody uczenia z tzw. ***momentum***. W metodzie tej proces aktualizacji wag uwzględnia nie tylko informację o gradiencie funkcji, ale również aktualny trend zmian wag.
Proces adaptacji wag z zastosowaniem momentum można zapisać przy pomocy zależności

$$\boldsymbol{W}_{\mathrm{k+1}} = \boldsymbol{W}_{\boldsymbol{k}} - \alpha e_k \boldsymbol{S}\left(\boldsymbol{V}^{\boldsymbol{T}} \boldsymbol{x}_{\mathrm{k}}\right) + \gamma\left(\boldsymbol{W}_{\boldsymbol{k}} - \boldsymbol{W}_{\boldsymbol{k-1}}\right), \tag{10}$$

gdzie $\gamma$ to tzw. *współczynnik momentum* o stałej wartości.

### 2.2. Sieci neuronowe z funkcjami Gaussa (RBF)

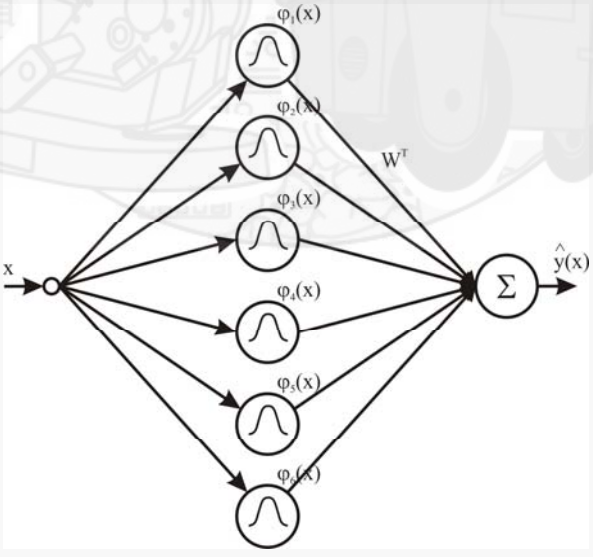
SN z lokalnymi funkcjami aktywacji neuronów typu ***funkcje Gaussa*** (ang. *Radial Basis Functions RBF NN*) można opisać równaniem

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{W}^{\boldsymbol{T}} \boldsymbol{S}(\boldsymbol{x}), \tag{11}$$

gdzie funkcję aktywacji *i*-tego neuronu SN typu Gaussa możemy zapisać za pomocą zależności

$$S_i(x) = exp\left[-\left(\frac{x - b_i}{a_i}\right)^2\right], \tag{12}$$

gdzie $a_i$ jest parametrem określającym szerokość krzywej gaussowskiej dla *i*-tego neuronu, natomiast $b_i$ parametrem określającym położenie środka krzywej. Schematycznie SN z radialnymi funkcjami aktywacji neuronów typu funkcja Gaussa, przedstawiono na rys. 4.




Rys. 4. Sieć RBF o jednym wejściu i jednym wyjściu ($\varphi_{\mathbf{i}}(.) = \mathbf{S_i}(.)$)

SN z funkcjami aktywacji neuronów typu funkcja Gaussa, przy założeniu stałych wartości parametrów funkcji Gaussa (szerokości funkcji $a_i$ oraz położenia środków funkcji $b_i$ ) jest liniowa ze względu na parametry i jest uniwersalnym aproksymatorem. W takim przypadku adaptowane są jedynie wagi warstwy wyjściowej z sieci. Schemat SN z funkcjami Gaussa o jednym wejściu i jednym wyjściu pokazano na rys.4 a funkcje aktywacji neuronów są opisane zależnością (12).
Wagi SN z radialnymi funkcjami aktywacji neuronów mogą być adaptowane przy pomocy metody gradientowej największego spadku

$$W_{\mathbf{k}+1} = W_k - \alpha e_k S(x_{\mathbf{k}}), \tag{13}$$

lub metody największego spadku z zastosowaniem tzw. momentum

$$W_{\mathbf{k}+1} = W_k - \alpha e_k S(x_{\mathbf{k}}) + \gamma(W_k - W_{k-1}). \tag{14}$$

Położenia środków $a_i$ oraz szerokości funkcji Gaussa $b_i$ są parametrami projektowymi sieci i mogą zostać dobrane przy pomocy różnych metod. Zazwyczaj nie dysponując wystarczającą wiedzą na temat aproksymowanej funkcji, można założyć równomierne rozmieszczenie funkcji aktywacji neuronów typu Gaussa w przestrzeni wejściowej do sieci.

**Przykład 1.** Realizacja SN *RVFL* w pakiecie Matlab, w postaci *m*-pliku

SN *RVFL* jednowarstwową, liniową ze względu na wagi, o losowym doborze stałych wag warstwy wejściowej ***V***, sigmoidalnych bipolarnych funkcjach aktywacji neuronów oraz wagach warstwy wyjściowej ***W*** adaptowanych przy pomocy metody gradientowej, zbudowano w formie *m*-pliku Matlab-a.
Przykładowy algorytm sieci *RVFL* zbudowano dla następujących danych:
- estymowana funkcja: $f(x) = e^{-(x-2)(x-2)} sin(3x) + 0.3x$,
- przedział zmienności wartości wejściowej: $x \in \langle 0,5 \rangle$,
- przyjęto liczbę neuronów warstwy ukrytej *SN* $l_{neur} = 8$,
- zastosowanej funkcje aktywacji neuronów: sigmoidalne bipolarne, $\boldsymbol{S}(\boldsymbol{x}) = \frac{2}{1 + exp(-\beta \boldsymbol{V}^T \boldsymbol{x})} - 1$,
- wartości stałych wag warstwy wejściowej *V* dobrane losowo z przedziału $V_{hm} \in \langle -0.5; 0.5 \rangle$,
- wartość współczynnika przegięcia funkcji sigmoidlanych: $\beta=1$,
- przyjęto zerowe warunki początkowe wag warstwy wyjściowej sieci $\boldsymbol{W_1}=[0,0,...,0]^T$,
- przyjęto wartość współczynnika wzmocnienia uczenia $\alpha$=0.4,
- przyjęto wartość współczynnik momentum $\gamma$=0.

Algorytm działania sieci oraz adaptacji jej wag można w uproszczony sposób opisać poniższą procedurą:

1. Deklaracja niezbędnych zmiennych oraz macierzy wag:

$\boldsymbol{W_{1(1xlneur)}}=[0,0,\ldots,0]^T$;
$l_{neur}$; $\beta$; $\alpha$ ; $\gamma$ ; V; S;
V=0.5-rand(2, $l_{neur}$);

2. Główna pętla programu:

*pętla 1 (dla i=1,2,…,N)* %gdzie *N* to liczba kroków dyskretnego procesu, np. *N*=500 dla $x \in \langle 0,5 \rangle$ )

2.1. Obliczenie wartości funkcji zadanej:

$x_i$=i/100; % zapamiętanie wartości zmiennej *x* w danym kroku iteracji, dla $x \in \langle 0,5 \rangle$,
$f_i(x_i)$=…; % obliczenie wartości zadanej $f_i(x_i)$ w *i*-tym kroku,

2.2. Obliczenia związane z SN,

2.2.a) skalowanie zmiennych wejściowych do SN, poszerzenie wektora wejściowego o „1”,

$x_{SNi}=[1,\ x_i/x_{max}]^T$; %gdzie $x_{max}$ to współczynnik skalowania,

2.2.b) obliczenie wartości wyjścia z SN dla aktualnego stanu

*pętla 2 (dla k=1,2,…, $l_{neur}$,)* %gdzie $l_{neur}$ to liczba neuronów sieci)

$$S_{k,i}\left(\boldsymbol{x}_{SNi}\right)=\frac{2}{1+exp\left(-\beta \boldsymbol{V}_{k,:}^{T}\boldsymbol{x}_{SNi}\right)}-1;$$ %wartość funkcji aktywacji dla *k*-tego neuronu

*end;*

$y_i=\boldsymbol{W_i^T}\cdot\boldsymbol{S}_{:,i}$; % wartość wyjścia z sieci,

2.2.c) obliczanie błędu odwzorowania zadanej funkcji przez SN,

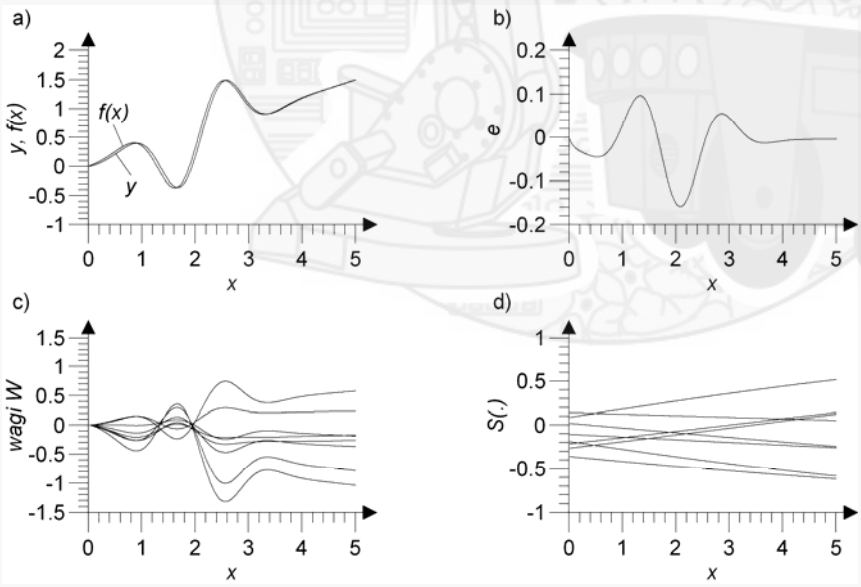
$e_i=y_i-f_i(x_i)$;

2.2.d) uczenie wag $\boldsymbol{W}$ warstwy wyjściowej SN metodą gradientową (10),

$$\boldsymbol{W}_{i+1}=\boldsymbol{W}_i-\alpha e_i \boldsymbol{S}\left(\boldsymbol{V}^T\boldsymbol{x}_i\right)+\gamma\left(\boldsymbol{W}_i-\boldsymbol{W}_{i-1}\right);$$

*end;* %koniec pętli głównej programu

W wyniku symulacji SN *RVFL* w zadaniu estymacji zadanej nieliniowej funkcji *f(x)* dla podanych wcześniej parametrów pracy, otrzymano następujące przebiegi wartości wyjścia z SN *y* (rys. 5.a)), błędu estymacji nieliniowego odwzorowania *e* (rys. 5.b)), wartości wag warstwy wyjściowej $\boldsymbol{W}$ dla poszczególnych neuronów (rys. 5.c)), oraz wartości sigmoidalnych bipolarnych funkcji aktywacji neuronów *S(.)* (rys. 5.d.)).




Rys. 5.a) Zadana nieliniowa funkcja *f(x)* oraz wartości wyjścia z SN *y*, b) błąd estymacji nieliniowego odwzorowania *e*, c) wartości wag warstwy wyjściowej $\boldsymbol{W}$, d) wartości sigmoidalnych bipolarnych funkcji aktywacji neuronów *S(.)*

Do oceny jakości aproksymacji nieliniowej funkcji *f(x)* przyjąć następujące **wskaźniki jakości**:
- maksymalna wartość błędu aproksymacji funkcji zadanej $e_{max}$,

- pierwiastek błędu średniokwadratowego aproksymacji funkcji zadanej *e* (RMSE - root mean square error of *e*): $\varepsilon = \sqrt{\frac{1}{N}\sum_{k=1}^{N} e_k^2}$ , gdzie *k* - numer kolejnych kroków iteracji procesu, *N* - całkowita liczba kroków iteracji.

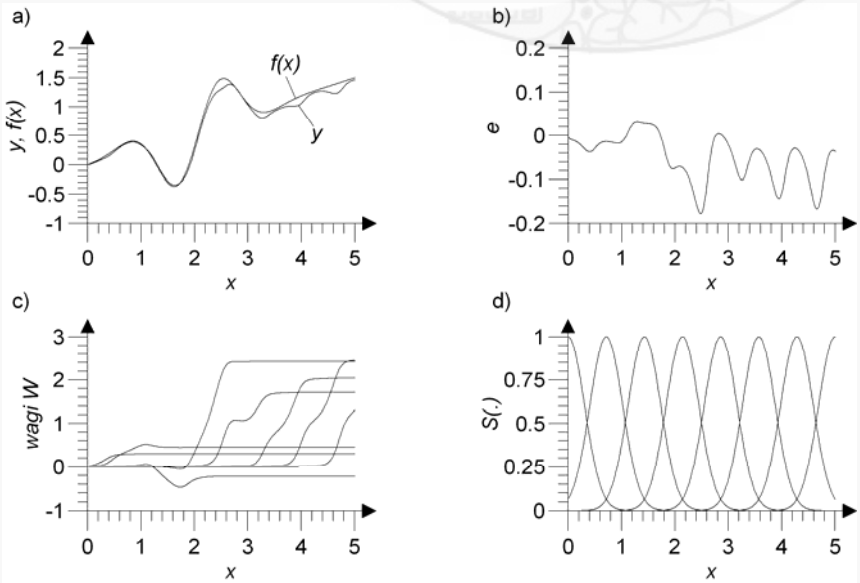
Uzyskano następujące wartości wskaźników jakości:
$e_{max}$=-0.11309
$\varepsilon$=0.00201

**Przykład 2.** Realizacja SN *RBF* w pakiecie Matlab, w postaci *m*-pliku

SN *RBF* jednowarstwową, liniową ze względu na wagi, o funkcjach aktywacji neuronów typu funkcje Gauss-a oraz wagach warstwy wyjściowej ***W*** adaptowanych przy pomocy metody gradientowej, zbudowano w formie *m*-pliku Matlab-a.
Przykładowy algorytm sieci *RBF* zbudowano dla następujących danych:
- estymowana funkcja: $f(x) = e^{-(x-2)(x-2)} sin(3x) + 0.3x$,
- przedział zmienności wartości wejściowej: $x \in \langle 0,5 \rangle$,
- przyjęto liczbę neuronów warstwy ukrytej *SN* $l_{neur}$ = 8,
- zastosowanej funkcje aktywacji neuronów: funkcje Gauss-a, $S_i(x) = exp\left[ -\left( \frac{x - b_i}{a_i} \right)^2 \right]$,
- wartość współczynnika szerokości funkcji Gaussa: $a_1=...=a_N$=0.0075,
- wartości współczynników położenia środków funkcji Gaussa: ***b***=*[0, 1/7, 2/7, ..., 7/7]*,
- przyjęto zerowe warunki początkowe wag warstwy wyjściowej sieci $\boldsymbol{W}_1=[0,0,...,0]^T$,
- przyjęto wartość współczynnika wzmocnienia uczenia $\alpha$=0.4,
- przyjęto wartość współczynnik momentum $\gamma$=0.

Algorytm działania sieci oraz adaptacji jej wag jest identyczny do przedstawionego w przykładzie 1.1., z tym że sigmoidalne funkcję aktywacji neuronów zastąpiono funkcją Gauss-a.

W wyniku symulacji SN *RBF* w zadaniu estymacji zadanej nieliniowej funkcji *f(x)* dla podanych wcześniej parametrów pracy, otrzymano następujące przebiegi wartości wyjścia z SN *y* (rys. 6.a)), błędu estymacji nieliniowego odwzorowania *e* (rys. 6.b)), wartości wag warstwy wyjściowej ***W*** dla poszczególnych neuronów (rys. 6.c)), oraz wartości funkcji aktywacji neuronów typu Gauss-a *S(.)* (rys. 6.d.)).



Rys. 6.a) Zadana nieliniowa funkcja *f(x)* oraz wartości wyjścia z SN *y*, b) błąd estymacji nieliniowego odwzorowania *e*, c) wartości wag warstwy wyjściowej ***W***, d) wartości funkcji aktywacji neuronów typu funkcja Gauss-a *S(.)*

Uzyskano następujące wartości wskaźników jakości:
$e_{max}$=-0.17767
$\varepsilon$=0.00287

## 3. Zadania do wykonania

1. Zbudować model SN liniowej ze względu na wagi, z zastosowaniem funkcji aktywacji neuronów w postaci:
a) funkcji sigmoidalnych bipolarnych (sieć *RVFL NN*),
b) funkcji Gaussa (sieć *RBF NN*),
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) $f(x)=sin(2x)+0.5cos(0.5x)$,
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. | 13. | 14. | 15. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $m=l_{neur}$ | 6 | 7 | 5 | 8 | 9 | 10 | 12 | 6 | 8 | 7 | 9 | 11 | 5 | 4 | 8 |
| $\alpha$ | 0.2 | 0.3 | 0.55 | 0.35 | 0.15 | 0.45 | 0.25 | 0.5 | 0.4 | 0.3 | 0.1 | 0.45 | 0.31 | 0.33 | 0.44 |
| $\gamma$ | 0.35 | 0.15 | 0.45 | 0.2 | 0.3 | 0.55 | 0.1 | 0.45 | 0.31 | 0.25 | 0.5 | 0.4 | 0.55 | 0.35 | 0.15 |

Przyjąć współczynnik pochylenia funkcji sigmoidlanych $\beta=1$.
Wartości współczynników położenia środków $a_i$ oraz szerokości funkcji Gaussa $b_i$ dla poszczególnych neuronów przyjąć wg. zasad podanych w rozdziale 2.2.
Przyjąć zerowe warunki początkowe wag warstwy wyjściowej sieci $\boldsymbol{W_0}=[0,0,...,0]^T$.
Przeprowadzić badania symulacyjne tak zbudowanych SN, dzieląc przedział wejściowy $x\in\langle 0,5\rangle$ na $N$=500 dyskretnych punktów.

Do oceny jakości aproksymacji nieliniowej funkcji *f(x)* przyjąć następujące **wskaźniki jakości**:
- maksymalna wartość błędu aproksymacji funkcji zadanej $e_{max}$,

- pierwiastek błędu średniokwadratowego aproksymacji funkcji zadanej *e* (RMSE - root mean square error of *e*): $\varepsilon = \sqrt{\frac{1}{N}\sum_{k=1}^{N} e_k^2}$ , gdzie *k* - numer kolejnych kroków iteracji procesu, *N* - całkowita liczba kroków iteracji.

**Należy przeprowadzić następujące badania:**

**I.** Należy przeprowadzić badania symulacyjne dla algorytmu **SN *RVFL*** zaprogramowanej w *m*-pliku *Matlab*-a:

**1)** symulację aproksymacji nieliniowej funkcji *f(x)* przez SN dla danych podanych w tab.1., przyjąć wartość współczynnika ***momentum*** $\gamma$ =0, [**symulacja 1**]

**2)** zbadać wpływ wartości wag początkowych warstwy wyjściowej ***W*** z SN na jakość aproksymacji, w tym celu należy przyjąć wagi początkowe $W_0=W_N$ z [symulacji 1], gdzie *N* to liczba kroków procesu iteracyjnego, [**symulacja 2**]

**3)** zbadać wpływ zmiany liczby neuronów SN na jakość aproksymacji nieliniowej funkcji *f(x)* (dla pozostałych parametrów projektowych SN jak w punkcie 1), w tym celu należy zmienić liczbę neuronów SN dodając do sieci 4 neurony ($l_{neur1}=l_{neur}+3$) oraz odejmując ze struktury sieci 3 neurony ($l_{neur2}= l_{neur}-3$), [**symulacja 3, symulacja 4**]

**4)** zbadać wpływ zmiany wartości współczynnika pochylenia funkcji sigmoidalnych $\beta$ na jakość aproksymacji (dla pozostałych parametrów projektowych SN jak w punkcie 1), w tym celu należy zmienić wartość współczynnika $\beta$ *do wartości* $\beta_1 = \frac{\beta}{nr\ zespolu}$, oraz $\beta_2 = nr\ zespolu$, [**symulacja 5, symulacja 6**] (Uwaga! Zespół nr 1 przyjmuje odpowiednio $\beta=1$, $\beta_1=0.2$, $\beta_2=5$)

**5)** zbadać wpływ zmiany wartości współczynnika ***momentum*** $\gamma$ na jakość aproksymacji (dla pozostałych parametrów projektowych SN jak w punkcie 1), w tym celu należy przeprowadzić symulację z wartością współczynnika ***momentum*** $\gamma$ jak w tab.1. [**symulacja 7**], następnie zmienić wartość współczynnika $\gamma$ do wartości $\gamma_1 = \frac{\gamma}{nr\ zespolu}$, [**symulacja 8**] (Uwaga! Zespół nr 1 przyjmuje odpowiednio $\gamma=0.35$, $\gamma_1=0.2$)

**II.** Należy przeprowadzić badania symulacyjne dla algorytmu SN z funkcjami aktywacji neuronów typu ***funkcje Gaussa*** zaprogramowanej w *m*-pliku Matlab-a:

**6)** symulację aproksymacji nieliniowej funkcji *f(x)* przez SN dla danych podanych w tab.1., przyjąć wartość współczynnika ***momentum*** $\gamma$ =0, [**symulacja 9**]

**7)** zbadać wpływ wartości wag początkowych warstwy wyjściowej ***W*** z SN na jakość aproksymacji, w tym celu należy przyjąć wagi początkowe $W_0=W_N$ z [symulacji 1], gdzie *N* to liczba kroków procesu iteracyjnego, [**symulacja 10]**

**8)** zbadać wpływ zmiany liczby neuronów SN na jakość aproksymacji nieliniowej funkcji *f(x)* (dla pozostałych parametrów projektowych SN jak w punkcie 5), w tym celu należy zmienić liczbę neuronów SN dodając do sieci 4 neurony ($l_{neur1}=l_{neur}+3$) oraz odejmując ze struktury sieci 3 neurony ($l_{neur2}= l_{neur}-3$), [**symulacja 11, symulacja 12**]

**9)** zbadać wpływ zmiany wartości współczynnika szerokości funkcji Gaussa ***a*** na jakość aproksymacji (dla pozostałych parametrów projektowych SN jak w punkcie 5), w tym celu należy zmienić wartość współczynnika ***a*** do wartości $a_1=3a$, oraz $a_2=0.5a$, [**symulacja 13, symulacja 14**]

**10)** zbadać wpływ zmiany wartości współczynnika ***momentum*** $\gamma$ na jakość aproksymacji (dla pozostałych parametrów projektowych SN jak w punkcie 5), w tym celu należy przeprowadzić symulację z wartością współczynnika ***momentum*** $\gamma$ jak w tab.1. [**symulacja 15**], następnie zmienić wartość współczynnika $\gamma$ *do wartości* $\gamma_1 = \frac{\gamma}{nr\ zespolu}$, [**symulacja 16**]

**III.** Należy przeprowadzić następujące badania symulacyjne dla algorytmu **SN *RVFL*** zaprogramowanej w Simulink-u:

**11)** symulację aproksymacji nieliniowej funkcji *f(x)* przez SN dla danych podanych w tab.1., przyjąć wartość współczynnika ***momentum*** $\gamma=0$, [**symulacja 17**]

**IV.** Należy przeprowadzić następujące badania symulacyjne dla algorytmu SN z funkcjami aktywacji neuronów typu funkcje ***Gaussa*** zaprogramowanej w Simulink-u:

**12)** symulację aproksymacji nieliniowej funkcji *f(x)* przez SN dla danych podanych w tab.1., przyjąć wartość współczynnika ***momentum*** $\gamma=0$, [**symulacja 18**]

**Sprawozdanie powinno zawierać:**

1. Wstęp teoretyczny
   - podstawowe wiadomości na temat zastosowanych SN liniowych ze względu na wagi,
   - podstawowe wiadomości na temat zastosowanej metody uczenia SN.
2. Przebieg ćwiczenia
   - przykładowy listing kodu Matlab-a służący do wygenerowania SN *RVFL* z procedurą uczenia, zastosowany do aproksymacji nieliniowej funkcji,
   - przykładowy listing kodu Matlab-a służący do wygenerowania SN z funkcjami *RBF*, z procedurą uczenia, zastosowany do aproksymacji nieliniowej funkcji,
   - graficzne przedstawienie realizacji SN *RVFL* w postaci schematów modelu *Simulink-a*,
   - graficzne przedstawienie realizacji SN z funkcjami *RBF* w postaci schematów modelu *Simulink-a*,
3. Wyniki symulacji

   3.1. Wyniki dla SN *RVFL:*
   - wykres (dla **symulacji 1,2,3,4**):
     a) funkcji zadanej *f(x)* oraz wyjścia z SN,
     b) przebiegu błędu *e*,
     c) przebiegi wartości wag warstwy wyjściowej ***W*** SN w procesie uczenia,
     c) w przypadku badania wpływu zmiany wartości współczynnika pochylenia funkcji sigmoidlanej $\beta$ oraz liczby neuronów SN *RVFL* na jakość aproksymacji - wykres funkcji aktywacji neuronów sieci *RVFL*, [**symulacja 1,2,3,4**]
   - dla wszystkich przeprowadzonych symulacji należy obliczyć wartości wskaźników jakości $e_{max}$, $\varepsilon$, wartości wskaźników jakości dla SN *RVFL* należy przedstawić w formie tab. 2. (dane dla zespołu nr 1).

Tab. 2. Wyniki przeprowadzonych symulacji dla SN *RVFL* (zespół nr 1) (symulacje 2 oraz 10 dla niezerowych wartości wag ***W***)

| nr symulacji | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 17. |
|---|---|---|---|---|---|---|---|---|---|
| $m=l_{neur}$ | 6 | 6 | 9 | 3 | 6 | 6 | 6 | 6 | 6 |
| $\beta$ | 1 | 1 | 1 | 1 | 0.2 | 5 | 1 | 1 | 1 |
| $\alpha$ | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $\gamma$ | 0 | 0 | 0 | 0 | 0 | 0 | 0.35 | 0.2 | 0 |
| $e_{max}$ | | | | | | | | | |
| $\varepsilon$ | | | | | | | | | |

   3.2. Wyniki dla *RBF* SN
   - wykresy (dla **symulacji 9,10,11,12**):
     a) funkcji zadanej *f(x)* oraz wyjścia z SN,
     b) przebiegu błędu *e*,
     c) przebiegi wartości wag warstwy wyjściowej ***W*** SN w procesie uczenia,
     d) w przypadku badania wpływu zmiany wartości współczynnika szerokości funkcji Gaussa oraz liczby neuronów SN *RBF* na jakość aproksymacji - wykres funkcji aktywacji neuronów sieci *RBF*, [**symulacja 9,10,11,12**]

- dla wszystkich przeprowadzonych symulacji należy obliczyć wartości wskaźników jakości $e_{max}$, $\varepsilon$ , wartości wskaźników jakości należy przedstawić w formie tab.2. (dane dla zespołu nr 1).

Tab. 3. Wyniki przeprowadzonych symulacji (zespół nr 1) (symulacje 2 oraz 10 dla niezerowych wartości wag ***W***)

| nr symulacji | 9. | 10. | 11. | 12. | 13. | 14. | 15. | 16. | 18. |
|---|---|---|---|---|---|---|---|---|---|
| $m=l_{neur}$ | 6 | 6 | 9 | 3 | 6 | 6 | 6 | 6 | 6 |
| $\beta$ | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $\alpha$ | a | a | a | a | 3a | 0.5a | a | a | a |
| $\gamma$ | 0 | 0 | 0 | 0 | 0 | 0 | 0.35 | 0.2 | 0 |
| $e_{max}$ | | | | | | | | | |
| $\varepsilon$ | | | | | | | | | |

4. Wnioski

Uwaga. Każdy realizowany podpunkt sprawozdania powinien być odpowiednio skomentowany.