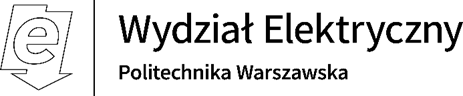
# Laboratorium 4

# Badanie zdolności generalizacyjnych sieci neuronowych

# Laboratorium „Modele Matematyczne w Uczeniu Maszynowym”

## **Zakład Elektrotechniki Teoretycznej i Informatyki Stosowanej**

Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych, PW



### **Ćwiczenie nr 4**

#### Badanie zdolności generalizacyjnych sieci neuronowych

### Autor: S. Osowski, K. Siwek, A. Osowska-Kurczab

##### Cel ćwiczenia

Badanie zdolności generalizacyjnych trzech rodzajów sieci neuronowych (MLP, RBF i SVM) w problemach regresji (aproksymacji).

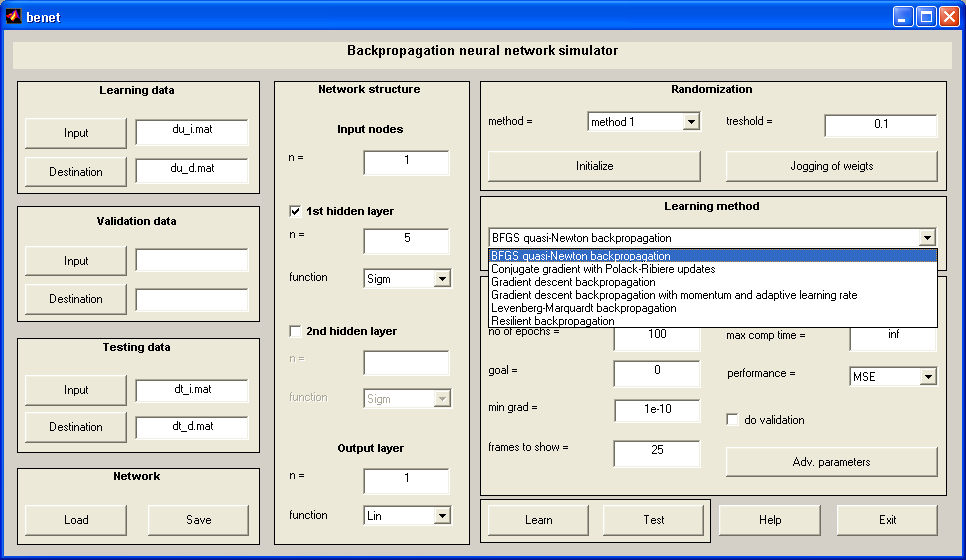
1. **Zdolności generalizacyjne sieci MLP**

**1.1 Opis programu**

Badania symulacyjne sieci MLP zostaną przeprowadzone przy wykorzystaniu programu *mlp* zaimplementowanego w Matlabie. Jest to program okienkowy, którego wygląd przedstawiony jest na rys. 1. Przed uruchomieniem programu należy wywołać polecenie dołączające do ścieżki Matlaba folder z funkcjami programu mlp. Dla zajęć stacjonarnych na PW:

***addpath('z:\neural\mlp')***

**Aby móc korzystać z interfejsów graficznych *mlp, rbf, svm\_win* należy albo wykonać polecenie *addpath* albo zmienić ścieżkę Matlaba na odpowiedni folder. Alternatywnie: *addpath(genpath(‘folder/nadrzedny’))***



Rys. 1 Wygląd głównego menu programu uczącego MLP

W programie posługiwać się będziemy algorytmem Levenberga-Marquardta (LM) i algorytmem gradientów sprzężonych.

**1.2 Badanie zdolności generalizacyjnych na przykładzie aproksymacji funkcji jednowymiarowych**

Zadanie dotyczy aproksymacji zbioru danych za pomocą sieci MLP (aproksymacja danych 1-wymiarowych i wielowyjściowych, np. zespół funkcji [*sin(x), sinc(x), sawtooth(x), square(x)*], jednocześnie przy jednym wspólnym wejściu *x*).

* wygenerować zbiór danych uczących (x**,d**) i testujących dla problemu, x – skalar, **d** – wektor 4-elementowy. Należy się upewnić, że uzyskane wektory **xu**, **du**, **xt**, **dt** są wektorami pionowymi, nie zawierającymi wartości NaN. Można wykorzystać poniższy program:

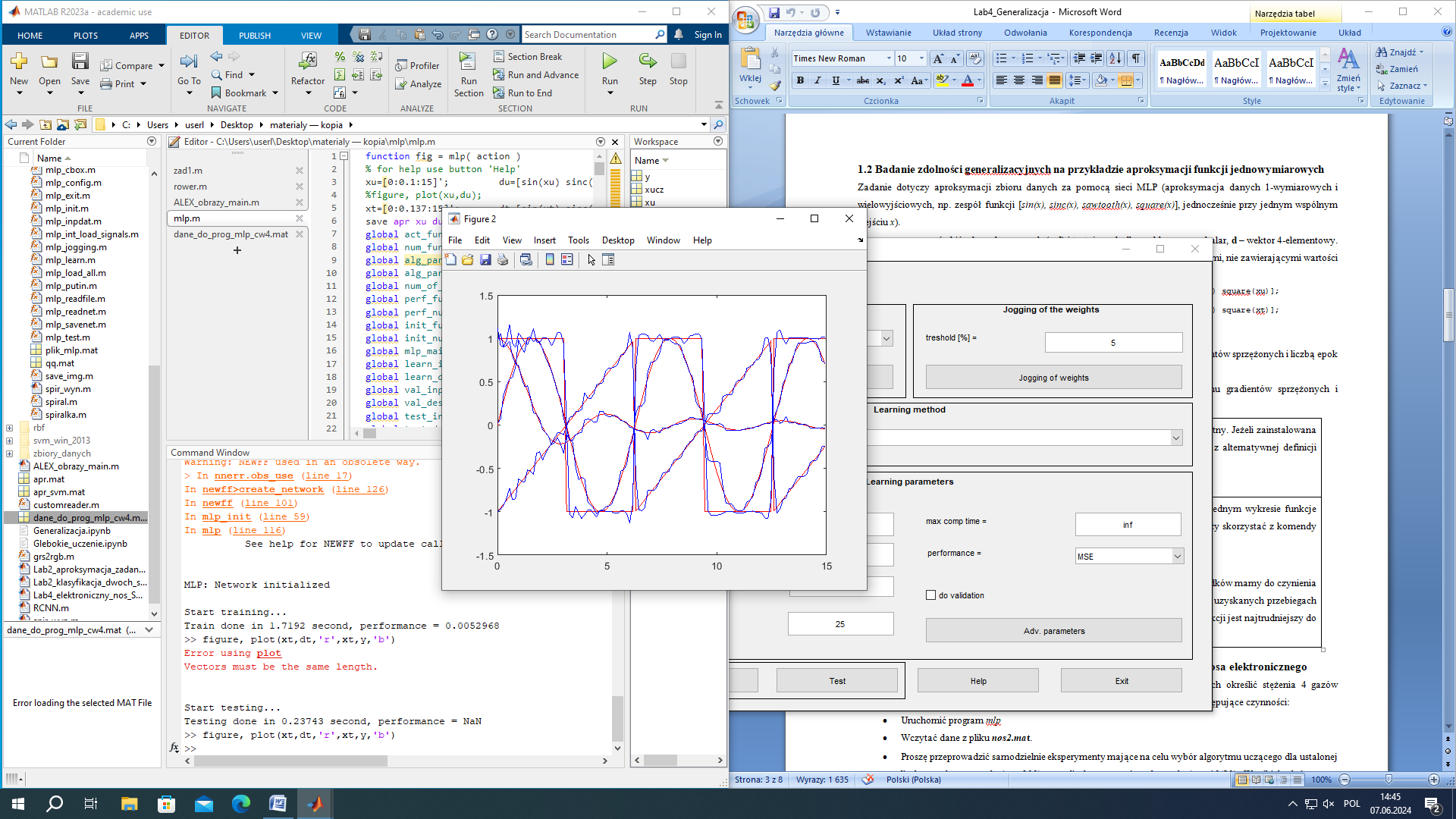
xu=[0:0.1:15]'; du=[sin(xu) sinc(0.5\*xu) sawtooth(xu) square(xu)];

figure, plot(xu,du);

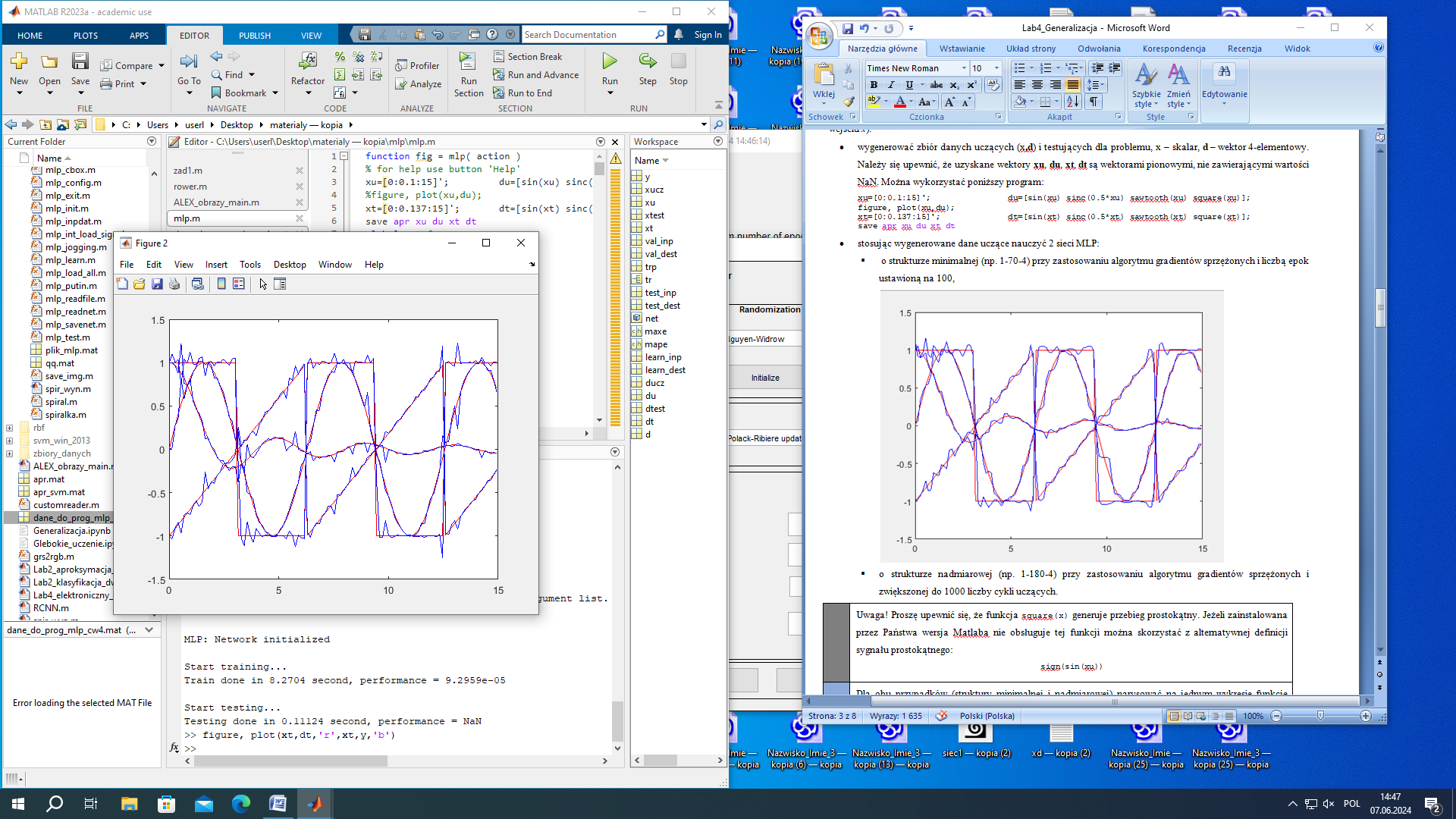
xt=[0:0.137:15]'; dt=[sin(xt) sinc(0.5\*xt) sawtooth(xt) square(xt)];

save apr xu du xt dt

* + stosując wygenerowane dane uczące nauczyć 2 sieci MLP:
    - o strukturze minimalnej (np. 1-70-4) przy zastosowaniu algorytmu gradientów sprzężonych i liczbą epok ustawioną na 100,



* + - o strukturze nadmiarowej (np. 1-180-4) przy zastosowaniu algorytmu gradientów sprzężonych i zwiększonej do 1000 liczby cykli uczących.



|  |  |
| --- | --- |
|  | Uwaga! Proszę upewnić się, że funkcja square(x) generuje przebieg prostokątny. Jeżeli zainstalowana przez Państwa wersja Matlaba nie obsługuje tej funkcji można skorzystać z alternatywnej definicji sygnału prostokątnego:  sign(sin(xu)) |
|  | Dla obu przypadków (struktury minimalnej i nadmiarowej) narysować na jednym wykresie funkcje zadaną i odtworzoną przez sieć dla danych testujących (zrzut ekranu). Należy skorzystać z komendy podobnej do:  figure, plot(xt,dt,'r',xt,y,'b')  Ocenić zdolności generalizacyjne sieci w obu przypadkach. W którym z przypadków mamy do czynienia z przeuczeniem (utratą generalizacji)? W jaki sposób można to zauważyć na uzyskanych przebiegach funkcji odtworzonej? Jaki parametr sieci MLP ma na to wpływ? Który typ funkcji jest najtrudniejszy do nauczenia dla sieci MLP? |

* 1. **Badanie zdolności generalizacyjnych MLP na przykładzie danych nosa elektronicznego**

Zadaniem sieci MLP jest na podstawie wskazań 5 sensorów półprzewodnikowych określić stężenia 4 gazów mieszaniny. Dane uczące zawierają 340 pomiarów nosa a testujące 80 Wykonać następujące czynności:

* + Uruchomić program *mlp*
  + Wczytać dane z pliku ***nos2.ma*t**.
  + Proszę przeprowadzić samodzielnie eksperymenty mające na celu wybór algorytmu uczącego dla ustalonej liczby epok uczących (np. 300) oraz liczby neuronów ukrytych (np. 10/20). Wyniki badań można przedstawić w postaci tabelarycznej lub opisać słownie jako jeden z paragrafów. Proszę zainspirować się badaniami przeprowadzonymi podczas laboratorium 2 – zadanie 1.
  + Dla wytypowanego najlepszego algorytmu uczącego zbadać wpływ liczby neuronów ukrytych na błąd uczenia i testowania przy stałej liczbie epok uczących (np. 300). Wyznaczyć błędy względne dopasowania dla każdego z 4 gazów dla danych testujących (tabela 1). Można do tego wykorzystać wzór:

|  |  |
| --- | --- |
|  | (1) |

WYBRANO LM

Tabela 1 Zależność błędów testowania sieci MLP od liczby neuronów ukrytych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Liczba neuronów ukrytych** | **Błąd względny**  **Gaz 1** | **Błąd względny**  **Gaz 2** | **Błąd względny**  **Gaz 3** | **Błąd względny**  **Gaz 4** |
| 2 |  |  |  |  |
| 5 |  |  |  |  |
| 10 |  |  |  |  |
| 200 |  |  |  |  |
| 500 |  |  |  |  |

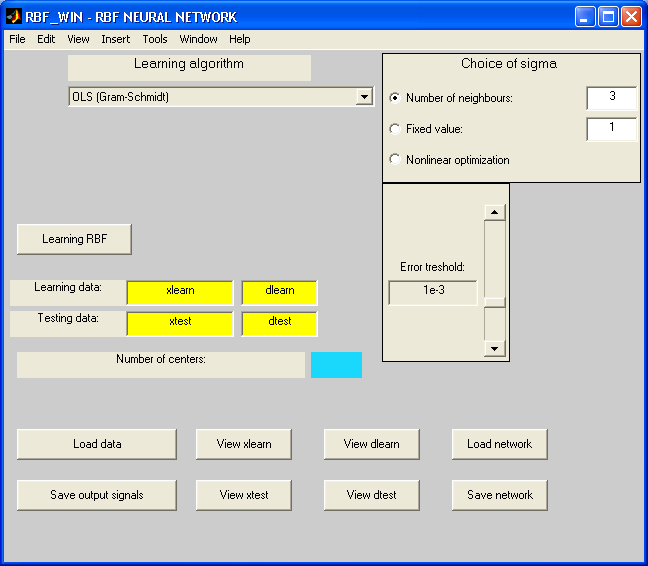
|  |  |
| --- | --- |
|  | Przedstawić wyniki badań nad wyborem algorytmu uczącego. Wyraźnie zaznaczyć, który algorytm został wybrany do kolejnych zadań.  Sporządzić wykresy błędów względnych w funkcji liczby neuronów ukrytych dla wszystkich gazów na podstawie tabeli 1 (*błąd względny = f(liczba neuronów ukrytych)*). Wykres powinien być wykresem typu ‘o-’. Dla czytelności można zastosować skalę logarytmiczną na osi x (Matlab: *semilogx*). Proszę, aby wykres w jednym oknie (subplot’cie) zawierał 4 serie odpowiadające 4 gazom. Określić optymalną z punktu widzenia generalizacji liczbę neuronów ukrytych.  Zanotować wykresy dopasowania wzorców zmian koncentracji gazów dla danych testujących przy optymalnej liczbie neuronów oraz dla sieci przeuczonej (2 zrzuty ekranu). Wykres dopasowania powinien mieć analogiczny wygląd do wykresów generowanych z użyciem przycisku *View dt* w interfejsie sieci RBF (4 subploty zawierające po 2 serie: *y* i *dtest*). |

|  |  |
| --- | --- |
|  | W sprawozdaniu skomentować wpływ liczby neuronów ukrytych na zdolność generalizacji sieci MLP. Czy na podstawie przeprowadzonych eksperymentów widoczna jest wyraźna korelacja? Wyjaśnić zjawisko utraty zdolności generalizacyjnych sieci przewymiarowanej. |

**2. Zdolności generalizacyjne sieci RBF**

Zadaniem sieci RBF jest na podstawie wskazań 5 sensorów półprzewodnikowych określić stężenia 4 gazów mieszaniny (identycznie jak w przypadku MLP). Dane uczące zawierają 340 pomiarów nosa a testujące 80 Wykonać następujące czynności:

* + Dodać do ścieżki folder zawierający program rbf. Dla zajęć stacjonarnych: ***addpath('z:\neural\rbf*** ***')***
  + Uruchomić program *rbf\_win* (rys. 2).

****

Rys. 2 Wygląd głównego menu programu *rbf\_win*

* + Wczytać dane z pliku ***nos2.ma*t**.
  + Dla algorytmu opartego na SVD i stałej szerokości funkcji gaussowskiej (=1) zbadać wpływ liczby neuronów ukrytych na błąd uczenia i testowania. Wyznaczyć błędy względne dopasowania dla każdego z 4 gazów dla danych testujących (tabela 2). Można do tego wykorzystać wzór 1.

Tabela 2 Zależność błędów testowania sieci RBF od liczby neuronów ukrytych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Liczba neuronów ukrytych** | **Błąd względny**  **Gaz 1** | **Błąd względny**  **Gaz 2** | **Błąd względny**  **Gaz 3** | **Błąd względny**  **Gaz 4** |
| 10 |  |  |  |  |
| 20 |  |  |  |  |
| 40 |  |  |  |  |
| 60 |  |  |  |  |
| 100 |  |  |  |  |
| 250 |  |  |  |  |
| 300 |  |  |  |  |

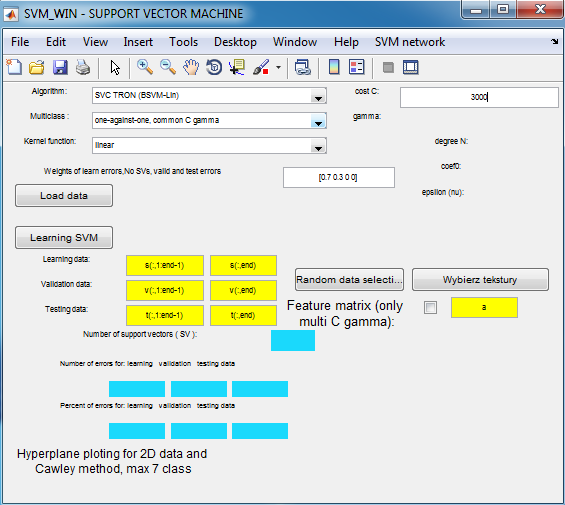
|  |  |
| --- | --- |
|  | Określić optymalną z punktu widzenia generalizacji liczbę neuronów ukrytych.  Sporządzić wykresy błędów względnych w funkcji liczby neuronów ukrytych dla wszystkich gazów. Oś x dla większej czytelności wykresu może mieć skalę logarytmiczną. Proszę, aby wykres wyglądał analogicznie do tego stworzonego w poprzednim podpunkcie z siecią MLP.  Do sprawozdania należy dołączyć wykresy dopasowania wzorców zmian koncentracji gazów dla danych testujących przy optymalnej liczby neuronów oraz sieci przeuczonej (2 zrzuty ekranu *View dt*). |

|  |  |
| --- | --- |
|  | W sprawozdaniu skomentować zdolności generalizacyjne sieci RBF. Jaki jest związek liczby neuronów ukrytych w sieci RBF z liczbą danych uczących? |

1. **Zdolności generalizacyjne sieci SVM**

Zadaniem sieci SVM jest na podstawie wskazań 5 sensorów półprzewodnikowych określić stężenia 4 gazów mieszaniny (identycznie jak w przypadku MLP). Dane uczące zawierają 340 pomiarów nosa a testujące 80. Ze względu na jedno wyjście sieci SVM procedura musi być powtarzana oddzielnie dla estymacji pojedynczego gazu przy tym samym wektorze wejściowym. Wykonać następujące czynności:

* + Dodać do ścieżki folder zawierający program *svm*. Dla zajęć stacjonarnych: ***addpath('z:\neural\svm\_2017')***
  + Uruchomić program *svm\_win* (rys. 3).

****

Rys. 3 Wygląd głównego menu programu *svm\_win*

* + Wczytać kolejno dane z pliku ***gaz1.mat***, ***gaz2.mat***, ***gaz3.mat***, ***gaz4.mat*** (sieć SVM ma tylko jedno wyjście, w związku z tym należy uruchomić ja dla każdego gazu oddzielnie).
  + Wytrenować sieć SVM korzystając z algorytmu SVR (BSVM) i zmiennej szerokości funkcji gaussowskiej (zadawanej wartością gamma, stanowiącą odwrotność szerokości funkcji gaussowskiej). Przyjąć stałą wartość współczynnika regularyzacji C=1000 i epsilon=0.005. Wyznaczyć błędy względne dopasowania dla każdego z 4 gazów dla danych testujących oraz zanotować liczbę wektorów nośnych NSV (tabela 3).

Tabela 3 Zależność błędów testowania sieci SVM od gamma przy ε=0.005 i C=1000

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **gamma** | **Gaz 1** | | **Gaz 2** | | **Gaz 3** | | **Gaz 4** | |
| NSV | Błąd względny | NSV | Błąd względny | NSV | Błąd względny | NSV | Błąd względny |
| 0.02 |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |
| 100 |  |  |  |  |  |  |  |  |
| 500 |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | Uwaga! Jeżeli wystąpią problemy z GUI svm\_win proszę skorzystać ze skryptu ***Lab4\_elektroniczny\_nos\_SVM.m***. Można rozszerzyć listę badanych gamm o wartości pośrednie, aby wykres błędów względnych w funkcji liczby wektorów nośnych miał więcej punktów pomiarowych. |
|  | Zbadać wpływ uzyskanej liczby wektorów nośnych Nsv (support vectors) na błąd uczenia i testowania sieci SVM z jądrem gaussowskim. Sporządzić wykresy błędów względnych w funkcji liczby wektorów nośnych dla wszystkich gazów. Wykres powinien być typu ‘-o’ (liniowy z punktowymi markerami) i powinien w jednym oknie zawierać wykresy wszystkich 4 gazów. Należy zadbać o to, aby uzyskana charakterystyka miała charakter funkcyjny (nie miała więcej niż jednej wartości dla każdego argumentu). Aby to uzyskać trzeba przesortować wspólnie wektory NSV oraz błędów względnych, aby były rosnące. Analogicznie jak w przypadku sieci MLP i RBF można zastosować skalę logarytmiczną x. Proszę skomentować uzyskany wykres. Jaka jest zależność między liczbą wektorów nośnych i błędem względnym?  Określić optymalną z punktu widzenia generalizacji wartość współczynnika gamma i towarzyszącą mu liczbę wektorów nośnych.  Do sprawozdania dołączyć wykresy dopasowania wzorców zmian koncentracji gazów dla danych testujących przy optymalnej liczby neuronów oraz sieci przeuczonej (2 zrzuty ekranu: dobre dopasowanie i przeuczenie). Powinny one wyglądać analogicznie do wykresów uzyskiwanych poprzez *View dt* interfejsu sieci RBF. Proszę skomentować uzyskane wykresy. |

|  |  |
| --- | --- |
|  | Sprawozdanie powinno zawierać wyniki badań wyszczególnionych w punktach, w tym wyniki liczbowe, wykresy, uzyskane obrazy.  W podsumowaniu proszę wyjaśnić zjawisko braku generalizacji i jego przyczyny dla indywidualnych zbadanych rozwiązań architektonicznych sieci. Które parametry w każdym z rodzajów sieci wpływają na zdolności generalizacyjne? Jak rodzaj sieci wpływa na liczbę neuronów ukrytych przy której sieć traci zdolność generalizacji? Jak można porównać zdolności generalizacyjne poszczególnych rodzajów sieci? |

**Literatura:**

1. Osowski S., Szmurło R., Modele matematyczne w uczeniu maszynowym, materiały elektroniczne, OWPW, 2023
2. Matlab user manual, MathWorks, Natick, 2019

**Materiały dodatkowe:**

1. **Machine Learning Yearning – o dobrych praktykach, jak badać generalizację (Andrew Ng)**

<https://d2wvfoqc9gyqzf.cloudfront.net/content/uploads/2018/09/Ng-MLY01-13.pdf>

1. **Generalizacja w sieciach neuronowych (ujęcie deep learningowe)** <https://deeplearningdemystified.com/article/fdl-5>   
   <https://towardsdatascience.com/understand-neural-networks-model-generalization-7baddf1c48ca>   
   <https://towardsdatascience.com/neural-networks-dont-generalize-the-way-you-think-they-do-de520bed2053>
2. **Bias variance trade-off**

<http://www.r2d3.us/visual-intro-to-machine-learning-part-2/>

1. **Dlaczego koncepcja bias-variance tradeoff może być niepoprawna w deep learningu?** <https://arxiv.org/pdf/1812.11118.pdf>