Ćwiczenie 1. Jednokierunkowa wielowarstwowa sieć neuronowa typu MLP

Celem ćwiczenia jest zaprojektowanie i zbadanie zdolności uogólniania sieci MLP w klasyfikacji danych. Wykorzystana sieć podlega procesowi uczenia w oparciu o metodę wstecznej propagacji błędu. Kod źródłowy na Listingu 1 przedstawia wykorzystane funkcje wymagane do realizacji ćwiczenia. Są to między innymi.

a. `crossvalind` – wygenerowanie indeksów do procedury walidacji krzyżowej,
b. `feedforwardnet` – stworzenie sieci o odpowiedniej architekturze,
c. `train` – przeprowadzenie procesu uczenia,
d. `sim` – zasymulowanie sieci.

W zaprezentowanym skrypcie należy dobrać wartość parametru **`K`** do przeprowadzenia procedury walidacji krzyżowej (zaleca się wartość **`K = 10`**).

```
clear
load('Zbiór');
X = Zbiór(:,1:end-1);
Y = Zbiór(:,end) ;

% Wygenerowanie indeksów do CV
% Wymagane jest zadanie wartości dla K.
indeksy = crossvalind('Kfold', Y, K);
Acc_CV = zeros(10,1);

for k = 1:K
    k
    % Indeksy do walidacji
    test_ind = (indeksy == k);
    % Indeksy do uczenia
    train_ind = ~test_ind;

    % Rekordy do walidacji
    X_Test = X(test_ind,:)';
    Y_Test = Y(test_ind)';

    % Rekordy do uczenia
    X_Train = X(train_ind,:)';
    Y_Train = Y(train_ind)';

    % Projekt i uczenie MLP:
    net = feedforwardnet(20);
    net.layers{1}.transferFcn = 'tansig';
    net = train(net, X_Train, Y_Train);

    % Walidacja sieci
    Y_Out = round(sim(net, X_Test));

    % Dokładnośc sieci
    Acc_CV(k) = sum(Y_Out == Y_Test)/length(Y_Test);
end

[mean(Acc_CV) std(Acc_CV)]
save('Wyniki.mat')
```

Listing 1. Zastosowanie jednokierunkowej sieci typu MLP w klasyfikacji danych w aplikacji Matlab.

Zdolność uogólniania sieci MLP należy wyznaczyć na podstawie kryterium dokładności. Aby wyniki były statystycznie wiarygodne, kod przedstawiony na Listingu 1 musi być wywołany/uruchomiony przynajmniej 10 krotnie.

Aby osiągnąć jak najwyższą predykcję sieci konieczne jest również:

1. zastosowanie rozbudowanej sieci na przykładzie architektury dwu-warstwowej; wywołanie funkcji: `feedforwardnet([10 20]);`
2. wykorzystanie różnych funkcji aktywacji, np.: `'logsig'`, `'purelin'`;
3. użycie kilku metod uczenia sieci, na przykład:
   - `'trainscg'` – algorytm gradientów sprzężonych ze skalowaniem,
   - `'traingd'` – algorytm spadku gradientu,
   - `'traingdm'` – algorytm spadku gradientu z składnikiem momentum,
   - `'trainlm'` – algorytm Levenberga-Marquardta,
   - `'trainbfg'` – algorytm quasi-Newtona.

   W tym celu wystarczy skorzystać z następującej instrukcji:
   ```
   net.trainFcn = 'traingd';
   ```

Wyniki analizy należy przedstawić w formie wykresów prezentujących zależności dokładności względem liczby neuronów dla sieci jedno- i dwu-warstwowej uwzględniając różne funkcje aktywacji w warstwach oraz różne algorytmy uczenia.