

Тема №1: Класификация с изкуствени невронни мрежи. Класификация с многослоен перцептрон (MLP).

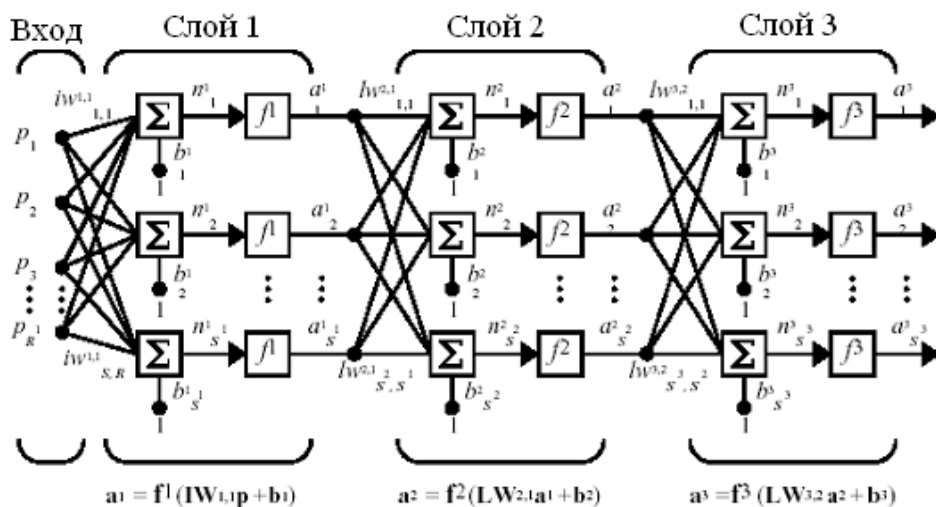
Цел на упражнението

Тема 1 запознава с концептуалните особености на архитектурата, обучението, и възможностите на невронни мрежи от типа многослоен перцептрон (*multilayer perceptron neural network*, MLP NN). Поясняват се допусканията, характерни за дискриминативните класификатори използващи многослойна невронна мрежа от перцептрони, добрите практики при избора на брой слоеве и количеството неврони във всеки слой. Изследват се основните методи за обучение и адаптиране на MLP, както и ограниченията и недостатъците, които определят областта на приложимост на този тип невронни мрежи.

Понятие за многослоен перцептрон

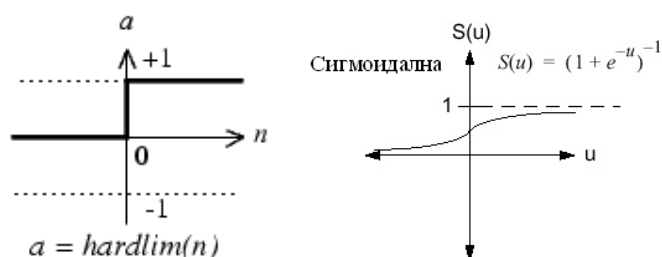
Невронните мрежи от типа многослоен перцептрон са широко използвани още от 80-те години на XX век, в приложения изискващи класификация на данни в два или повече класа. Често тези невронни мрежи се наричат с термина *feedforward*, с което се подчертава факта, че информацията се разпространява еднопосочно от входа към изхода. За всеки входен вектор от данни мрежата изчислява изхода си, без да използва информация от предишни или бъдещи моменти от време. В традиционният вариант броя на слоевете най-често е два, три или четири, като ограничението за повече слоеве произлиза главно от методите за обучение (виж. Фиг. 1.1).

В общия случай, невронните мрежи от типа многослоен перцептрон се обучават с някой от разновидностите на градиентните методи, използващи обратно разпространение на грешката от изхода към входа (*backpropagation*). При обучението на невронната мрежа, тегловните коефициенти и отместванията на невроните се настройват така че да осигурят най-добро възможно разделение на представителите на отделните класове, т.е. минимална грешка върху набора от обучаващи данни.



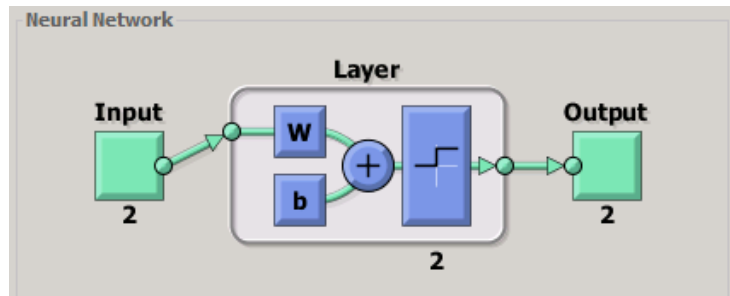
Фиг. 1.1. Трислойна невронна мрежа

Слой 1 приема данните дефинирани с входния вектор $\mathbf{P}=[p_1, p_2, \dots, p_R]$, поради което се нарича входен слой. Изходът на Слой 1 е свързан към входовете на невроните от втория слой, означен като Слой 2. Втория слой няма директен достъп до входните данни или до изхода на мрежата, поради което се нарича *скрит слой*. Изходите на скрития слой се подават на входовете на невроните от третия слой, като в случая. изходите на невроните от Слой 3 се явяват изход на мрежата от многослойни перцептрони. Най-често активиращите функции на невроните от всички слоеве f_1, f_2, f_3 се избират еднотипни и се реализират или със сигмоидална функция (soft-limiter) или с прагова ограничаваща активираща функция от типа hardlimiter. И в двата случая, активиращата функция може да бъде едностранна, приемаща само нулеви и положителни стойности в изхода) или двустранна.



Пример за класификация с невронна мрежа от тип многослоен перцептрон (MLP NN)

Нека да разгледаме пример за класификация на двумерни обекти в два класа.



Всеки обект $X(:,k)$ се описва с два параметъра $\{X(1,k), X(2,k)\}$, които определят неговото местоположение в равнината.

```
X = [1 1; 2 1; 1 2; 2 2; 1.5 1.5; 2.5 1.5; 1.2 2.1; 2 2; 2 3; 3 2; 3 3; 3 4; 4.1
3.9; 2.7 2.2; 2.6 3.6; 3.6 2.4; 3.4 3.5; 3.8 4.9; 4 3; 3 5; 3 6; 4 4; 4 5; 4 6;
5 6; 6 5; 5 4]';
Tc = [1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1];
```

Визуализираме данните $X(:,k)$ в двумерна координатна система

```
Co = {'or', '*g', 'xm'};
lenX=length(Tc);
figure, plot(X(:,1),Co{1}), axis([0 7 0 7]), hold on
for k=2:lenX,
    plot(X(1,k),X(2,k),Co{Tc(k)})
end
```

Избираме подмножество от първите 15 образа

```
X=X(:,1:15);
Tc=Tc(1:15);
T = ind2vec(Tc);
figure
plotpv(X,T)
```

Задаваме произволен тестов вектор за да проверим работоспособността на невронната мрежа.

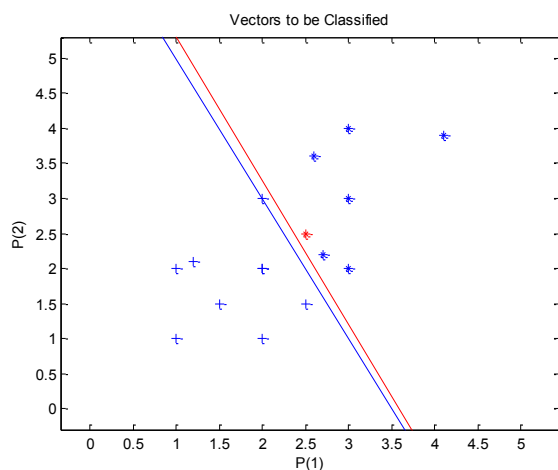
```
%Xtest = [2.4, 2.1, 2.4; 2.7, 2.4, 2.5];
Xtest = [2.5; 2.5];
%Xtest = [2.4; 2.5];
net = newp(X,T);
plotpv(X,T);
%plotpc(net.IW{1},net.b{1});

%net.adaptParam.passes = 30;
%net = adapt(net,X,T);
Y = net(Xtest)
net.trainParam.epochs = 300;
net = train(net,X,T);
Y = net(Xtest)
%plotpc(net.IW{1},net.b{1});

a = sim(net,Xtest)
```

```
plotpv(Xtest,a);
point = findobj(gca,'type','line');
set(point,'Color','red');
hold on;
plotpv(X,T);
plotpc(net.IW{1},net.b{1});
hold off;
```

Невронната мрежа връща резултата от разпознаването във вектора **a**, която съдържа 1 в позицията която има смисъл на *индекса на класа към който е причислен тестовия вектор*. Ако са подадени няколко тестови вектора наведнъж, тогава **a** е матрица с брой елементи равни на броя тестови вектори.



Задачи за изпълнение

Задача 1. Като използвате данните записани в MATLAB файла „zad1.mat“, създайте невронна мрежа за класификация в три класа.

Проверете работоспособността на невронната мрежа за пет тестови вектора

```
Xtest=[3.5 4.2]';  
Xtest=[0.5 4.1]';  
Xtest=[5 3.5]';  
Xtest=[2.9 1.6]';  
Xtest=[4 4.6]';
```

Изберете настройки на невронната мрежа, така че да се минимизира грешката от класификация.

Задача 2. Създайте еднослойна невронна мрежа за класификация в два класа с неврони от тип перцептрон за решаване на проблема дефиниран в Задача 1. Използвайте първите два от трите класа дефинирани в MATLAB файла „zad1.mat“.

Използвайте следните 3 тестови вектора за да изследвате работоспособността на мрежата.

```
Xtest = [2.4, 2.1, 2.4; 2.7, 2.4, 2.5];
```

Определете оптималната архитектура на невронната мрежа при която класификацията е с минимална грешка.

Задача 3. Създайте MLP класификатор в два класа като използвайте данните от файл „diabetes.mat“