در این تکلیف، شما باید یک ANN پرسپترون چندلایه (MLP) را پیاده‌سازی کنید و آن را با استفاده از الگوریتم Back Propagation (BP) آموزش دهید. پایگاه داده Normal قرار است برای آموزش طبقه‌بندی کننده و سپس آزمایش آن استفاده شود. نمونه‌های پایگاه داده Normal دوبعدی هستند و دو کلاس دارند. بنابراین، MLP دارای سه گره ورودی (x_1، x_2 و یک گره بایاس با مقدار ثابت ۱+) و دو گره خروجی مربوط به کلاس ۱ و کلاس ۲ خواهد بود. شما باید از چهار گره پنهان به‌اضافه یک گره بایاس (۱+) در لایه پنهان استفاده کنید. برای تابع انتقال گره‌های لایه پنهان و خروجی از یک تابع Sigmoid به شکل $f(x) = 1/(1+\exp(-x))$ استفاده کنید.

صفحه گسترده اکسل ارائه‌شده حاوی وزن‌های اولیه و داده‌های آموزشی و آزمایشی نرمال شده است که در ادامه توضیح داده شده است.

۱. برگه "وزن‌های اولیه" حاوی وزن‌های اتصال اولیه به‌طور تصادفی است. از این وزن‌ها برای مقداردهی اولیه الگوریتم BP استفاده کنید.

۲. تب "Training Set" شامل جفت‌های نمونهٔ آموزشی است که هرکدام شامل دو ویژگی به همراه مقادیر گره خروجی موردنظر برای این نمونه است. مقادیر x_1 و x_2 با مقادیری که قبلاً در پایگاه دادهٔ Normal داشتید مطابقت ندارند. دلیل آن این است که آن‌ها به‌صورت خطی تبدیل‌شده‌اند تا در محدوده ۰٫۲ تا ۰٫۸ قرار بگیرند تا از اشباع توابع سیگموئید در لایه پنهان جلوگیری شود. همچنین توجه داشته باشید که ترتیب نمونه‌ها به‌طور متناوب بین کلاس‌های ۱ و ۲ تغییر می‌کند. شما باید از این ترتیب برای آموزش استفاده کنید تا از سوگیری آموزش به یک کلاس خاص جلوگیری کنید. مقادیر موردنظر برای گره‌های خروجی به ترتیب (۰٫۰۵، ۰٫۹۵) و (۰٫۹۵، ۰٫۰۵) برای کلاس ۱ و کلاس ۲ تنظیم شده است.

۳. برگه "Testing Set" شامل نمونه‌های تست نرمال شده است. ترتیب تست مهم نیست.

این MLP را دقیقاً با استفاده از ۵۰۰ دوره (تکرار در مجموعه آموزشی) و نرخ یادگیری ۰٫۲ آموزش دهید. سپس وزن‌های به‌دست‌آمده در پایان دوره ۵۰۰ به‌عنوان وزنه‌های تمرین شده در نظر گرفته می‌شوند تا برای طبقه‌بندی مجموعه آزمایش استفاده شوند.

برای طبقه‌بندی یک نمونهٔ آزمایشی، آن را روی لایهٔ ورودی اعمال کنید و با استفاده از وزن‌های آموزش داده‌شده، مقدار مربوط به دو گرهٔ خروجی را بیابید. سپس این دو مقدار خروجی را مقایسه کنید و ببینید کدام‌یک بالاتر است. اگر مقدار اولین گره خروجی بیشتر باشد، نمونه به کلاس ۱ طبقه‌بندی می‌شود و بالعکس.

آمار زیر را ارائه دهید:

۱- مقادیر وزن‌های آموزش‌دیده با فرمت مشابه وزن‌های اولیه داده‌شده در فایل اکسل.

۲- ماتریس سردرگمی برای نتیجهٔ طبقه‌بندی.

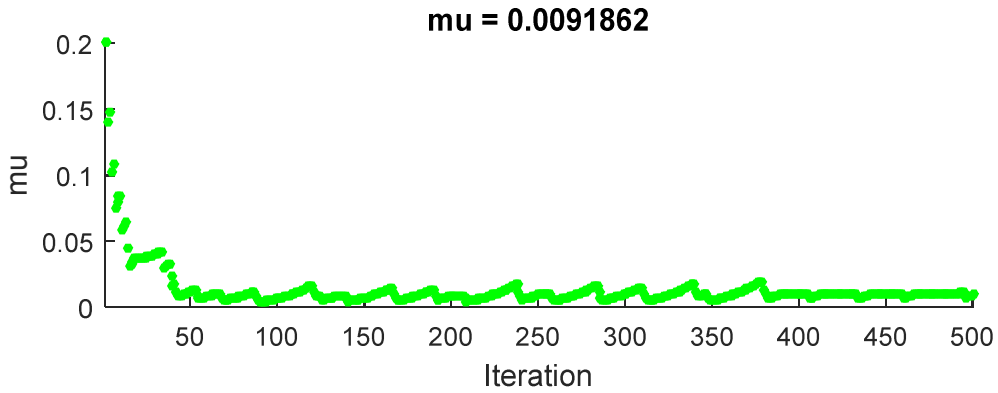
۳- فهرست نمونه‌هایی که به‌اشتباه طبقه‌بندی شده‌اند، به‌عنوان‌مثال شماره نمونهٔ ۳۵ از کلاس ۱، نمونهٔ شمارهٔ ۲۳۴ از کلاس ۲ و غیره.

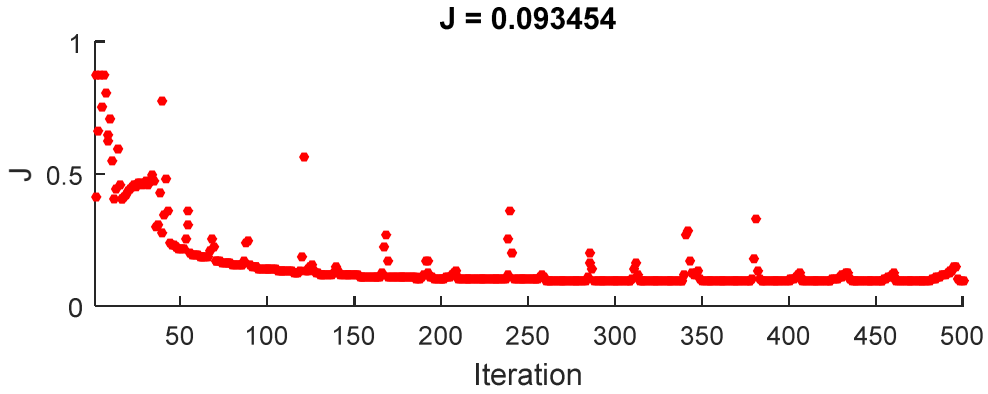
```
clc;
clear;
close all;

filename = 'training_data_set.csv';
delimiter = ',';
startRow = 2;
formatSpec = '%f%f%f%f%f%[^\n\r]';
fileID = fopen(filename, 'r');
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, 'HeaderLines', startRow-1,
'ReturnOnError', false);
fclose(fileID);
class_train = dataArray{:, 1};
x1 = dataArray{:, 2};
x2 = dataArray{:, 3};
```

```
target1 = dataArray{:, 4};
target2 = dataArray{:, 5};
clearvars filename delimiter startRow formatSpec fileID dataArray ans;
train_data = [x1, x2, class_train, target1, target2];

mu = 0.2;
ri = 1.05;
rd = 0.7;
c = 1.05;
J = 0;
Jtm1 = 0;
MaxIter = 500;

ntrain = numel(train_data(:, 1));

v1_ = zeros(ntrain, 4);
f1_ = zeros(ntrain, 4);

v2_ = zeros(ntrain, 2);
f2_ = zeros(ntrain, 2);

e_out = zeros(ntrain, 2);
delta_out = zeros(ntrain, 2);
rEi_d_rW2_ = zeros(ntrain, 10);

e_h1 = zeros(ntrain, 4);
delta_h1 = zeros(ntrain, 4);
rEi_d_rW1_ = zeros(ntrain, 12);

Ei = zeros(ntrain, 1);

                                                    %  -                       -
W1_ = [+0.248493, -0.397141, +0.027034, -0.115785; ... % | W1_11 W1_12 W1_13 W1_14 |
       +0.169035, +0.371143, -0.173352, -0.408767; ... % | W1_21 W1_22 W1_23 W1_24 |
       -0.357132, +0.360178, -0.474102, -0.048102];    % | W1_b1 W1_b2 W1_b3 W1_b4 |
                                                    %  -                       -
                                  %  -          -
W2_ = [+0.012515, -0.065186; ... % | W2_11 W2_12 |
       +0.448336, +0.306281; ... % | W2_21 W2_22 |
       +0.091863, -0.059266; ... % | W2_31 W2_32 |
       +0.143707, -0.045253; ... % | W2_41 W2_42 |
       -0.191953, +0.290301];    % | W2_b1 W2_b2 |
                                  %  -          -

for i = 1:MaxIter
    for j = 1:ntrain
        v1_(j, :) = [train_data(j, 1:2), +1] * W1_;
        f1_(j, :) = sigmoid(v1_(j, :));

        v2_(j, :) = [f1_(j, :), +1] * W2_;
        f2_(j, :) = sigmoid(v2_(j, :));

        e_out(j, :) = f2_(j, :) - train_data(j, 4:5);
        delta_out(j, :) = e_out(j, :) .* sigmoid_gradient(v2_(j, :));
        rEi_d_rW2_(j, :) = [delta_out(j, 1) .* [f1_(j, :), +1], delta_out(j, 2) .* [f1_(j, :),
+1]];

        e_h1(j, :) = delta_out(j, :) * W2_(1:4, :)';
        delta_h1(j, :) = e_h1(j, :) .* sigmoid_gradient(v1_(j, :));
```

```
        rEi_d_rW1_(j, :) = [delta_h1(j, 1) .* [train_data(j, 1:2), +1], delta_h1(j, 2) .*
[train_data(j, 1:2), +1], delta_h1(j, 3) .* [train_data(j, 1:2), +1], delta_h1(j, 4) .*
[train_data(j, 1:2), +1]];

        Ei(j) = sum(e_out(j, :).^2);
    end
    J = sum(Ei) / ntrain;
    Delta_W1_ = -mu .* sum(rEi_d_rW1_);
    Delta_W2_ = -mu .* sum(rEi_d_rW2_);
    W1_ = W1_ + [Delta_W1_(1, 1:3)', Delta_W1_(1, 4:6)', Delta_W1_(1, 7:9)', Delta_W1_(1,
10:12)'];
    W2_ = W2_ + [Delta_W2_(1, 1:5)', Delta_W2_(1, 6:10)'];

    subplot(2, 1, 1);
    hold on;
    plot(i, mu, 'og', 'MarkerSize', 3, 'MarkerFaceColor', 'g');
    xlabel('Iteration');
    ylabel('mu');
    title(['mu = ', num2str(mu)]);
    axis([1, MaxIter, 0, 0.2])

    subplot(2, 1, 2);
    hold on;
    plot(i, J, 'or', 'MarkerSize', 3, 'MarkerFaceColor', 'r');
    xlabel('Iteration');
    ylabel('J');
    title(['J = ', num2str(J)]);
    axis([1, MaxIter, 0, 1])
    pause(eps);

    if i == 1
        Jtm1 = J;
    end
    jt_d_jtm1 = J / Jtm1;
    if jt_d_jtm1 < 1
        mu = ri * mu;
    elseif jt_d_jtm1 > c
        mu = rd * mu;
    end
    Jtm1 = J;
end

c_train = zeros(ntrain, 1);

for i = 1:ntrain
    [~, c_train(i)] = max(f2_(i, :));
end

train_acc = sum(c_train == class_train) / ntrain;
disp(['Train Accuracy = ', num2str(train_acc)]);

filename = 'testing_data_set.csv';
delimiter = ',';
startRow = 2;
formatSpec = '%f%f%f%[^\n\r]';
fileID = fopen(filename, 'r');
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, 'HeaderLines', startRow-1,
'ReturnOnError', false);
fclose(fileID);
class_test = dataArray{:, 1};
x1_test = dataArray{:, 2};
```

```
x2_test = dataArray{:, 3};
clearvars filename delimiter startRow formatSpec fileID dataArray ans;

test_data = [x1_test, x2_test, class_test];
ntest = numel(test_data(:, 1));

vv1_ = zeros(ntest, 4);
ff1_ = zeros(ntest, 4);

vv2_ = zeros(ntest, 2);
ff2_ = zeros(ntest, 2);

c_test = zeros(ntest, 1);

for i = 1:ntest
    vv1_(i, :) = [test_data(i, 1:2), +1] * W1_;
    ff1_(i, :) = sigmoid(vv1_(i, :));

    vv2_(i, :) = [ff1_(i, :), +1] * W2_;
    ff2_(i, :) = sigmoid(vv2_(i, :));
end

for i = 1:ntest
    [~, c_test(i)] = max(ff2_(i, :));
end

test_acc = sum(c_test == class_test) / ntest;
disp(['Test Accuracy = ', num2str(test_acc)]);

filename = 'weights';
W1_Re = reshape(W1_', 1, []);
for i = 1:numel(W1_)
    xlswrite(filename, W1_Re(i), 'weights', ['C', num2str(i + 2)]);
end
W2_Re = reshape(W2_', 1, []);
for i = 1:numel(W2_)
    xlswrite(filename, W2_Re(i), 'weights', ['G', num2str(i + 2)]);
end

disp('Wrongly Classified Test Data');
disp('Sample no. x from class y');
disp([find(c_test(1:500) ~= class_test(1:500)), class_test(find(c_test(1:500) ~=
class_test(1:500)))]);
disp([find(c_test(501:1000) ~= class_test(501:1000)), class_test(find(c_test(501:1000) ~=
class_test(501:1000))+500)]);

Confusion_matrix{1, 1} = 'P\A';
Confusion_matrix{2, 1} = 'C1';
Confusion_matrix{3, 1} = 'C2';

Confusion_matrix{1, 2} = 'C1';
Confusion_matrix{1, 3} = 'C2';

Confusion_matrix{2, 2} = 500 - numel(find(c_test(1:500) ~= class_test(1:500)));
Confusion_matrix{2, 3} = numel(find(c_test(1:500) ~= class_test(1:500)));

Confusion_matrix{3, 2} = numel(find(c_test(501:1000) ~= class_test(501:1000)));
Confusion_matrix{3, 3} = 500 - numel(find(c_test(501:1000) ~= class_test(501:1000)));

display(Confusion_matrix);
```






```
Train Accuracy = 0.917
Test Accuracy = 0.931
Wrongly Classified Test Data
Sample no. x from class y
    11     1
    16     1
    25     1
    37     1
    56     1
    61     1
    66     1
    94     1
    98     1
   120     1
   123     1
   133     1
   134     1
   137     1
   148     1
   151     1
   154     1
   197     1
   211     1
   212     1
   214     1
   221     1
   229     1
   269     1
   282     1
   295     1
   298     1
```

```
   310     1
   318     1
   320     1
   347     1
   376     1
   377     1
   380     1
   398     1
   451     1
   465     1
   481     1
   486     1
   490     1
   491     1
   492     1

    10     2
    15     2
    29     2
    53     2
    71     2
    82     2
    98     2
   146     2
   191     2
   217     2
   241     2
   247     2
   287     2
   291     2
   333     2
   336     2
   346     2
   350     2
   352     2
   371     2
   386     2
   409     2
   417     2
   440     2
   463     2
   464     2
   483     2


Confusion_matrix =

    'P\A'    'C1'     'C2'
    'C1'     [458]    [ 42]
    'C2'     [ 27]    [473]
```