Problem5

Eta: the learning rate

H: the number of hidden units

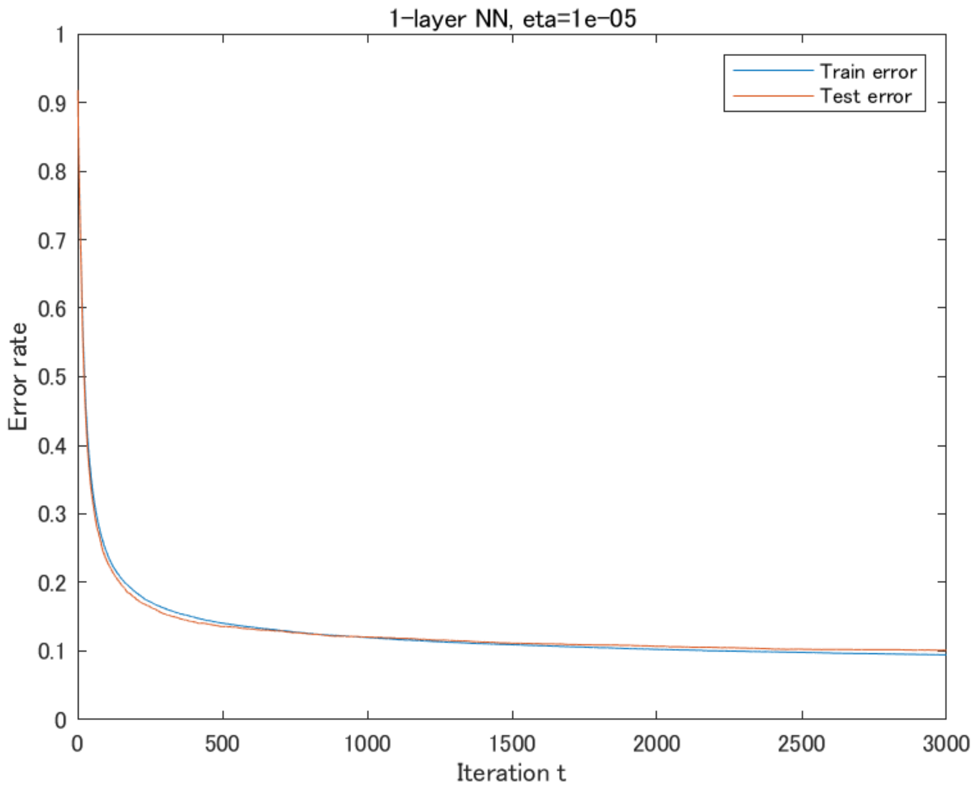
Lambda: the weight decay rate

(a) ii 1

1-layer NN, eta=1e-05

The final train error rate is 0.094583.

The final test error rate is 0.099596.



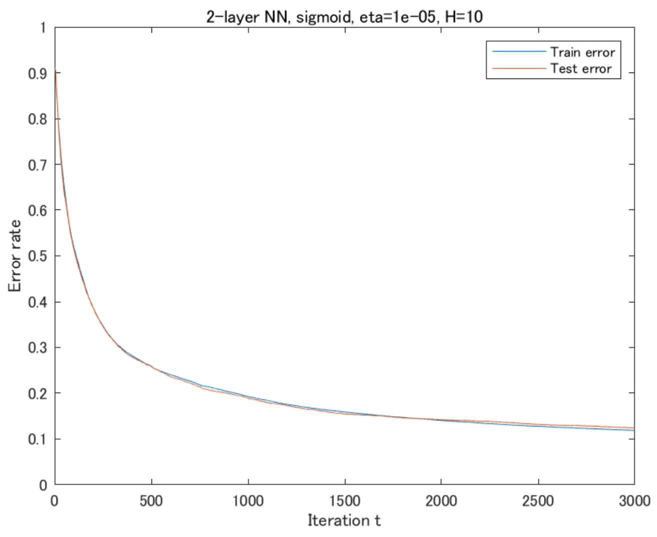
(b) ii 3

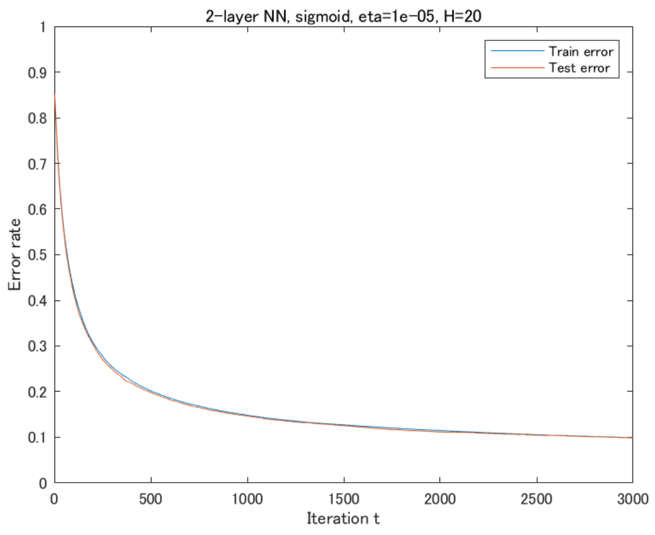
2-layer NN, sigmoid, eta=1e-05, H=10
The final train error rate is 0.1189.
The final test error rate is 0.1209.

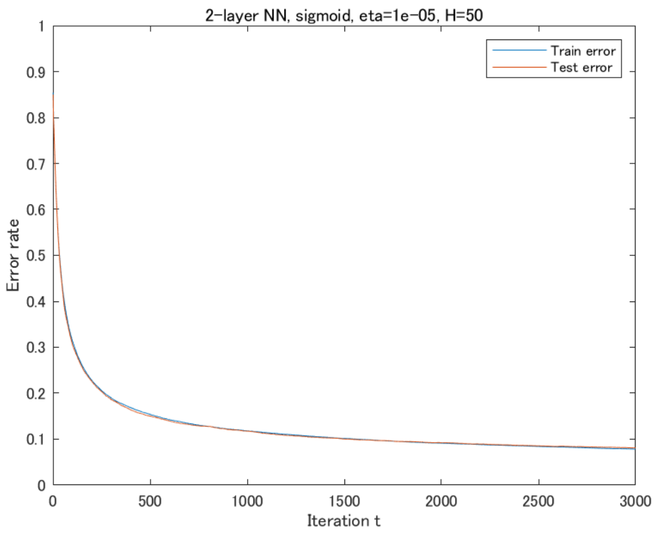
2-layer NN, sigmoid, eta=1e-05, H=20
The final train error rate is 0.099083.
The final test error rate is 0.099373.

2-layer NN, sigmoid, eta=1e-05, H=50
The final train error rate is 0.07845.
The final test error rate is 0.07855.

The larger H is, the lower the error rate is. This holds both for evaluation with training data and that with test data.


2-layer NN, sigmoid, eta=1e-05, H=10
Train error
Test error
Error rate
Iteration t
0
500
1000
1500
2000
2500
3000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


2-layer NN, sigmoid, eta=1e-05, H=20
Train error
Test error
Error rate
Iteration t
0
500
1000
1500
2000
2500
3000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


2-layer NN, sigmoid, eta=1e-05, H=50
Train error
Test error
Error rate
Iteration t
0
500
1000
1500
2000
2500
3000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(b) iii 6

2-layer NN, sigmoid, eta=2e-06, H=10, lambda=0.001
The final train error rate is 0.15502.
The final test error rate is 0.15778.

2-layer NN, sigmoid, eta=2e-06, H=20, lambda=0.001
The final train error rate is 0.12795.
The final test error rate is 0.12907.

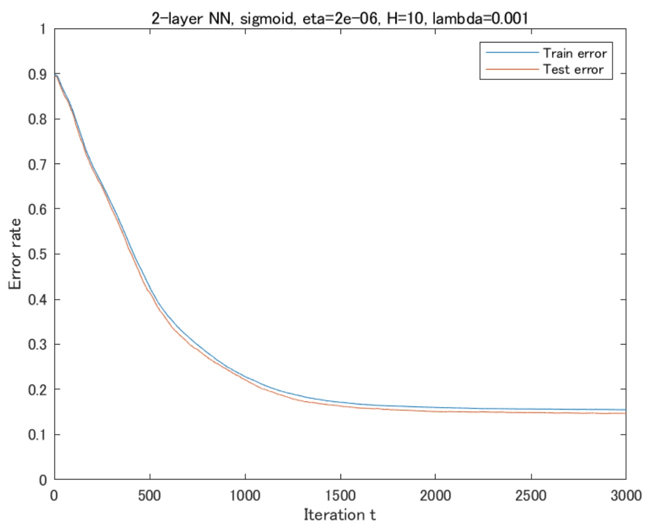
2-layer NN, sigmoid, eta=2e-06, H=50, lambda=0.001
The final train error rate is 0.12508.
The final test error rate is 0.12900.

2-layer NN, ReLu, eta=2e-06, H=10, lambda=0.001
The final train error rate is 0.094617.
The final test error rate is 0.094617.

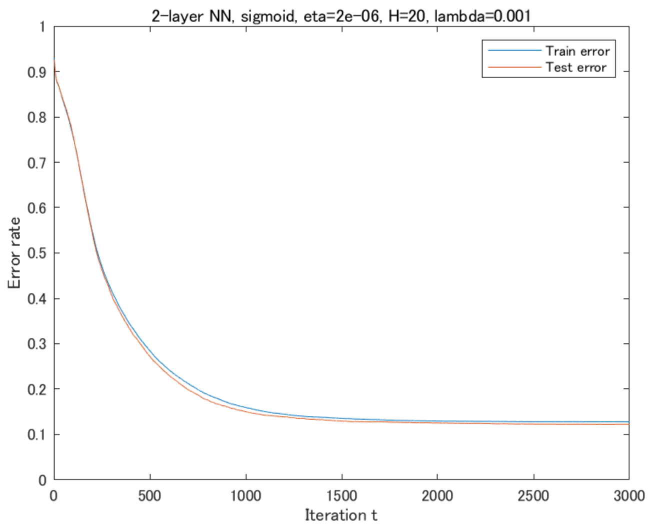
2-layer NN, ReLu, eta=2e-06, H=20, lambda=0.001
The final train error rate is 0.08655.
The final test error rate is 0.08885.

2-layer NN, ReLu, eta=2e-06, H=50, lambda=0.001
The final train error rate is 0.08105.
The final test error rate is 0.08235.

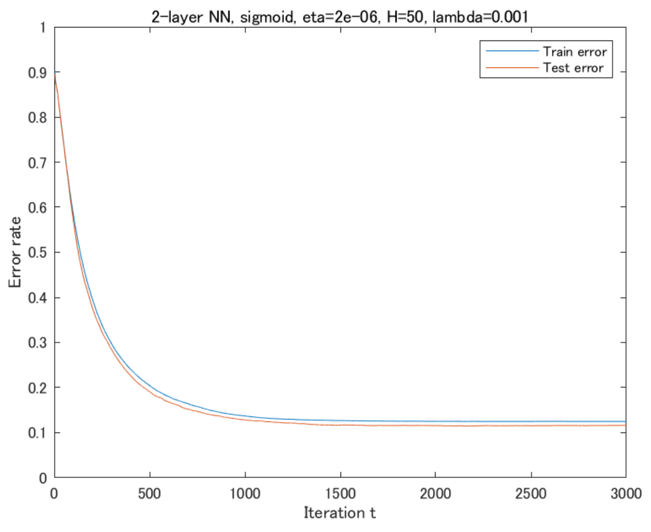
The error rates of ReLu cases are lower than those of Sigmoid cases with the same H. This holds both for evaluation with training data and that with test data.
Also, for both Sigmoid and ReLu cases, the larger H is, the lower the error rate is. This holds both for evaluation with training data and that with test data.
About the convergence speeds, the larger H is, the faster the convergence is. This holds both for Sigmoid and ReLu cases. The Models in RaLu cases converges faster than those in Sigmoid cases.



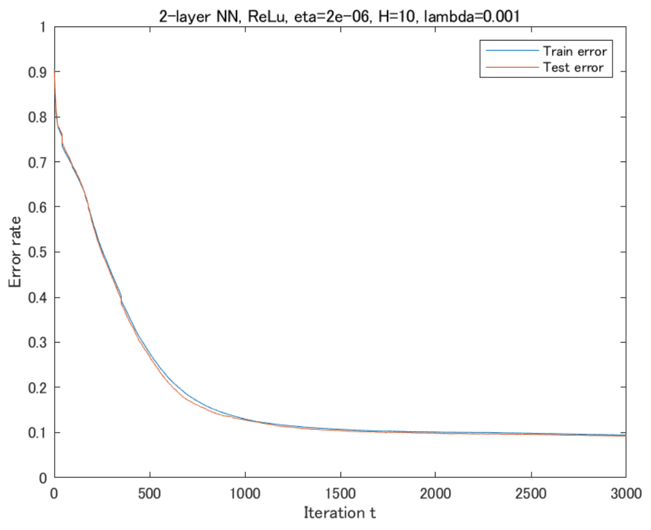
2-layer NN, sigmoid, eta=2e-06, H=10, lambda=0.001
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000

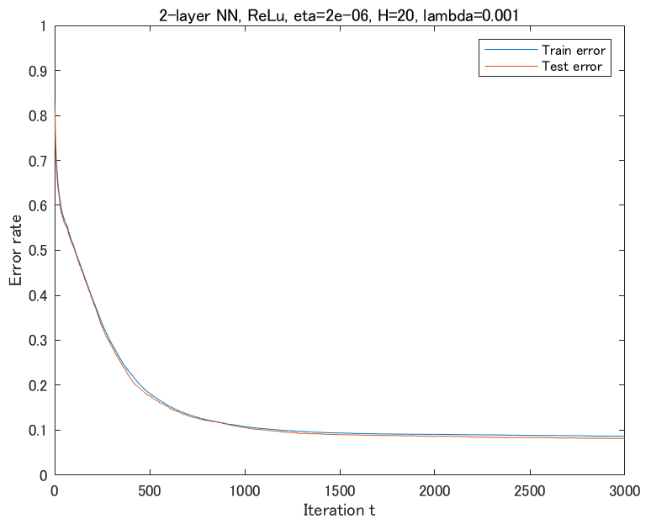


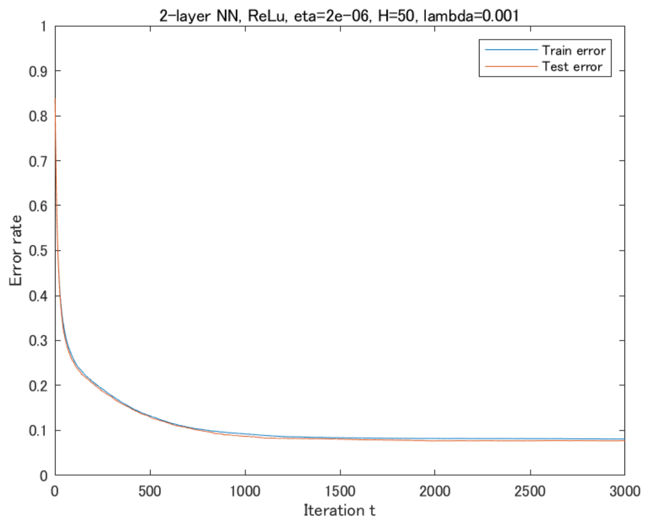
2-layer NN, sigmoid, eta=2e-06, H=20, lambda=0.001
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000



2-layer NN, sigmoid, eta=2e-06, H=50, lambda=0.001
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000


2-layer NN, ReLu, eta=2e-06, H=10, lambda=0.001
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000


2-layer NN, ReLu, eta=2e-06, H=20, lambda=0.001
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000


2-layer NN, ReLu, eta=2e-06, H=50, lambda=0.001
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000

(b) iv 6

2-layer NN, sigmoid, eta=2e-06, H=10, lambda=0.0001
The final train error rate is 0.17173.
The final test error rate is 0.17973.

2-layer NN, sigmoid, eta=2e-06, H=20, lambda=0.0001
The final train error rate is 0.13837.
The final test error rate is 0.14036.

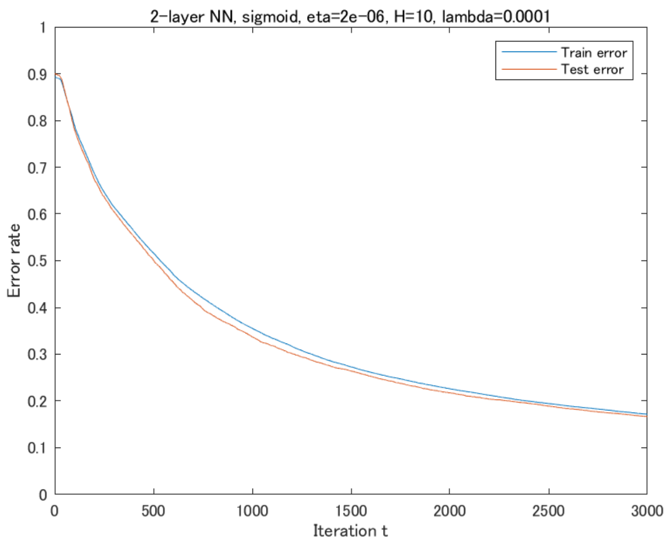
2-layer NN, sigmoid, eta=2e-06, H=50, lambda=0.0001
The final train error rate is 0.12268.
The final test error rate is 0.12778.

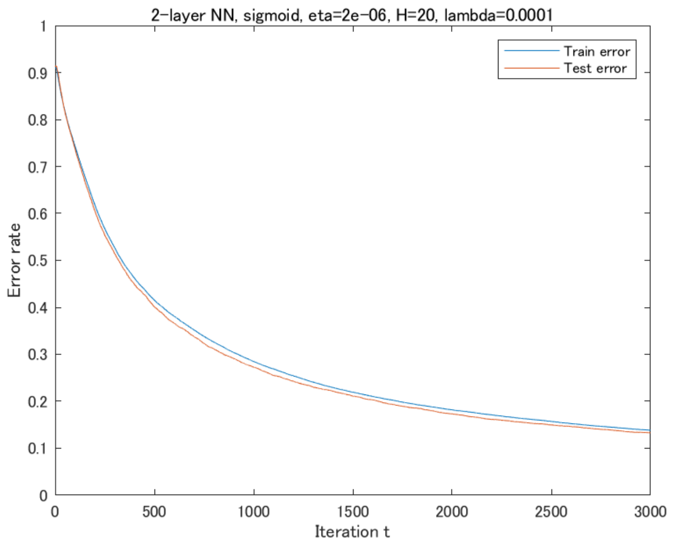
2-layer NN, ReLu, eta=2e-06, H=10, lambda=0.0001
The final train error rate is 0.12877.
The final test error rate is 0.12997.

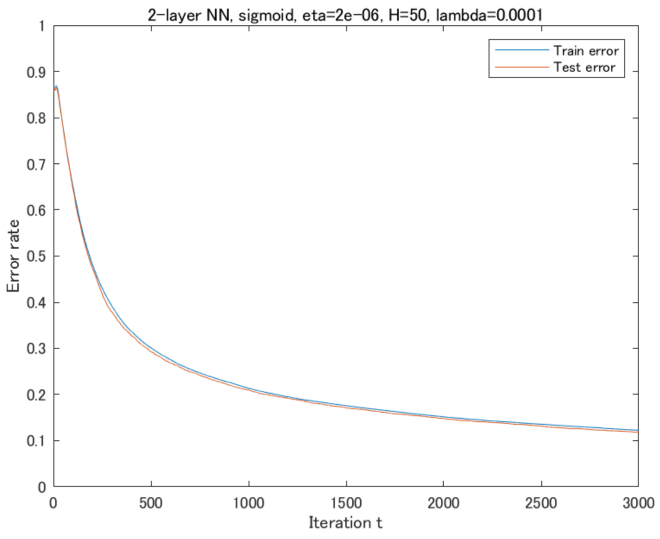
2-layer NN, ReLu, eta=2e-06, H=20, lambda=0.0001
The final train error rate is 0.119.
The final test error rate is 0.12033.

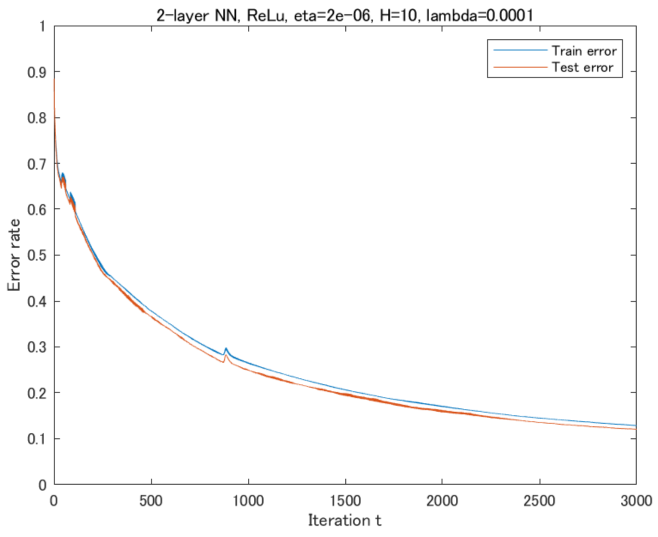
2-layer NN, ReLu, eta=2e-06, H=50, lambda=0.0001
The final train error rate is 0.091583.
The final test error rate is 0.091583.

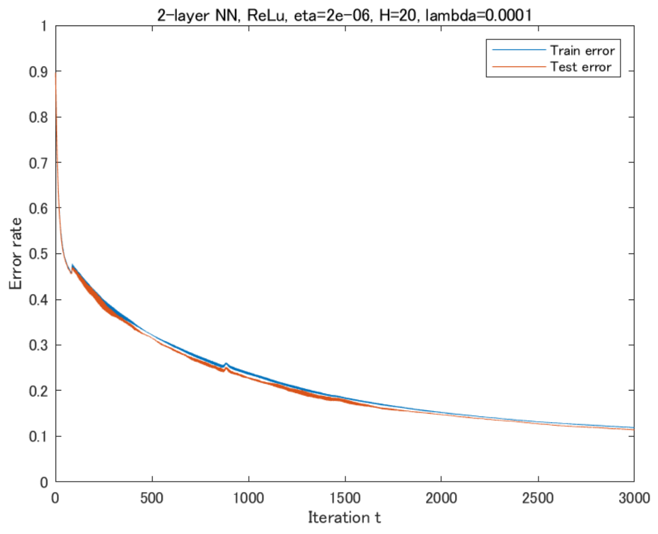
The regulation with large lambda can improve the performance.
In all cases with small lambda, the convergence speeds are slower than those in large lambda cases. Also, the error rates in small lambda cases are higher than those in large lambda cases.
In ReLu cases, the error rates are not stable.

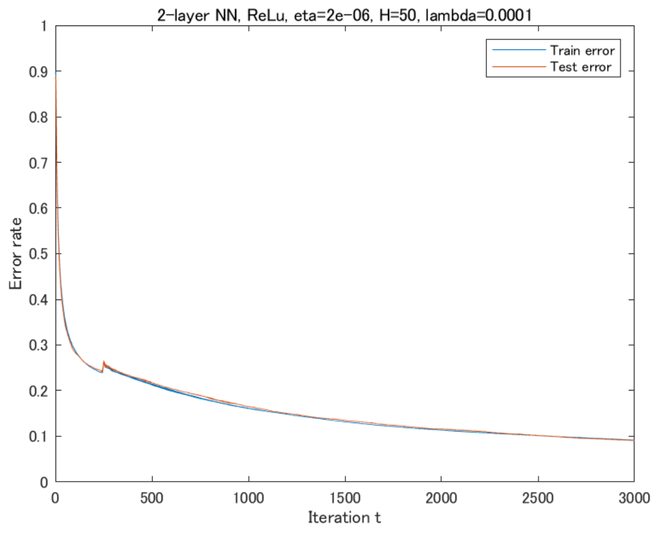

2-layer NN, sigmoid, eta=2e-06, H=10, lambda=0.0001
Train error
Test error
Error rate
Iteration t


2-layer NN, sigmoid, eta=2e-06, H=20, lambda=0.0001
Train error
Test error
Error rate
Iteration t


2-layer NN, sigmoid, eta=2e-06, H=50, lambda=0.0001
Train error
Test error
Error rate
Iteration t


2-layer NN, ReLu, eta=2e-06, H=10, lambda=0.0001
Train error
Test error
Error rate
Iteration t


2-layer NN, ReLu, eta=2e-06, H=20, lambda=0.0001
Train error
Test error
Error rate
Iteration t


2-layer NN, ReLu, eta=2e-06, H=50, lambda=0.0001
Train error
Test error
Error rate
Iteration t

(c) redo (a) ii 1

1-layer NN, SGD, eta=1e-05

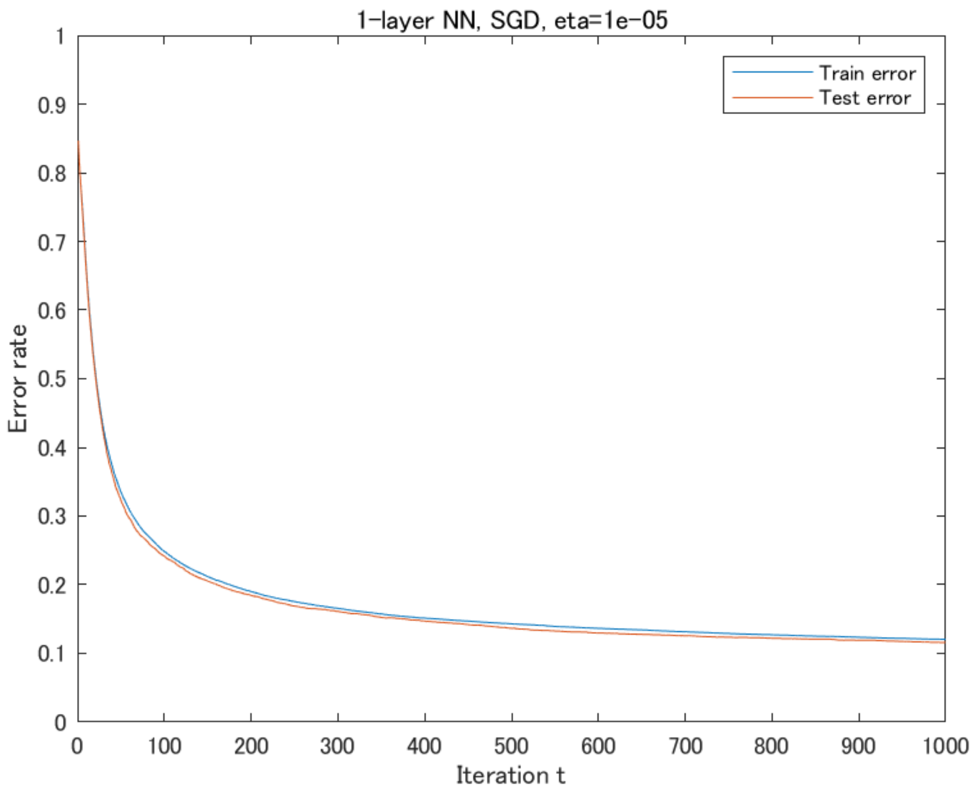
The final train error rate is 0.12055.

The minimum train error rate is 0.12055 at the 1000th step.

The final test error rate is 0.12635.

The minimum test error rate is 0.12635 at the 1000th step.

The error rate is slightly higher than that in the batch-approach case.



(c) redo (b) iii 6

2-layer NN, SGD, sigmoid, eta=0.01, H=10
The final train error rate is 0.20203.
The minimum train error rate is 0.14618 at the 9th step.
The final test error rate is 0.2244.
The minimum test error rate is 0.1433 at the 7th step.

2-layer NN, SGD, sigmoid, eta=0.01, H=20
The final train error rate is 0.051.
The minimum train error rate is 0.051 at the 1000th step.
The final test error rate is 0.0909.
The minimum test error rate is 0.0793 at the 61th step.

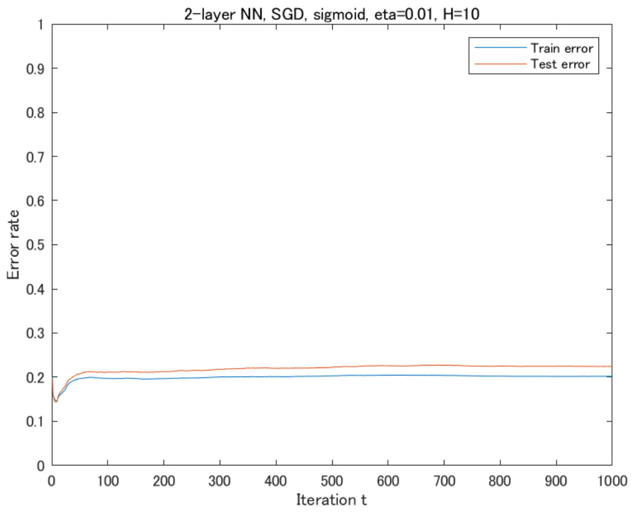
2-layer NN, SGD, sigmoid, eta=0.01, H=50
The final train error rate is 1.6667e-05.
The minimum train error rate is 1.6667e-05 at the 721th step.
The final test error rate is 0.0503.
The minimum test error rate is 0.0486 at the 581th step.

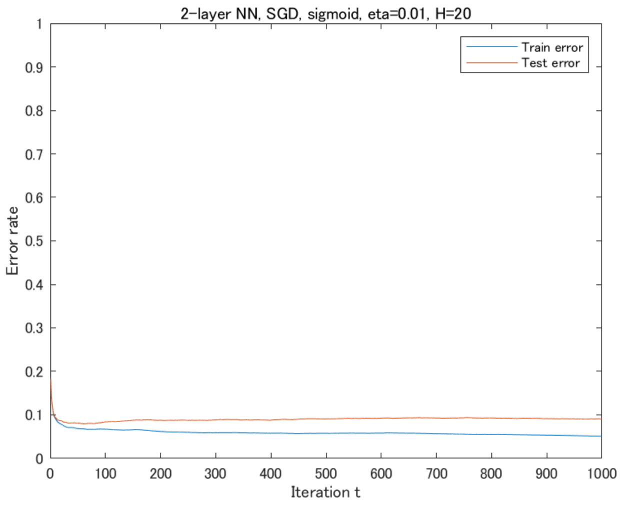
2-layer NN, SGD, ReLu, eta=0.002, H=10
The final train error rate is 0.88657.
The final test error rate is 0.88531.

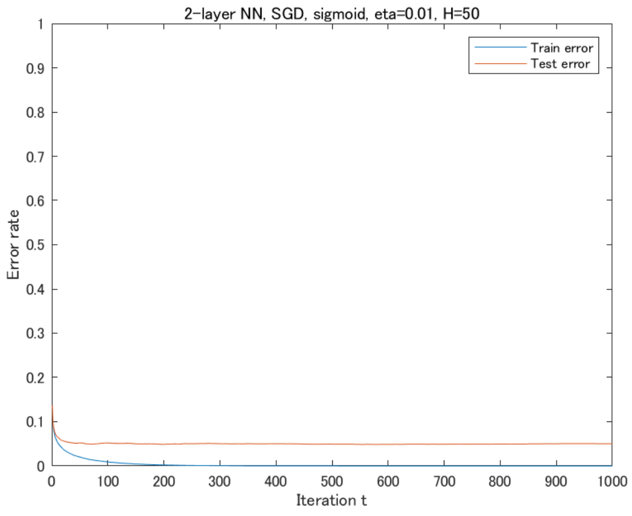
2-layer NN, SGD, ReLu, eta=0.002, H=20
The final train error rate is 0.88763.
The final test error rate is 0.88657.

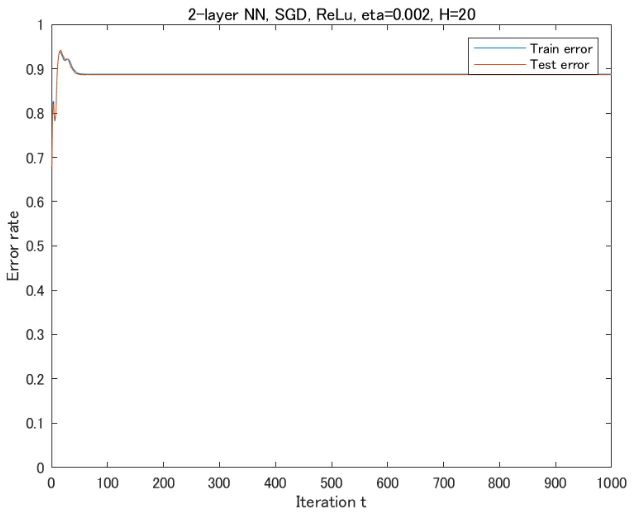
2-layer NN, SGD, ReLu, eta=0.002, H=50
The final train error rate is 0.34518.
The final test error rate is 0.39223.

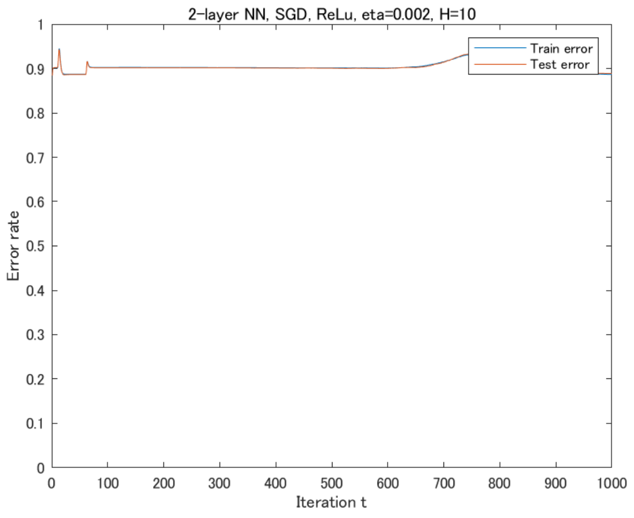
In the Sigmoid cases, the convergence speeds of SGD approach are faster than those of batch-approach. The error rate is the lowest in the case of H=50. The error rate in the case of H=10 increases after some steps.
In the ReLu cases, the algorithm does not work because it does not have weight-decay.

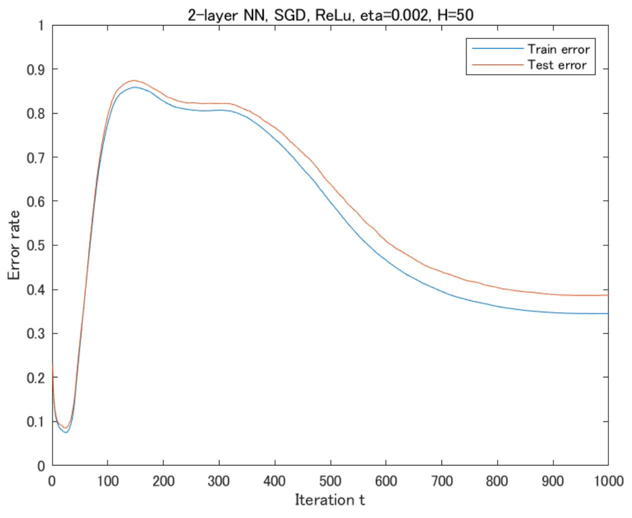

2-layer NN, SGD, sigmoid, eta=0.01, H=10
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800
900
1000


2-layer NN, SGD, sigmoid, eta=0.01, H=20
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800
900
1000


2-layer NN, SGD, sigmoid, eta=0.01, H=50
Train error
Test error
Error rate
Iteration t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800
900
1000


2-layer NN, SGD, ReLu, eta=0.002, H=20
Train error
Test error
Error rate
Iteration t


2-layer NN, SGD, ReLu, eta=0.002, H=10
Train error
Test error
Error rate
Iteration t


2-layer NN, SGD, ReLu, eta=0.002, H=50
Train error
Test error
Error rate
Iteration t

Code

```
% clc;
clear all;
trainImgFile = 'training set¥train-images-idx3-ubyte';
trainLblFile = 'training set¥train-labels-idx1-ubyte';
[trainImgs, trainLabels] = readMNIST(trainImgFile,trainLblFile,60000,0);

testImgFile = 'test set¥t10k-images-idx3-ubyte';
testLblFile = 'test set¥t10k-labels-idx1-ubyte';
[testImgs, testLabels] = readMNIST(testImgFile,testLblFile,10000,0);

% %tmp
% trainImgs = trainImgs(1:100,:);
% trainLabels = trainLabels(1:100,:);
% testImgs = testImgs(1:100,:);
% testLabels = testLabels(1:100,:);

t_train = zeros(size(trainImgs,1), 10);
for i = 1:size(trainImgs,1)
    t_train(i,trainLabels(i)+1) = 1;
end
x_train = [trainImgs ones(size(trainImgs,1),1)];
x_test = [testImgs ones(size(testImgs,1),1)];

% nn_1layer(30, 0.00001, x_train, t_train, trainLabels, x_test, testLabels);
% nn_2layer_sigmoid(30, 0.001, 10, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_sigmoid_srm(30, 0.001, 10, 0.0001, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu(30, 0.001, 10, 0.0001, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_1layer_sgd(30, 0.01, x_train, t_train, trainLabels, x_test, testLabels);
% nn_2layer_relu_sgd(30, 0.01, 10, x_train, t_train, trainLabels, x_test,
testLabels);
```

```
% nn_2layer_sigmoid_sgd(30, 0.01, 10, x_train, t_train, trainLabels, x_test,
testLabels);

% % (a) ii 1
% nn_1layer(3000, 0.00001, x_train, t_train, trainLabels, x_test, testLabels);
% % (b) ii 3
% nn_2layer_sigmoid(3000, 0.00001, 10, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_sigmoid(3000, 0.00001, 20, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_sigmoid(3000, 0.00001, 50, x_train, t_train, trainLabels, x_test,
testLabels);
% % (b) iii 6
% nn_2layer_sigmoid_srm(3000, 0.000002, 10, 0.001, x_train, t_train, trainLabels,
x_test, testLabels);
% nn_2layer_sigmoid_srm(3000, 0.000002, 20, 0.001, x_train, t_train, trainLabels,
x_test, testLabels);
% nn_2layer_sigmoid_srm(3000, 0.000002, 50, 0.001, x_train, t_train, trainLabels,
x_test, testLabels);
% nn_2layer_relu(3000, 0.000002, 10, 0.001, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu(3000, 0.000002, 20, 0.001, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu(3000, 0.000002, 50, 0.001, x_train, t_train, trainLabels, x_test,
testLabels);
% % (b) iv 6
% nn_2layer_sigmoid_srm(3000, 0.000002, 10, 0.0001, x_train, t_train, trainLabels,
x_test, testLabels);
% nn_2layer_sigmoid_srm(3000, 0.000002, 20, 0.0001, x_train, t_train, trainLabels,
x_test, testLabels);
% nn_2layer_sigmoid_srm(3000, 0.000002, 50, 0.0001, x_train, t_train, trainLabels,
x_test, testLabels);
% nn_2layer_relu(3000, 0.000002, 10, 0.0001, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu(3000, 0.000002, 20, 0.0001, x_train, t_train, trainLabels, x_test,
testLabels);
```

```
% nn_2layer_relu(3000, 0.000002, 50, 0.0001, x_train, t_train, trainLabels, x_test,
testLabels);
% % (c) redo (a) ii 1
% nn_1layer_sgd(1000, 0.00001, x_train, t_train, trainLabels, x_test, testLabels);
% % (c) redo (b) iii 6
nn_2layer_sigmoid_sgd(1000, 0.01, 10, x_train, t_train, trainLabels, x_test,
testLabels);
nn_2layer_sigmoid_sgd(1000, 0.01, 20, x_train, t_train, trainLabels, x_test,
testLabels);
nn_2layer_sigmoid_sgd(1000, 0.01, 50, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu_sgd(1000, 0.002, 10, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu_sgd(1000, 0.002, 20, x_train, t_train, trainLabels, x_test,
testLabels);
% nn_2layer_relu_sgd(1000, 0.002, 50, x_train, t_train, trainLabels, x_test,
testLabels);


function nn_1layer(nstep, eta, x_train, t_train, trainLabels, x_test, testLabels)
    w = normrnd(0, 1, [28*28+1,10]);
%     dE = zeros([28*28+1,10]);
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        a = x_train*w;
        exp_a = exp(a);
        y = exp_a ./ sum(exp_a, 2);
        dE = - x_train'*(t_train-y);
        w = w - eta*dE;
        trainError(step) = 1-accuracy(x_train, trainLabels, w);
        testError(step) = 1-accuracy(x_test, testLabels, w);
    end
    description = ['1-layer NN, eta=', num2str(eta)];
    showResult(trainError, testError, nstep, description);
end
```

```
function nn_2layer_sigmoid(nstep, eta, H, x_train, t_train, trainLabels, x_test,
testLabels)
    w1 = normrnd(0, 1, [28*28+1,H+1]);
    w2 = normrnd(0, 1, [H+1,10]);
%     dE1 = zeros([28*28+1, H+1]);
%     dE2 = zeros([H+1, 10]);
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        a1 = x_train*w1;
        sigmoid_a1 = 1./(1 + exp(-1.*(a1)));
        u = sigmoid_a1;
        a2 = u*w2;
        exp_a2 = exp(a2);
        y = exp_a2 ./ sum(exp_a2, 2);
        d_sigmoid = u.*(1-u);
        delta = (t_train-y)*w2';
        tmp = delta.*d_sigmoid;
        dE2 = - u'*(t_train-y);
        dE1 = - x_train'*tmp;
        w1 = w1 - eta*dE1;
        w2 = w2 - eta*dE2;
        trainError(step) = 1-accuracy2(x_train, trainLabels, w1, w2, 'sigmoid');
        testError(step) = 1-accuracy2(x_test, testLabels, w1, w2, 'sigmoid');
    end
    description = ['2-layer NN, sigmoid, eta=' num2str(eta) ', H=', int2str(H)];
    showResult(trainError, testError, nstep, description);
end

function nn_2layer_sigmoid_srm(nstep, eta, H, lambda, x_train, t_train, trainLabels,
x_test, testLabels)
    w1 = normrnd(0, 1, [28*28+1,H+1]);
    w2 = normrnd(0, 1, [H+1,10]);
%     dE1 = zeros([28*28+1, H+1]);
%     dE2 = zeros([H+1, 10]);
```

```
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        a1 = x_train*w1;
        sigmoid_a1 = 1./(1 + exp(-1.*(a1)));
        u = sigmoid_a1;
        a2 = u*w2;
        exp_a2 = exp(a2);
        y = exp_a2 ./ sum(exp_a2, 2);
        d_sigmoid = u.*(1-u);
        delta = (t_train-y)*w2';
        tmp = delta.*d_sigmoid;
        dE2 = - u'*(t_train-y);
        dE1 = - x_train'*tmp;
        w1 = w1 - eta*dE1 - 2*lambda*w1;
        w2 = w2 - eta*dE2 - 2*lambda*w2;
        trainError(step) = 1-accuracy2(x_train, trainLabels, w1, w2, 'sigmoid');
        testError(step) = 1-accuracy2(x_test, testLabels, w1, w2, 'sigmoid');
    end
    description = ['2-layer NN, sigmoid, eta=' num2str(eta) ', H=', int2str(H) ',
lambda=' num2str(lambda)];
    showResult(trainError, testError, nstep, description);
end

function nn_2layer_relu(nstep, eta, H, lambda, x_train, t_train, trainLabels, x_test,
testLabels)
    w1 = normrnd(0, 1, [28*28+1,H+1]);
    w2 = normrnd(0, 1, [H+1,10]);
%     dE1 = zeros([28*28+1, H+1]);
%     dE2 = zeros([H+1, 10]);
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        a1 = x_train*w1;
        u = max(a1,0);
        a2 = u*w2;
        exp_a2 = exp(a2);
        y = exp_a2 ./ sum(exp_a2, 2);
        d_relu = u>0;
        delta = (t_train-y)*w2';
        tmp = delta.*d_relu;
        dE2 = - u'*(t_train-y);
        dE1 = - x_train'*tmp;
        w1 = w1 - eta*dE1 - 2*lambda*w1;
        w2 = w2 - eta*dE2 - 2*lambda*w2;
        trainError(step) = 1-accuracy2(x_train, trainLabels, w1, w2, 'relu');
        testError(step) = 1-accuracy2(x_test, testLabels, w1, w2, 'relu');
    end
    description = ['2-layer NN, ReLu, eta=' num2str(eta) ', H=', int2str(H) ', 
lambda=' num2str(lambda)];
    showResult(trainError, testError, nstep, description);
end

function nn_1layer_sgd(nstep, eta, x_train, t_train, trainLabels, x_test, testLabels)
    w = normrnd(0, 1, [28*28+1,10]);
%     dE = zeros([28*28+1,10]);
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        for i = 1:size(x_train,1)
            t = zeros(10, 1);
            t(trainLabels(i)+1) = 1;
            x = x_train(i,:)';
            y = forward(x, w);
            dE = - x*(t-y)';
            w = w - eta*dE;
        end
        trainError(step) = 1-accuracy(x_train, trainLabels, w);
        testError(step) = 1-accuracy(x_test, testLabels, w);
    end
    description = ['1-layer NN, SGD, eta=' num2str(eta)];
    showResult(trainError, testError, nstep, description);
```

```
end

function nn_2layer_relu_sgd(nstep, eta, H, x_train, t_train, trainLabels, x_test,
testLabels)
    w1 = normrnd(0, 1, [28*28+1,H+1]);
    w2 = normrnd(0, 1, [H+1,10]);
%     dE1 = zeros([28*28+1, H+1]);
%     dE2 = zeros([H+1, 10]);
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        for i = 1:size(x_train,1)
            t = zeros(10, 1);
            t(trainLabels(i)+1) = 1;
            x = x_train(i,:)';
            [y,u] = forward2(x, w1, w2, H, 'relu');
            dE2 = - u*(t-y)';
            dE1 = - x*((t-y)'*w2'.*(u'>0));
            w1 = w1 - eta*dE1; % does not use weight decay
            w2 = w2 - eta*dE2;
        end
        trainError(step) = 1-accuracy2(x_train, trainLabels, w1, w2, 'relu');
        testError(step) = 1-accuracy2(x_test, testLabels, w1, w2, 'relu');
    end
    description = ['2-layer NN, SGD, ReLu, eta=' num2str(eta) ', H=', int2str(H)];
    showResult(trainError, testError, nstep, description);
end

function nn_2layer_sigmoid_sgd(nstep, eta, H, x_train, t_train, trainLabels, x_test,
testLabels)
    w1 = normrnd(0, 1, [28*28+1,H+1]);
    w2 = normrnd(0, 1, [H+1,10]);
%     dE1 = zeros([28*28+1, H+1]);
%     dE2 = zeros([H+1, 10]);
    trainError = zeros(nstep, 1);
    testError = zeros(nstep, 1);
    for step = 1:nstep
        for i = 1:size(x_train,1)
            t = zeros(10, 1);
            t(trainLabels(i)+1) = 1;
            x = x_train(i,:)';
            [y,u] = forward2(x, w1, w2, H, 'sigmoid');
            dE2 = - u*(t-y)';
            dE1 = - x*((t-y)'*w2'.*u'.*(1-u'));
            w1 = w1 - eta*dE1;
            w2 = w2 - eta*dE2;
        end
        trainError(step) = 1-accuracy2(x_train, trainLabels, w1, w2, 'sigmoid');
        testError(step) = 1-accuracy2(x_test, testLabels, w1, w2, 'sigmoid');
    end
    description = ['2-layer NN, SGD, sigmoid, eta=' num2str(eta) ', H=', int2str(H)];
    showResult(trainError, testError, nstep, description);
end

function showResult(trainError, testError, nstep, discription)
    figure;
    plot(1:nstep, trainError, 1:nstep, testError);
    legend('Train error', 'Test error');
    title(discription);
    xlabel('Iteration t');
    ylabel('Error rate');
    ylim([0 1]);
    saveas(gcf, [discription, '.png']);

    finalTrainError = trainError(end);
    minTrainError = min(trainError);
    argminTrainError = find(trainError==min(trainError), 1);
    finalTestError = testError(end);
    minTestError = min(testError);
    argminTestError = find(testError==min(testError), 1);
    disp(discription);
    disp(['The final train error rate is ' num2str(finalTrainError) '.']);
    disp(['The minimum train error rate is ' num2str(minTrainError) ' at the '
num2str(argminTrainError) 'th step.']);
    disp(['The final test error rate is ' num2str(finalTestError) '.']);
    disp(['The minimum test error rate is ' num2str(minTestError) ' at the '
num2str(argminTestError) 'th step.']);
end

% for SGD
function y = forward(x, w)
    a = zeros(10, 1);
    for j = 1:10
        a(j) = w(:,j)'*x;
    end
    y = softmax(a);
end

% for SGD
function [y,u] = forward2(x, w1, w2, H, actfunc)
    u = zeros(H+1, 1);
    a1 = zeros(H, 1);
    a2 = zeros(10, 1);
    for j = 1:H
        a1(j) = w1(:,j)'*x;
        if strcmp(actfunc, 'sigmoid')
            u(j) = 1/(1+exp(-a1(j)));
        elseif strcmp(actfunc, 'relu')
            u(j) = max(a1(j), 0);
        end
    end
    u(H+1) = 1;
    for j = 1:10
        a2(j) = w2(:,j)'*u;
    end
    y = softmax(a2);
end
```

```
function res = accuracy(x, labels, w)
    a = x*w;
    exp_a = exp(a);
    [~, result] = max(exp_a,[],2);
    res = sum(labels+1 == result) / numel(labels);
end

function res = accuracy2(x, labels, w1, w2, actfunc)
    a1 = x*w1;
    if strcmp(actfunc, 'sigmoid')
        sigmoid_a1 = 1./(1 + exp(-1.*(a1)));
        u = sigmoid_a1;
    elseif strcmp(actfunc, 'relu')
        u = max(a1, 0);
    end
    a2 = u*w2;
    exp_a2 = exp(a2);
    y = exp_a2 ./ sum(exp_a2, 2);
    [~, result] = max(y,[],2);
    res = sum(labels+1 == result) / numel(labels);
end
```