Table of Contents

Spring 2026 EELE-491

```
%Drew Currie Homework-3
```

Problem 1

```
close all; clear; clc;
%Create a bias range that allows for negative values.
%Clamp random values to -1 to 1
bias_min = -1;
bias_max = 1;
bias = (bias_max-bias_min).*rand+bias_min;
% Call SLP forward pass with random values.
% Weights are locked to be positive only because I think that is
% required?
SLP_forward_pass(rand(1,3), rand(3,3), bias)


ans =

     1     0     1
```

Problem 2

```
close all; clear; clc;

x = [0.05, 0.10];
W1 = [0.15, 0.25; 0.20, 0.30];
W2 = [0.40, 0.50; 0.45, 0.55];
b = [0.35, 0.60];

[y,h] = MLP_forward_pass(x, W1, W2, b)


y =

    0.7514    0.7729
```

h =

   0.5933   0.5969

Problem 3

```
close all; clear; clc;

x = [0.05, 0.10];
t = [0.01, 0.99];
W1 = [0.15, 0.25; 0.20, 0.30];
W2 = [0.40, 0.50; 0.45, 0.55];
b = [0.35, 0.60];
alpha = 0.5;

[W1_new, W2_new, b_new] = MLP_backprop(x,t,W1,W2,b,alpha);
```

Problem 4, run problem 3 200 times

```
close all; clear; clc;

x = [0.05, 0.10];
t = [0.01, 0.99];
W1 = [0.15, 0.25; 0.20, 0.30];
W2 = [0.40, 0.50; 0.45, 0.55];
b = [0.35, 0.60];
alpha = 0.5;

[W1_new, W2_new, b_new] = MLP_backprop(x,t,W1,W2,b,alpha);
error = zeros(1,201);
for i = 1:200
    [y,h] = MLP_forward_pass(x, W1_new, W2_new, b_new);
    error(i) = sum((0.5)*(t-y).^2);
    [W1_new, W2_new, b_new] = MLP_backprop(x, t, W1_new, W2_new, b_new,
alpha);

end
figure("name", "Problem 4, Error Plot")
plot(error);
title("Error over training iteraitons")
xlim([0,200])
```
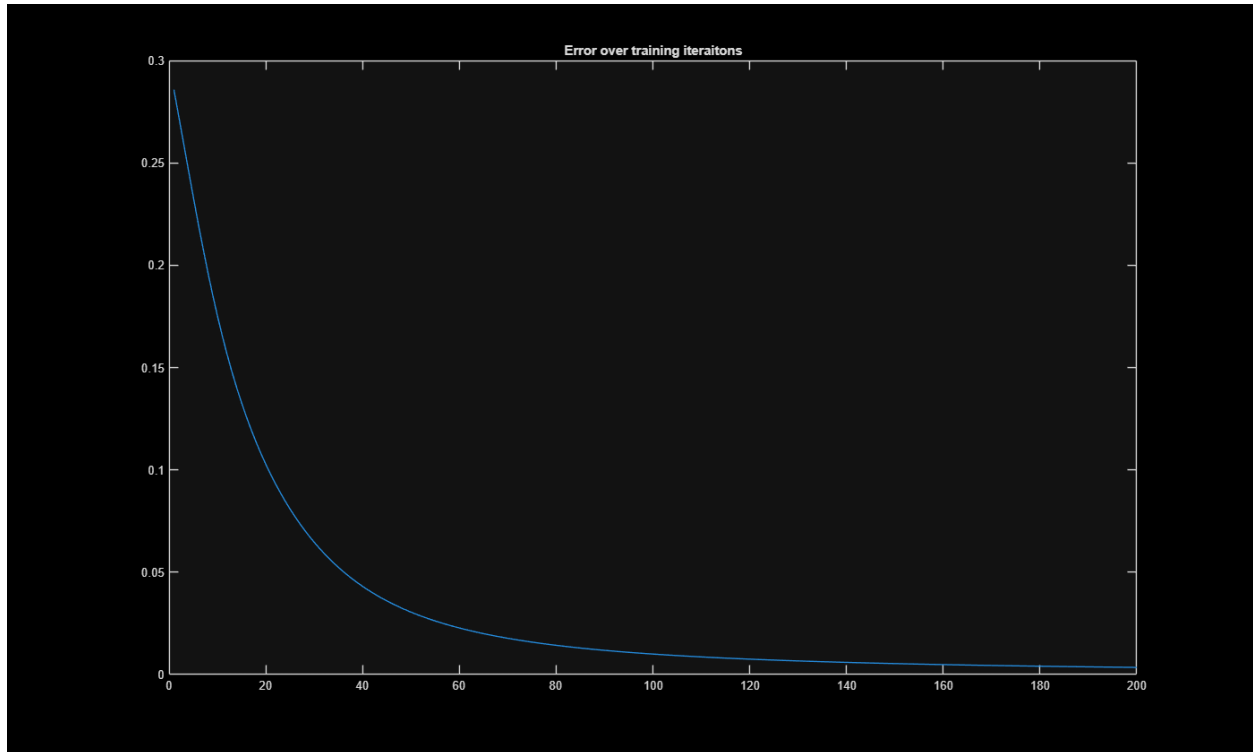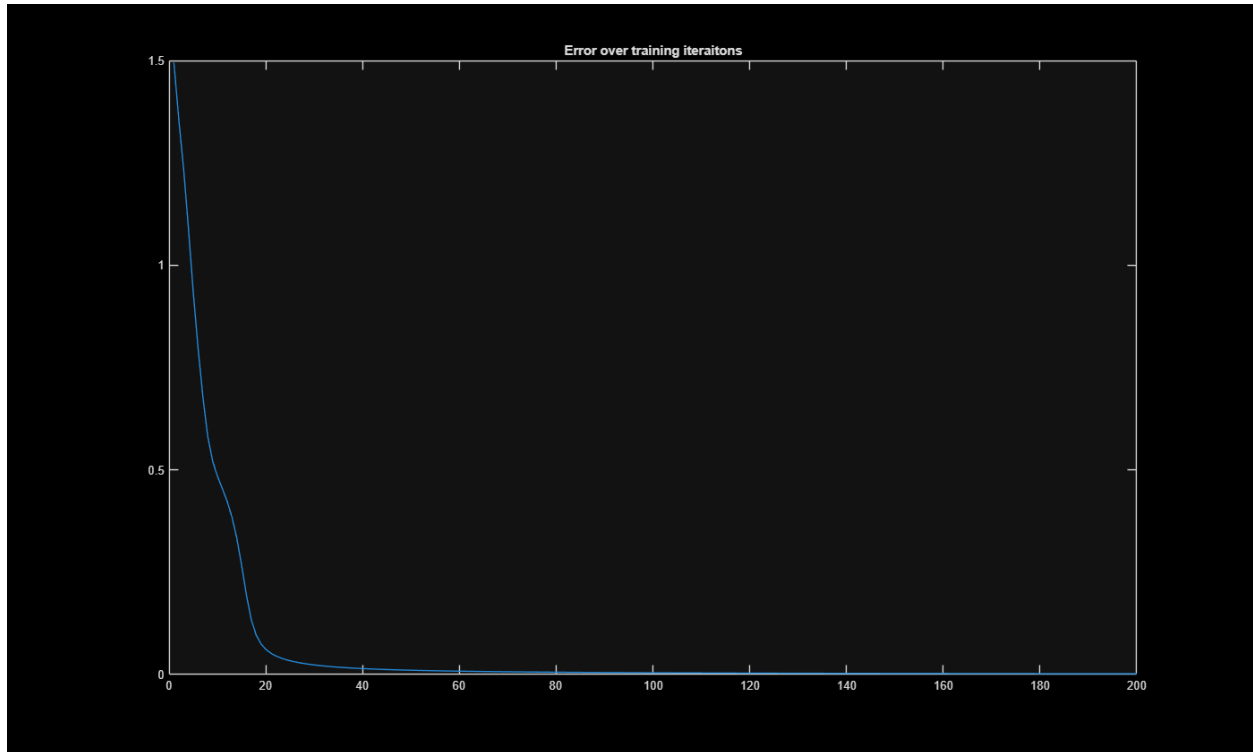
Problem 5, repeat problem 4 but with random initial values

```matlab
close all; clear; clc;

x = rand(1,8);
t = round(rand(1,6));
W1 = randn(8,12);
W2 = randn(12,6);
b = randn(1,2);
alpha = 0.5;

[W1_new, W2_new, b_new] = MLP_backprop(x,t,W1,W2,b,alpha);
error = zeros(1,201);
for i = 1:200
    [y,h] = MLP_forward_pass(x, W1_new, W2_new, b_new);
    error(i) = sum((0.5)*(t-y).^2);
    [W1_new, W2_new, b_new] = MLP_backprop(x, t, W1_new, W2_new, b_new, alpha);

end
figure("name", "Problem 5, Error Plot")
plot(error);
title("Error over training iteraitons")
xlim([0,200])
```

Problem 6, make it do something that isn't just theory.

```matlab
close all; clear; clc;
load("hw03_data/hw03_problem06_data.mat")
%Setup
sample = randi([1 length(meas)]);
x = meas(sample,:);
target = t(sample,:);
W1 = randn(4,16);
W2 = randn(16,3);
b = [1,1];
alpha = 0.125;

% Train
[W1_new, W2_new, b_new] = MLP_backprop(x,target,W1,W2,b,alpha);
error = zeros(1,1500);
for i = 1:1500
    sample = randi([1 length(meas)]);
    x = meas(sample,:);
    target = t(sample, :);
    [y,h] = MLP_forward_pass(x, W1_new, W2_new, b_new);
    error(i) = sum((0.5)*(target-y).^2);
    [W1_new, W2_new, b_new] = MLP_backprop(x, target, W1_new, W2_new, b_new,
alpha);

end
```
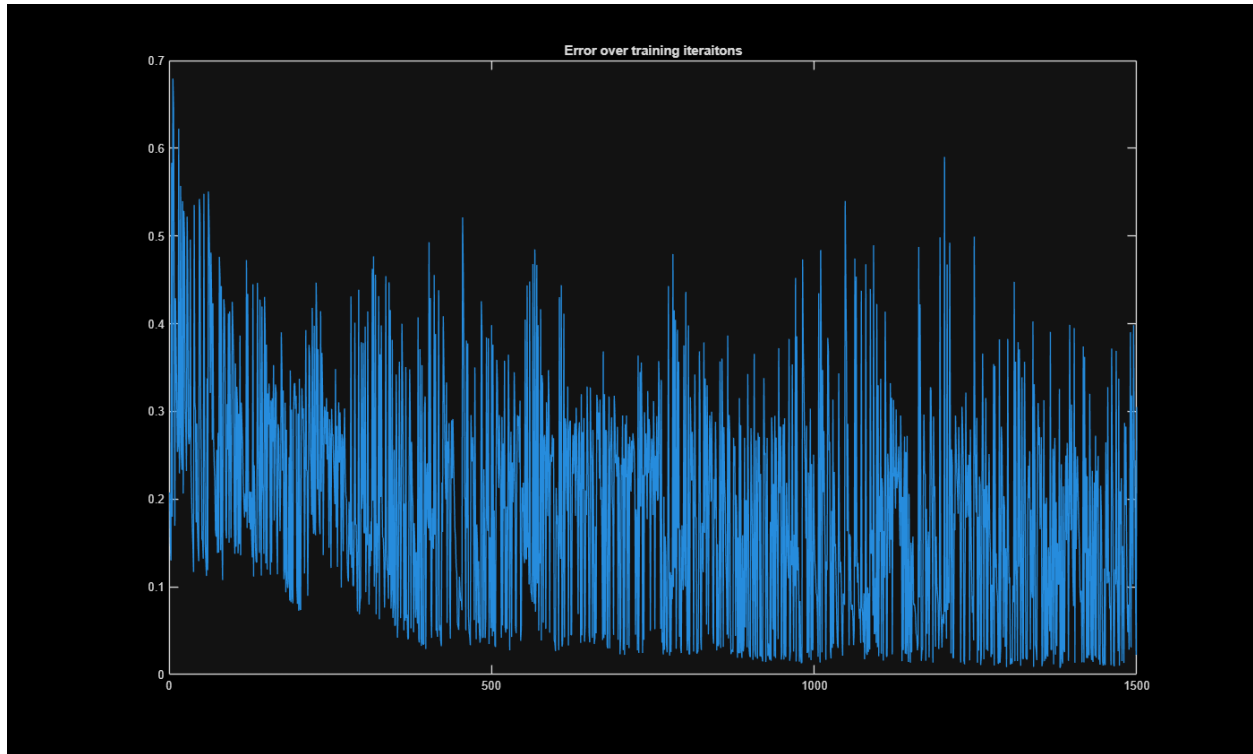
```matlab
W1_final = W1_new;
W2_final = W2_new;
b_final = b_new;
figure("name", "Problem 5, Error Plot")
plot(error);
title("Error over training iteraitons")

% Run on data set
predictedResults = cell(length(meas), 1);
for sample = 1:length(meas)
    [y,h] = MLP_forward_pass(meas(sample,:), W1_final, W2_final, b_final);
    [maxValue, maxIndex] = max(y);
    mask = y < y(maxIndex);
    y(maxIndex) = 1;
    y(mask) = 0;
    if maxIndex == 1
        predictedResults{sample} = "setosa";
    elseif maxIndex == 2
        predictedResults{sample} = "versicolor";
    elseif maxIndex == 3
        predictedResults{sample} = "virginica";
    end
end

C = confusionmat(categorical(species),
categorical(string(predictedResults)), 'Order', categorical(["setosa",
"versicolor", "virginica"]));
fprintf("Confusion matrix for flower identification deep-nueral network\n")
disp(C);

Confusion matrix for flower identification deep-nueral network
    50     0     0
     0    45     5
     0     0    50
```

Published with MATLAB® R2025b