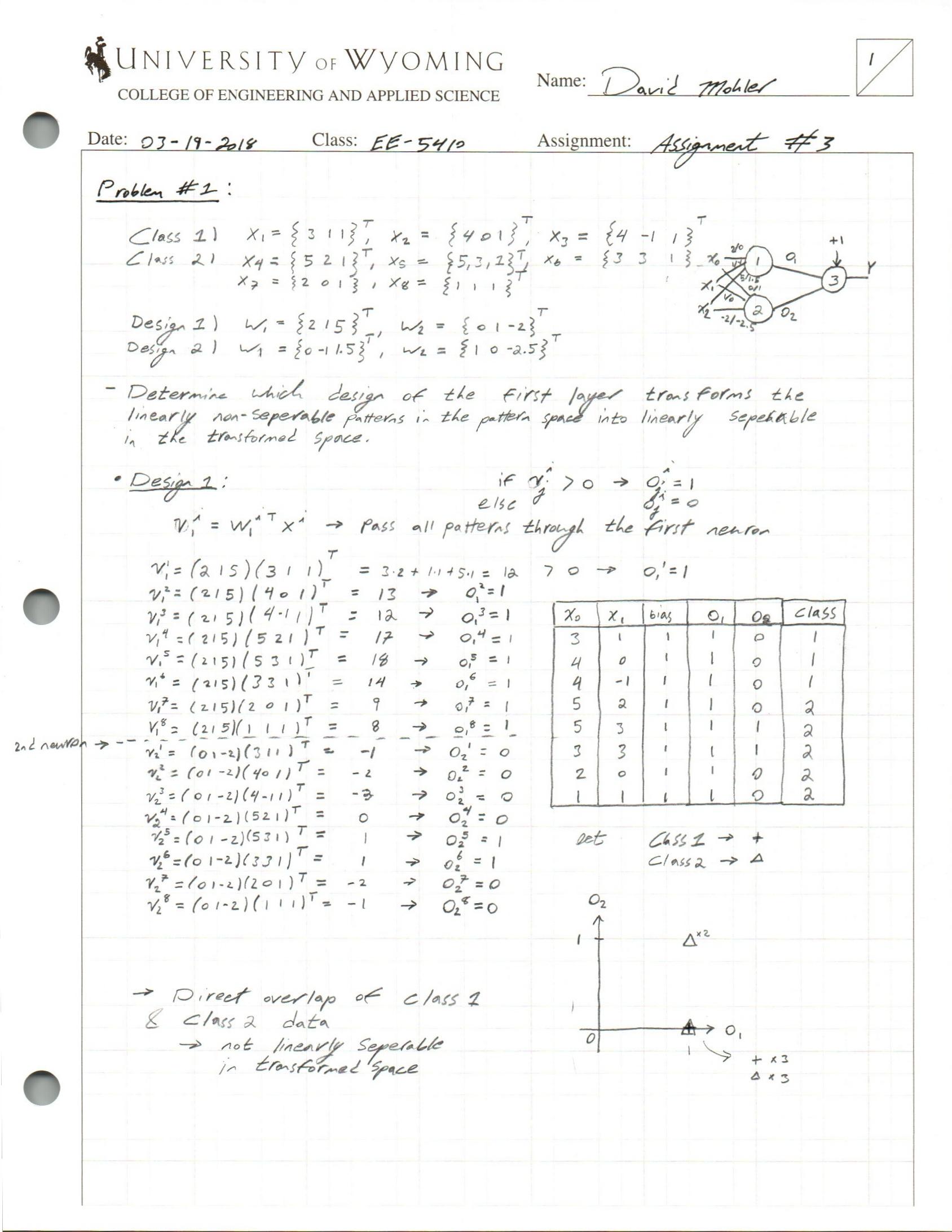
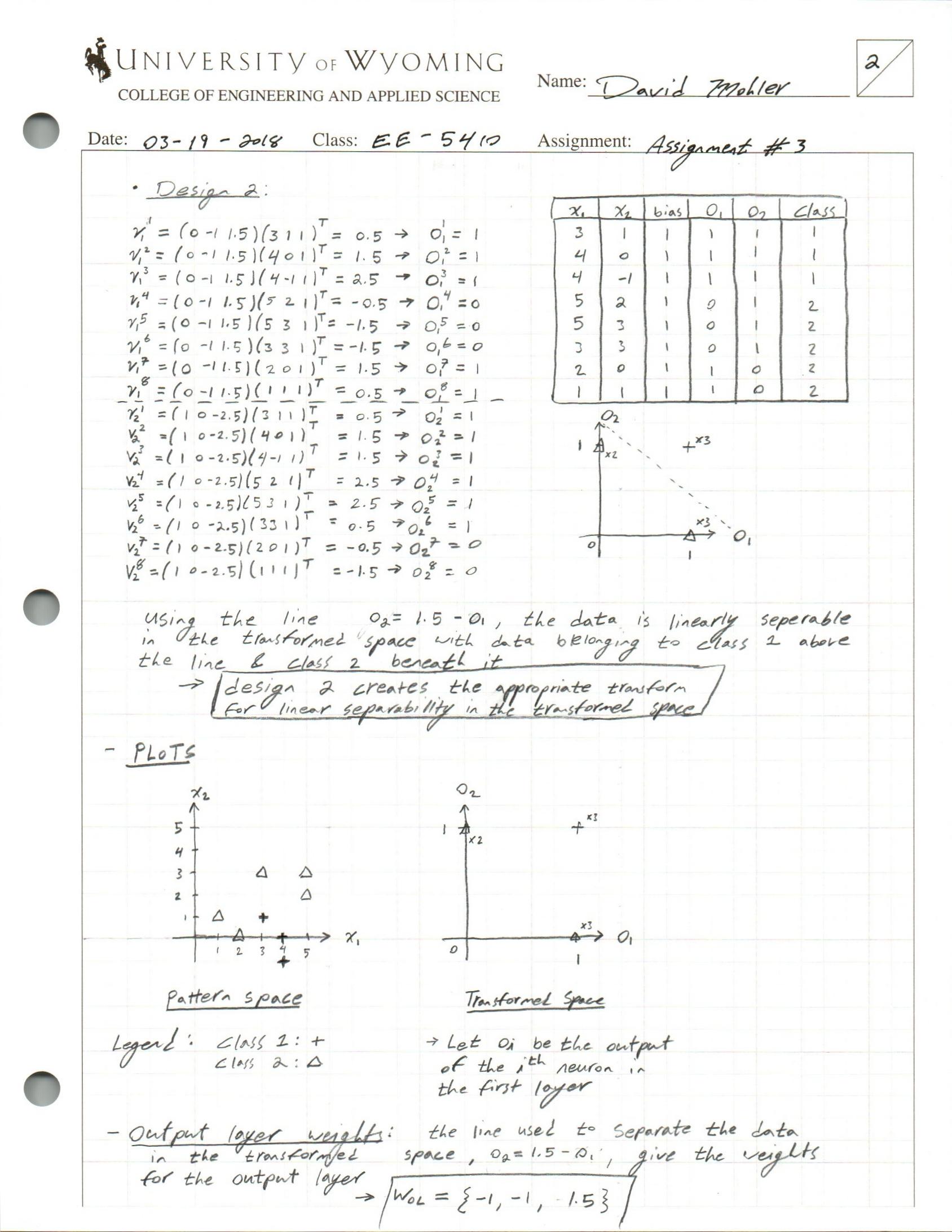
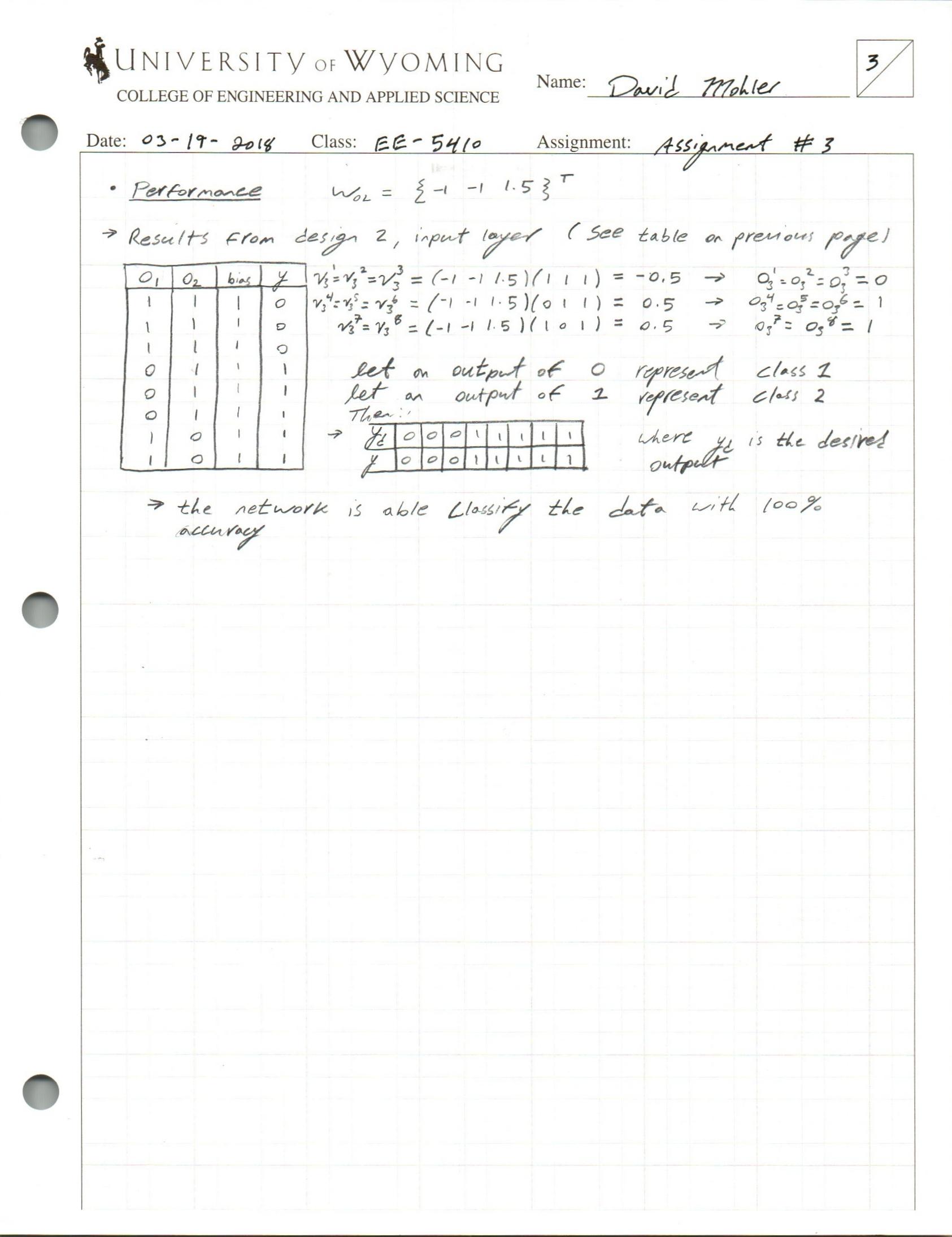
**David R Mohler**

**EE 5410: Neural Networks**

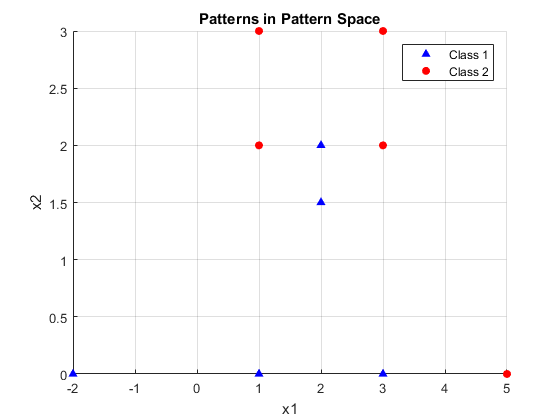
**Assignment #3**

**03-20-2018**

****

****

**Problem #2**

**\*NOTE: Most calculations in problem 2 are completed in the attached Matlab code. \***

|  |  |  |
| --- | --- | --- |
| **X1** | **X2** | **Class** |
| **1** | **3** | **2** |
| **3** | **3** | **2** |
| **1** | **2** | **2** |
| **2** | **2** | **1** |
| **3** | **2** | **2** |
| **2** | **1.5** | **1** |
| **-2** | **0** | **1** |
| **1** | **0** | **1** |
| **3** | **0** | **1** |
| **5** | **0** | **2** |

*Input Data Patterns and Associated Classes Input Patterns in the pattern space*

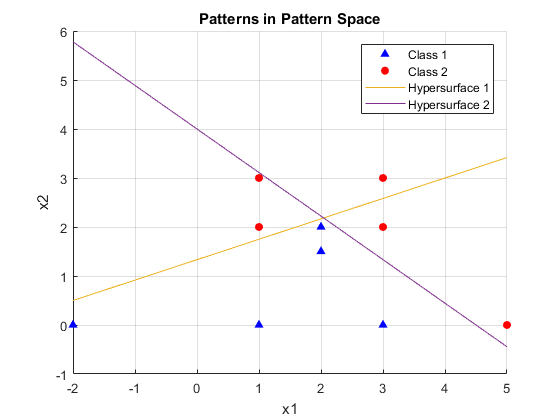
* From examination of the plotted data it is possible to classify the data appropriately with two linear hypersurfaces such that all the data in Class 1 will lie below the logical intersection of both curves.
* Using the parametric approach, I select a series of two candidate points in the pattern space from which to construct each respective decision boundary.
  + For the first hypersurface, let the analytically chosen points p1 and p2 be:

Using the standard equation of a line given two points, , we are able to obtain the equation of the first linear hypersurface as:

* + - .

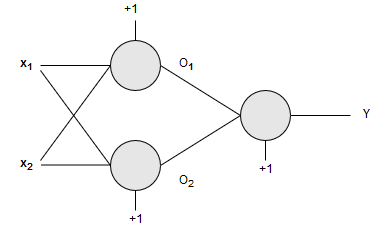
Similarly, it is possible to obtain the second hypersurface equation using the candidate points:

From these candidate points the second hypersurface equation can be represented as:

* ****The two hypersurface can then be superimposed in the pattern space as shown below:

*Hypersurfaces and Input Patterns in Pattern Space*

* These two surfaces dictate the number of neurons used in the first layer of the network. This gives us an over a three perceptron network, with the first two in the input layer and a single perceptron as the output layer to appropriately classify a two class problem.

******

*Neural Network Diagram*

* Using the analytically generated hyper surface we extract the coefficients from the hypersurface equations, these describe the respective weight vectors for each neuron in the input layer
* Using the input data augmented with a bias of positive 1, we next begin to compute the induced local field for each pattern presented to the network. The following equation can be used to calculate the Net, where is the net related to the pattern presented to the neuron. Following this the Heaviside activation function is applied to binarize the output of each neuron.

  + Heaviside activation function
* The results of passing all the patterns through the local induced field and subsequently the activation function are summarized in the table below: