# Final Exam: Math 472

Spring 2011

For this exam, you may use our class notes, anything on our class website, and any homework. You are expected to work on your own.
For each problem, create a published script file, and email me the result. If you email me them separately, please indicate it on the subject line- For example, `Final Exam, Problem 1`.

**Deadlines:** Turn in it by Tuesday, May 17 at 11PM.

## Problem 1

If you're solving the matrix equation $A\mathbf{x} = \mathbf{b}$, and $A$ is a wide matrix, what solution are you finding when you use the pseudo-inverse? There are two difficulties here: There is a combination of having no solution (if $A$ is not full rank), and having an infinite number of solutions (since $A$ has free variables).
Discuss what single solution you get by using the pseudo-inverse (hint: think about the four fundamental subspaces) and illustrate your answer using the matrix $A$ and vector $\mathbf{b}$ in `FinalExamData01.mat`

## Problem 2

Consider the function

$$F(x_1, x_2) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

We wish to approximate the minimum, and we are at the point $\mathbf{x} = (3/4, -5/4)$.

1. Use 10 iterations of gradient descent with an appropriately sized step-size (try some, use Matlab).
2. Perform 4 iterations of Newton's Method using the same initial point.
3. Which method ends with a better approximation?

## Problem 3

In this problem, we review linear classification using Widrow-Hoff. A good summary is on our class website from Feb 11. Download the data `FinalExamData02.mat`, which has two sets, `Mix1` and `Mix2`, each represent 200 points in $\mathbb{R}^4$, and they are samples from two classes. Write your own Matlab code (don't use `wid_hoff1`) to find $W$, $\mathbf{b}$ using online updates. Use 30% of the data for training, 70% for testing. Have Matlab output an appropriate confusion matrix.

## Problem 4

Create a $2-3-2$ feedforward neural network. Look at the script file from April 28th as a template, and modify the script to do backpropagation of error using the single data point:

$$\mathbf{x} = [2, 1]^T \qquad \mathbf{y} = [-1, 3]^T$$

The transfer function at the hidden layer should be the usual, $\sigma(x) = \frac{1}{1+e^{-x}}$.

## Problem 5

In this project, we use a neural net to perform classification on mushroom characteristics. That is, the neural net should input several characteristics of a mushroom, and output whether or not that mushroom is poisonous or edible.
The data consists of a matrix of 4062 entries each containing 22 characteristics of a mushroom (i.e., cap size, cap shape, stalk color, etc.). The range is either 0 or 1, depending on whether the mushroom is edible or not. Obtain this data by loading `mushroom_pre.mat`.
Project Questions:

1. Retain 25% points of data for training, 25% for testing and reserve the rest for validation of the network model. Use as few nodes in the hidden layer as necessary in order to get an error of 0 in classification. (You'll need to train the network a few times, and change the values that are used in `dividerand`.

2. Plot the results of the training. That is, first plot the network output for those mushrooms corresponding to class 1 as blue dots. Plot the network output for those mushrooms corresponding to class 0 as red dots. You should see a wide separation between red and blue.

3. Does preprocessing the data help? Rerun the network training, except now use enough dimensions to retain 90% of the energy. Use the projected data for the domain of your network (do it by hand and not using Matlab's built-in preprocessing functions).