Lab and Homework
Modeling

1. Finish the M-file below that computes the pseudo inverse of a matrix A using the SVD. Some comments about the code:

   - If a vector `s` contains the singular values of the matrix, then the normalized eigenvalues of the covariance matrix are:

     ```
     L=s.^2./(sum(s.^2));
     ```

   - Matlab has a command to compute the *cumulative sum* of a vector. For example, the command `cumsum([1 2 3 4])` returns the cumulative sum in a vector, `[1 3 6 10]`

   - Here's part of your code to get you started:

     ```
     function [pA, k]=pinv2(A,p)
     %function [pA, k]=pinv2(A,p) gives the pseudo inverse of the matrix A
     %  Input:  Matrix A and percentage p (between 0 and 1) for the rank
     %             Inputting p is optional- the default value is 0.98.
     %  Output:  Pseudo inverse as pA, rank is k.

     %Check the number of incoming arguments:
     if nargin<2
        p=0.98;  %default value
     end

     [m,n]=size(A);
     [U,S,V]=svd(A,'econ');
     s=diag(S);
     L=cumsum((s.^2)./(sum(s.^2)))

     for j=1:length(L)
         if L(j)>p
             k=j;    %Set the rank
             break %Break out of the loop
         end
     end

     %Now finish by computing the pseuo inverse:
     pA=
     ```

2. Write a script file that will perform gradient descent on the function:

$$f(x_1, x_2) = 5x_1^2 + x_2^2 + 4x_1x_2 - 14x_1 - 6x_2 + 20$$

The starting point should be $(0, 10)$, and set α to be a small (fixed) number. Stop when the difference between consecutive $x-$values is small- Say 10^{-8}.

Extra: Plot the trajectory that the x's take, along with some level curves of f.

3. Practice with the SVD: Write a Matlab script that illustrates two ways of obtaining a rank k approximation to the matrix A (assumes $A = USV^T$). In this case, the matrix A_k is obtained numerically by either a sum of rank 1 matrices:

$$A_k = S(1,1)\mathbf{u}_1\mathbf{v}_1^T + S(2,2)\mathbf{u}_2\mathbf{v}_2^T + \ldots S(k,k)\mathbf{u}_k\mathbf{v}_k^T$$

or all at once (Matlab notation):

```
Ak=U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
```

Using the matrix representing the clown image in Matlab, discuss what these rank 1 matrices are (intuitively), and illustrate your answer using images of these matrices[1]. What is the approximate rank of the clown matrix so that the clown is recognizable (by eye)?

4. Practice with Neural Networks (assigned earlier):

   (a) Code up the solution to the pattern classification problem on the new handout, pgs 13-14 (this was on the old handout, pg. 11). Be sure to use the examples in the text to help you!

   (b) The last problem used online training. Solve the same pattern classification using batch training. In place of Matlab's slash, use the pseudo-inverse code that you wrote in Exercise 1.

5. Be sure you can do the exercises in Appendix A (The Derivative), but only turn in the solutions to Exercise 7 and 8.

[1] You might look at the first few images and compare them to the last few.