MATLAB NOTES: Final Projects

1. Classification By Gender

- **Description of the Data:** To load the data, type `load MenAndWomen` The main data file is in the 3-d array `Photos`, and the gender (0 or 1) is in the vector `classes`.

  The data file is $162 \times 149 \times 26$ of unsigned integers, meaning that there are 26 photos, each 162×149. There are 13 men, 13 women. To work with these, we'll have to change the `uint8` format to `double` by typing: `Photos=double(Photos);`

  To view the images, for example the third one, type: `imagesc(Photos(:,:,3))` followed by `colormap(gray);` To make the image look better, you might also use `axis square` and `axis off`
- **Conversions:** To make the data set into a regular matrix (and have each photo as a column), you can use the command:

  ```
  X=reshape(Photos,162*149,26);  %X is 24138 x 26
  ```

  You might verify that you can undo this by typing:

  ```
  Y=reshape(X,[162,149,26]);     %Y is 162 x 149 x 26
  ```

  Once you've verified that Y is the same as `Photos`, be sure and delete Y to save space: `clear Y`
- To separate into genders, you can use two vectors, one for boys and one for girls:

  ```
  Boys=find(classes==1);
  Girls=find(classes==0);
  ```

  so that `X(:,Boys)` will contain all the photos for the males, `X(:,Girls)` will contain all the photos for the females.
- To get multiple graphs in the same figure, use the `subplot` command. For example, we could put 4 images in the same figure:

  ```
  subplot(2,2,1)
  imagesc(Photos(:,:,1));
  subplot(2,2,2)
  imagesc(Photos(:,:,2));
  subplot(2,2,3)
  imagesc(Photos(:,:,3));
  subplot(2,2,4)
  imagesc(Photos(:,:,4));
  ```
- To obtain a random sampling of 20 photos for training, we could write:

  ```
  idx=randperm(26);  %idx is a vector with 1-26 in random order
  TrainingPhotos=X(:,idx(1:20));
  TrainingClasses=classes(idx(1:20));
  ```

2. heart1

(1) To load the data, download it from the class website and in Matlab, type `load heart1` There will be two data sets, X which is 91×6 represents the 91 patients, each with 6 measurements. The data set in T (targets) are 1 or 0.

(2) You should try several networks, with different numbers of neurons in the hidden layer and use a couple of different optimization methods (for example, the default is `trainlm` for Levenberg-Marquardt, `traingdx` is gradient descent, and `trainrp` is for "resilient backpropagation"). For each, keep track of relative speed of the algorithm versus ending error.

(3) To visualize the results, you might plot T with the values given by the network. Also check the values of the weights to see if there are certain input parameters that seem to matter more than others (you're looking for the sizes of the weight values).