Homework Set 9

Problem 1 (MATLAB neural network toolbox)
This problem helps you get familiar with MATLAB neural network (NN) toolbox. Suppose you are using a DAQ to measure a voltage on a thermal couple (TC) over a period of time. After the data collection, you want to predict the TC voltage over time. Assume you collect the voltage for a period of 2 seconds, with a sampling rate of 0.01 Hz. Create the input time in MATLAB by typing

```
>>X = 0:.01:2;
```

Create the TC voltage as sin function plus some noise, i.e.,

```
>>Y = sin(2*pi*X)+.1*rand(size(X));
```

Initialize a MLP NN with 1 input, 3 neurons in the hidden layer, and 1 neuron in the output layer.

```
>>net = newff([0 2],[3 1],{'tansig' 'purelin'});
```

Here the `tansig` function uses hyperbolic tangent as the transition function instead of a hard threshold, and `purelin` is a linear relationship outputs a linear function of its inputs. See MATLAB help for more information about `tansig` and `purelin`. Assign the training parameters by typing

```
>>net.trainParam.epochs = 2000; %maximum training epochs
>>net.trainParam.goal = 0.001;  %goal for the error function
```

Train the network using the given input and training data by following

```
>>net = train( net, X, Y);
```

Simulate the network output given another set of test data

```
>>simY = sim(net, X);
```

In how many epochs your NN converges? Calculate the root-mean-squared error of simulation result between true output and predicted output.

Problem 2 (Convolutional neural network)
This problem helps you to get familiar with the CNN toolbox. CNN is a popular classification tool. In this problem, we will practice using the tools for recognizing hand written digits. Download the file "08HW2_CNN.rar" from the course webpage and unpack it to your working directory. This toolbox contains several functions (in \CNN\CNN\) that we will use.

Use the hand-written figure dataset "06HW1_digit.mat" from homework 6. Again, the dataset contains 500 training samples and 100 test samples for each digit. In this dataset, each image is 28×28 (784 pixels) in size and 256 in grayscale. Load this file into your MATLAB workspace.

Before the development of a CNN classifier, you will need to arrange the input data into appropriate format. For the CNN toolbox, the input training data should be put into one variable `train_x`. In

our case, this variable should include all the 5,000 images from various digits, and should be 28×28×5000 in dimension. Use concatenation and reshape operations to create the training data variable. Remember to normalize your data before training. The labels of the digit images for training should be stored in one variable `train_y`. This variable only contains values of 0 and 1 and is 10×5000 in dimension. In each row, there exists only one entry with value 1. This entry corresponds to the label (digit) of the image.

Next, you will need to assign the parameters of the CNN classifier. The parameters include the layer arrangements, layer types, feature map numbers, and kernel sizes. See the MATALB code below for example. You will also need to provide the training options, including learning rate, batch size, and number of epochs.

```
cnn.layers = {
    struct('type', 'i')                                      %input layer
    struct('type', 'c', 'outputmaps', 6, 'kernelsize', 5)    %convolution layer
    struct('type', 's', 'scale', 2)                          %sub sampling layer
    struct('type', 'c', 'outputmaps', 12, 'kernelsize', 5)   %convolution layer
    struct('type', 's', 'scale', 2)                          %subsampling layer
};

opts.alpha = 1;
opts.batchsize = 50;
opts.numepochs = 1000;
```

To develop a CNN classifier, you will need to use the functions `cnnsetup` and `cnntrain`. See the MATALB code below for example.