**CS4981 Week 2 Lab**

Signal Representation Option

**Due Date**

Wednesday, March 23 at 11:59pm. Submit your file or a link to your file/video via Canvas.

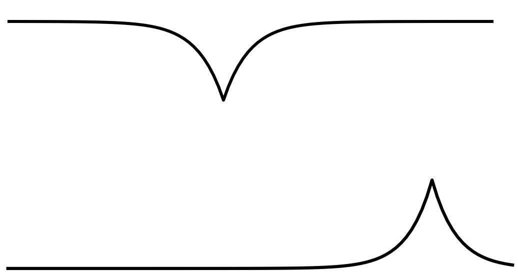
**Problem Description**

In this lab, you are to classify 1-dimensional signals as representing a "hill" or a "valley".

We will be using the [UCI Machine Learning Repository: Hill-Valley Data Set], and I will provide several versions of this data to assist you in the various approaches you may choose to classify the signals.

The dataset consists of 1212 signals of "clean" data, in which each signal is smooth and clearly shows a hill or valley. Each signal consists of 100 points.

Examples of the signals when plotted:



For additional challenge, the dataset also provides 1212 signals of "noisy" data, where the hill or valley can still be discerned by the human eye, but the classification is more difficult.

You may use the original text files from the UCI site, the MATLAB files I have created containing the signals and labels in various forms, or the image files I have created containing plots of the signals. A detailed description of the available formats will be included at the end of this document. You may use any computational environment you wish; MATLAB is not required.

**Assignment**

Use any automated method to classify the signals. Your method may include machine learning, but it is not required. For example, you could:

- Come up with a heuristic calculation based on the data (for example, I might be tempted to use numerical differentiation to classify the signals, but that is hard to interpret with the noisy data)

- Use shallow or deep neural networks to classify the 1-D signals, either in the time domain, or convert them to the frequency domain before classification[1]
- Use deep networks to classify plot images (2-D versions of the signals)

*You will not be graded on how successful you are in classifying the signals.* You will be graded on your description of your approach to the problem and the analysis of the results you obtained, regardless of whether they are accurate or inaccurate. This problem is meant to get you thinking about data representation and the variety of approaches one can take for classification of signals. You do not need to try all the approaches listed above; pick an approach and do your best.

**Deliverables**

Provide a report, in the form of a document or a video, describing your approach, your results, and (if applicable) difficulties you encountered in accurately classifying the signals. If writing a report, it should be no more than 4 pages. If recording a video, it should be no more than 10 minutes.

**Available Data Files**

In addition to the original text files from the UCI site, I have saved the signals in a variety of formats, to facilitate different approaches to the classification problem. Link to Folder

- MATLAB file hillvalley.mat contains the signals in both the time domain and frequency domain formats. The noisy signals have _n in the variable names. These representations are the most convenient for heuristic calculations and shallow neural networks. The "target" variables contain the signal labels in the format needed by MATLAB's shallow neural networks, and the signals are arranged so that each column of the matrix is an individual signal.
- MATLAB files hillvalley_cells.mat and hillvalley_n_cells.mat contain the signals in cell array form. The data in this file would be compatible with the approach taken in the MATLAB tutorials on sequence-to-label classification using the Japanese Vowels example dataset: Create Simple Sequence Classification Network Using Deep Network Designer - MATLAB & Simulink (mathworks.com) Note the separate cell arrays for training and validation, which were taken from the training/testing breakdown given in the original dataset. I have created both time domain and frequency domain representations for you to use in your experiments.
- The zip files hillvalley_plots and hillvalley_noisy_plots contain plots of the signals with the y axis starting at zero and ending at 1.2 times the max value, similar to a standard MATLAB plot. The signal is positioned in roughly the same place in the plot for each of these images. The files hillvalley_zoomed_in and hillvalley_noisy_zoomed_in plot the signals in a more "zoomed-in" fashion. These images could be classified with a deep network with convolutional layers.

[1] The fft() function in MATLAB will provide the coefficients for the frequency components of the signal, as a vector going from the lowest to highest frequency components. These coefficients are produced as complex numbers. It is common to represent the coefficients using a magnitude value and phase value, specifying the amplitude and phase of the sinusoidal components. Thus, the single 1-D signal sequence in the time domain becomes two real-valued sequences in the frequency domain. The function abs() provides the magnitude of a complex number and the function angle() provides the phase.