Course Project: Classification Programming

Neural Networks

Classification and prediction tools are important components of typical decision-support systems. In this assignment, you will be collaborating with 1 to two other students to develop an appropriate code base, perform experiments and write-up results. This handout describes these elements and involves applying your knowledge of neural networks to create, train and evaluate two neural network designs based on data (see below) to classify and predict data into one of two categories. One category indicates that investment in advertising for a targeted advertising campaign is not worthwhile and coded as a 0. The other category indicates an advertising campaign **is** worthwhile and coded as a 1. Below is a scenario and problem motivation along with data with which you will train and evaluate your neural networks. The second part outlines the basic methodology for training and evaluating the performance of your neural network.

This assignment is designed to utilize your own code base used for the Module 7 Programming assignment and will require relatively simple modifications to it to do the training and analysis.

1. Data/Motivation/Scenario

Advertising companies, organizations, associations, and credit providers are interested in determining the best groups of consumers that should be exposed to advertising campaigns. These groups of consumers can be induced to engage in financial transactions such as purchases of goods and services by specially **targeted** advertising campaigns and other inducements. Advertising groups are motivated to generate as much revenue as possible per dollar spent on advertising that includes these targeted advertising campaigns. In other words, these groups want to get the 'biggest bang per buck' from their advertising expenditures.

Targeted advertising campaigns are especially profitable because groups to which these campaigns are directed are most likely to respond to advertising campaigns by engaging in financial transactions. It is therefore important to determine which of these groups and/or individuals should be targeted for these advertising campaigns. There are several factors used to determine which groups are most profitable to target. The effort/cost of determining these factors varies.

The first metric used to assess whether a targeted campaign is warranted is referred to as the household's **Size of Wallet** (SOW) and is an estimate of a household's disposable income or money that is likely to be spent on non-essential goods and services. This metric is associated with or based on several complex socioeconomic factors but is principally based on a household's estimated **gross income** (GI), the number of people living in the household, and other socioeconomic and demographic data. Basically, this metric SOW $\in$ [0, 3] (higher values indicate a larger SOW).

Another important metric is the **local affluence code** (LAC). This is a very complicated metric to accurately determine, but rough estimates are obtainable by looking at neighborhood housing sales, comparable housing prices, and other factors. This value also ranges from 0 – 3, *i.e.,* LAC $\in$ [0, 3] where numbers near 0 indicate a very low level of affluence typically associated with lower-grade building structures, lower housing prices, and some degree of neighborhood blight and crime. A grade near 1 corresponds to higher housing prices, typically in the range of state average prices per square foot, higher quality housing structures, and lower crime rates. A grade near 2 or above (where 3 is the highest possible LAC score), corresponds to a higher affluence score typified by housing prices over $500k, newer constructions (although this rating can encompass very old, stable, and high-priced housing), and little or no crime.

These two inputs, SOW and LAC map to the investment opportunity, the **targeted advertising code assignment** (TACA). The TACA $\in$ {0, 1} where a 0 indicates a negative return on targeted advertising campaigns. A 1 indicates a positive return on advertising campaigns. These codes are based on historical data and can be used to train a neural network so that using inputs for SOW and LAC, yields an output value for the TACA. In other words, the neural network can classify (and predict) which households should be targeted for advertising campaigns.

2. Neural Network Data

Accurately determining estimated SOW, LAC, and TACA values involves collecting and analyzing a host of data and is relatively expensive. The whole point of using a neural network in this context is to serve as a decision support system that can quickly and accurately identify households with a TACA score of 1 based on their LAC and SOW scores. Consequently, the TACA score can serve as an output of a neural network while the SOW and LAC can serve as inputs. The following data shows the associations of these three values for 20 households. **Use this data to train and evaluate two neural networks: one that uses a single perceptron and one that uses the network in the Module 7 Programming assignment.** The same basic methodology for each network is described below.

Data Item	LAC	SOW	TACA
1	0.90	0.87	1
2	1.81	1.02	0
3	1.31	0.75	1
4	2.36	1.60	0
5	2.48	1.14	0
6	2.17	2.08	1
7	0.41	1.87	0
8	2.85	2.91	1
9	2.45	0.52	0
10	1.05	1.93	0
11	2.54	2.97	1
12	2.32	1.73	0
13	0.07	0.09	1
14	1.86	1.31	0

15	1.32	1.96	0
16	1.45	2.19	0
17	0.94	0.34	1
18	0.28	0.71	1
19	1.75	2.21	0
20	2.49	1.52	0

3. Methodology

Your **training data** will use the odd-numbered data points. To train your network, use the online training technique as in the Module 7 Programming Assignment. This involves using the approach in Method 1 of that assignment, only here, each cycle involves applying an FFBP cycle using the **first input/output** pair with inputs 0.90 and 0.87 with the desired output of 1 followed by using the third input/output pair 1.31 and 0.75 with the desired output of 0 and so on up to applying the 10^{th} (row 19) input/output pair with the desired output of 0. Perform 30 such cycles to minimize the value of Big E. The above sequence can be repeated with different initialized random weights in order to obtain the best weights.

Once you have trained the weights in your networks with the best weights, you must perform an optimization of the trained network to determine the best threshold for using threshold logic for the output node which maps some value from the Sigmoid activation function to either a 0 or a 1. This will involve setting some reasonable threshold and using the network in a feed-forward mode to present each set of inputs to see how many correct mappings to 0 or 1 occur from the training data. This will allow you to determine the Receiver Operating Characteristics (ROCs) of your network. You will have to experiment with adjusting this threshold to obtain the best ROCs. Once you have determined the best weights and threshold value, evaluate your network using the **testing data** in the even-numbered rows 2, 4, …, 20 to determine the actual ROCs based on the testing data.

4. Submission Requirements

The following guidelines describe the requirements for project submission. Only 1 member of your team should submit a pdf formatted file as described below.

Documentation

Your final reports should use 1-inch margins and use 10 pt. font preferably in Times New Roman and include the following elements:

- An "executive summary" that describes the goals and purpose of your network on 1 page.
- A description of your problem, the network design, and how it addresses the problem in 1-2 pages or less.
- A description of the computational performance of your two networks in 3 pages or less. This should include some comparisons of the results.

- An analysis of the performance and how it might be improved in the future. This analysis should include appropriate statistical methodologies in 3 pages or less.
- A summary and conclusion addressing the overall performance of the networks and what might be done to improve it in the future. This could take the form of conjectures, references to other works, etc., and should be 1-2 pages in length.

  Appropriate references, graphics, and sample code output should also be included **in an appendix**. I do not want to be deluged with reams of code and output data. The final report should be written using word processing software. And remember, the work must be your own or your team's own work. The final reports will be checked using Turnitin to ensure that there is no plagiarism.

  This assignment is due in Module 14

Grading Criteria

Your project grade will be based, in part, on the following considerations. These are not set in stone of course, but will be considered:

- **Imagination** - your **inventiveness**, *i.e.,* how well you implemented **ideas** that support a stated **purpose or goal** for your network.
- **Implementation** - how well you have implemented your ideas using methods discussed in class, in the book, and from other sources. This means that you simply cannot take a canned or COTS package and run the back-propagation algorithm on it with some training set that you find on the internet. You should add value to the neural network world with your work.
- **Analysis** - your project should include appropriate analysis of your network, data, and performance. If your network is your own invention, then some attempts to analyze it mathematically should be included; however, your own inventiveness may preclude rigorous analysis, and this will be taken into account. In other words, new inventions may not readily be amenable to analysis. Nonetheless, some appropriate mathematical statements should be included.

  The following table describes the relative weights that will be used in assessing your project. This involves four main categories each of which is further broken down into sub-categories. Note the main categories have associated weights and each sub-category also has associated weights given the category it is in. Thus, each sub-category is given a final relative weight factor as indicated and based on the product of the category weight and the relative weight within a category.

Category		Score	Cat. %	Rel.	Factor	
Technical Content	Topic Mastery		0.4	0.4	0.16	0
	All components		0.4	0.25	0.10	0
	Appropriate Detail		0.4	0.1	0.04	0

	Completeness of Analysis	0.4	0.25	0.10	0
Organization	Clear goals and approach	0.2	0.3	0.06	0
	Content organized	0.2	0.3	0.06	0
	Uniform writing style	0.2	0.2	0.04	0
	Good transition	0.2	0.1	0.02	0
	Intro/Conclusion appropriate	0.2	0.1	0.02	0
Imagination/Insight/Inventiveness	Scientific Method	0.3	0.5	0.15	0
	Originality	0.3	0.2	0.06	0
	Insight	0.3	0.3	0.09	0
Layout/Visuals	Quality of graphics	0.1	0.7	0.07	0
	Uniform document design/layout	0.1	0.3	0.03	0
		Total		1.000	0

Each row will be initially evaluated using a Likert scale of 1 – 5. The resulting score will therefore vary between 1 and 5 and will then be adjusted based on class performance.

5. Written Report

The written report will indicate the weights for each perceptron after final training, and the measures of performance such as the mean squared error for the TACA value and the data corresponding to the **receiver operating characteristics** for the classification problem such as the sensitivity, the specificity and you **may** include the **negative predictive probability** and the **positive predictive probability** if you desire.

In your report, indicate any particulars you think are relevant to using the neural network as a decision support tool and limit your report to no more than 10 pages. You may include your programming code as an appendix.