Should I Buy That CD?

A Neural Network Project

By Lucas Divine

ECE 539 -12/19/03

Introduction

The following is a project report for my Album Rating Neural Network. It has been developed for ECE 539: Neural Networks and Fuzzy Systems in the Fall semester of 2003. The Neural Net is a simple node based network to decide if Music Albums are worth listening to. The material is intended for academic and enjoyment purposes; none of the material or ideas my be taken and used for profit. All material, ideas, and code are owned by Lucas Divine.

Table Of Contents

Summary

The following project melds the concepts of Artificial Neural Networks and Music Album Selection. A Neural Network has been developed to take in data about a particular music album and to report back to the user whether or not the album is: worth buying, worth listening to, or not worth either.

The basis of the Neural Network was to develop a model that would closely resemble how a human mind decides on album buying. This encompassing idea stretches throughout the whole project – from the data selection, weight derivation, and final outputs.

The data inputted into the neural network consists of 11 pieces of information based on facts, opinions, and statistics. These pieces of information have been compiled from Amazon.com, RollingStone.com, and Billboard's Research Division. They range from customer ratings of the albums, how many millions of albums have been sold, to the current prices of the albums. Another input to the neural nets are also the inputted biases.

The last piece of data is dynamic based on the user. This is the 'gut feeling' indicator. Since much of music buying depends on if a person has heard a single, a friend has recommended the album, or various other reasons, the neural network needed to account for this. The gut feeling has sway over the final results in most cases. Obviously, though, in some cases, the album is either quite good or bad, which makes the static data very much overpower the 'gut feeling'.

The derivation of weights was dependant on what types of information needs to be stressed more than others. Such items as number of weeks on the Billboard Top 200 charts has more sway than number of CDs that come with the album.

And lastly, the neural network doesn't just give a yes or no to buying the album. It also has a 'grey' area, which, like music, means that it would be good to just listen to the CD. This could mean checking it out from the library or borrowing it from a friend. This closely resembles how music is handled by the human brain.

Starting Point

Problem Statement

If I pick a certain Music Album, I would like a neural net to tell me if it is worth buying or listening to.

Problem Motivation

This project was approached from the basic premise of creating a Neural Network that could be applied to things that most people understood. It was created as something that anyone could give a go. This would allow myself to explain the basics of Neural Networks and show people first hand one way it could be applied.

This network could be applied especially well in the holidays. With people doing a lot of online shopping and research, they want to know whether the person they are buying the Compact Disc for will like it. By using mostly data found online, people could just throw the data in the network and not be faced with the agonizing pain of deciding for themselves. This also would help parents or grandparents who are not 'in the know' with the latest music to decide which CD on a Christmas list to buy their children or grandchildren.

Another feature that helps make this network appeal to the public is the 'gut feeling' input. If they have heard something good or bad about it and have a gut feeling, they put that in and it does affect the outcome of the decision. This is just like in the brain, how gut feelings contribute, sometimes more than the data, to decisions.

Beginning Ideas

The basic ideas I had going into the project revolved around thinking about how the human brain dissects data in regards to music albums. I wanted my neural network to function in many of the same ways that a network of biological neurons would. That is how I approached my data and the concepts of the project. I originally wanted a Neural Network that would judge songs as well, but the data collection is harder for that type of network and that could be done in a whole separate project.

Most neural nets are brought about by developing algorithms that apply to the real world. This is one of them.

Theory

The implemented program is an Artificial Neural Network. It is modeled after the human brain in many ways, trying to simulate how the brain would attack the same issue. It is comprised of nodes, called neurons. Each neuron makes some decision and passes that decision on to other nodes. Finally a decision is made on whether or not to keep interest in the music album.

The neural network is a multilayer perceptron with hidden nodes in one hidden layer. The layer is referred to as hidden because all the user is aware of are the inputs and the final output. The perceptron part of the name stems from the fact that if the final output of the output node is above zero, the neural net will tell you to either listen to the album, or go all out and buy it. If the output is less than zero, you should do neither.

The network is also memoryless. It is feed-forward, acyclic, and does not apply learning errors to update the weights. This is due to the nature of the problem. Album purchasing is based generally on music opinions. The data generated, such as billboard stats and critic ratings, are all based on judgments. This leads to not having a 'right' or 'wrong,' just making decisions based on some data given. This also means, though, that no training data is available that would give a 'factual' picture of what the outputs should be based on the inputs.

The only learning done is by the programmer who sets up the network to function. Thus weights are static and apply to all albums equally. It is good to have this so one albums training data can't adjust the weights so that they don't apply to other albums.

The activation function, or net function, used is the hyperbolic tangent. This non-linear net function could have been selected from many other types. For the final output, after the hyperbolic tangent is applied to the summation results, there is a final threshold that makes the final decision. These activation functions serve to make the results more manageable and comprehensive. It also functions to not let one bad piece of data skew the results of the whole network. Thus, if a certain album was only on the Billboard Top 200 for 2 weeks but ended up selling 8 million copies and costs only $10, the final output isn't skewed.

Biases that shift the output a certain way are applied to each neuron in the network. These biases are applied to aid in the summation steps and are all set to one. The weights decide how affective a certain bias will be.

Data

Choosing

Choosing what data to use and retrieving it are often the hardest parts of the neural network creation process. Without data, you can have the best idea, and it will go nowhere. Almost all Scientists - Social, Astronomical, Computer, and from all other fields - crave data. This was the big task that would determine how effective the Neural Net would be.

In choosing the data for music albums some types were automatic to chose, while some were debated on. For example, the highest the album climbed on the Billboard Top 200 charts was a given to have, but the current sales ranking on amazon.com or what the album debuted at on the charts might not need to be viable data. This also affected the weights later on – the more relevance, the more the weights reflected that.

The final data types selected are shown in the Arrangement section.

Acquisition

Acquiring the necessary data is a nuisance. This goes for most people in search of data. It has to come from a variety of sources, it has to be credible, and it must be accurate. These are the criteria that the data search was built around. My sources also had to have all of the albums I was going to be working with, simply because a partial data set would lead to bad results.

I acquired my information from three primary sources:

1. Amazon.com: Amazon.com is an online store that has almost anything that can be thought of. They did have much of the information I needed because they pride themselves in providing good information to their customers. The information I got off Amazon.com were a lot of the general statistics such as current price and total songs. This is because the data is very easy to read on their site and they do have very competitive market prices. They also had all the albums; they didn't have just the data on one album or artist and not another.

2. Rolling Stone: Rolling Stone is a music publication that comes out every month, to which millions subscribe. It is always on the leading edge of the music industry. They have reviewed all of the albums and given them ratings, so this was one metric I got from them. This rating was more of an 'official critics' opinion. I also got the customer ratings from RollingStone.com customers that write their own reviews and post their

own ratings. The final ratings are then tallied up as the average of all of the customer ratings.

3. Billboard: The final place data was acquired from was Billboard. Billboard is a industry standard for music song and album statistics. They have publications, a large online presence, and are quoted on almost every program on music. This includes news stories on albums to all of the 'behind the music' specials on artists. On many shows, if you hear about 'this song was on the charts this long' or 'this debuted this high on the charts,' this data was from Billboard. In order to get such information the Billboard research department had to be contacted. It is not just free information. They sell research data based on artist and this is why I only have a certain number of artists to select from. Billboard also provides what certifications the album received (gold, platinum, etc…). This is how the total sold data was acquired.

The reason why I chose the customer ratings on specifically RollingStone.com and Amazon.com is based on how many reviews that each site had. They both had such high traffic, that usually thousands of people had cast their ratings and so it is a pretty good sampling of the population's opinion.

Arrangement

The arrangement of data in the files is not something that matters to a high degree. As the data is handled by the network, the row of data is just selected by the number of the row. This means that the computer does not care in which order it gets the data, as long as it knows how to handle it. This leaves the programmer flexibility in setting up the data and the neural network. It is the weights of the synaptic connections that truly determine how each input node affects the hidden nodes.

The data is out in this specific way because I tried to keep the data near the hidden layer node that it is close too in terms of type. The hidden layer nodes will described more in the Development section, but in general, the data near the top focuses on facts, middle on opinions, and the bottom on statistics. Thus, such items as ratings are mostly opinions and are towards the middle of the data layout due to the fact that the middle hidden node is the 'opinions node.'

The arrangement of the data in the data files is the following:

Album	U2 – All That You Can't Leave Behind
Current Price	14
Number Of Discs	1
Total Songs	11
Original Release Year	2000
Amazon Customer Rating	4
Amazon Sales Ranking (thousands)	0.7
Rolling Stone Rating	4
Rolling Stone Customer Rating	4.5
How Many Sold (millions)	3
Number of weeks on Top 200	94
Highest on Top 200	3
User Gut Feeling	?

Choice of Artists

The choice of artists was based on varying factors. Artists were chosen from many genres, with varying success commercially, and with varying target generations. U2, Paul Oakenfold, Elvis Presley, Metallica, Frank Sinatra, Garbage, Tupac Shakur, and Goldfinger were all selected. This is sure to please almost all music enthusiasts. For a better description of the artists and their type of music, one can check out my sources in Appendix B.

Note: ****Due to the poor statistics given by billboard and much confusion over what releases were singles, movie soundtracks, special editions, and so forth, Elvis and Frank Sinatra don't have as many albums as I was hoping for. Pre 1985, the data gets a bit more 'fuzzy'. *****

Preprocessing

Preprocessing of the data involved was very important. This is because the weights used merely applied a multiplication factor upon the preprocessed inputs.

The preprocessing in general can be done in two ways. One way is to have them all enter a first layer of neurons which perform a function on them. The second is to preprocess them so that they are not 'considered' part of the Neural Network. I chose the second route because almost all of the input data had different preprocessing functions that needed to be applied to them. Another reason for this was that I wanted to give the user and tester the feel that the neural network began with the scaled input, not with the raw unscaled data.

An example of the preprocessing is how the ratings are handled. On each rating, the average is 3, so 3 was subtracted from each rating to prepare it for the neural net. This is due to the fact that negative is bad and positive is good, so a rating of 2 is not good, so it translates to -1 after being scaled.

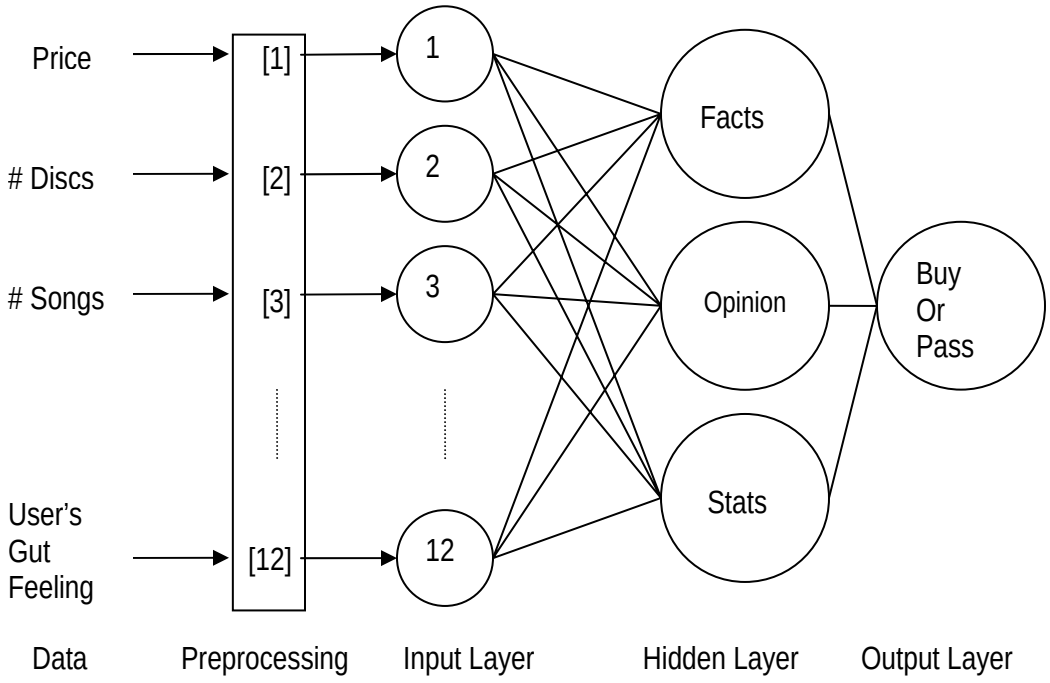
Another example is price. Some albums have two compacts discs with them, so I divide the price and total songs by the number of compact discs to scale the data to more equally compare against the others.

And obviously, I had to switch around the price, because lower is better, so I subtracted the number from 13. This made higher priced albums not as attractive as their lower priced competition. Other preprocessing techniques are commented in the Matlab code for musicNeuralNetwork.m. This function is described in Program Development section.

Development

Artificial Neural Network

The Neural Network used in this project is shown below.



The type of neural network could have been almost anything. This specific version was chosen for one main reason: comprehensibility. This means that its function should be easy to understand for a human. This aids when it is being programmed, when it is being thought about, when it is being described, and when it is being graded.

The hidden layer nodes are the key players in the system. They are arranged as alluded to before, with facts, opinion, and stats neurons. There are lines from every input layer node to every hidden layer node, but many of them have weights equal to zero if they do not fall into the category for the hidden layer node. This is a neat feature because then the outputs of each hidden node can be viewed to see if, for instance, the facts node has a high output. In this manner, an album can be chosen and the user could see which of its data helps support the claim of buying it, and which hurts the claim.

The hidden nodes then send their outputs into the output node, which makes the final decision whether to buy the CD, just give it a listen, or have nothing to do with it.

Weights

After data collection, weight derivation was the most crucial part of the project. All the data needs to play a role, but each to a different extent. The weights multiply with the output of the source neuron and that result is summed with all of the other results from other source neurons. Then that node's activation function applies to the summation output and on to the next layer of neurons the data flows.

The weights in this project were developed through human learning and application. Since I had no training data to use any back propagation methods, I had to designate what the final weights were going to be myself. Many, many considerations came into play. A neural network which did adjust well to the change in user input is something that was necessary. Other requirements included making sure all the data had some effect, but none 'drowned out' the rest.

The weights were statically programmed in a 13x4 matrix with the 13^{th} value being the bias at that neuron that aids in the summation step. Each column represented a destination neuron, and the row the source neuron. These weights were tested with all inputs and all albums to make sure the results were accurate. They can be viewed in musicNeuralNetwork.m.

Program Development

Two programs and a bunch of data files have been developed to handle this project:

1. musicInput.m: This Matlab file handles the user interfacing while they pick which artist and album they would like to test. It also receives their gut feeling. It passes the albumData matrix to musicNeuralNetwork.m which does the rest.

2. musicNeuralNetwork.m: This Matlab file is the workhorse of the project. It handles the preprocessing, sets up the weight matrix, does all the math, and gets a final result back to the user. This is the neural network in action.

3. musicU2.txt: This is an example of a data file that is read in. U2 stands for the artist because each artist has their own data file. These files are created by copying the data from the Excel file where all of the data has been gathered. The data is arranged as is show in the data section. It can be more properly viewed in the Excel file, musicData.xls.

Results

Below is an excellent sampling of what this neural network produces. It performs as expected. See the notes below the table for a more in depth look at the Table.

Gut Feeling	1	2	3	4	5
1. U2 - Pop	SORRY	SORRY	SORRY	LISTEN	BUY
2. Paul Oakenfold – Perfecto Presents Another World	LISTEN	LISTEN	LISTEN	LISTEN	BUY
3. Metallica - Metallica	BUY	BUY	BUY	BUY	BUY
4. Metallica - Kill 'em All	SORRY	SORRY	SORRY	LISTEN	BUY
5. Elvis - 30 #1 Hits	LISTEN	LISTEN	BUY	BUY	BUY

To make sense of the data above, it will be slowly diagnosed. A sampling has been taken from each artist, and I could fill up page after page if I investigated all the possible albums.

- Row 1: It is shown what an ‘average’ album would be like. If the user has a high ‘gut feeling’ about it, then go for it, otherwise just listen to it or hold off.

- Row 2: Some CDs just flat out deserve a listen. In this case, other than solid ratings, you get 2 CDs with 12 songs each. That is a good deal and it carries the load to stay a worth a listen.

- Row 3: With 13 million copies sold, 5 star ratings from every source, having been on the Billboard top 200 for 281 straight weeks, and still only being 13.5 bucks, if a person doesn’t buy it, it is their loss. That is the neural net at work.

- Row 4: This just shows the varying results, that the neural net works well, and the results truly do take into account all the data.

- Row 5: With high ratings, even if you don’t buy the Elvis Hits, you should at least listen to them no matter what. Most people would have to agree with that opinion.

More results….

Gut Feeling	1	2	3	4	5
6. Elvis – The Christmas Album	SORRY	LISTEN	BUY	BUY	BUY
7. Frank Sinatra – The Very Best of Frank Sinatra	SORRY	SORRY	BUY	BUY	BUY
8. Frank Sinatra – The Very Good Years	SORRY	SORRY	SORRY	SORRY	LISTEN
9. Garbage - Beautifulgarbage	SORRY	SORRY	SORRY	LISTEN	BUY
10. Tupac Shakur – 2Pacalypse Now	SORRY	SORRY	SORRY	SORRY	LISTEN
11. Goldfinger – Open Your Eyes	SORRY	SORRY	SORRY	SORRY	SORRY

- Row 6: This shows that the Neural Net doesn't always go one way or the other, it does lean towards one end or the other depending on the data. The listen is in the 2 spot. That means that only if you heard that this specific album was terrible should you avoid it. This 'moving around' of the LISTEN is the sign of a robust design.

- Row 7: This is an example of how some don't have any LISTEN buffer in between BUY and SORRY.

- Row 8: Compared to Row 6, this CD is at the opposite end of the spectrum. This is mainly because of the customer ratings and because Sinatra's 'The Very Good Years' sold 1 million fewer copies. This shows that depending on the album, some factors have a larger contribution than others.

- Row 9-10: These just show the varying results, that the neural net works well, and the results truly do take into account all the data.

- Row 11: Some CDs are just bad and should be not even be considered. In this case, 'Open Your Eyes' was only on the Top 200 charts for 1 week at spot 136. This and a terrible Rolling Stone ranking just sink this Goldfinger Album. You could still buy it, but you'd be going against the recommendation of the Neural Network.

For more results, give the neural network program files a run through.

Discussion

So as you can see, the results:

- Vary quite a bit (so does the data)
- Are specifically affected by varying stats
- Show that the ANN does tell you to BUY good CDs and SORRY for bad ones
- A ‘linear’ pattern does not occur, thus demonstrating the non-linear element of the perceptron network
- User ‘gut feeling’ input has the effect that it should

As stated above, the user input in some cases will determine whether to BUY the album or not. In other cases, like the Metallica case, the album should be bought no matter what the user has heard. This is definitely what is expected from a Neural Network that makes decisions that are based on opinions to begin with.

In the end, music selection is heuristic. If the computer or the brain is doing it, it is still heuristic, based on heuristic data. What has been created is a Neural Network that is responsive to many factors in music selection to help weed out some albums that you might want to wait on listening too. This is very valuable to people with not as much time as they would like. The strength of this ANN is the quality and volume of data. One CD might have been #1, but another CD might have sold 2 million more copies. All is taken into account. Sometimes even humans can’t digest all this data.

The user ‘gut feeling’ is another area where this network shines. As show in the Results section, the neurons are quite responsive to the change in user input. But, for some albums, if they are entirely bad, even a great gut feeling can’t save the album from the ‘trash’. This is also as it should be.

Conclusion

In conclusion, I found out that I decided on a project that was doable and not too insane to be completed in the allotted time. I also learned that data on songs is not as easy to come by as data on albums. This lead me to drop that idea from the overall project.

I also learned about how researchers do their work. This, I feel, was the most valuable part of the project. From the beginning thesis to the final result is a trip that I will most definitely have to do all the time in the future. The steps of data collection, handling, and calculation were all done thoroughly and effectively. In doing, I have learned. I also liked developing an ANN that my friends and family can check out and try.

One neat aspect of this Neural Network is that you can now pick any artist and album, snag all the data and this network can apply. It is very open to any data, but for this project, only a certain number of albums could be tested. So this will encourage others to use and play around with the programs provided.

Appendix A: Matlab Files Code

musicNeuralNetwork.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% musicNeuralNetwork.m
% Lucas Divine
%
% Description: This is the main file that handles the neural network.
%
%
% Created 11/26/03
% Due 12/19/03
% Part of ECE 539 Final Project
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close, clear all

%%%
% Set up variables
%%%
global albumData;
nodeFacts = 0; nodeOpinion = 0; nodeStats = 0; output = 0;


disp(' ')
disp('********************************************************************************')
disp(' Welcome to the Divine Neural Network for deciding whether or not you should ')
disp('  listen/buy a certain Album.')

%%%
% Run program to get the data to run through the network
%%%
musicInput

% Save input for output later
save = albumData;

%%%
% Create and Initialize all Weights in a Weight Matrix
%%%
w = zeros(13,4);
w = [  1   0   0  .5 ;
      .5   0   0   1 ;
     .25   0   0  .5 ;
```

```
    .02   0   0   0 ;
     0   1   0   0 ;
     0   0   1   0 ;
     0   1   0   0 ;
     0   1   0   0 ;
     0   0  .5   0 ;
     0   0 .02   0 ;
     0   0 .33   0 ;
     0   1   0   0 ;
     0 -4.5 -4   1 ];

%%%
% Set up offsets (bias) that come in on w0 which is w(13,:) for us
%%%
albumData(13,1) = 1;
offsetOutput = 1;

%%%
% Preprocess data
%%%
% price input = 13 - Price per CD
albumData(1,1) = 13 - (albumData(1,1)/albumData(2,1));

% #songs input = 12 - songs per CD
albumData(3,1) = 12 - (albumData(3,1)/albumData(2,1));

% Year = distance between then and now
albumData(4,1) = abs(2003 - albumData(4,1));

% avg cust ranking = ranking - 3
albumData(5,1) = albumData(5,1) - 3;
albumData(7,1) = albumData(7,1) - 3;
albumData(8,1) = albumData(8,1) - 3;
albumData(12,1) = albumData(12,1) - 3;

% amazon sales ranking = scaled down version
albumData(6,1) = tanh(2 - albumData(6,1));

% # weeks on the chart = weeks - 20
albumData(10,1) = albumData(10,1) - 50;

% highest in the top 10 = 10 - highest
albumData(11,1) = 10 - albumData(11,1);

% the rest stay as they are
```

```
%%%
% Get the output of the hidden layer neurons
%%%
for i=1:13
  n = albumData(i,1) * w(i,:);
  nodeFacts = n(1) + nodeFacts;
  nodeOpinion = n(2) + nodeOpinion;
  nodeStats = n(3) + nodeStats;
end

% Smooth the answer to either 1 or 0 for each node
nodeFacts   = tanh(nodeFacts);
nodeOpinion = tanh(nodeOpinion);
nodeStats   = tanh(nodeStats);


%%%
% Get Final Ouput
%%%
output = tanh(w(1,4)*nodeFacts + w(2,4)*nodeOpinion...
   + w(3,4)*nodeStats + w(13,4)*offsetOutput);

%%%
% Report Back to User
%%%
disp('Hmmm... Interesting....')
disp(' ')
disp(' - - - - - - - - - - - - - - - - - -')
if (output > 0.5)
   disp('     BUY!   ')
   disp('Get this CD! Buy it, listen to it! Enjoy!')
elseif (output < 0)
   disp('     SORRY!  ')
   disp('It is recommended that you pass on this CD...' )
   disp(' Try other CDs to find sure winners')
else
   disp('     Just LISTEN to it   ')
   disp('The library probably has this one you can check out free and listen to it')
   disp(' That would probably be the recommended action')
end
disp(' - - - - - - - - - - - - - - - - - -')


%%%
% Show them the Album Data if they are interested
%%%
disp(' ')
```

```
disp('Would you like to see the data that brought about this decision? ')
disp(' 1 - Yes, 0 - No ')
yes_or_no = input('Your Choice: ');

if (yes_or_no == 1)
   disp([' Current Price of the Album: $',num2str(save(1))]);
   disp([' Number of Discs: ',num2str(save(2))]);
   disp([' Total Songs: ',num2str(save(3))]);
   disp([' Original Release Year: ',num2str(save(4))]);
   disp([' Amazon.com Customer Rating (out of 5 stars): ',num2str(save(5))]);
   disp([' Amazon.com Sales Ranking (in thousands): ',num2str(save(6))]);
   disp([' Rolling Stone Rating (out of 5 stars): ',num2str(save(7))]);
   disp([' Rolling Stone Customer Rating (out of 5 stars): ',num2str(save(8))]);
   disp([' How Many Albums Sold (in millions): ',num2str(save(9))]);
   disp([' Number of Weeks on the Billboard Top 200: ',num2str(save(10))]);
   disp([' Highest Ranking on the Billboard Top 200: ',num2str(save(11))]);
   disp([' Your Gut Rating (out of 5 stars): ',num2str(save(12))]);
end


%%%
% Say Goodbye and Thanks, people like that
%%%
disp(' ')
disp(' ')
disp('Thanks for testing this CD, ')
disp('       Run this program again for many others,')
disp('               Have a nice day! ')
disp(' ')
disp('*********************************************************************************')
```

---

musicInput.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% musicInput.m
% Lucas Divine
%
% Description: Describes the input file and helps users pick which CD they would
%  like to test.
%
%
% Created 11/26/03
% Due 12/19/03
% Part of ECE 539 Final Project
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
global albumData;

% Set up Band Variables
U2 = 1;
Oakenfold = 2;
Elvis = 3;
Metallica = 4;
Sinatra = 5;
Garbage = 6;
Tupac = 7;
Goldfinger = 8;


%%%
% Let the user select which artist to choose from. Each artist loads its own data file.
%%%
disp('Please select the artist whose album you would like to test of the following: ');
disp(' U2 = 1 ');
disp(' Paul Oakenfold = 2 ')
disp(' Elvis Presley = 3')
disp(' Metallica = 4')
disp(' Frank Sinatra = 5')
disp(' Garbage = 6')
disp(' Tupac Shakur = 7')
disp(' Goldfinger = 8')
bandChoice = input('Your Choice: ');
disp(' ');

%%%
% The Data in the musicData files are formatted in the following way.  Each column is
%  a different album for that Artist.  Each Row constitutes an input to the neural network.
%  They all signify the following:
%
% Row 1  - Current Price (@ Amazon.com) as of Dec 5th, 2003
% Row 2  - Number of Discs
% Row 3  - Total Songs
% Row 4  - Original Release Year
% Row 5  - Average Amazon.com customer rating (out of 5 stars)
% Row 6  - Amazon.com sales ranking (in thousands)
% Row 7  - Rolling Stone Rating (out of 5 stars)
% Row 8  - Rolling Stone Reader Rankings (out of 5 stars)
% Row 9  - How many sold (Gold, Platinum etc...) in terms of millions
% Row 10 - Number of weeks the album stayed on the Billboard top 200 chart
% Row 11 - Highest it got on the Billboard Charts
%
%
%%%

%%%
% Set up case statement to handle specific CD choosing
% Each Section lists the Artist's Albums and the user selects one of them
%%%
switch (bandChoice)
   %%%
   % U2 Section, a list of CDs is given by column
```

```
%%%
case U2
   load musicU2.txt
   disp('You Chose U2 - They have had 15 Platinum Titles!')
   disp('Please select which Album to test from the following: ')
   disp(' All That You Can''t Leave Behind = 1')
   disp(' Pop = 2')
   disp(' Zooropa = 3')
   disp(' The Joshua Tree = 4')
   disp(' War = 5')
   disp(' Boy = 6')
   albumChoice = input('Your Choice: ');
   % Make sure input is valid
   if (albumChoice > 6)
      disp('Please only input valid options. ')
      break;
   end
   albumData = musicU2(:,albumChoice);
%%%
% Paul Oakenfold Section
%%%
case Oakenfold
   load musicOakenfold.txt
   disp('You Chose Paul Oakenfold - Good Dance Beats! ')
   disp('Please select which Album to test from the following: ')
   disp(' Bunkka = 1')
   disp(' Swordfish: The Album = 2')
   disp(' Perfecto Presents Another World = 3')
   albumChoice = input('Your Choice: ');
   % Make sure input is valid
   if (albumChoice > 3)
      disp('Please only input valid options. ')
      break;
   end
   albumData = musicOakenfold(:,albumChoice);
%%%
% Elvis Presley Section
%%%
case Elvis
   load musicElvis.txt
   disp('You Chose Elvis Presley - The King of Rock and Roll!')
   disp('Please select which Album to test from the following: ')
   disp(' Elvis: 30 #1 Hits = 1')
   disp(' The Christmas Album = 2')
   albumChoice = input('Your Choice: ');
   % Make sure input is valid
   if (albumChoice > 2)
      disp('Please only input valid options. ')
      break;
   end
   albumData = musicElvis(:,albumChoice);
%%%
% Metallica Section
%%%
case Metallica
   load musicMetallica.txt
```

```
   disp('You Chose Metallica - Their self titled album sold 13 million copies!')
   disp('Please select which Album to test from the following: ')
   disp(' Reload = 1')
   disp(' Load = 2')
   disp(' Metallica = 3')
   disp(' ...and Justice For All = 4')
   disp(' Master of Puppets = 5')
   disp(' Ride the Lightning = 6')
   disp(' Kill em All = 7 ')
   albumChoice = input('Your Choice: ');
   % Make sure input is valid
   if (albumChoice > 7)
      disp('Please only input valid options. ')
      break;
   end
   albumData = musicMetallica(:,albumChoice);
%%%
% Frank Sinatra Section
%%%
case Sinatra
   load musicSinatra.txt
   disp('You Chose Frank Sinatra - Smooth... just smooth')
   disp('Please select which Album to test from the following: ')
   disp(' The Very Best of Frank Sinatra = 1')
   disp(' Duets = 2')
   disp(' Sinatra Reprise -- The Very Good Years = 3')
   albumChoice = input('Your Choice: ');
   % Make sure input is valid
   if (albumChoice > 3)
      disp('Please only input valid options. ')
      break;
   end
   albumData = musicSinatra(:,albumChoice);
%%%
% Garbage Section
%%%
case Garbage
   load musicGarbage.txt
   disp('You Chose Garbage - Their last CD came out in 2001')
   disp('Please select which Album to test from the following: ')
   disp(' Beautifulgarbage = 1')
   disp(' Version 2.0 = 2')
   disp(' Garbage = 3')
   albumChoice = input('Your Choice: ');
   % Make sure input is valid
   if (albumChoice > 3)
      disp('Please only input valid options.')
      break;
   end
   albumData = musicGarbage(:,albumChoice);
%%%
% Tupac Section
%%%
case Tupac
   load music2Pac.txt
   disp('You Chose Tupac Shakur - All Eyez On Me went 9 times Platinum!')
```

```
        disp('Please select which Album to test from the following: ')
        disp(' The Don Killuminati = 1')
        disp(' All Eyez On Me = 2')
        disp(' Me Against The World = 3')
        disp(' Stricty 4 My N.I.G.G.A.Z = 4')
        disp(' 2Pacalypse Now = 5')
        albumChoice = input('Your Choice: ');
        % Make sure input is valid
        if (albumChoice > 5)
            disp('Please only input valid options. ')
            break;
        end
        albumData = music2Pac(:,albumChoice);
    %%%
    % Goldfinger
    %%%
    case Goldfinger
        load musicGoldfinger.txt
        disp('You Chose Goldfinger - Punk Genre!')
        disp('Please select which Album to test from the following: ')
        disp(' Open Your Eyes = 1')
        disp(' Stomping Ground = 2')
        disp(' Hang - Ups = 3')
        disp(' Goldfinger = 4')
        albumChoice = input('Your Choice: ');
        % Make sure input is valid
        if (albumChoice > 4)
            disp('Please only input valid options. ')
            break;
        end
        albumData = musicGoldfinger(:,albumChoice);
    otherwise
        disp('You need to select one of the options specified!  ')
        albumData = 0;
        return;
end


%%%
% Get the last input, which is the person's personal opinion
%%%
disp(' ')
disp('What is your gut feeling about the CD? (out of 5 stars) ')
disp(' Put 3 if you have no opinion ')
albumData(12,1) = input('Your Choice: ');
```

Appendix B: Sources

Neural Networks: A Comprehensive Foundation by Simon Haykin

ECE 539 Class Notes by Yu Hen Hu

Amazon.com

RollingStone.com

Billboard’s Research Department