

# **Back-propagation Neural Network Homework**

T. Nathan Mundhenk and Michael A Arbib  
University of Southern California  
Department of Computer Science

Copyright (c) 2006 T. Nathan Mundhenk, Michael A Arbib  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover  
Texts. A copy of the license is included in the section entitled "GNU  
Free Documentation License".

Brain Theory and Artificial Intelligence cs564

Homework 2

Fall 2003

For this homework download the file *BackPropAug.zip* from the DEN web site. This is essentially the same as the BasicBookModel back prop network with a few changes for this assignment. Notable changes for this network are the introduction of a bias term for pre-processing of data, you do not need to do anything with this, it just makes the data train better. Additionally, post output is sent through a step function. This is there to make the decisions of the network more clear and shorten the time of training. The step function itself does not play a role in training the network, so again, during training, pretend it is not there. It is only for post processing data so that interpretation of data is consistent for everyone doing the homework.

In *BackPropAug* you will find three folders, *nsls*, *2layer* and *3layer*. The folder marked *nsls* has the training and testing data for this homework along with the scripts for running the back propagation network. *2layer* is a two layer back prop you will use for question two and *3layer* is an incomplete three layer back prop you will work with on question three.

Each question will ask you to plot out data in relation to what NSL does. You may do this in any manner you wish, so long as the plots are clear and easy to interpret. Try and keep each plot consistent and at the same scale. Points will be deducted for poor graphing style.

The point value for each question is next to each question in *italics*. The total points for this homework equal 100. Extra credit may be given if you do anything cool. If you do anything REALLY cool, copious amounts of extra credit may be given. Use your discretion; I will not help you with the extra credit. Note: the cool thing must have something to do with NSL.

Hand in your homework by dropping it in the box in front of HNB 06 before 5:00 pm or in class on **Oct. 21**. Since the building locks at 5:00 pm this will be strictly enforced. **Late homework** will not be accepted without a really good excuse. For remote students, email your homework to me at [Nathan@mundhenk.com](mailto:Nathan@mundhenk.com) before the due date. Please include your email address on your homework so I can easily contact you if there is a problem.

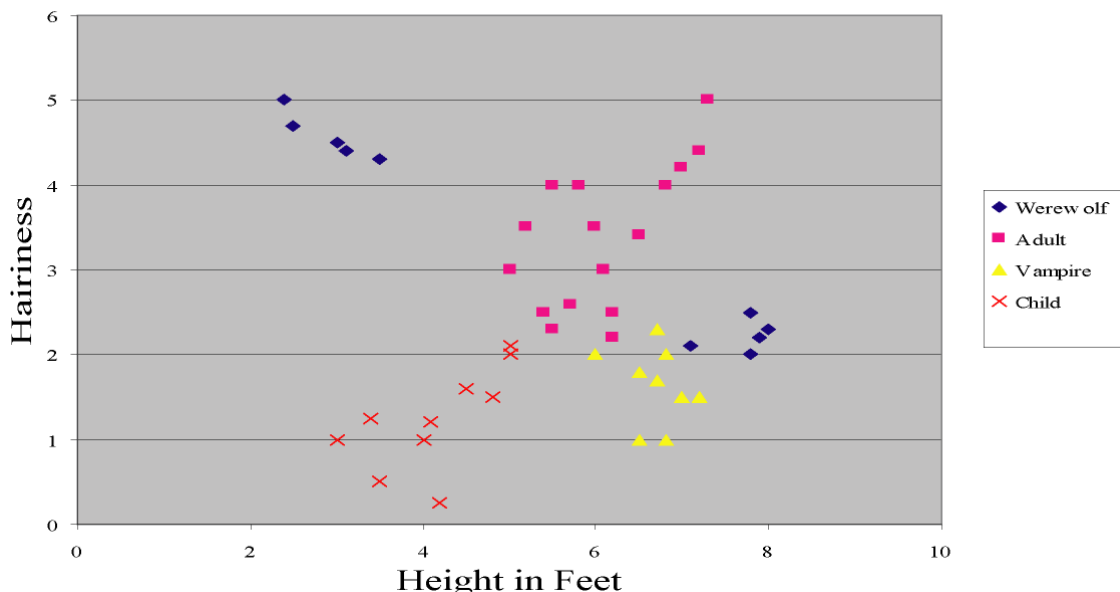
1) (15 points)

- a. Describe the difference between learning in a back propagation neural network and learning in a Hebbian neural network.
- b. Which one is a more biologically plausible system and why?
- c. What is the purpose of adding a second layer to a single layer perceptron

### The Well Behaved Robot:

Your great uncle Otto recently passed away leaving you his mansion in Transylvania. When you go to move in, the locals warn you about the Werewolves and Vampires that lurk in the area. They also mention that both vampires and werewolves like to play pool, which is alarming to you since your new mansion has a billiard room. Being a savvy computer scientist you come up with a creative solution. You will buy a robot from Acme Robotics (of Walla Walla Washington). You're going to use the robot to guard your billiard room and make sure nothing super natural finds its way there.

To train your robot you need to select a set of features which the robot can detect and which also can be used to tell the difference between humans, vampires and werewolves. Further, after having your nephew Scotty ruin one of your priceless antique hair dryers, which you keep in the billiard room you decide that the robot should also detect children entering the room. After reading up on the nature of the undead and after taking careful measurement, you realize that the two best features for detection are how tall a person is who is entering the room is and how hairy they are. This works because vampires are tall and completely bald, werewolves are either short and totally covered in fur, or they are the mutant type that are extremely tall, but no more hairy than a human. An adult human is taller than a child and slightly hairier. The chart below shows samples that you took to validate your hypothesis.



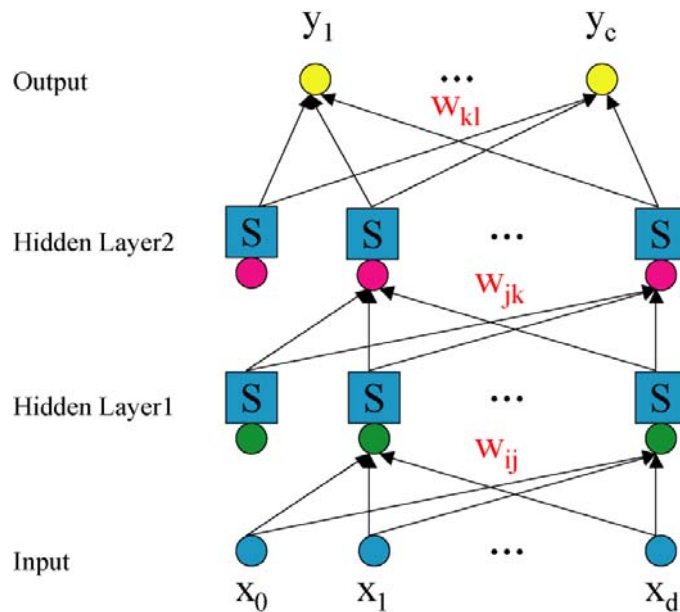
The next thing your robot will need to do in addition to detecting what creatures enter your billiard room is take an action that is appropriate for the situation. Since you want your robot to be polite, it will greet every human that enters the room. Additionally, if a child enters the room, when it greets the child it will scream so that you know to look in your closed circuit television and see what is happening in the room. When the robot detects a vampire, it will scream and impale it with a stake. Since robots are no match for werewolves, if the robot detects a werewolf, it will scream and then run away. Thus, your robot can take any of four actions, it can impale something entering the room, it can scream, it can run away and it can greet people. Any of these action can be performed following the detection of anything entering the room. Your job is to train the robot so that it performs the correct actions whenever it detects something entering the room.

- 2) (35 points) Your first task is to train your robot. Take the training data marked *train1.dat* and plug it into *bpt1.nsls* located in 2layer. Compile the model and run it. After you train the model, test it with *bpr.nsls*. The output from testing can then be found in *out.bin.dat* or *out.dec.dat*. These are tab-delimited files with the test results. You will want to take this data and make a scatter plot which shows a map of how the robot will react when it observes different heights and different amounts of hairiness.

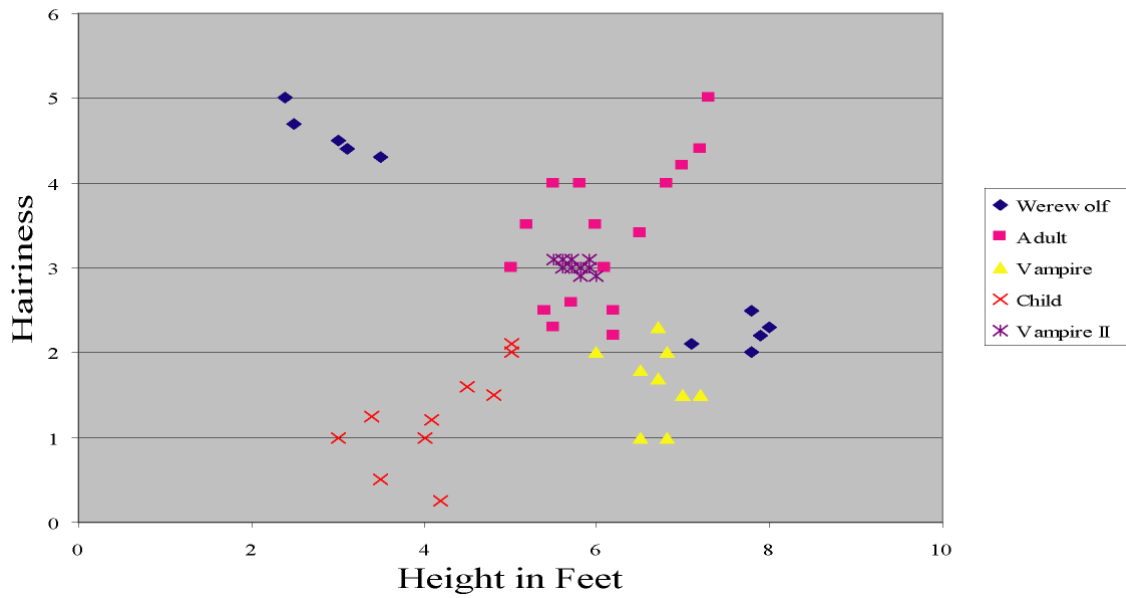
The way to interpret the output is as follows, there are four actions the robot can take, if the robot will take that action, the output is a 1, if the robot will not take that action, the output is a 0. The four actions in order are *Impale*, *Scream*, *Run Away* and *Greet*. Thus, if the output is 0,1,0,1 then that means the robot will scream and greet.

- a. Take and plot the actions the robot takes over the space of possible inputs. *Out.bin.dat* and *out.dec.dat* contain the same information. However, *out.bin.dat* contains the binary coarse code for the output while, *out.dec.dat* contains the decimal equivalent. You can use either for creating the plot. The decimal version may be easier to use. It's up to you. For the plot, make the *x*-axis the height of the visitor and the *y*-axis the amount of hair measured. Each point on the plot should show the robots action for that input. **Note:** you may create the plot with any method you choose, just so it is neat and clear.
- b. Compare the plot of the robot's test actions against the training data. Does the network do a good job of generalizing over the training data? Why or why not?
- c. Does the robot always behave as programmed or does it commit actions that do not fit the patterns for people, children, werewolves or vampires? Explain.
- d. Notice that it reacts to things entering the room as if they were vampires in two regions of space not visibly connected to the vampire training data. Why is that?

- 3) (50 points) Being an inquisitive lad or lass you decide you would like to find out how your robot would perform if you added a third layer to your back prop.
- Recall the equations from the NSL back prop lecture. Derive equations for a 3 layer back prop. This can be done by extending a two layer perceptron to three layers the same way a one-layer perceptron is extended to two layers. The figure below shows the schematic of the three-layer perceptron. **Define:**  $\Delta w_{ij}$ ,  $\Delta w_{jk}$ ,  $\Delta w_{kl}$ ,  $\delta_j$ ,  $\delta_k$ ,  $\delta_l$  and  $y_l$  using the same notation from the NSL slides.
  - Using figure 2 as a guide and your results from 3.a , extend the 2 layer back prop model to a three layer back prop model. Do this on the model in the folder 3layer. Some parts have been filled in already to help guide the process. When you are finished, run the model on the same testing and training data as question 2. Create a scatter plot in the same manner and compare the two.
    - How good a job does the three layer network do on generalizing on the problem?
    - How does it compare to the results from the two layer network?
    - Does it do a better job? Why or why not?



- As it turns out the evil Dr. Moriarty has created vampires that are similar in height and hairiness as adult humans. You have discovered his fiendish plan and must now train a new network. The figure below shows a scatter plot of the new training data. How fortunate for you that almost no one in Transylvania is the same height and hairiness as the mutant vampires, but which neural network should you use? Train both the two layer network and the three layer network on the new data. Create scatter plots for both results in the same manner as before.
  - How well do the two networks perform on the new data?
  - Which of the two networks performs better on generalization?
  - Specifically, why does the one that performs better do so?



- d. Having analyzed the outcome of two networks on two different sets of data list several pro's and con's to using either network and explain with each one why it is the case that it is either a pro or a con.