Spring 4-2001

Simulated SCN: Biological Modeling with Neural Networks

Mark Eric Hixson
University of Tennessee-Knoxville

Appendix E - UNIVERSITY HONORS PROGRAM
SENIOR PROJECT - APPROVAL

Name: Mark Hixson

College: Engineering Department: Engineering Science

Faculty Mentor: Dr. Rebecca Prosser

PROJECT TITLE: Simulated SCN: Biological Modeling with Neural Networks

I have reviewed this completed senior honors thesis with this student and certify that it is a project commensurate with honors level undergraduate research in this field.

Signed: _____, Faculty Mentor

Date: 4/26/01

General Assessment - please provide a short paragraph that highlights the most significant features of the project.

Comments (Optional):

General Assessment

Senior Honors Project
Mark Hixson

Simulated SCN

This project merges engineering analysis with biological data gather techniques. The use of a neural network for pattern analysis is not new, but the area of application to biological data is quite unique. An intimate knowledge of neural networks was required to accomplish this project along with experience working in a biology lab environment and researching the biological mechanisms underlying circadian rhythms.

Simulated SCN:
Biological Modeling with Neural Networks

Mark Hixson
27 April 2001

Abstract: Multiunit recording experiments on the suprachiasmatic nucleus located in the hypothalamus have revealed some of the direct relationships between drug application and cell firing rate. This nucleus is associated with entrainment and maintenance of circadian rhythms in mammals. A neural network might help to predict and better understand the reaction a drug or drug combination will produce on this population of cells. Two drugs were isolated for this experiment, NMDA, which acts as an agonist on a subset of glutamate receptors, and (+) DPAT has been shown to interact with 5-hydroxytryptamine receptors, more specifically the subset of 5-HT_{1A} and 5-HT_7 receptors.

Data were taken from five different days of recordings and used to train a multilayer perceptron. Input data consisted of the time of day in minutes, a predicted output ten minutes in the future, NMDA drug concentrations, (+) DPAT drug concentrations, time since end of last NMDA application and time since last (+) DPAT application. Levenberg Marquardt training was used, due to its speed in training the large input/output files. It also consistently performed best in simulating the network against other training algorithms such as backpropagation. Once the network was trained in this manner, it was tested against a previously untrained data. The new data were still within the range of the network. The simulation appeared to be successful in predicting expected responses to certain drugs and contained an acceptable error compared to the observed output for that day. Two-drug interactions may not have been completely and properly mapped due to a lack of input data for those situations.

Table of Contents

Introduction: Neural networks came about through the application of a biological model in order to solve problems. It seems almost fitting that this field is applied back towards better understanding biology. Biological systems are inherently filled with complex interactions on a very small scale. Recent research into neuroscience has attempted to cut through much of this complexity. The brain is still a big black box system, which is far from well understood. Current research has delved into the black box to pull out specific components to attempt to ascertain what that component does on its own. Recordings from these components tend to contain a great deal of noise that can interfere with a clean interpretation of any data gathered. By running many experiments certain trends tend to emerge from the data. Usually it is easiest to interpret simple 1 to 1 relationships: drug X causes excitation. Still, even after numerous applications and trials, certain drugs and drug combinations fail to reveal any noticeable input/output correlation.

Chronobiologists have focused their energy on studying specific areas of the brain responsible for generating biological rhythms. Biological rhythms are "recurrent events within a biological system (usually an organism) (Aschoff 1981)." Those internal biological rhythms running on roughly a daily cycle, such as body temperature fluctuations, are called circadian rhythms (Refinetti 2000). The pacemaker that controls the operation of the circadian rhythms within mammals was discovered in the early 1970s via lesioning of experiments carried out on rats (Moore and Eichler 1972). It was discovered that upon lesioning the suprachiasmatic nucleus (SCN), located in the hypothalamus, previously rhythmic behaviors and rhythmic biological functions ceased to be rhythmic (Moore and Eichler 1972). This collection of neurons in the brain is the primary center responsible for controlling the biological rhythms of mammals. It

receives inputs from a variety of sources to adjust itself as conditions change. One example of such control comes from the optic nerve; light and dark cycles are very important for controlling the internal rhythms of an animal.

In our lab studies different drugs were applied in vitro to a 500 μm slice containing rat SCN. The experiments singled out for this project used NMDA and (+) DPAT. Limiting the scope seemed essential for early modeling and testing of the system. NMDA is an excitatory drug. It is not found in the body on its own, but it is an agonist for a neurotransmitter, glutamate, which is found in the body (Colwell et al 1990, Vindlacheruvu et al 1992). The acute effects of (+) DPAT remain unknown, however it is thought to act as an agonist at the 5-HT_{1A} receptor (Pauwels 2000).

In the lab it became apparent that NMDA applications led to reliable increases in activity for the SCN. The application of (+) DPAT, however, did not produce consistent effects. The combination of the two drugs at the 100 μM level usually resulted in a marked excitation immediately followed by a large inhibition of the firing rate. Often the system would recover from the inhibition over the course of a few hours. This response is both puzzling and difficult to explain. These relationships are unexplainable using linear models, therefore leading one to use a neural network to try to determine an approximating function. One major concern was whether this response was merely due to random noise.

By taking the lab data and training it into a neural network one could potentially filter through some of the noise and reveal the underlying relationship between the drug applications and the SCN firing rate. In addition, one could determine the feasibility for

future and more extensive system modeling using various algorithms. Potential applications of this technique are limitless in scope.

Objectives: The goal for this project was to partially simulate the acute actions of neurotransmitters on the electrical activity of the SCN through a neural network. Only two different types of drugs were used to train the network. The training period included only a small portion of the day. By giving the trained network an input matrix with time in minutes as well as the concentration of drug application as well as time since the last drug application it is hoped that it will provide an accurate output of SCN firing activity.

Methodology: Initially data from lab experiments were examined to single out those experiments with the fewest errors inherent in them due to noise and other externally caused events. A clean experiment with clear input data and relatively clear output data was sought. In the end five particular experiments were used to train the neural network. They all had similar ranges for their times and all used only two different drugs, NMDA and (+) DPAT.

To best train the network, the input matrix was constructed to have size different input types. First the time of day was entered in minute intervals. Time zero corresponded to the beginning of the day for the rat, 7 AM. Time 60 would then be 8 AM, and so on until time 1439, which would correspond to 6:59 AM. For this set of data though no time exceeded 720 minutes.

The second input was the predicted value for y (SCN activity) 10 minutes after the current time. This is one input, which could be adjusted to attempt to better map the

inputs and outputs. Maybe a larger or smaller prediction step-size would be ideal. The initial results were within an acceptable range to not warrant the added time to adjust this variable although at some point it would be wise to experiment with this value (Uhrig and Tsoukalas 1997).

The third and fourth inputs were both drug concentration values applied at the time given by the time input. The concentrations were based on a micro molar scale: 100 = 100 μM. The third input corresponds to the NMDA concentration and the fourth input corresponds to the (+) DPAT concentration.

The fifth and sixth inputs are again times. These inputs are also in minutes, just as the first. The times correspond to the time since the application of the drug and are reset whenever the drug is applied again. This input is included because some of the influences of these drugs on the system seem to take place long after their application. It seemed necessary to provide the network with inputs as to how long ago a treatment was applied. Especially if training is to occur with a non-time based network, one without memory, then the inputs would need to provide the necessary information as to past events (Uhrig and Tsoukalas 1997).

Another set of inputs considered was a pair that would relate the concentration of drug just applied. This would maybe help in the delayed response recognition. The time since an application is an input but will the network know to respond differently to a 10 μM vs a 100 μM application of (+) DPAT? These inputs were not considered since most of the input data were at a single concentration. Only a very few situations were used that had an input different from 100 μM for the drug concentration.

The output was in the form of average cell firing rate for a minute interval. This output was scaled to a 0 to 1 scale, since the firing rate could vary from day to day based on different factors and no one day's firing rate should have anymore weight than another day's rate. The values used for the prediction input were also scaled in a similar way to provide a standard scale for the SCN activity. No other input values were scaled at all since most of them were time values.

At first the different days were placed in their own individual input and output matrices. This was done so that each could be trained and then tested against expected values. This process helped to figure out which type of network to best use with the system. The types of input and output matrices were well suited to training with a multilayer perceptron. A time delay neural network or even a recurrent network may have worked for training these data. But the MLP seemed the most reliable and best-understood option in this situation. Something was needed that could approximate a very complex nonlinear function (Uhrig and Tsoukalas 1997).

The data were trained using the Levenberg Marquardt training algorithm. It is a fast and fairly reliable method for training data. With the large output/input matrices it seemed ideal to pick a faster training method. Some attempts were made at training with backpropagation. These attempts never matched the accuracy of the LM training when the network was simulated.

Training took place assuming an SSE based off of a 10% error per output value. For a single day's worth of data, the SSE was typically around 5 or less. For the total merged matrix the SSE was around 23 typically. There was always a danger of training

too much noise while also not allowing the network to miss important relationships. The SSE had to be able to pick up trends but not too much noise (Uhrig and Tsoukalas 1997).

The number of neurons in the neural network seemed to have little influence on the training so long as the number was typically kept between two and six. As long as the error goal was kept the same the simulation seemed to stay generalized. Still, the network was typically trained with either two or three hidden neurons. Hidden neurons do the majority of the work within the neural network as opposed to input and output neurons, which merely feed data to and from the hidden neurons (Uhrig and Tsoukalas 1997). The fewer neurons did seem in the end to best generalize for unseen data.

During training the momentum and learning rate were left unaltered from their default values in the lmtrain program: lr=0.03 and momentum=0.9. The maximum number of epochs was kept to within 200. There never seemed to be any drastic changes in SSE above this point. If the network was going to train the data it was usually below this value. It was set as the ceiling to save time. The hidden layer was a logsig layer and the output layer was linear. The difference between logsig and tansig would have probably been negligible, but it seems that logsig would have an advantage since it operates between 0 and 1 and the outputs are also scaled between 0 and 1. It is doubtful that the choice of logsig or tansig would have made any major difference (Uhrig and Tsoukalas 1997).

Once a few of the network input and output matrices were tested and the best training and simulation method were developed for known data, all of the input matrices were merged into one large input matrix as were the output matrices. The total number of data points per input or output column numbered about 2300. These merged matrices

were tested with known data using lmtrain and the networks were subsequently simulated using simuff.

For situations where the outputs were supposed to be known it was easy to test the simulation of the network because an input matrix could be used that contained known predicted values of the output. However, in simulating the network for an unknown, these predicted values were completely unavailable. In the final testing of the network, these predictions were simply random numbers and as the network trained these random values were replaced with simulated values based on the random values. The network was then trained again and the simulated values were tested against the predicted values used for the training. If there was a large discrepancy then the predicted values were replaced with the simulated values and the network was tested again. This process looped until the error between predicted and simulated values was reduced below a certain level. This simulation approach worked fairly well for this network, but it could be modified to find the least error more quickly using different techniques, such as Newton's method. In the end though a very simple error-testing algorithm got the network to approximate the actual data to within an acceptable error range. Once the simulated and predicted values matched closely the simulated outputs were checked against the actual output values.

Results: The earliest results obtained from the network were in training the first inputs and testing those against the expected outputs. These results were used to primarily judge the effectiveness of the various network properties: learning method, number of neurons and SSE (Fig. 1, 2 & 3).

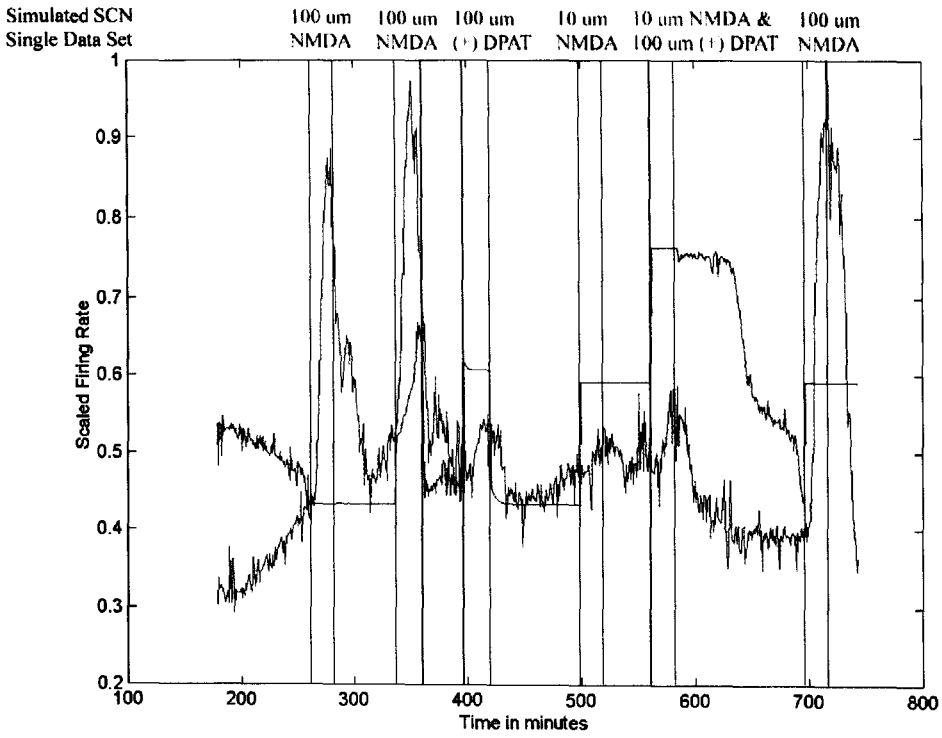

Figure 1: An example of training with too high as SSE~23. The red line represents predicted values while the blue line represents the actual data.

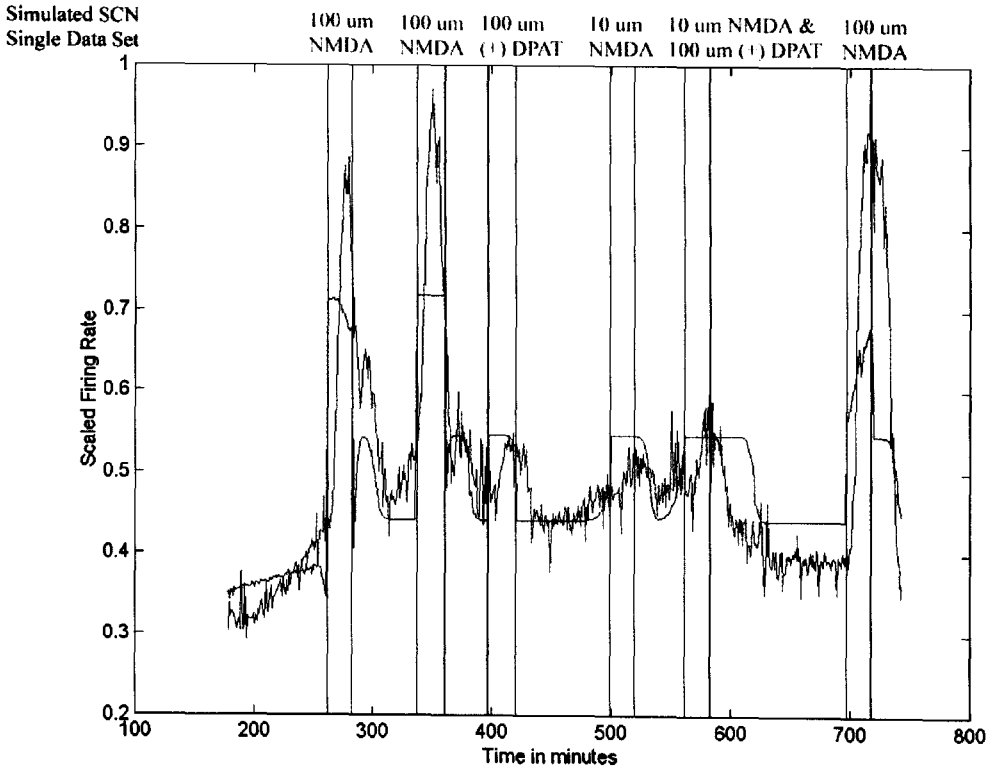

Figure 2: Training took place with 3 hidden neurons and an SSE of 5. The red represents the predicted value while the blue represents the actual data.

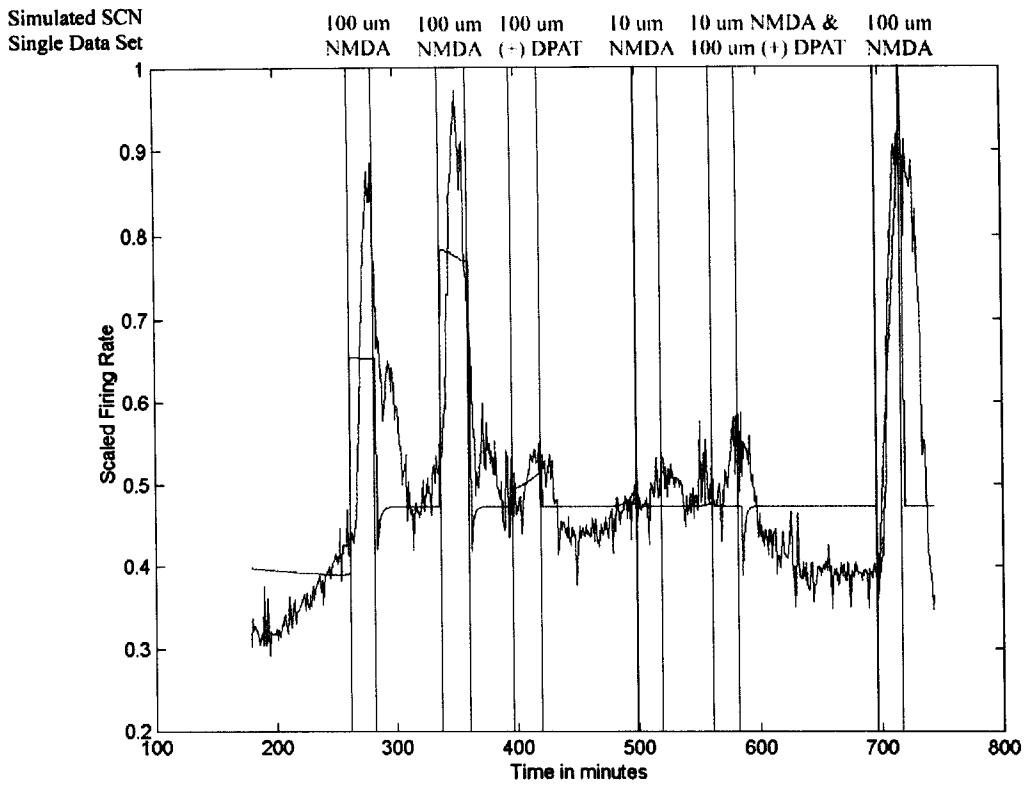

Figure 3: The same network as before only trained with 2 hidden neurons and an SSE of 5. Red represents predicted values and blue represents actual data.

In the data shown above it seems that the network trained best with a SSE around 5 and with as few as two or three hidden neurons (Fig. 1, 2 & 3). Among these examples the three-neuron run looks best when comparing the simulation to the actual data. Some of the actual SCN responses to drug applications were filtered out in the two neuron run (Fig. 3). During the 500 and 600-minute time intervals, there were two drug applications that were not simulated at all in the two-neuron run (Fig. 3). Still, this was not always observed to occur. Only in a few runs did the two-neuron system not work as well as the three-neuron network. For the most part there was little noticeable difference between the networks operating with 2, 3 or 4 hidden neurons (Fig. 2 & 3). Another observation about these data is that the simulated network seems to lead the actual data by about a minute (Fig. 2 & 3). This is probably due to the drug not having a directly noticeable

response from the brain slice. Usually it takes a minute for the drug to cause a noticeable effect.

Next, the individual inputs were combined into a single large input matrix. The known outputs were also combined into a large single output matrix. The merged data were trained and then tested against a known input/output pair that had been included in the training set (Fig. 4 &5).

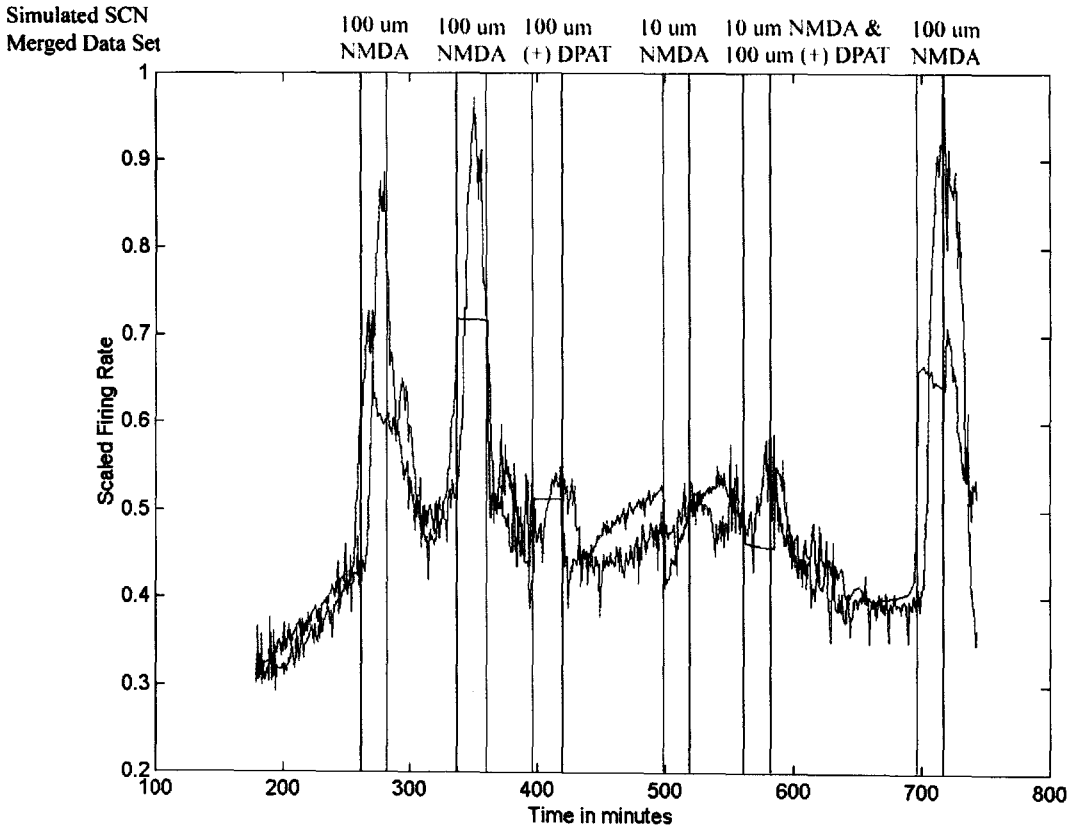

Figure 4: Merged data has been used to simulate a single day previously trained into the network: SSE=23 N=20. Red represents predicted values and blue represents actual data.

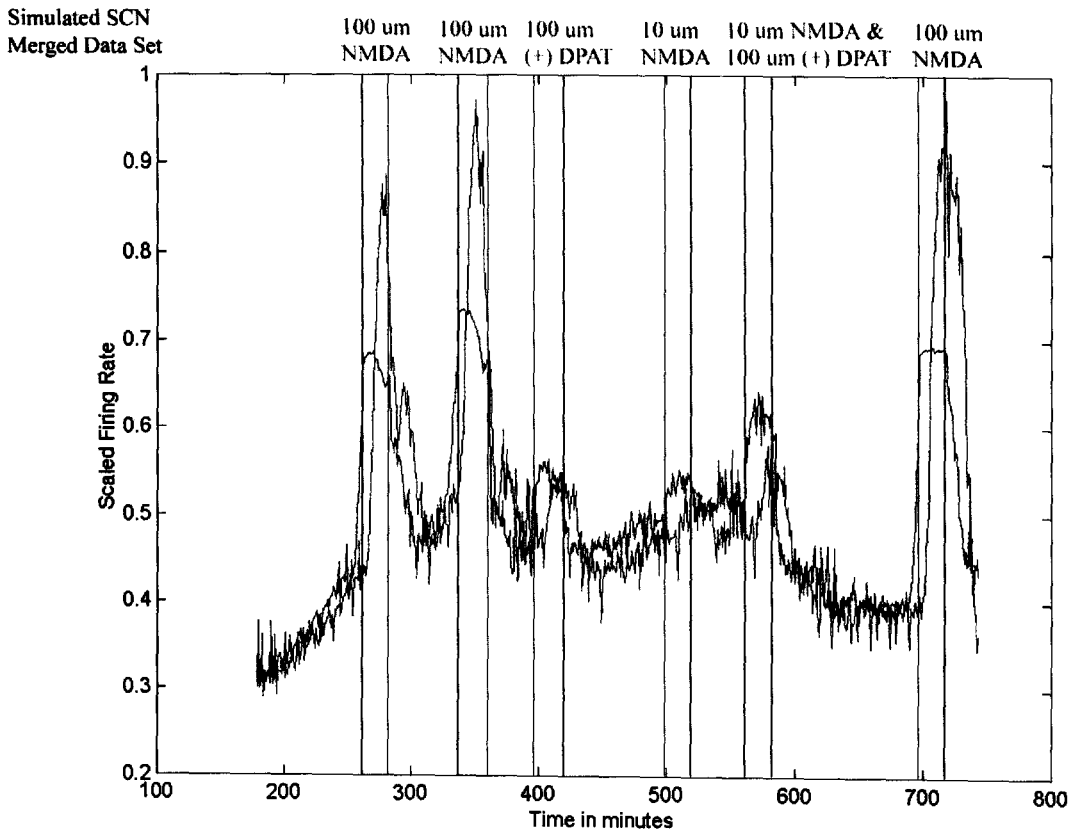

Figure 5: Merged data simulate a single recording day: SSE=23 N=3. Red represents predicted values and blue represents actual data.

There was little difference visibly between training the network with 20 neurons versus the standard 3 or 2 in this situation (Fig. 4 &5). This is probably due to the fact that the network has seen these data before and the SSE is at the same value for both training sessions, hindering the network from training too closely even when more neurons are present. It would be expected that the number of neurons has a larger influence on the network when it sees data for the first time.

Finally, the network was trained with the merged input/output matrix at an SSE of 23, and then was introduced to new data (Fig. 6 & 7). The data were still within the range of the network. It was necessary to loop the output as described previously so that

the network could reduce the error of the simulation and generate better predicted firing rate values to feed back into the network.

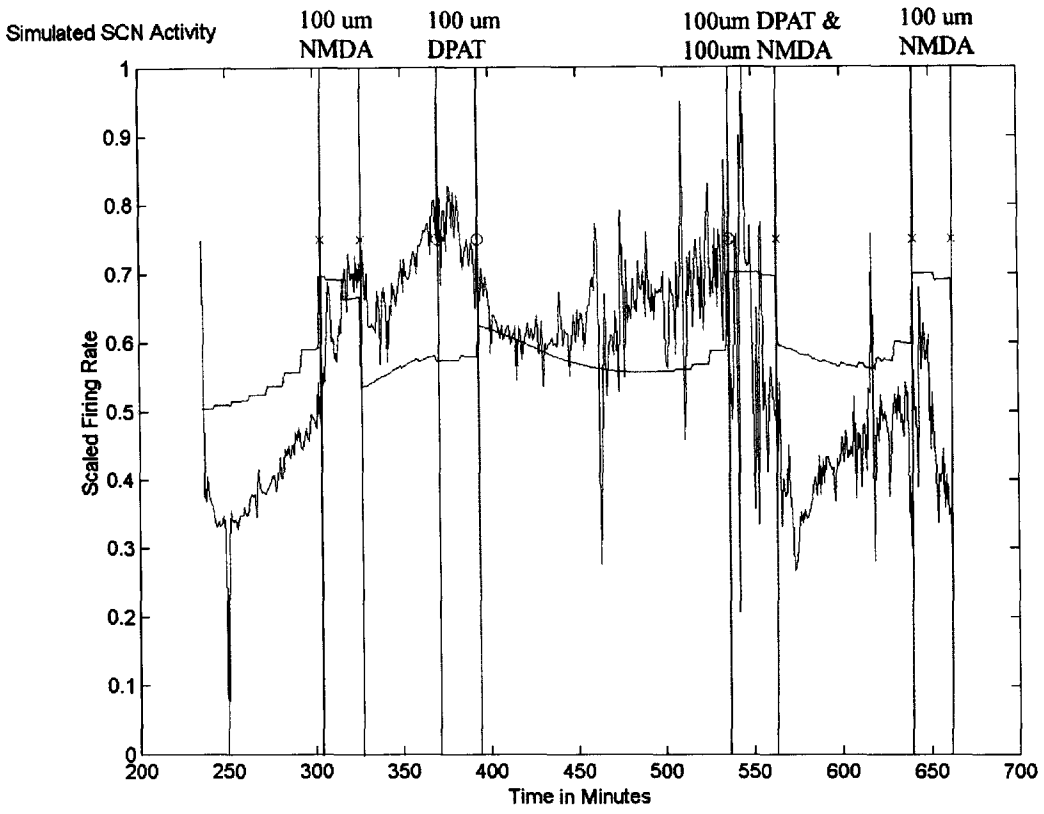

Figure 6: Graph showing Simulated SCN vs. actual SCN output: SSE=23 N=2 LM training. Red represents predicted output and blue represents the actual data.

This output from the simulated network looks very promising (Fig. 6). The training seems to have been successful in both generalization and preventing overfitting. The simulation does not follow the actual data exactly and it really shouldn't do that. There is always short-term noise inherent in these signals that could never be trained or predicted from day to day. It is good to see that the signals seem to follow the same base line; output scaling had something to do with that as well. The network responds in this run to the proper drug inputs. No response is really expected from DPAT although at

times it does seem to excite the SCN. One fault in the simulation that seemed unavoidable is in the randomization of the predicted outputs. Initially all of the predicted outputs are set to random values and are later replaced as the simulation generates its own predicted values. The way the simulation is designed, the last ten predicted values are always random. The simulation cannot generate these values so they must always be random values. This would cause the last ten data points to maybe be erratic. The simulation overall was a success.

On a side note, it became clear when simulating the network that the Levenberg Marquardt training algorithm was better suited to these data compared to backpropagation.

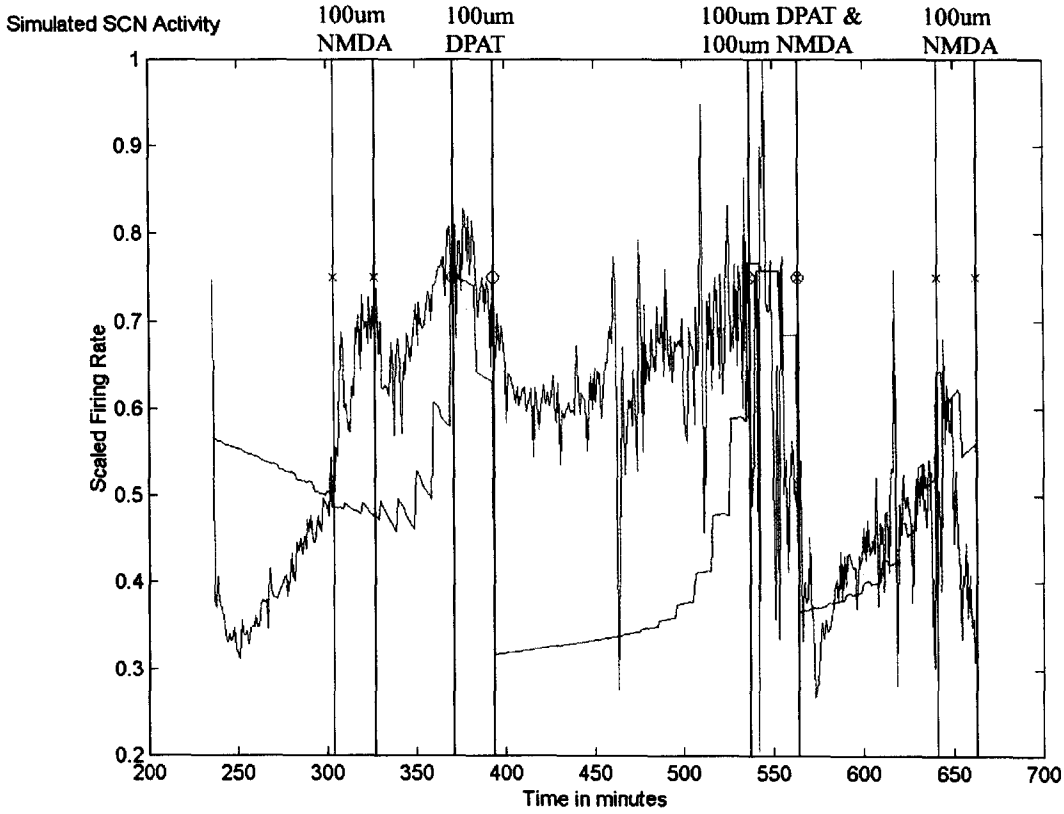

Figure 7: Simulation of the SCN when trained with BP: SSE=45 N=2. Red represents predicted values and blue represents the actual data.

It was nearly impossible to train the backpropagation network to an acceptable SSE. In the end an SSE of 45 was the lowest it could train to. Admittedly this is probably the reason for the poor matching of actual SCN versus simulated SCN outputs. Still, if the network cannot train to an acceptable level then it is not useful in simulating an output (Uhrig and Tsoukalas 1997).

Conclusion: The multilayer preceptron was able to successfully model the SCN in a limited capacity. The limitation in this situation was that only two types of drugs were trained into the network. Also the drugs were really trained in at primarily a single concentration: 100 μM. On a very small scale this network modeled a part of the brain. In order to create a full model of this or any part of the brain, tremendous amounts of data would need to be trained into a network. All the different types of neurotransmitters and other inputs would need to be trained at all the different concentrations that could conceivably be delivered to that part of the brain. One would also need to map the specific sites where these inputs were received by the particular brain component. The same drug applied to a different part of a slice could cause a different response.

The results from this project helped to more clearly show the SCN's response to input from NMDA and DPAT. It was hoped that a clear relationship between the interaction of these drugs in combination could be revealed. With more training data focusing on the combination of these two drugs, that interaction could be better realized. Overall that seemed to be the biggest downfall of this project, a lack of clear and good data. There were only five experiment days that were clear enough and also used a

limited number of drugs. A sixth day was withheld so it could be used in testing the simulation of the network.

Limiting the drug inputs to train to only two drug varieties probably helped in more quickly training the network and in helping to better analyze the results (Uhrig and Tsoukalas 1997). This project is more of a first step than it is the definitive solution to whether a part of the brain can be modeled with a neural network.

Scaling the outputs helped to keep all the input and output pairs equally weighted. Having the simulated output match the actual output so closely on their baselines was probably due to the scaling of the outputs. It is still unknown why the backpropagation failed to match as closely as Levenberg Marquardt when it came to the baseline of the simulation and the actual data. Both training algorithms seemed to match when they expected excitations, but they were on different baselines. Certainly by tweaking different properties of bptrain it would be possible to both speed up the training and increase its accuracy. Given time constraints it was easiest to stick with lmtrain, especially after looking at early simulation results, which indicated it was most accurate (Uhrig and Tsoukalas 1997).

Generating the final network simulation was the most telling part of the experiment by far. All of the work depends on how well the network can generate those outputs without having seen those specific data before. The Matlab program used in that portion of the experiment is very simple and slow acting, but it did generate a decent simulation. Alternative programs should be looked into for generating the best-simulated output. A genetic algorithm is one possibility. The SSE for the simulation compared to the actual was typically around 45. With more inputs for the network to observe it is

hoped this might be reduced. Also tweaking the simulation program could help to reduce this error (Uhrig and Tsoukalas 1997).

Fully translating a component of the brain to a series of computer algorithms, even if that site were incredibly small, would take a great deal of time and effort. This project was a small step towards a larger goal of better understanding the components of the brain and their interactions with each other. Creating a completely functional artificial brain component seems like an achievable goal.

References:

Aschoff, J., A survey of biological rhythms, *Handbook of Behavioral Neurobiology, Volume 4: Biological Rhythms*, New York: Plenum, pp.3-10, 1981

Colwell, C. S., Ralph M. R., Menaker, M., Do NMDA receptors mediate the effects of light on circadian behavior? *Brain Research* 523, pp.117-120, 1990

Moore, R. Y. and Eichler, V. B., Loss of circadian adrenal corticosterone rhythm following suprachiasmatic lesions in the rat, *Brain Research* 42, pp.201-206, 1972

Paulwels, P. J., Diverse signaling by 5-hydroxytryptamine (5-HT) receptors, *Biochemical Pharmacology,* v. 60, pp.1743-1750, Dec. 15, 2000.

Refinetti, R., *Circadian Physiology*, Boca Raton, Fl, pp.19-38, 2000.

Uhrig, R. E. and Tsoukalas, L. H., *Fuzzy and Neural Approaches in Engineering*, John Wiley and Sons, Inc., New York, NY, pp. 229-256, 1997.

Vindlacheruvu, R. R., Ebling, F. J. P., Maywood, E. S., Hastings, M. H., Blockade of glutamatergic neurotransmission in the suprachiasmatic nucleus prevents cellular and behavioral responses of the circadian system to light, *European Journal of Neuroscience*, v. 7, pp.673-679, 1992

Appendix:

```
%Generates the simulated network outputs for data collected on
10/30/00.
%The program then test the simulated against the actual output.
load dat1030
x1=x7';
a=size(x1,2);
x1(2,:)=rands(1,1:a);
err=100;
a=0;
hold off
while err>=0.001;
   a=1+a;
   ysim=simuff(x1,W1,B1,F1,W2,B2,F2);
   yp=x1(2,:);
   errc=0;
   for i=1:427;
      k=i+10;
      errc=((ysim(1,k)-yp(1,i))/ysim(1,k))^2+errc;
   end
   x1(2,1:437)=ysim(11:447);
   x1(2,438:447)=rands(1,1:10);
   err=errc;
erro(a,1)=errc;
end
ytst=ysim';
plot(x7(1:427,1),yans(1:427,1),
x7(1:427,1),ytst(1:427,1),'r',304,0.75,'x',327,0.75,'x',371,0.75,'o',39
4,0.75,'o',537,0.75,'x',537,0.75,'o',564,0.75,'x',564,0.75,'o',641,0.75
,'x',663,0.75,'x')
erra=0;
for i=1:427
   erra=((yans(i,1)-ysim(1,i))/yans(i,1))^2+erra;
end
errc
erra
title('Actual SCN Against Simulated SCN');
xlabel('Time in minutes');
ylabel('Scaled Firing Rate');
```

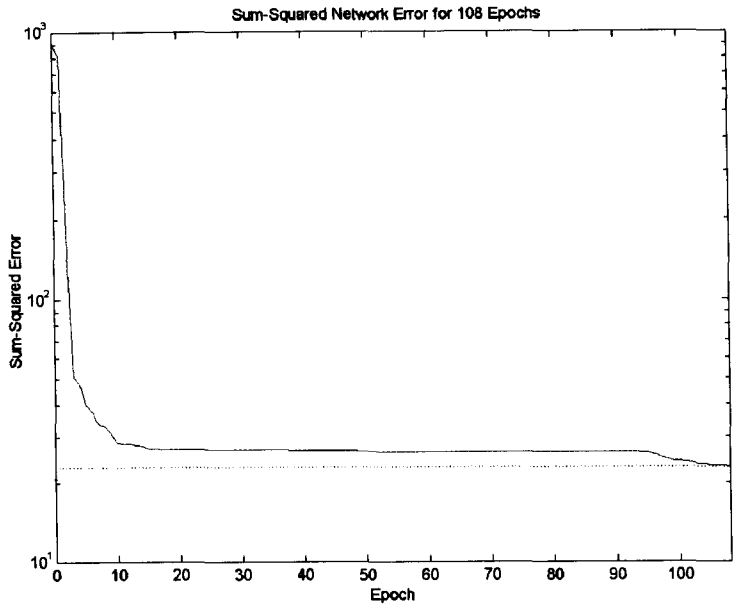

Figure 8: Graph for LM training with the merged data: SSE=23 N=3.

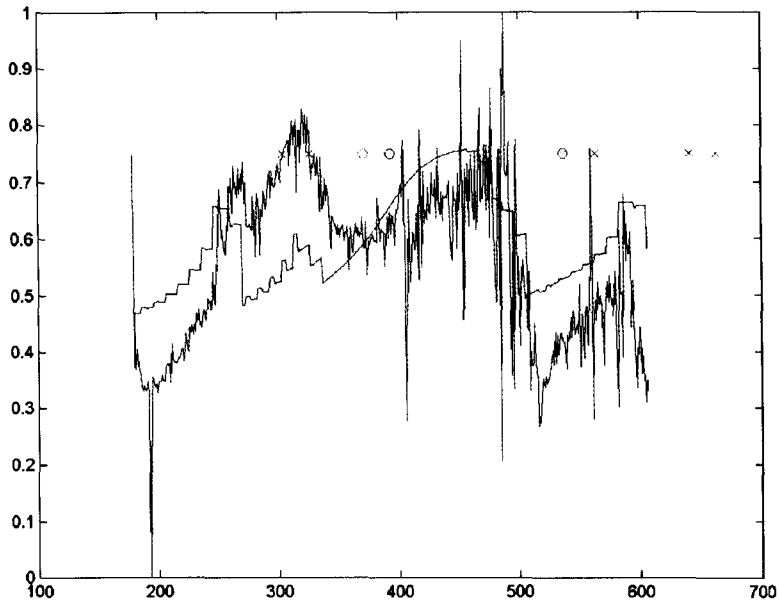

Figure 9: Final simulated output using two neurons for training. Red represents predicted output and blue represents the actual data.

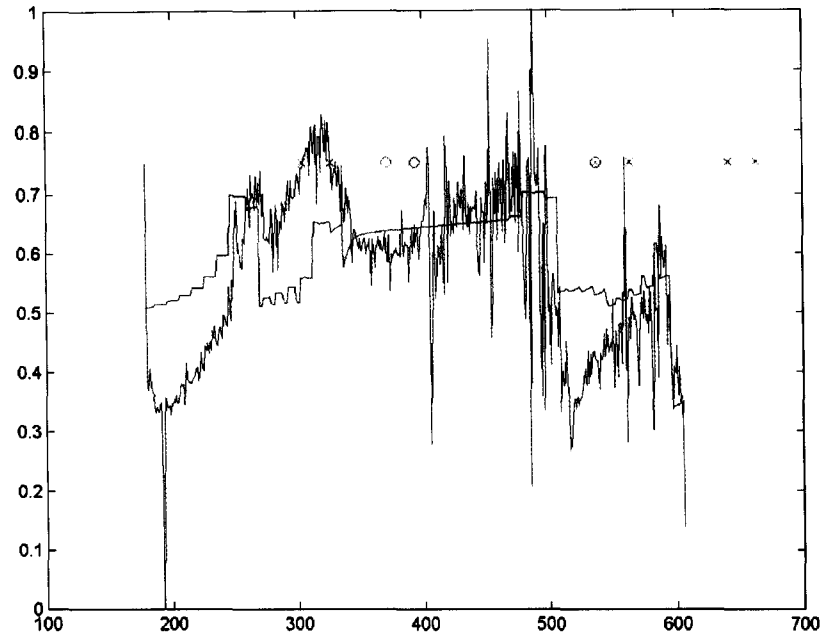

Figure 10: Final Simulated output using three neurons for training. Red represents predicted values and blue represents actual data.

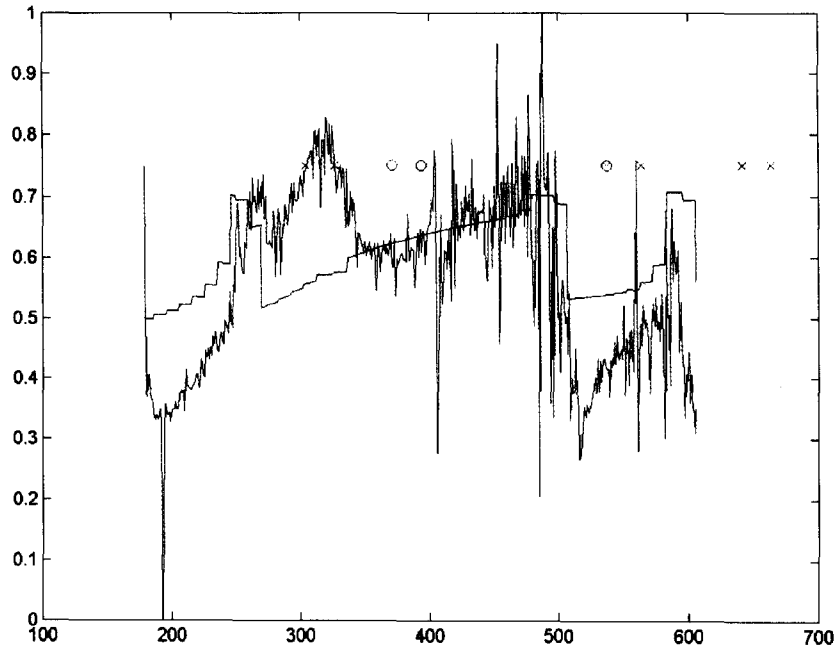

Figure 11: Final simulated output using four neurons for training. Red represents predicted values and blue represents actual data.