Research with an Undergraduate Student: Using Entropy to Assess the Training of a Neural Network

G. Beate Zimmer, Jeremy S. Flores and Alexey L. Sadovski

Department of Computing and Mathematical Sciences
Texas A&M University – Corpus Christi

Philippe E. Tissot

Department of Physical and Life Sciences
Texas A&M University – Corpus Christi

Abstract

This paper reports on enhancing undergraduate education by doing research with students. The work presented was done with a freshman mathematics major at Texas A&M–Corpus Christi. The student joined a continuing project of the Division of Nearshore Research which uses various models to predict water levels along the Texas coast. The most successful models use neural networks written in Matlab and are trained with a backpropagation algorithm. The training set consists of one year's worth of hourly water level and wind data. Initially the weights and biases of all neurons are assigned at random or with the Nguyen–Widrow algorithm. With these weights and biases, the forecast is computed and then compared to the actual water levels. In each training epoch, the weights and biases are updated following the gradient of the mean square error between calculated forecast and actual data. The size of the mean square error is the main quality criterion for the success of the training; if the mean square error is small enough, the training stops. Shannon's definition of entropy as the sum of terms –p*log(p) measures the entropy of a probability distribution. The probability distribution of the discrepancies between forecasts and actual measurements over the whole training year is used to calculate the entropy for the error distribution after each training epoch. Once the entropy ceases to change from one training epoch to the next, no new information is learned by the neural network and the training can end. Different forecast times, different locations, different training algorithms and different initial weights are used to illustrate the changes in entropy during the training of a neural network. This investigation of a very experimental character proved to be suitable for research with a freshman.

Introduction

Research with a Hispanic Freshman Mathematics Major

Often when thinking about recruiting a research student, we only consider students in the upper–level classes, assuming that they already have some background knowledge, a proven track record of academic performance and a clearer understanding of “research”. However, if the experiment of picking out a freshman succeeds, a wealth of positive effects ensues.

The faculty member can work with this student for up to four years - which is about the time equivalent of having a Ph.D. student - and this time span allows for serious progress. Faculty members at primarily undergraduate institutions often complain about the lack of research interactions with students and successfully including freshmen and sophomores in their research can bring continuity to their group and an opportunity for more in–depth work.

The student benefits by gaining individual attention and mentoring, as well as by learning outside the classroom and having publications. Such experience will likely have an influence in the student’s career choice. Research students also get financial support and sometimes the benefit of office space. A study[1]of science, mathematics, and engineering majors concludes: (p. 384) “All students, regardless of race or ethnicity, appreciated those departments and faculty who had given them a discipline–specific work–study opportunity or the chance to be involved with a research project. We found that research opportunities for any undergraduates were rare on all seven campuses. However, students who had been involved in departmental research were very enthusiastic about their experience and felt it greatly enhanced their interest in the field.” Another book[2] on retention management also stresses the importance of the faculty: “Retention management can not succeed without faculty input and advising. A school’s greatest attrition weapon is its faculty.”

The university gains, as research projects are part of a student–centered approach to retention. In “Keeping Students in Higher Education”[3], five forms of supportive retention practices are outlined: Emotional Support and sustenance, Informational support, Instrumental support, Material support and Identity support. An individual research project will include several of those practices. Retention can happen in two forms: retaining a student in academia or retaining a student in the chosen major. Most university wide efforts are centered at retaining a student enrolled, regardless of the major. Faculty efforts should address retention within the major. Starting in 1990, E. Seymour and N.M. Hewitt conducted a three–year study[1]of 335 students at seven universities to discover why undergraduates leave the sciences, mathematics or engineering (S.M.E.). On p. 32 they list “the most–commonly cited factors contributing to switch decisions, namely

- Lack or loss of interest in science
- Belief that a non–S.M.E. major holds more interest, or offers a better education
- Poor teaching by S.M.E. faculty
- Feeling overwhelmed by the pace and load of curriculum demands”

Most interestingly, the study does not find switchers and non–switchers to be two different kinds of people. Many of the concerns are shared by both groups. The authors report that “Science and Mathematics switchers more commonly left their majors because neither the career options and material rewards nor the personal satisfaction of careers open to them, appeared to justify the effort involved in graduating”.

A possible conclusion from these findings is that involving a student in research may retain a potential switcher within the major by adding interest, respect, personalized advising and sharing the enthusiasm for the subject. And starting retention efforts in a student’s freshman year is the optimal timing to not lose them to other majors or careers. Such efforts can be applied equally well, or targeted toward, minority students.

The setting of this student project

Texas A&M University–Corpus Christi ("The Island University") is a comprehensive four–year university with more than 8,000 students. A&M–Corpus Christi's proximity to water has enabled the university to become a hub of the latest coastal, marine and environmental research. For this public university, 37% of the student body consists of Hispanic students, a group whose retention is very important for a university in an overwhelmingly Hispanic region such as South Texas. This paper describes an example of research done with a Hispanic freshman at Texas A&M University–Corpus Christi during the Fall 2004 semester. While this is not part of a larger retention program, it may illustrate some possibilities of working with freshmen. The student, Jeremy Flores, attended Texas A&M University–Corpus Christi in the Fall 2004 and took three courses, two of them in Mathematics. One of the mathematics courses was Advanced Calculus, for which he lacked the formal prerequisites, but wanted to take regardless. In Fall 2003 he had attended MIT, but for health reasons did not finish his first semester there. In Spring 2004 he had taken some classes that sparked his interest, for example in music, and helped tutor mathematics in the tutoring center on campus. This is certainly not the usual background or situation of a freshman. Having been recruited in the Advanced Calculus class for a research project in Applied Mathematics, Jeremy was supported as a research student under a NASA grant which allowed him to be paid for up to 19.5 hours of research activity per week.

Research Set–up

The student was incorporated into an integrated research environment within the Division of Nearshore Research and the Department of Computing and Mathematical Sciences at TAMU-CC. Within the academic unit he was included in weekly research meetings with Dr. Zimmer, Dr. Sadovski and their research students. On the Division of Nearshore Research side, Dr. Tissot helped him understand the Matlab program for neural networks and gave him a share of an office to further interaction with other research students. Jeremy Flores was included in the meetings of the whole DNR research group and pointed towards the web pages for the Division of Nearshore Research (http://lighthouse.tamucc.edu) with past presentations and background material on water level forecasts. Jeremy was also given some literature to study, starting with a textbook on entropy[4].

Theoretical Background

Neural Networks

A neural network mimics the function of the human brain. It takes its inputs and processes them through a network of neurons, usually arranged in two or more layers. The neurons combine weighted inputs, add a bias and then apply a transfer function before giving their output as an input for the next layer. Figure 1 shows a schematic of a two–layer feed–forward neural network with two neurons in the first layer and one in a second layer, using transfer functions f and g. The weights are denoted by $\mathbf{a}_{i,j}$ and the biases are denoted by $\mathbf{b}_j$.

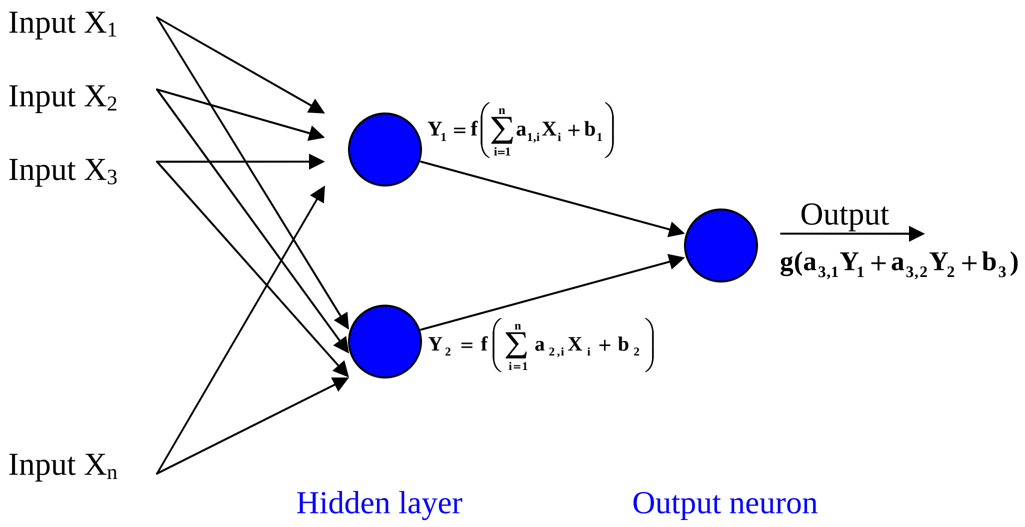
Input X_1
Input X_2
Input X_3
Input X_n
$Y_1 = f\left(\sum_{i=1}^{n} a_{1,i}X_i + b_1\right)$
$Y_2 = f\left(\sum_{i=1}^{n} a_{2,i}X_i + b_2\right)$
Output
$g(a_{3,1}Y_1 + a_{3,2}Y_2 + b_3)$
Hidden layer
Output neuron

Figure 1: Schematic of a feed–forward neural network with one hidden layer and transfer functions f and g.

Initially the weights and biases are assigned random values, which are balanced by the Nguyen–Widrow algorithm to avoid starting the minimization process at a local minimum. Then the network uses its training inputs and compares the output to the target outputs. After comparing, the weights and biases areadjusted to reduce the mean square error between outputs and targets. This process is called one training epoch. To work well, the network goes through a series of training epochs. There are a few conditions which will abort the training process, for example reaching a zero mean square error, a singular gradient matrix, or a repeating pattern of adjustments. Otherwise the decision of how long to run the training rests with the designer of the neural network. Usually the size of the mean square error is used as the decision factor on how long to train. But this one number does not always tell the full story of how accurate the predictions are.

Different training algorithms are implemented in Matlab. A slow but sturdy algorithm is the gradient descent method. The fastest algorithm is the Levenberg–Marquardt algorithm, which does not have to compute a Hessian Matrix, using only the Jacobian matrix of first derivatives of the network errors with respect to the weights and biases.

Dr. Tissot has written a Matlab program to produce water level forecasts along the Texas coast[5]. In Jeremy's work, he used a scaled–down version of this program, which only uses past water levels at one station as inputs, and does not test the network performance for other years or calculate all the performance criteria for the forecast.

One of the goals in forecasting is to have a forecast that is within 0.15m of the actual water level at least 90% of the time. We define Central Frequency (CF) as the percentage (represented by a decimal in the figures below) of forecasts that are within 15 cm of the actual water levels. MSE denotes the Mean square error between forecasts and water levels.

Entropy

From a discrete probability distribution with probabilities $\mathbf{p_1,...,p_n}$, one can calculate the entropy **E** according to Shannon's definition:

$\mathbf{E = -\sum_{i=1}^{n} p_i * \ln(p_i)}$ or equivalently, $\mathbf{E = \sum_{i=1}^{n} p_i * \ln\left(\frac{1}{p_i}\right)}$. The part $\mathbf{\ln\left(\frac{1}{p_i}\right)}$ is sometimes called the surprise factor: the smaller a probability is, the larger the surprise is when an event with that probability occurs. Entropy is the expectation of the surprise factor. Entropy units when a natural logarithm is used are nats. For a uniform probability distribution, $\mathbf{p_i = \frac{1}{n}}$ for $\mathbf{i = 1,...,n}$ and in that case, $\mathbf{E = \sum_{i=1}^{n} \frac{1}{n} * \ln(n) = \ln(n)}$. If only one outcome is possible, for example $\mathbf{p_1 = 1, p_2 = ... = p_n = 0}$, which is the case for a deterministic process, then we need the special definition $\mathbf{0 * \ln(0) = 0}$, which can be justified by L'Hôpital's Rule, to obtain $\mathbf{E = 1 * \ln(1) = 0}$. These are the two extreme possibilities; all other possible values of the entropy lie in between. For a normal distribution, the entropy is a function of the standard deviation: $\mathbf{E = \frac{1}{2} \ln\left(2\pi e \sigma^2\right)}$. Probability distributions with fewer nonzero probabilities have smaller entropy. Entropy measures the sharpness of the distribution, while change in entropy measures the change in information. If the entropy decreases, information is acquired.

Combining Entropy and Neural Networks

Dr. Sadovski had suggested the combination of the two ideas as a method of studying the training of a neural network. A literature search found that other groups have previously combined entropy and neural networks, but in different ways. An overview of interactions between neural networks and entropy can be found in a textbook[6]on the subject. One group[7]was studying how efficient a neural network is in using its neurons by calculating the entropy of the network. Pruning a neural network describes one possible design process, which starts with a large neural network and discards minimally used neurons until a smaller, more efficient network is obtained. For feature extraction via a recurrent Boltzmann machine without hidden units mutual entropy has been used[8] to extract statistically independent features without loss of information.

The student project

Project description

The project described here used an existing Matlab neural network program for water level forecasts and modified it with stops and calculations after each training epoch. The new program reads the training data, initializes the neural network and does one epoch of back–propagation training. One epoch of training refers to one update of the weights and biases for the entire neural network. After the training epoch, the program then simulates the neural network and calculates the forecast error for every hour of the training year. The resulting forecasting errors are grouped into 20 bins: (–10m, –1m), (–1m, –0.8m), (–0.8m, –0.6m), (–0.6m, –0.4m), (–0.4m, –0.3m), (–0.3m, –0,2m), (–0.2m, –0.15m), (–0.15m, –0.1m), (–0.1m, –0.05m), (–0.05m, 0m) and the corresponding positive error ranges.

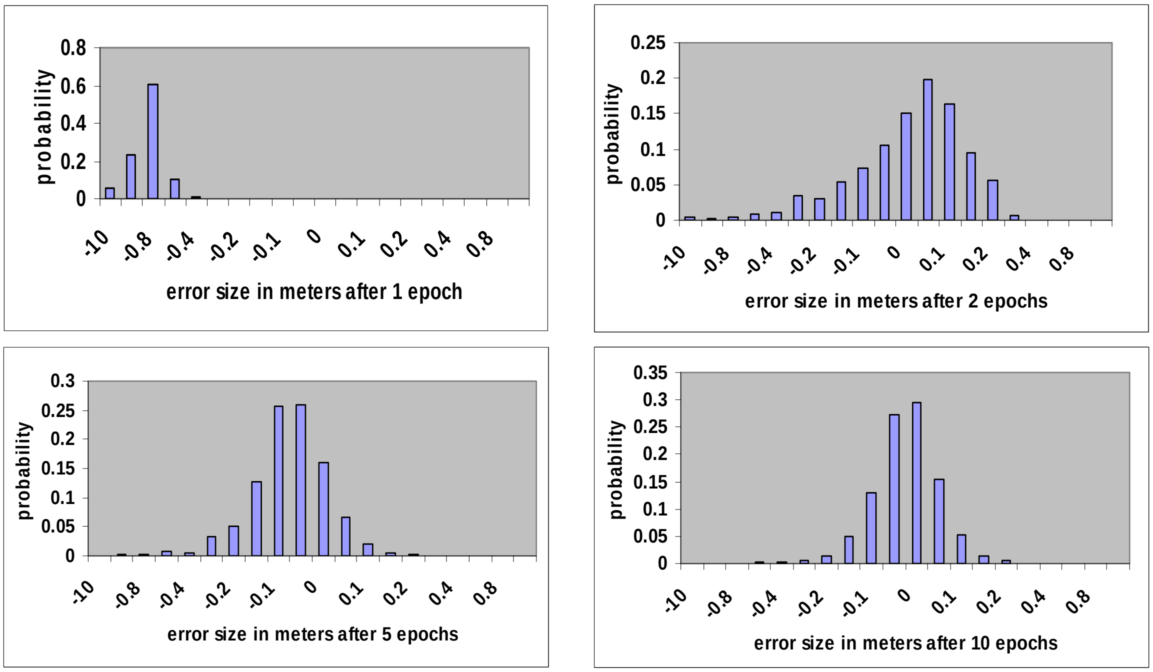

Figure 2. Error probability histograms for a neural network trained on 1998 data to forecast hourly water levels at Bob Hall Pier, shown after 1, 2, 5 and 10 training epochs, trained with the Levenberg–Marquardt algorithm.

The entropy can be calculated from the error histogram. The program writes the epoch number, number of past water levels used, forecast time, error distribution, entropy, MSE and CF into an array that at the end of the whole program is exported as an Excel spreadsheet.

Once the first training epoch is done and the error data stored, the network trains for a second epoch, without reinitializing the network. At the end of the second epoch, the entropy and other values are calculated and added as a new row in the array. Then the training process is continued following the same procedure. Training and simulation for 40 epochs in this fashion takes only a few minutes.

Part of the student’s assignment was to include loops into the existing Matlab program to calculate entropy at the end of the training process. The initial setup was to train for one epoch, and calculate the entropy of the resulting error distribution, then restart, train for two epochs and calculate the entropy, then restart and train for three epochs and so on. This did not produce satisfactory results, as the random choice of initial weights and biases produced unrelated distributions and the entropies showed no clear patterns. A next attempt was to start with an initial value of zero for all initial weights and biases. This zero initialization was found to be a local minimum for the mean square error and the training algorithm was unable to leave this minimum. Sometimes the initial guess is significantly better than it is at other times.

Finally, the spreadsheet that the Matlab the program created during the training is used to analyze the changes in entropy through Excel charts. The figures 2 to 6 were created within this setting.

The student's role

The student was given a working version of the neural network program. His first task was to understand how the program functions by performing some calculations for a different research project in which the effect of different performance functions on the forecasts generated by neural network models were studied. This included rerunning the program with different parameters and training functions. After the student could competently handle the existing program, he was asked to add loops to the program that would train for different numbers of epochs and calculate the entropy. The student himself had the idea of exporting the data on entropy, epochs and errors into an Excel spreadsheet. Not having taken a statistics class yet, he needed a bit of a background lecture on relative frequencies and probability, which he quickly absorbed. He also read literature on entropy himself and even suggested the use of mutual information as an improvement for this study. In each of the weekly meetings he had progress to show and he approached the project with great enthusiasm.

Here is how the student feels about this project: "I believe that I have gained much from my experience with the project. Working on research so early in my college career has given me a better qualification for any internships and summer programs that I might choose to apply for since I have had prior experience with something as complex as neural networks. Furthermore, I feel that this research was an excellent starting point for understanding AI systems, which is a field that I am considering as a career choice. The professors and research assistants proved valuable resources whenever I had problems with both theory and applications of the ANN, and they also provided me with online resources and printed publications which gave me a better background on subjects. The possibility for me to receive pay and the flexible schedule were also most welcome. Over all, it was - and still is - a pleasant experience that has bolstered my understanding of research projects and ANN structures."

Research findings

Intuitively, one expects that initially with random weights and biases, the forecast is far off and large errors account for the bulk of the histogram. Then the program adjusts the weights and centers the errors around 0, which will increase the entropy. Then a third phase happens: as the forecasts improve, the errors get smaller and the entropy decreases again. The example in Figure 2 above illustrates the change in entropy during the training of a neural network computing 12 hour forecasts for Bob Hall Pier based on 1998 hourly data.

In the spreadsheet exported from Jeremy's program it is possible to generate plots of the entropy, the Central Frequency and the mean square error of the forecasts as a function of the training epochs.

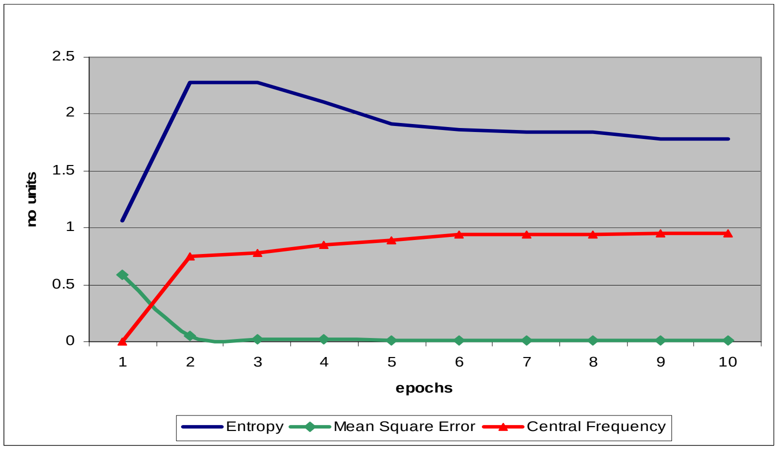

Figure 3. Graphs of the entropy of the error distribution, the mean square error and the Central Frequency of neural net water level forecasts for 1998 Bob Hall Pier data, trained with the Levenberg–Marquardt algorithm for 10 epochs. The central frequency is measured in decimals, not percent.

Figure 3 illustrates that entropy is a different criterion than the mean square error: by the nature of the training algorithm, the mean square error is (generally) a decreasing function, whereas entropy is not monotonic. This, however, only affects the early stages of the training process. Once the errors are centered at zero, both the entropy and the mean square error should decrease.

Figure 4 illustrates the usefulness of entropy over sole reliance on the mean square error. The example uses a different station and a different training algorithm.

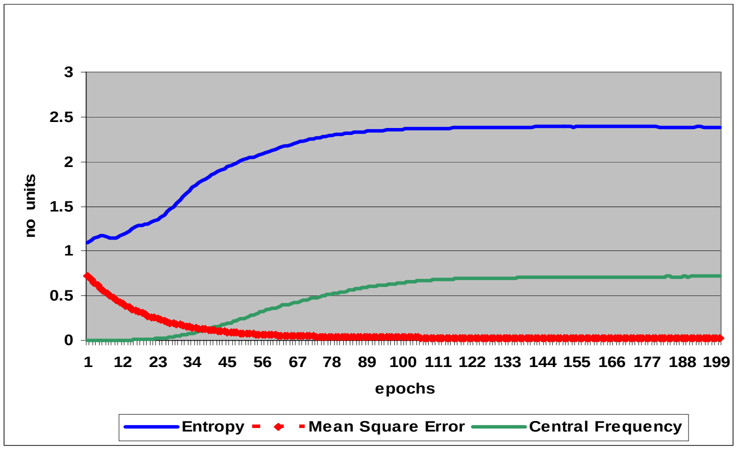

Figure 4. Graphs of the entropy of the error distribution, the mean square error and the Central Frequency of neural network water level forecasts for 1998 Morgans Point data, trained with the gradient descent algorithm for 200 epochs. The central frequency is measured in decimals, not percent.

In the example shown in Figure 4, the mean square error alone is not an indicator of sufficient training: even though the mean square error is relatively small, the entropy is still increasing, whereas towards the end of the training phase, the entropy should decrease. The increasing entropy is explained by the histograms in Figure 5, which show that after 20 epochs, the errors are all on the large positive side, and have just started spreading out, whereas after 200 epochs, the error distribution is wide and centered near zero. The width of the distribution will need to be decreased for a good neural network model.

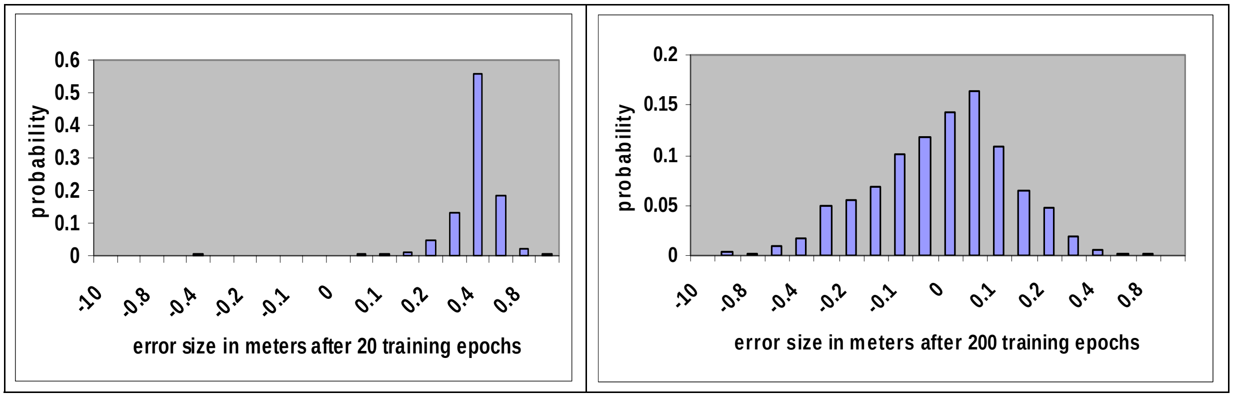

Figure 5. Error Probability Histograms for a neural network trained on 1998 data to forecast hourly water levels at Morgans Point, after 20 and 200 training epochs, trained with the gradient descent algorithm.

The difference between this example (Figures 4 and 5) and the first example (Figures 2 and 3) is that here we used a slower training algorithm and a station, Morgans Point, for which neural network models and other models have more difficulties computing accurate forecasts. In this example 200 training epochs were not sufficient, as the errors are still skewed to the right and have not yet entered the phase of reducing the standard deviation. The fact that the central frequency after 200 epochs is only 71.54% is another indication of likely insufficient training.

To illustrate the statement, a longer training session for the Morgans Point data was executed, training for 2000 epochs instead of the 200 in the previous example. The neural network had one hidden neuron and one output neuron and was trained using on hourly data for 1998. It used 12 hours of previous water levels as input data for a 12-hour forecast. At epoch 1500, the MSE is 0.0298, the CF is 71.58%, and the entropy is 2.3945 nats. At epoch 2000, the MSE is 0.02817, the CF is 72.31% and the entropy is 2.3788 nats. The entropy is decreasing during these 500 training epochs, but very slowly. Even after 2000 epochs, the distribution is not symmetric and hence not a normal distribution. The still decreasing entropy suggests that even 2000 epochs were not sufficient to train the neural network as well as possible.

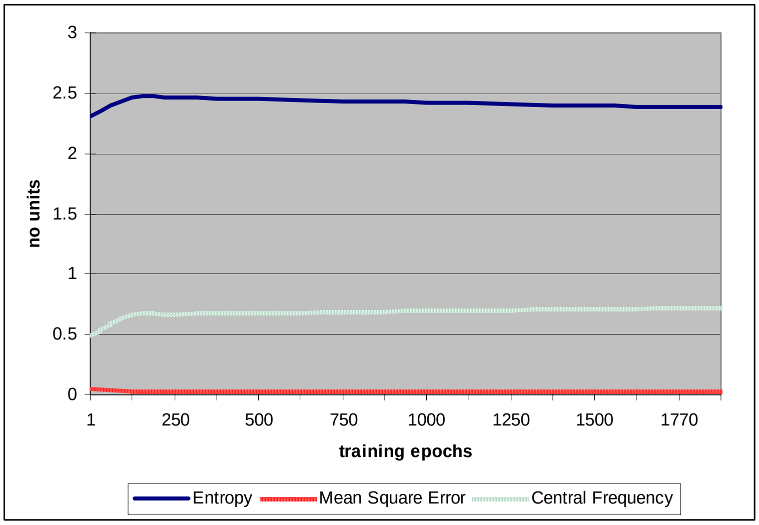

Figure 6. Graphs of the entropy of the error distribution, the mean square error and the central frequency of neural net water forecasts for 1998 Morgans Point data, trained with the gradient descent algorithm for 2000 epochs. The central frequency is measured in decimals, not percent.

Conclusions

Research with a freshman student can be mutually rewarding. Although Jeremy Flores may not be a typical student for this institution, other freshman students could quite conceivably also be nurtured into good research students on similar projects. The NASA grant that supported Jeremy Flores has also supported a number of other research students who were only a little further ahead in their undergraduate program. There is no official minimum qualification for a research student; the faculty member who assumes responsibility for the student needs to decide, on prerequisite information versus in-project learning. The student's comments on the project indicate favorable results on S.M.E. retention

A project with an experimental character rather than a purely theoretical endeavor may be better suited for such a purpose. The first few steps for Jeremy Flores consisted of operating an existing software program and learning how the code functions. No theoretical background was initially required, but as the project progressed, all sides felt the need for better knowledge of the existing literature. There was no prerequisite knowledge, but clearly a corequisite, which was fulfilled by discussions with the faculty members, with other research students or researchers and by independent reading. Questions from the student were encouraged and quickly answered.

Students are fast learners and can come up with good programming tricks. Jeremy Flores successfully adapted the existing neural network program to the specialized program needed for this project.

For the faculty members involved a student project asks for fast paced work; often things must be accomplished in just one semester. This is a great motivator to not rest on or laurels.

This student's work showed that the change in entropy can help to understand the training phase of a neural network better. Once the entropy decreases and then stabilizes, the useful part of the training is over and no new information is obtained. Entropy is not in a linear relation to the mean square error. This project, which lasted for less than one semester, has produced results that will be used in further development of neural network models for water level forecasts.

Jeremy Flores hopes to return to MIT and has applied for readmission at MIT for Fall 2005. He presented a paper on his work with different performance functions for the training of a neural network at the 13th Meeting of the South Texas Mathematics Consortium in Harlingen on February 19, 2005. When he leaves TAMU-CC, his research position will be offered to another student who shows enthusiasm and ambition to learn.

References

1. Seymour, E. and Hewitt, N. M., 1997, Talking About Leaving, Why Undergraduates Leave the Sciences, Westview Press, A Division of Harper Collins Publishers, Boulder, Colorado.
2. Dennis, M. J., 1998, A Practical Guide to Enrollment and Retention Management in Higher Education, Bergin& Garvey, Westport, Connecticut.
3. Moxley, D., Najor–Durack, A. and Dumbrigue, C., 2001, Keeping Students in Higher Education, Successful Practices & Strategies for Retention, Kogan Page Ltd., London, UK.
4. Saridis, G. N., 2001, Entropy in Control Engineering, Series in Control and Intelligent Automation, Vol. 12, World Scientific Publishing, Singapore.
5. Tissot P., Cox, D., Sadovski, A., Michaud, P., and Duff, S., Performance of Water Level Forecasting for the Texas Ports and Waterways, proceedings of the PORTS 2004 Conference, Houston, TX, May 23–26, 2004.
6. Deco, G., Dragan, O., 1996, An Information–Theoretic Approach to Neural Computing, Perspectives in Neural Computing, Springer, New York.
7. Barlow, H., Kaushal, T. and Mitchinson, G., 1989 "Finding Minimum Entropy Codes". *Neural Computation*, Vol. 1, pp. 412–423.
8. Ng, G. S., Wahab, A. and Shi, D., 2003, "Entropy Learning and Relevance Criteria for Neural Network Pruning", *International Journal of Neural Systems*, Vol. 13, No. 5, pp. 291–305.

Biographical Information

G. BEATE ZIMMER is a Visiting Associate Professor of Mathematics in the Department of Computing and Mathematical Sciences, Texas A&M University–Corpus Christi, 6300 Ocean Drive, Corpus Christi, TX 78412. Her research interests are Mathematical Modeling, in particular neural networks, as well as applications of Nonstandard Analysis to Functional Analysis.

PHILIPPE E. TISSOT is an Assistant Professor of Physics and Physical Science in the Department of Physical and Life Sciences, Texas A&M University–Corpus Christi, 6300 Ocean Drive, Corpus Christi, TX 78412. He is a member of the research faculty at the Division of Nearshore Research, and the Center for Water Supply Studies and applies artificial intelligence techniques such as artificial neural networks to the modeling of environmental systems.

JEREMY S. FLORES is a freshman mathematics major at Texas A&M University – Corpus Christi, 6300 Ocean Drive, Corpus Christi, TX 78412. He is supported by a NASA grant through the Department of Computing and Mathematical Sciences to work on modeling water levels with neural networks.

ALEXEY L SADOVSKI is a Professor of Mathematics in the Department of Computing and Mathematical Sciences, Texas A&M University – Corpus Christi, 6300 Ocean Drive, Corpus Christi, TX 78412. He is one of the principal researchers in the TAMU – CC NASA project, in the Division of Nearshore Research and Interim Director of the Center for Statistics and Quality Improvement Services.