Identification of Low Frequency Patterns in Backpropagation Neural Networks

Lucila Ohno-Machado, MD, MHA
Section on Medical Informatics, Stanford University School of Medicine
Stanford, CA 94305 machado@camis.stanford.edu

Although neural networks have been widely applied to medical problems in recent years, their applicability has been limited for a variety of reasons. One of these barriers has been the inability to discriminate rare classes of solutions (i.e., the identification of categories that are infrequent). In this article, I demonstrate that a system of hierarchical neural networks (HNN) can overcome the problem of recognizing low frequency patterns, and therefore can improve the prediction power of neural-network systems. HNN are designed according to a divide-and-conquer approach: Triage networks are able to discriminate supersets that contain the infrequent pattern, and these supersets are then used by Specialized networks, which discriminate the infrequent pattern from the other ones in the superset. The supersets that are discriminated by the Triage networks are based on pattern similarity. The application of multilayered neural networks in more than one step allows the prior probability of a given pattern to increase at each step, provided that the predictive power of the network at the previous level is high. The method has been applied to one artificial set and one real set of data. In the artificial set, the distribution of the patterns was known and no noise was present. In this experiment, the HNN provided better discrimination than a standard neural network for all classes. In a real data set of nine thousand patients who were suspected of having thyroid disorders, the HNN also provided higher sensitivity than its corresponding standard neural network (without a corresponding decay in specificity) given the same time constraints. I discuss the reasons why the sensitivity achieved by systems of divide-and-conquer hierarchical neural networks is superior to that of non-hierarchical neural network models, the conditions in which the algorithm should be applied, potential improvements, and current limitations.

NEURAL NETWORKS AND CLASSIFICATION

Neural networks, also known as connectionist systems, or parallel distributed processing models, are computer-based, self-adaptive models of artificial intelligence (AI) that were first developed in the sixties, but that reached great popularity only in the mid-eighties, after the development of the backpropagation algorithm by Rumelhart et al. [1]. Initially derived from neuroscientists' models of human neurons, neural networks now encompass a wide variety of systems (many of which have no intention to mimic the human brain). Classification, or pattern recognition, is one of the most common uses of neural networks in medicine, and is usually implemented as a supervised machine-learning method because the system needs a set of training cases to estimate its internal parameters, or weights. Typically, no rules or other traditional AI knowledge representation schemes are used in neural networks (with the exception of hybrid models). Figure 1 shows the basic components of a neural network. Input values are multiplied by weights that are adjusted iteratively every time a set of patterns is presented. The results of the multiplication are passed through an activation function in each "hidden" unit of the intermediate layer of nodes (in our figure, the activation function is the sigmoid). The activation values for the units of the "hidden" layer will then be multiplied by the weights of the second layer, and the results of these operations will subsequently pass through the activation function of the output layer, providing the final solution.

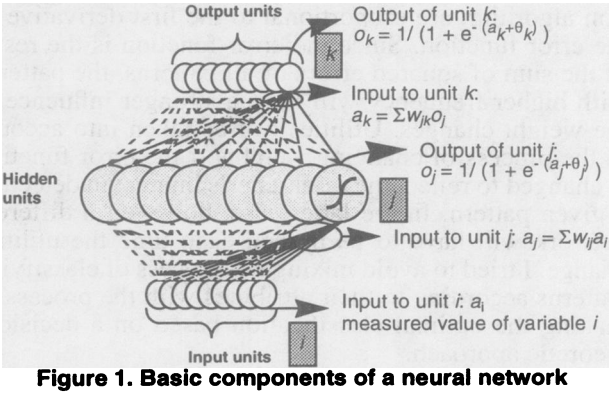

Figure 1. Basic components of a neural network

Usually, the output node that has the highest activation at the end of the training phase will indicate the predicted category. In a classification application, inputs are generally composed of the attributes of each instance in a data set, and outputs constitute classification categories. For example, a neural network that was designed to classify patients suspected of having thyroid disease, such as the one depicted in Figure 2, may have as inputs laboratory values, history data, and physical-examination items. The outputs are the classes *hypothyroidism*, *hyperthyroidism*, and so on. In medicine, neural networks have been used in many different applications, such as automated diagnosis of myocardial infarction [2], prediction of length of stay in intensive care units [3], decision support for assessing the adequacy of weaning patients from ventilators [4], prediction of the mechanism of action of new drugs [5], and radiology applications [6,7,8,9]. The backpropagation algorithm for supervised classification is the most frequent algorithm employed in medical applications of neural networks [10].

The backpropagation algorithm applies a steepest-descent (or hill-climbing) method to minimize an error function, and therefore it inherits steepest-descent's well-known problems: the existence of local minima, the possibility of having multiple solutions, and the difficulty of assuring that the solution found is optimal. Nevertheless, none of the limitations mentioned above has prevented backpropagation-based neural networks from being useful in a variety of real-world settings. However, researchers should fully understand the limitations of the backpropagation algorithm and its multiple variants to benefit maximally from its use.

0195-4210/94/$5.00

RECOGNITION OF RARE PATTERNS

Even though researchers in medical informatics are often looking for low frequency data or rare patterns, the latter are difficult to recognize in certain types of machine learning methods, including backpropagation-based neural networks. The difficulty is often due to the fact that the utility of a classification is not taken into account by the methods employed, and that the error that needs to be minimized is not weighted accordingly. The standard error function to be minimized in a back-propagation-based neural network is usually

$$E(w) = \frac{1}{2}\Sigma[\zeta_i - O_i]^2 \quad (1)$$

where w is the weight matrix, ζ_i is the expected output for pattern i, and O_i is the output provided by the network [11]. The changes in weights in the backpropagation algorithm are proportional to the first derivative of the error function. Since the error function is the result of the sum of squared errors of all patterns, the patterns with higher frequency will have a stronger influence in the weight changes. Utilities can be taken into account in the process of changing weights if the error function is changed to reflect the researcher's interest in detecting a given pattern. In the latter case, however, a different network will have to be trained each time the utilities change. I tried to avoid mixing the process of classifying patterns according to their attributes with the process of making the optimal classification based on a decision-theoretic approach.

Machine-learning methods of classification provide inexpensive means to perform classification. Backpropagation-based neural networks are able to perform classification reliably, provided that the frequency of the relevant patterns is not low. With the increasing number of electronic clinical databases, and the increasing costs of manual processing, it is likely that machine-learning applications will be necessary to detect deviant procedures and unexpected outcomes. These patterns are infrequent, but their detection is important. Therefore, there is a need to enhance the predictive power of the machine learning methods, especially the detection of low frequency patterns, without a decrease in specificity.

Traditional classification methods, such as linear-discriminant analysis, also have difficulties in detecting infrequent patterns [12]. If the variability of the most frequent classes is high, then a rare class may be considered just another instance of the most frequent class, and no discrimination will be possible. On the other hand, if all classes are equally represented and they are separable (linearly separable, if the simplest form of neural networks — the perceptron — is used), then the neural network should be able to make the distinction. A large number of medical applications in which classification is desired have the goal of discriminating a pattern of low frequency (e.g., "thyroid disease", "bad prognosis") from a pattern with high frequency (e.g., "no disease", "good prognosis"). For example, if only a very small group of patients who have undergone by-pass surgery have prolonged lengths of stay in hospital, this category will hardly be recognized by most machine learning methods. These are, however, exactly the patterns that need to be studied and followed more closely. Another example is screening for certain diseases, which is considered beneficial even when the prevalence of the condition is low but the overall benefit of detecting a case justifies the costs (e.g, screening for congenital hypothyroidism, a disease that has a prevalence of 1/4,000) [13]. Unless neural network applications address the problem of discriminating low frequency patterns, their use in medical applications will not scale up to useful real-world applications. The issues of (a) considering the utility of a classification and (b) creating mechanisms to allow the discrimination of rare patterns must be addressed. In this article, I will focus on the latter.

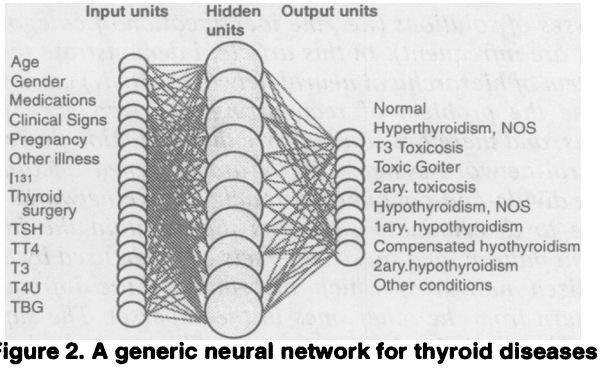

Figure 2. A generic neural network for thyroid diseases

HIERARCHIES OF NEURAL NETWORKS

The HNN is an architecture of neural networks in which the problem is divided and solved in more than one step. Figure 3 shows how a hierarchical system of neural networks should operate: the first classifier, or Triage Network, divides the data set in smaller subsets, which will then constitute the inputs for the Specialized networks.

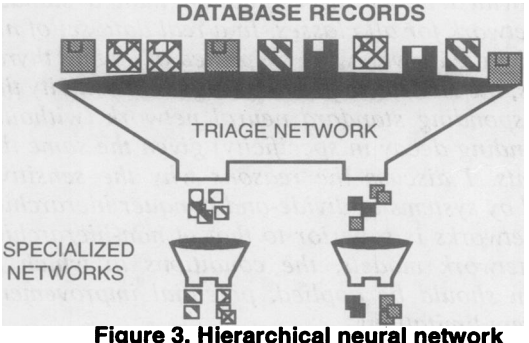

Figure 3. Hierarchical neural network

Electronic data from medical records are entered in a Triage network. This network filter instances that should be further processed by Specialized networks.

The application of multilayered neural networks in more than one step allows the prior probability of a given pattern to increase at each step, provided that the predictive power of the network at the previous level is high (i.e., the area under the ROC curve is greater than 0.5). For example, suppose a researcher needs to discriminate four categories of patterns in a given data set. Among the patterns, there exists one that corresponds to only 1 percent of the patterns. The other categories have prior probabilities of 5, 44, and 50 percent. By applying a classifier that can reliably discriminate a set of two categories from the other patterns, and applying another classifier to the results of this pre-classification, the total number of patterns in the second step is decreased, and consequently the frequency of a given pattern increased. This increase in frequency allows a hierarchical neural network classifier to discriminate patterns faster, as I will demonstrate. The hierarchical model assumes that

the first classifier is able to discriminate a superset of some categories, which includes the desired one, from the other ones. Since in any of these reliably constructed supersets the prior probability of a category in the set is higher than that of the initial sample, this process will yield higher posterior probabilities for the desired class than the one used by the classifier that attempts to make all distinctions in one single step.

Using Bayes rule, where X is a vector of attributes, and C_i is a category, we have:

$$P(C_1|X) = \frac{P(X|C_1)P(C_1)}{\Sigma P(X|C_i)P(C_i)} \quad (2)$$

In the two-category case, the equation becomes:

$$P(C_1|X) = \frac{P(X|C_1)P(C_1)}{P(X|C_1)P(C_1) + P(X|\neg C_1)P(\neg C_1)} \quad (3)$$

Assuming that $k_1 = P(X|C_1)$ and $k_2 = P(X|\neg C_1)$ are constants, and that $P(\neg C_1) = 1 - P(C_1)$, we can see in Eq.4 that whenever $P(C_1)$ is increased, the posterior probability $P(C_1|X)$ is also increased.

$$P(C_1|X) = \frac{k_1 P(C_1)}{(k_1 - k_2)P(C_1) + k_2} \quad (4)$$

Therefore, if the prior probability of a class is augmented in the training set and the sensitivity and specificity of the network remain unchanged, the posterior probability of the class is increased. In other words, if the Triage and the Specialized networks of the hierarchical system in Figure 3 each have the same number of weights as that of the generic system (and consequently the same potential for achieving the same sensitivity and specificity after training), they can perform better than the non-hierarchical system can.

This process confirms the intuition that if by any reason the prevalence of a pattern is increased, while everything else remains unchanged, the posterior probability of that pattern, given the same set of attributes, is increased. Therefore, if a Triage network is applied and is able to reliably discriminate a set that contains the desired pattern, an increase in the prior probability of that pattern will occur, also causing an increase in the posterior probability of that pattern in the corresponding Specialized network. The question remains whether Triage and Specialized networks with a smaller number of weights than that of the corresponding Generic network can also perform better than the non-hierarchical system. If the *total* number of free parameters (weights) in both systems is the same, the Triage and Specialized networks will certainly have *fewer weights* than the Generic network. The two following experiments were designed to answer this question.

EXAMPLE I: SORTING BINARY NUMBERS

In order to evaluate the power of HNN in classifying low frequency patterns, and to compare it to a standard neural network, I created an artificial data set using a known distribution. In the artificial data set, four categories (Category 0, Category 1, and so on) have to be discriminated. There were two attributes for each pattern, which constituted the binary representation of the number assigned to each of the classes ("00" was the pattern that corresponded to Category "0," "01" corresponded to Category "1," "10" corresponded to Category "2," and "11" corresponded to Category "3"). Each input unit corresponded to one digit of the binary number. All the units were binary. The inputs patterns, frequency of each type of pattern, and the expected output categories are shown in Table 1.

Table 1: Distribution of patterns for Example I

Pattern	Frequency	Output (Category)
00	44%	0
01	1%	1
10	5%	2
11	50%	3

I tested the hypothesis that the HNN could discriminate low frequency patterns earlier (i.e., requiring fewer training cycles) than a standard neural network could, provided that the systems had the same number of weights. Figure 4 shows how the hierarchical system of neural networks works. A standard feed-forward neural network that tries to classify the patterns in just one step was created for comparison. Classification in the HNN was done in a supervised manner in each step. The neural networks of the first-level (Triage networks) discriminate patterns 0 and 1 from patterns 2 and 3. The two networks for the second-level (Specialized networks), discriminate between patterns 0 and 1 and patterns 2 and 3, respectively. Note that the *total* number of weights in the HNN is the same as that of the standard neural network (i.e., the total number of parameters that needed to be estimated in each of the systems is controlled to be the same).

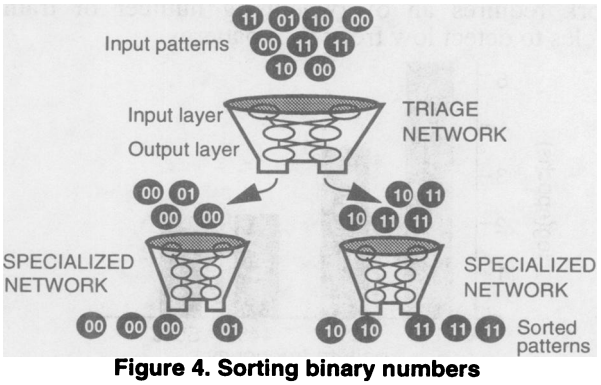

Figure 4. Sorting binary numbers

Table 2 displays the number of parameters to be estimated (weights), the number of training cycles (epochs), and the average time that each system took to converge to a perfect solution. A perfect solution was defined to be achieved when the activation of the correct output unit was at least twice that of the other output units. No noise was added to the data. Training was done by epochs. I performed 10 simulations for each system, starting with different initial weights. All networks were trained with a fixed learning rate of 0.01 and no momentum term. The overall time spent for making the perfect classification was significantly reduced ($p < 0.01$) with the use of HNN. I did not run Specialized networks in parallel, even though by doing so time could be reduced even more. It must also be taken into account that one epoch in the non-hierarchical network takes longer than one epoch in any of the networks in the hierarchical system, given the smaller number of weights in each of the networks of the latter, and the smaller number of pat-

terns in the Specialized networks.

Table 2: Comparison of systems for Example I

System	Units	Weights	Epochs†	Time ‡
Standard NN	10	24	148,791	50 min 53 sec
Hierarchical NN	18	24	14,623	2 min 37 sec
Perceptron	6	8	11,119	2 min 00 sec
Hierarchical Perceptron	12	12	6,437	36 sec

† Average of 10 runs. Refers to the detection of pattern 1.
‡ Average time on an HP9000 workstation. Considers longest epoch in the hierarchical system.

Although the nature of the problem allows a simple perceptron (a one-layered neural network) to converge to a solution, my study focused on the behavior of the backpropagation algorithm for multilayered neural networks. The perceptron's performance on this problem (see Table 2) was extremely good, as expected, but it would not be as good in the case of a non-linearly separable problem, as we will see in Example II. A multilayered neural network that has enough hidden units can approximate any function [14], and its applicability is therefore much broader than that of a perceptron. Furthermore, a hierarchical system of perceptrons also proved to converge faster than a standard perceptron did in this example.

Figure 5 displays the number of epochs (in fact, the logarithm of the number of epochs, given the orders of magnitude involved) required for the standard neural network to learn patterns that have different frequencies in Example I. As we can see, the standard neural network requires an overwhelming number of training cycles to detect low frequency patterns.

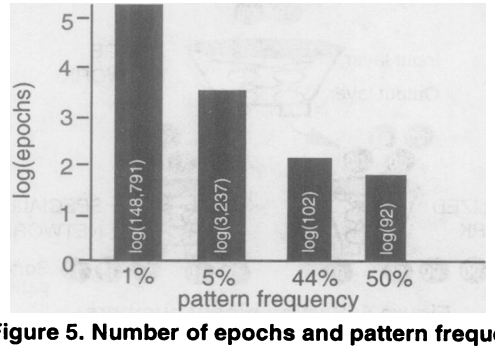

Figure 5. Number of epochs and pattern frequency

One might still argue that the pre-selection of subsets that were themselves linearly separable introduced a bias in favor of the hierarchical system. I also ran the same experiments dividing the subsets in a different way, such that patterns "00" and "11" would be separated from patterns "01" and "10" in the Triage network. This grouping would require that the Triage network would be able to solve a non-linearly separable problem first, and is by far the worst possible grouping: the Hamming distance between patterns in the same group is twice that of patterns in other groups. Furthermore, the proportions involved would require the Triage network to detect a subgroup that had a low frequency value itself (the patterns "01" and "10" constitute only six percent of the total number of patterns). The HNN exhibited a peculiar behavior: four of the ten networks converged to a solution after relatively few epochs (mean: 34,944), but the other six did not converge to a perfect solution even after $4x10^5$ epochs. This result indicates that the groupings should be done by similarity of features, rather than be based purely on pattern frequencies. Therefore, merging rare patterns that do not share similarities into a group simply to increase their frequency in the training set does not help. Patterns have to be similar for the Triage network to work.

Another experiment, in which the pattern distribution was changed to the one shown in Table 3, proved that the difficulties encountered by the Triage network were not related to the combined low frequency of the group "01" and "10", but to the fact that the similarities within the groups were low. None of the ten Triage networks built for this experiment converged to a perfect solution after $4x10^5$ epochs. Pattern similarity seems to be the key factor in determining the success of HNN.

Table 3: Another distribution of patterns for Example I

Pattern	Frequency	Output (Category)
00	1%	0
01	45%	1
10	5%	2
11	49%	3

Evaluation of a test set was not necessary in this artificial example because the categories were *defined* as being the decimal representation of the binary numbers. The systems would have exhibited the same performance on any test set composed of the same patterns, independent of their distribution. Overfitting was not a concern for exactly the same reason.

In order to determine whether (a) the difficulties that standard neural networks had to detect low-frequency patterns in the artificial data set would be reproduced in a real-world data sets, which often contain missing values and noise, and (b) the proposed solution would also be applicable in more complex problems, the following experiment was designed.

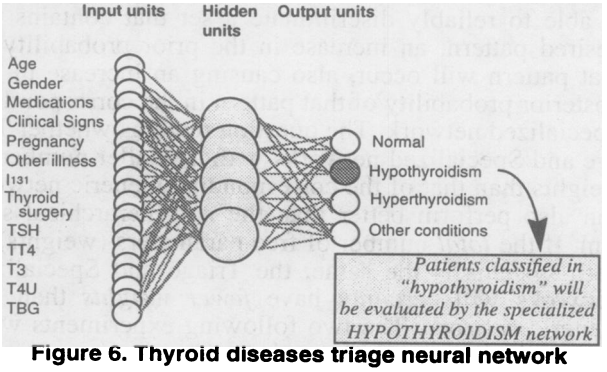

Figure 6. Thyroid diseases triage neural network

EXAMPLE II: THYROID DISEASES

I used a set of 9,172 patients suspected of having thyroid diseases, obtained from the data repository at University of California at Irvine [15]. The same data set was used by Quinlan to demonstrate the performance of decision trees in diagnosing hypothyroidism [16]. I used a subset of 4,586 patients to train the networks. A standard neural network discriminated ten different diagnoses. It consisted of 22 inputs, 10 hidden units, and 10 outputs. The standard neural network, or Generic network, was shown in Figure 2. In the HNN, the Triage network was dedicated to discriminate patterns of

hypothyroidism, hyperthyroidism, normality, and other thyroid conditions. The rationale for establishing these groupings was based on the assumptions that (a) patients in each group shared similar attribute values, and (b) even if not all the specialized networks were able to refine the solution and obtain a final diagnosis, the partial diagnoses provided by the Triage network could be clinically useful. Figure 6 shows the Triage network. The Specialized network for hypothyroidism, shown in Figure 7, takes as inputs all patients that were classified in *hypothyroidism* in the Triage network and discriminate the patterns of *primary hypothyroidism, secondary hypothyroidism, compensated hypothyroidism, and hypothyroidism not otherwise specified.*

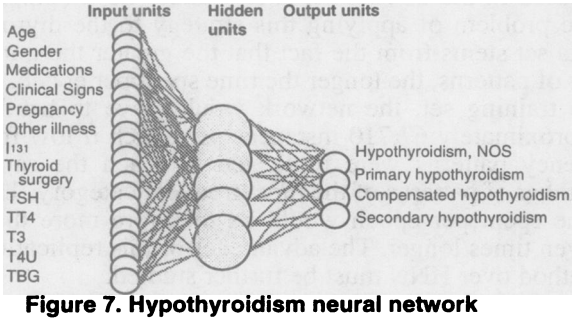

Figure 7. Hypothyroidism neural network

Table 4 shows the distribution of the output categories in the training set. Some patients had more than one diagnosis. Input attributes included age, gender, current medications, pregnancy status, previous thyroid surgery, presence of other illness, treatment with ^{131}I, clinical signs, and laboratory values for TSH, T_4, T_4U, T_3, and TBG. Missing values were imputed as their means (in the case of continuous variables) or their mode (in the case of categorical variables).

Table 4: Distribution of patterns for Example II

Output Category	Frequency	Percentage
normal	6771	72.52
Hyperthyroidism, NOS	193	2.07
Primary hyperthyroidism	21	2.25×10^{-3}
Toxic goiter	18	1.93×10^{-3}
Secondary hyperthyroidism	9	9.64×10^{-4}
Hypothyroidism, NOS	1	1.07×10^{-4}
Primary hypothyroidism	239	2.56
Compensated hypothyroidism	419	4.49
Secondary hypothyroidism	8	8.57×10^{-4}
Other conditions	1658	17.76

The networks were trained as long as the error rate in a test set of 4,586 patients was declining. When the error in the test set started to increase again, the stopping criterion was reached, and training was discontinued. The networks were not trained up to convergence to avoid overfitting [17]. Figure 8 illustrates the stopping criterion used on our networks. More details on an earlier implementation of HNN and the data set used for making the automated diagnosis of thyroid conditions can be found in [18]. Table 5 shows the time that the different systems took to reach the stopping criterion.

Table 5: Comparison of systems for Example II

System	Weights	Epochs	Time†
Standard NN	426	37,948	56 h 5 min 19 sec
Hierarchical NN	410	18,511	4 h 59 min 45 sec

† Time on an HP9000 workstation.

The time performance of hierarchical systems was clearly the best. The perceptron was not able to discriminate rare patterns even after 4×10^5 epochs, indicating that the problem was probably non-linearly separable.

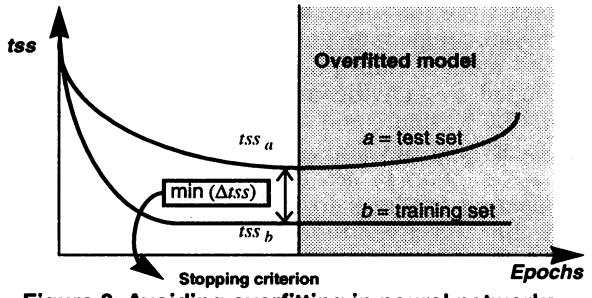

Figure 8. Avoiding overfitting in neural networks

Table 6 shows the sensitivities and specificities of the different systems after 90 minutes of training for the class *hypothyroidism*. Table 7 shows the equivalent numbers for the pattern *compensated hypothyroidism*. These numbers are based on the test set. Note that the increase in sensitivity obtained by using HNN is not coupled with a marked decrease in specificity. The superiority of the hierarchical system was clearly demonstrated in this complex problem. Not all possible subsets of variables were tried, but the results clearly confirm what was learned from the experiment using the artificial data set: HNN can learn rare patterns faster than their non-hierarchical counterparts, provided that the groupings are defined based on pattern similarity.

Table 6: Prediction of class *Hypothyroidism*

System	Sensitivity	Specificity	Epochs	Time†
Standard NN	49.25%	98.97%	650	90 min
Hierarchical NN	79.35%	98.82%	1,800	90 min

† Approximate time on an HP9000 workstation.

Table 7: Prediction of *Compensated Hypothyroidism*

System	Sensitivity	Specificity	Epochs	Time†
Standard NN	41.83%	98.45%	650	90 min
Hierarchical NN	65.87%	98.79%	3,800	90 min

† Approximate time on an HP9000 workstation.

DISCUSSION

Several authors have dealt with the decomposition of complex problems inside and outside the field of neural networks. The reasons for developing the hierarchical models of neural networks were in general very different from the ones presented in this article. Fukushima [19] developed the Neocognitron for eliminating the problem of space variations in the visual recognition of handwritten digits. The author was not specifically concerned with the frequencies of

the patterns involved. He has also suggested that there were similarities between his architecture and the human visual cortex. Ballard [20] also developed a system of hierarchical neural networks for applications in machine vision, and he was particularly concerned with the problem that the backpropagation algorithm might not scale-up to complex networks. Hrycej [21] discussed modularization in neural networks. In his system, preprocessing of inputs was done in an unsupervised manner by a neural network, and the results of this factoring process were then imputed in the following networks. Frean was concerned with the problem of establishing the necessary number of units in a neural network, and consequently developed an algorithm for incremental addition of hidden units [22]. Romaniuk and Hall [23] developed the Divide and Conquer Network algorithm (DCN) that could also be related Frean's work. Hripcsak [24] developed a connectionist model for decision-support in medicine based on several backpropagation modules to incorporate real-valued and uncertain data. Even though many of the works mentioned above carried the name "Hierarchical Neural Networks", the systems developed by Jordan et al. [25] and Curry and Rumelhart [26] bear the most similarity to the one described in this article. Jordan proposed a system where many networks of experts would receive the system's inputs and compete for providing the best solution. A gating network decided among the experts' solutions. The system proposed in this article is different. Even though I propose a system were Specialized networks refine the partial solutions proposed by the Triage network, the decision on which network to use is done first, so not all experts need to be overburden with all data.

Curry and Rumelhart's work on the Mass Spectrometry Network (MSNet) is closely related to the one presented here. In that system, categories of chemical compounds are determined in a Top level network. The probability of belonging to a given group, allied to the original input attribute vector were then used by Specialized networks to refine the solution and get a final diagnosis. The authors were concerned with the fact that low frequency patterns would cause the performance of the network to decay, and they solved the problem of dealing with infrequent patterns by using a different strategy: they trained the network to recognize low frequency patterns by assigning a higher utility to these patterns. This procedure was done by modifying the learning algorithm, and processing the final output to reflect the consequent changes in posterior probabilities. My system, however, tried to disambiguate the process of diagnosing the categories from the process of using utilities while training to make an optimal decision based on a decision-theoretic approach. In my system, the diagnosis is based on the similarities between the patterns, and not on their relative utility. Once the diagnostic process is proven to be reliable and based mainly on the features presented by the inputs, the use of utilities and the decision on which category to choose should be straightforward. The selection of the best grouping at the Triage level may involve human participation, as in this study, or the clustering of examples by similarity-based algorithms, such as multidimensional scaling [27]. Rumelhart has also proposed the preprocessing of input patterns to eliminate the problem of low-frequency–pattern detection[1]. The preprocessing involves the replication of rare patterns up to the point where all categories have equal prior probabilities. A full investigation on the implications of this approach in terms of loss of specificity still needs to be done. As occurs in other systems, the rise in sensitivity of a classifier is tightly coupled with a decay in specificity. In screening large data bases, it is desirable that the rate of false-positives not be too high. Although I tried the replication method in the artificial data set — obtaining very good results with the standard neural network, as shown in Table 8 — application and evaluation of the method in the thyroid set is still under development. The problem of applying this strategy to the thyroid data set stems from the fact that the greater the number of patterns, the longer the time spent per epoch. In the training set, the network would have to handle approximately 67,710 instances per epoch if low frequency patterns were replicated to reach the same number of patterns of the most frequent category. The time spent per epoch would be therefore more than seven times longer. The advantages of this replication method over HNN must be further studied.

Table 8: Another system comparison for Example I

System	Weights	Epochs†	Time‡
Standard NN	24	304.8	6 sec
Hierarchical NN	24	857.4	9 sec
Perceptron	8	124.4	2 sec

† Average of 10 runs. Refers to the detection of all patterns.
‡Average time on an HP9000 workstation. Considers longest epoch in the hierarchical system.

Although I have demonstrated the superiority of HNN over standard neural networks, given specific time constraints, further enhancement of classification results could be achieved by implementing methods for pruning small weights and therefore reducing the number of free parameters allowed in the system [28]. Future work includes the study of misclassified cases to make sure that the gold-standard was correct and comparison with other statistical methods of pattern recognition. A principled way to establish the groupings at intermediate stages of the hierarchical systems needs to be developed. The adequacy of clustering methods for this purpose has to be tested. I am currently working on an implementation of HNN in the analysis of a large data set of HIV infected patients, in which investigation of these issues will be pursued. There are a number of medical applications other than the ones mentioned here that could benefit from HNN. As structured electronic medical records become more common, screening large data sets for unusual patterns may be greatly enhanced by the use of HNN. The unusual patterns detected by the neural networks can then be processed by a number of manual or computer-based decision-support applications. Database mining for knowledge discovery in large databases may also benefit from the power and simplicity of HNN.

1. Rumelhart DE. Personal communication, 1994.

CONCLUSION

The number of epochs required to train a neural network to detect patterns increases exponentially with the decrease in pattern frequency. To minimize this problem, a HNN can be used. Two examples, which used an artificial data set to classify binary numbers and a real-world complex data set of patients suspected of having thyroid disease, indicate that hierarchical systems of neural networks can overcome the problem of low frequency pattern detection in backpropagation neural networks if the selection of groupings at each step is based on pattern similarity. Many medical problems are amenable to such decomposition and should benefit from the use of HNN, especially if the detection of low frequency patterns is required. Furthermore, a rational choice of groupings may be useful for providing partial diagnoses and even for explanation purposes.

Acknowledgments

I thank Prof. David Rumelhart, Dr. Michael Walker, Prof. Mark Musen, Prof. Edward Shortliffe, Prof. Les Lenert, and Prof. Nils Nilsson for useful discussion in different stages and different aspects of the present work. I am solely responsible for any errors in the text. This work has been funded by the Conselho Nacional de Pesquisa (CNPq), Brazilian Ministry of Education. Computing facilities were provided by CAMIS, through grant LM05305 from the National Library of Medicine.

Reference

[1] Rumelhart DE; Hinton GE; Williams RJ. Learning internal representation by error propagation. In Rumelhart, D.E., and McClelland, J.L. (eds) *Parallel Distributed Processing*. MIT Press, Cambridge, 1986.

[2] Baxt WG. Use of an artificial neural network for the diagnosis of myocardial infarction. *Annals of Internal Medicine*, 1991, 115(11):843–8.

[3] Tu JV; Guerriere MR. Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery. *Computers and Biomedical Research*, 1993, 26(3):220–9.

[4] Ashutosh K; Lee H; Mohan CK; Ranka S; Mehrotra K; Alexander C. Prediction criteria for successful weaning from respiratory support: statistical and connectionist analyses. *Critical Care Medicine*, 1992, 20(9):1295–301.

[5] Weinstein JN; Kohn KW; Grever MR; Viswanadhan VN; Rubinstein LV; Monks AP; Scudiero DA; Welch L; Koutsoukos AD; Chiausa AJ. Neural computing in cancer drug development: predicting mechanism of action. *Science*, 1992, 258(5081):447–51.

[6] Miller AS; Blott BH; Hames TK. Review of neural network applications in medical imaging and signal processing. *Medical and Biological Engineering and Computing*, 1992, 30(5):449–64.

[7] Scott JA; Palmer EL. Neural network analysis of ventilation-perfusion lung scans. *Radiology*, 1993, 186(3):661–4.

[8] Maclin PS; Dempsey J. Using an artificial neural network to diagnose hepatic masses. *Journal of Medical Systems*, 1992, 16(5):215–25.

[9] Wu Y; Giger ML; Doi K; Vyborny CJ; Schmidt RA; Metz CE. Artificial neural networks in mammography: Application to decision making in the diagnosis of breast cancer. *Radiology*, 1993, 187(1):81–7.

[10] Reggia JA. Neural computation in medicine. *Artificial Intelligence in Medicine*, 1993, 5(2):143–57.

[11] Hertz JA; Palmer RG; Krogh, AS. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, 1991.

[12] Gray NAB. Constraints on "learning machine" classification methods. *Analytical Chemistry*, 1976, 48(14):2265–8.

[13] U.S. Preventive Services Task Force. *Guide to clinical preventive services*. William and Wilkins, Baltimore, 1989.

[14] Hornik K; Stichcombe M; White H. Multilayered feedforward networks are universal approximators. *Neural Networks*, 1989, 2:359–66.

[15] Murphy PM; Aha DW. *UCI Repository of Machine Learning Databases* (on-line directory), University of California at Irvine, Department of Information and Computer Science, Irvine, CA, 1993.

[16] Quinlan JR Induction of decision trees. *Machine Learning*, 1986, 1, 81–106.

[17] Hecht-Nielsen, R. *Neurocomputing*, Addison-Wesley, Reading, 1990.

[18] Ohno-Machado L; Musen MA. Hierarchical neural networks for partial diagnosis in medicine. *Proceedings of the 1994 World Congress on Neural Networks*, San Diego, in press.

[19] Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1988, 1:119–30.

[20] Ballard D. Modular learning in hierarchical neural networks. In Schwartz, E.L. (ed) *Computational Neuroscience*, Bradford, London, 1990.

[21] Hrycej T. *Modular Learning in Neural Networks*. John Wiley and Sons, New York, 1992.

[22] Frean M. The Upstart Algorithm: A method for constructing and training feedforward neural networks. *Neural Computation*, 1990, 2:198–209.

[23] Romaniuk SG; Hall LO. Divide and conquer neural networks. *Neural Networks*, 1993, 6(8): 1105–16.

[24] Hripcsak G. Using connectionist modules for decision support. *Methods of Information in Medicine*, 1990, 29:167–81.

[25] Jordan RA; Nowlan SJ; Hinton SJ. Adaptive mixtures of local experts. *Neural Computation*, 1991, 3:79–87.

[26] Curry B; Rumelhart DE. MSnet: A neural network that classifies mass spectra. *Tetrahedron Computer Methodology*, 1990, 3:213–37.

[27] Shepard RN. Multidimensional scaling, tree-fitting, and clustering. *Science*, 1980, 210:390–8.

[28] Weigend AS; Rumelhart DE; Huberman BA. Generalization by weight-elimination applied to currency exchange rate prediction. *1991 IEEE International Joint Conference on Neural Networks*, 3:2374–9.