The Effect of Hidden Units in Neural Networks on Identifying Data Duplication Records

Abdullah Al-Namlah, Ministry of Defense and Aviation, RSADF/Computer Directorate, P.O. Box 21386, Riyadh, 11475, Saudi Arabia; E-mail: alnamlah@hotmail.com

ABSTRACT

Learning algorithms have been widely used to solve different problems in the field of Artificial Intelligence. Presently there are many learning algorithms; each is used depending on specifics of the problem to be solved. Examples of learning algorithms can be found in the field of Artificial Neural Networks (Neural Nets) where these algorithms are used to train the neural nets (as an example, Backpropagation algorithm). Neural nets have been used in data quality problems where a complex database has a lot of duplicate data (dirty data). By using neural nets, it was demonstrated that they can be a very useful tool to identify duplicate and non-duplicate records in the database. In this paper, we show the impact of internal architecture of neural network (hidden units) on the accuracy of results.

INTRODUCTION

Neural Networks are one of the most popular advanced modeling techniques (Barth, 1997). A neural network is an information processing system that can be used to store and recall data or patterns and classify them. It has the capability to learn by examples. Neural networks have proven to be quite effective for a broad range of problems, and are especially useful for predicting events when there is a large pool of data during the learning process. Neural Networks are of interest to both academics and practitioners in many areas like signal processing, medicine, pattern recognition, speech recognition, and even in business (Harston, 1990). Chiang, Urban and Baldridge (1996) developed a neural network to forecast the net asset value of mutual funds, and found the model to perform well in forecasting processes. Another example of using neural nets in business was to predict daily stock prices for three German stocks (Schoneburg, 1990). In 1995, Jain and Nag applied a neural net to the problem of pricing initial public offerings (Jain and Nag, 1995).

Lately, neural networks have been used in the field of Computer Science to address data quality during the software maintenance process (Al-Namlah and Becker,2003). Neural nets also have been used in the data quality field where a complex database has a lot of duplicate data (dirty data), and showed that they can be a very useful tool to identify duplicate and non-duplicate records in a database (Al-Namlah, Becker and Koksal,2002) and (Al-Namlah,2003).

DATA DUPLICATION PROBLEM

Data duplication means the database has stored duplicate data about an object. Conversely, non-duplication is defined by English (1999), as "The degree to which there is a one-to-one correlation between records and the real-world object or events being represented" (English, 1999, p. 142).

There are many processes that lead to having duplicate data in a database. Common processes that lead to this situation (Milrud, 2001) are:

1. Merging two or more databases as in the case of creating a data warehouse.
2. Using the system to generate a unique number for each row and assign it as a primary key.

Data duplication adds costs in at least two ways: First, it leads the organization to have more data than it needs. Brauer (2001) reports that an acquiring company learned long after the deal was closed that their new consumer business only had 50% of the customers as they thought because of the large amount of duplicate data in their customer database (Brauer, 2001). Second, data duplication affects the correctness of all the processes that depend upon this data, such as business reports.

Sending duplicate mail to customers leads to additional costs that can be alleviated if the company cleans their customer database. Here are some of the costs involved in having duplicate data:

- The cost of duplicate faxes, mailing and other forms of communication
- The cost of printing and production of additional mail services
- The cost of inaccurate results from analysis of data and subsequent reports
- The cost of inaccurate forecasting due to the misleading number of records
- The negative impact presented to potential clients receiving duplicate mail
- Time consumed by salespeople to contact the same customer

To prevent these additional costs, business organizations should consider eliminating duplicate records before they start using their data warehouses.

DATA DUPLICATION METRICS

Before highlighting the past efforts and the results of our work, we introduce metrics that are used to evaluate the data duplication solution. Researchers in the area of solving data duplication problems have used the following metrics:

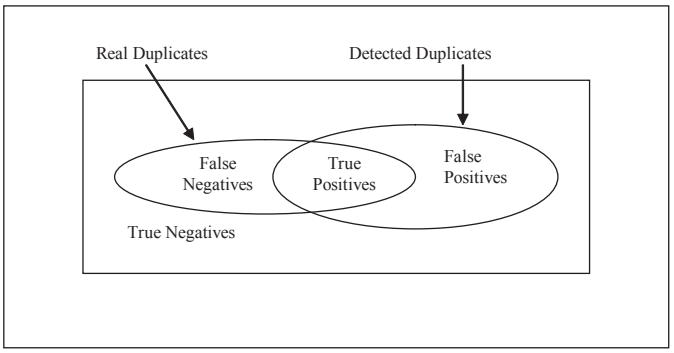
1. ***False negatives*** are also called *missed matches* (Winkler, 1995) and some researchers call them *misses*- which are those duplicate records where the approach fails to identify them as duplicates (see Figure 2).
2. ***False positives*** are also known as *false matches* (Winkler, 1995), which include those records that are not duplicates but the approach wrongly identifies them as duplicates (see Figure 2).
3. ***True positives*** *are* those records that are duplicates and the approach correctly identifies them as duplicates (see Figure 2).
4. ***True negatives*** are those records that are not duplicates and the approach correctly identifies them as not duplicates (see Figure 2).
5. ***Recall*** is also known as *percentage hits* (Lee et al., 2000), and is defined as the percentage of duplicate records being correctly identified. Higher recall is achieved by accepting records with low degrees of similarity as duplicates. Recall is computed as follows:

$$Recall = \frac{|True\ Positives|}{|Misses| + |True\ Positives|}$$

6. ***Precision*** is contrasted with recall; the percentage of correct predictions among all pairs of records that have been identified as duplicates (McCallum, Nigam and Ungar, 2000). Higher precision is achieved by accepting records with a higher degree of similarity as duplicates (Monge & Elkan, 1996; Lee et al., 2000; Do et al. 2002). Precision is computed as follows:

$$Precision = \frac{|True\ Positives|}{|True\ Positives| + |False Positives|}$$

Figure 1. Data duplication metrics (from Do et al. (2002), pg. 224)



PREVIOUS EFFORTS TO SOLVE DATA DUPLICATION PROBLEM

Record duplication is a complex problem that many researchers have tried to solve, using a variety of approaches. One of the most effective approaches is (Hernandez & Stolfo, 1995), and almost all subsequent researches have referenced this approach as a unique and effective way of solving the data duplication problem. Since the process of identifying data duplicates in databases involves matching the corresponding attributes in two different records, some efforts have focused on the field matching algorithms used to find the degree of similarity between two corresponding database fields. Monge and Elkan (1996) describe three record matching algorithms and evaluate their performance on real-world datasets. These are the basic field matching algorithm, recursive field matching algorithm and Smith-Waterman algorithm. Monge and Elkan (1996) found that recursive field matching and Smith-Waterman algorithms could achieve 100% recall while the basic algorithm could only achieve 90% recall. One fact to consider is that the Smith-Waterman algorithm has lower precision than the other two algorithms. A main contribution of the Monge and Elkan (1997) study is that it gives a relatively domain-independent algorithm to detect approximate duplicate records. It also shows how to compute transitive closure of the "is duplicate of" relationship by incrementally using a union-find data structure.

Lee et al. (2000) presents a knowledge-based framework for intelligent data cleaning. The framework consists of three stages: pre-processing, processing, and (validation and verification) stages. In the preprocessing stage, data records are first conditioned and scrubbed of any anomalies, and then data type and format are standardized. In the second stage, conditioned records are fed into an expert system engine together with a set of rules. The rules are fired in an opportunistic manner when conditioned records are fed into the expert system engine. These rules are responsible for identifying duplicate records, updating records that have missing data, and raising certain alert rules when some constraints are violated. The third and last stage is to generate a log report, which is used as an audit trail for all actions that have been done to the database records.

McCallum et al. (2000) used a technique for clustering, called canopies, to solve the problem of grouping large, high-dimensional data sets such as clustering textual bibliographic references. A canopy is a subset of the data elements to be clustered, and each data item that appears in a canopy is within some distance threshold from the center of the canopy (which is another data item).

Cohen and Richman (2002) have used the canopy approach, and presented an adaptive scheme for entity-name matching and clustering. What is meant by adaptive in this paper is that accuracy can be improved by training, like the nature of most learning methods. The entity-name matching means matching names for two different sources to identify those names that belong to the same object. The main use of the canopy approach in their scheme is to compute the set of candidate pairs to be compared in a subsequent stage. This way, the canopy approach restrains the number of items in each canopy and then another expensive edit distance is used to compare the items under each canopy. By doing this, overall time complexity is reduced since not all data items in the two resources are compared against each other using an expensive edit distance.

In (Al-Namlah, 2003) we showed that combining neural nets with other methods such as the one used by (Hernandez & Stolfo, 1995), was a powerful mechanism in uncovering data duplication. Our results showed that this approach reduced time complexity, uncovered duplicate records, and reduced the number of false positives and misses when uncovering duplicate records. However, there were some variables related to the neural net that were not studied in detail in this effort. One of them is the effect of the internal architecture (hidden layer) of the neural net on accuracy of the results. This paper details the effect of the number of hidden units in the hidden layer on overall results for resolving the data duplication problem.

NEURAL NETWORK ARCHITECTURE

Artificial neural nets are mathematical models that have been developed to imitate the biological neural net and they share common properties. The following assumptions are made to generalize artificial neural nets so that they are similar to human neural biology (Fausett, 1994):

- The information is processed in multiple central processing units called neurons (units).
- Neurons interchange signals through a highly connected net.
- Each connection between two neurons has an associated weight.
- Each neuron computes the output by summing all incoming signals (net input) and applies an activation function to the net input

The internal architecture of the neural net consists of nodes that are highly connected. Each connection has a weight, and as the neural network is trained, the weights are adjusted. When these weights no longer need to be adjusted during the training phase, the neural net has learned from provided examples. Then, it should be able to recognize (memorize) exact and (generalize) similar (generalize) patterns when it sees them in future applications (Fausett, 1994).

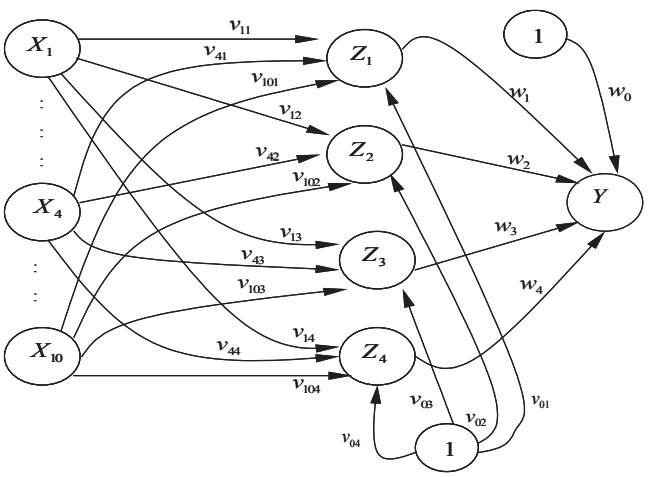
Neural nets are usually characterized by their internal architectures and methods that are used to train them. Training a neural net involves changing the weights to reflect current understanding to the behavior of the problem under investigation. In general, there are two methods of training: *supervised* and *unsupervised*. *Supervised training* is accomplished when the provided training examples (also can be referred to as training vectors or training patterns) consist of two parts: input example and output target.

In order to have a reasonable number of training pairs, we use the following formula (Baum and Haussler, 1989)

$$\frac{W}{P} = e$$

where W is the number of weights in the net, P is the number of training pairs, and e is the accuracy of classification expected. In our study, we start the algorithm

Figure 2. Neural net architecture



by assuming e=0.01 and the total weights W= 44, which suggests 4,400 training pairs. We train the net using a back propagation algorithm with Nguyen-Widrow (1990) initial weights. After the net is trained, we capture the final weights and use them in performing ten separate tests for different sets of data. The net has a ten-unit input layer (X_i), variable of units in the hidden layer (Z_i), and a single unit output layer (Y). As an example, Figure 2 shows the neural net architecture with four hidden units.

To test the effect of the hidden layer on accuracy of the end result of the net, number of the hidden units in the hidden layer will vary. Each time we change the number of hidden units, we train the net until it reaches the stability phase and then weights are captured to be used in the test phase. In this work we tested the results when the hidden layer had 2, 3, 4, 5, 6, 8, 10 and 12 units. Results of these tests are detailed in the Experimental Results Section of this paper.

EXPERIMENTAL RESULTS

The database used to test our proposed method was generated by the same database generator used by Hernandez (1996). This database generator allowed us to generate data with prior knowledge about duplicate data records. Furthermore, the database generator as described in (Hernandez, 1996) provided a large number of parameters that helped us perform controlled studies. These parameters include size of the database, percentage of duplicates in the database, and amount and type of error in any attribute to be introduced**.**

The layout of generated records consists of the following fields: social security number, first name, middle initial, last name, street number, street address, apartment #, city, state, and zip code. Some of these fields can contain null values as a simulation of errors that can happen in real life databases. The names were chosen randomly from a list of 63,000 real names. The cities, states and zip codes are all from the U.S.A.

In order to test effect of the hidden layer on accuracy of the neural net in identifying duplicate records, we built 8 neural nets with 10 input units, one hidden layer and one output unit. The difference between these neural nets is the number of units in the hidden layer. The 8 neural nets have 2, 3, 4, 5, 6, 8, 10 and 12 units respectively. After building each one of these neural nets we trained it using the 4,400 training examples. After the training we tested the neural net by feeding it with 501,360 records. Among these 501,360 there were 295,689 duplicate records, and 205,671 non-duplicate records. Table 1 shows the results of these tests.

Table 1 shows that when the neural net has only 2 units in the hidden layer, 294,842 out of 295,689 duplicate records were correctly identified by the neural net as duplicates (true positives), where 847 duplicate records were misses, i.e. the neural net failed to identify them as duplicates (false negatives). Furthermore, 205,639 out of 205,671 non- duplicate records were correctly identified by the neural net as non-duplicates (true negatives), where only 32 non-duplicate records were wrongly identified as duplicates (false positives). As a result of the above identification, recall was computed as follows:

$$Recall = \frac{|True\ Positives|}{|Misses| + |True\ Positives|} = \frac{294{,}842}{847 + 294{,}842} = 0.997$$

Table 1. The result of identifying duplicate records with different number of hidden units

Metric	Number of Hidden Units							
	2	3	4	5	6	8	10	12
True Positives	294,842	294,607	295,037	295,037	294,927	294,644	294,932	294,867
False Negatives	847	1082	652	652	762	1045	757	822
True Negatives	205,639	205,651	205,621	205,637	205,649	205,386	205,534	205,393
False Positives	32	20	50	34	22	285	137	278
Recall (%)	99.7	99.6	99.8	99.8	99.7	99.6	99.7	99.7
Precision (%)	100	100	100	100	100	99.9	99.9	99.9

and the precision was computed as follows:

$$Precision = \frac{|True\ Positives|}{|True\ Positives| + |False\ Positives|} = \frac{294{,}842}{294{,}842 + 3} \approx 1$$

We should take notice that the best overall results were obtained when the neural net had 3, 4 and 5 hidden units. The net with 3 hidden units was the best in identifying non-duplicate records, while the net with 4 and 5 hidden units was the best in identifying duplicate records. By reviewing the overall results, all 8 neural nets were excellent in their recall and precision. This is almost complies with (Rumelhart, McClelland, & PDP Research Group, 1986) that a set of N orthogonal input patterns can be mapped onto $\log_2 N$ hidden units to form a binary code with distinct patterns for each of the *N* input patterns.

It should be noticed that in our proposed solution we are always trying to obtain the maximum value of both precision and recall metrics together by doing a balance between them. It might be observed as an example, Monge and Elkan (1996) found that recursive field matching and Smith-Waterman algorithms could achieve 100% recall whereas our proposed method the best achieved 99.8%, however, we should consider the other metric (precision) in both solutions as well as other advantages such as the ability to improve the results through learning among others, that are not the subject of this paper. Detailed advantages of our solution can be found at Al-Namlah.(2003).

SUMMARY AND FUTURE WORK

In this study we found that internal architecture has an effect on the results of a neural net in identifying duplicate records. What we mean by internal architecture involves number of hidden units in the hidden layer. It was also observed that it is not necessarily the increase of hidden units that makes a neural net more capable of identifying duplicate records as noticed when the neural net had 8, 10, and 12 hidden units.

Theoretical results show that one hidden layer is sufficient for a backpropagation net to approximate any continuous mapping from the input patterns to the output patterns, to an arbitrary degree of accuracy (Fausett, 1994). Future work will include a practical study regarding the effect of number of hidden layers on the accuracy of a neural net in identifying duplicate records.

REFERENCES

Al-Namlah A., Becker S., and Koksal S., (2002). "Eliminating Data Duplication Using A Back propagation Neural Net," Proceedings of the 2nd International Conference on Neural, Parallel, and Scientific Computations, Vol. 2, pages 37-41, Atlanta, GA, August 07-10, 2002.

Al-Namlah, A.(2003). "Solving the Data Duplication Problem for Complex Databases Using Neural Networks." Ph. D. dissertation, Florida Institute of Technology, December, 2003.

Al-Namlah, A., and & Becker S. (2003). "Using Neural Networks for Addressing Data Quality During the Software Maintenance Process Proceedings of the 2003 IRMA Conference, Vol. 1, pages 1-4, Philadelphia, Pennsylvania, USA, May 18-21, 2003.

Baum, E. and Haussler (1989). "What Size Net Gives Valid Generalization?" Neural Computation, Vol. 1, No.1, Pages 151-160.

Barth P., (1997). Mining for Profits in the Data Warehouse. In Barquin R. & Edelstein H. (Eds.), Building, Using, and Managing The Data Warehouse, Prentice-Hall, Inc., Upper Saddle River, NJ.

Brauer Bob (2001). "Data quality, Spinning Straw Into Gold", Data Flux Corporation. http://www.dataflux.com/data/spinning.pdf

Chiang, W., Urban T., and Baldridge G. (1996). A neural network approach to mutual net asset value forecasting. Omega 24: 205-215.

Cohen W. and Richman J. (2002). "Learning to Match and Cluster Large High-Dimensional Data Sets for Data Integration". In Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, Edmonton, Alberta, Canada, July 23-26, 2002.

Do H., Melnik S., Rahm E. (2002). "Comparison of Schema Matching Evaluation." In Proceedings of the 2nd Int. Workshop on Web Databases (German Informatics Society), Pages 221-237, 2002.

English Larry P. (1999). Improving Data Warehouse and Business Information Quality, Methods for Reducing Costs and Increasing Profits, John Wiley & Sons, Inc., New York, NY.

Fausett L. (1994). Fundamentals of Neural Networks, Architectures, Algorithms, and Applications, Prentice-Hall, Inc., Upper Saddle River, NJ.

Harston, C. T. (1990). "Business with Neural Networks" In A. J. Maren, C. T. Harston, & R. M. Pap, eds., Handbook of Neural Computing Applications. San Diego: Academic Press, pp. 391-400.

Hernandez M. and Stolfo S., (1995). "Merge/Purge Problem for large databases", Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 127-138, May 1995.

Hernandez M. (1996). "A Generalization of Band Joins and the Merge/Purge Problem", Ph.D. thesis, Columbia University, 1996.

Jain, B. and Nag, B. (1995). Artificial neural network models for pricing initial public offerings. Decision Sciences 26: 283-302.

Lee M.L., Ling T. W. and Low W.L. (2000). "IntelliClean: A Knowledge-Based Intelligent Data Cleaner ", Proceedings of the 6th ACM SIGMOD International Conference on Knowledge discovery and Data Mining, pages 290-294, August 2000.

McCallum A., Nigam K., and Ungar L. (2000). "Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching". In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining, pages 169-178, 2000.

Milrud B. (2001). "Finding and Eliminating Duplicate Data." Retrieved November 13, 2001 on World Wide Web. http://gethelp.devx.com/techtips/oracle_pro/10min/10min0501/10min0501.asp

Monge A. E., and Elkan C. P. (1996). "The field matching problem: Algorithms and applications", Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pages 267-270, AAAI Press, August 1996.

Monge A. E., and Elkan C. P. (1997). "An efficient domain-independent algorithm for detecting approximately duplicate database records", Proceedings of the ACM-SIGMOD workshop on Research Issues on Knowledge Discovery and Data Mining, Tucson, AZ, 1997.

Nguyen, D., and B. Widrow. (1990). "Improving the Learning Speed of Two-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights." International Joint Conference on Neural Networks, San Diego, CA, III: 21-26, 1990.

Rumelhart, D.E., J. L. McClelland, & PDP Research Group. (1986). Parallel Distributed Processing, Explorations in the Microstructure of Cognition; Vol. 1: Foundations. Cambridge, MA : MIT Press.

Schoneburg, E. (1990). Stock price prediction using neural networks: A project report. Neurocomputing 2: 17-27.

Winkler, W. E. (1995). "Matching and Record Linkage," in B.G. Cox et al. (ed.) Business Survey Methods, New York: J. Wiley, 335-384.