Quick Convergence of FFNN by Weight Optimization Technique in Data Mining

S.S.Baskar
Research Scholar, Department of computer, St. Joseph's college(Autonomus), Trichy, India

Dr L. Arockiam
Associate Professor,Department of computer, St. Joseph's college(Autonomus), Trichy, India

S.Charles
Associate Professor,Department of computer, St. Joseph's college(Autonomus), Trichy, India

Abstract-In this paper, we present a new method that provides quick convergence for a feed forward neural networks system using feature selection and weight optimization techniques. A neural networks system functions on the basis of weights presented to the neurons. These weights are fine tuned to the most possible accurate level using the training data set. But this process is time and resource intensive, and increases exponentially with the increase in number of attributes. Further, the irrelevant attributes present in the attribute set reduces the level of accuracy and increases the computation time. Hence we present a method that initially shortlists the valid candidates and their dependencies and finally assign them with the appropriate weights for providing to the Neural networks system.

Keywords: Feature selection; Neural Networks; fuzzy association rule mining

1. INTRODUCTION

1.1. Soil Classification

Soil classification [8]&[9] refers to grouping soils based on their physical and chemical characteristics that distinguish each soil type. For soil resources, grouping soils by their inherent properties, behaviors, or genesis, can provide better results. Despite the differences, classification criteria can group similar concepts so that interpretations do not vary widely. Natural system approaches to soil classification, such as the French Soil Reference System are based on presumed soil genesis. Systems have developed, such as USDA soil taxonomy and the World Reference Base for Soil Resources, which use taxonomic criteria involving soil morphology and laboratory tests to inform and refine hierarchical classes.

Another approach is numerical classification [13], also called ordination, where soil individuals are grouped by statistical methods such as cluster analysis. This produces natural groupings without requiring any inference about soil genesis.

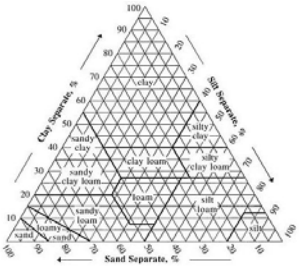

Fig 1: Soil Classification

1.2. Neural Networks

An Artificial Neural Network (ANN), [1]&[2] often just called a "Neural Network" (NN), is a mathematical model or computational model based on biological neural networks, i.e. it functions similar to that of a biological neural system. It consists of an interconnected group of nodes called artificial neurons and these nodes processes information and provide the output to the user. These nodes communicate with each other and every node has an assigned functionality and it performs the process on the data provided to it.

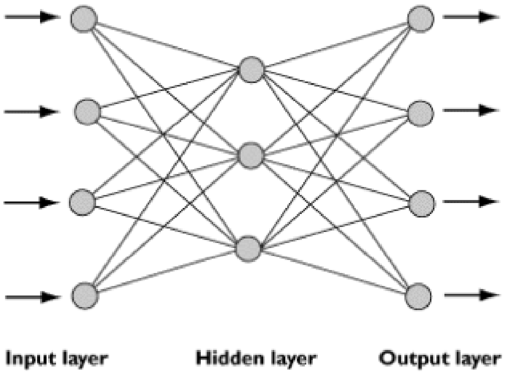

Fig 2: Structure of an Artificial Neural Network

The ANN changes its structure based on external or internal information that flows through the network during the learning phase. A basic ANN consists of three layers, the input, hidden and the output layers. The hidden layer performs the processing in the network.

1.3. Feed Forward Neural Networks

A feed-forward neural network is an artificial neural network where connections between the neurons do not form a directed cycle. This is different from the simple neural networks. The feed-forward neural network is the simplest type of artificial neural network. In this network, the information moves in only in the forward direction. i.e. information moves from the input nodes, to the processing nodes and then to the output nodes. There are no cycles or loops in the network.

1.4. Feature Selection

Feature selection [10], also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features. In general, all data contains both relevant and irrelevant information. The data might also contain redundant features, which may not be of much use. Hence we use the feature selection methods to eliminate the noisy data (i.e. irrelevant and redundant information) from the set. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features.

1.4.1. Feature Selection Methods

Feature selection algorithms can be categorized into three broad types:

i. **Wrapper method:** They use a predictive model to score feature subsets. Each new subset is used to train a model, which is tested on a hold-out set. Counting the number of mistakes made on that hold-out set gives the score for that subset.

ii. **Filter method:** Filter methods use a proxy measure instead of the error rate to score a feature subset. This measure is chosen to be fast to compute, still capturing the usefulness of the feature set.

iii. **Embedded methods:** Embedded methods are a catch-all group of techniques which perform feature selection as part of the model construction process.

1.5. Relief-F

Relief algorithms [3] are used for the estimation of the attributes in a system. Estimation of attributes is performed by evaluating the level of dependencies between the attributes. In addition, their quality estimates have a natural interpretation. While they have commonly been viewed as feature subset selection methods that are applied in prepossessing step before a model is learned, they have actually been used successfully in a variety of settings, e.g., to select splits or to guide constructive induction in the building phase of decision or regression tree learning, as the attribute weighting method and also in the inductive logic programming.

1.6. Fuzzy Logic

Fuzzy Logic [12] incorporates a simple, rule-based IF X AND Y THEN Z approach to solving a control problem rather than attempting to model a system mathematically. The Fuzzy Logic model is empirically-based, relying on an operator's experience rather than their technical understanding of the system. Generally, Fuzzy Logic is so forgiving that the system will probably work the first time without any tweaking.

The remaining paper organized as follows, Section 2 discusses the overall system architecture and the overall functioning of the system. Section 3 describes the weight optimization technique in detail, explaining all the involved components. Section 4 provides the simulation results and their explanations.

2. SYSTEM ARCHITECTURE

The number of available inputs present in soil data varies significantly with samples. Further, the number of attributes can be very high. Hence an effective system for pre-processing this data is a must for providing a better performing system. Our system analyzes this data, removes the irrelevant and repetitive data and finally provides us with appropriate data and its corresponding weights.

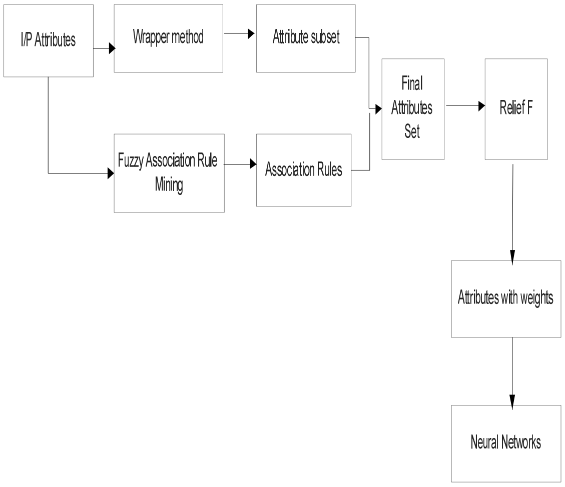

Fig 3: Weight Optimization

Figure 3 shows the overall system architecture of our weight optimization technique. Initially all the input attributes are provided to the wrapper method. The wrapper method has the property of dividing the attributes into subsets and finally finding the subset with the maximum score. This subset contains the attributes with maximum relevance to the result. All these attributes are added to the final attribute set. The input attributes are then given to the fuzzy association rule mining algorithm to find the association between various attributes. The attributes with highest confidence values are added to the final attribute set. This attribute set is then passed to the Relief-F algorithm for weight calculation. Finally, weights are calculated and incorporated into the Neural Networks for further processing.

3. WEIGHT OPTIMIZATION TECHNIQUE

The weight optimization for a Neural Networks is performed in three phases. They are, feature subset selection using the wrapper method, attribute relationship discovery using fuzzy association rule mining and weight incorporation using the Relief-F method.

All the available inputs are taken and passed to the feature subset selection and the attribute relevance discovery.

3.1. Feature Subset selection

The wrapping method is used for the feature selection process. The following are the conditions that are to be considered for finding attribute relevance.

Almuallim and Dietterich [4] define relevance under the assumptions that all features and the label are Boolean and that there is no noise.

A feature Xi is said to be *relevant* to a concept C if Xi appears in every Boolean formula that represents C and *irrelevant* otherwise.

Gennari et al. [5] allow noise and multi-valued features and define relevant features as those whose "values vary systematically with category membership".

We formalize this definition as follows.

X_i is *relevant* iff there exists some x_i and y for which p(X_i = x_i) > 0 such that

$$p(Y = y \mid X_i = x_i) \neq p(Y = y).$$

Under this definition, Xi is relevant if knowing its value can change the estimates for the class label Y, or in other words, if Y is conditionally dependent on X;. Note that this definition fails to capture the relevance of features in the parity concept where all unlabeled instances are equi-probable, and it may therefore be changed as follows.

Let $S_i = \{ X_1,\ldots,X_{i-1}, X_{i+1},\ldots,X_{m\}}$, the set of all features except X_i. Denote by s_i a value-assignment to all features in S_i.

Xi is *relevant* iff there exists some Xi, y, and si for which p(Xi = xi) > 0 such that

$$p(Y=y, S_i=s_i \mid X_i=x_i) \neq p(Y=y, S_i=s_i).$$

Under the following definition, Xi is relevant if the probability of the label (given all features) can change when we eliminate knowledge about the value of Xi.

Xi is *relevant* IIF there exists some xi, y, and si for which p (Xi = xi, Si =si) > 0 such that

$$p(Y=y, \mid X_i = x_i, S_i = s_i) \neq p(Y=y, S_i = s_i).$$

The wrapper approach conducts a search in the space of possible parameters. A search requires a state space, an initial state, a termination condition, and a search engine [6]&[7]. The search space organization that we chose is such that each state represents a feature subset. For n features, there are n bits in each state, and each bit indicates whether a feature is present (1) or absent (0).

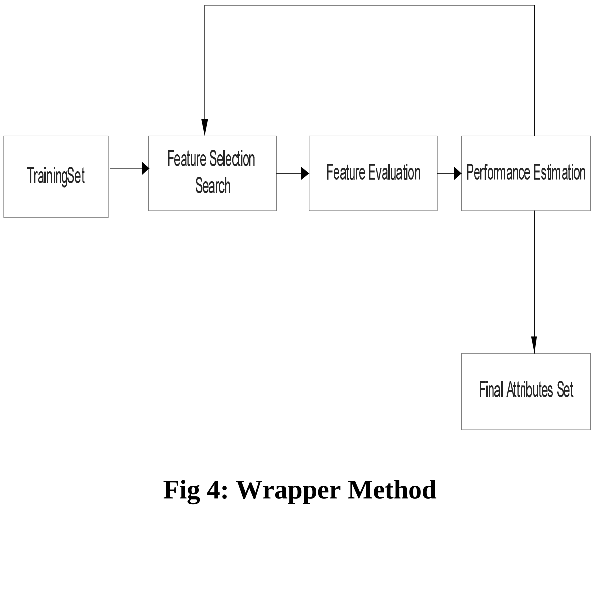

Fig 4: Wrapper Method

3.2. Attribute relationship discovery

This method finds all the large itemsets for the given transactions by comparing the fuzzy count of each candidate itemset with its support threshold. Furthermore, some pruning strategies are used to reduce the number of candidate S itemsets generated.

Initially, taxonomy encoding [11] is performed using a sequence of numbers and the symbol ‘‘*’’, with the ***l*** th number representing the branch number of a certain item at levels.

Determine $x \in \{2,3,4,\ldots\}$ (Maximum item threshold). x is a threshold to determine maximum number of items in a transaction by which the transaction may or may not be considered in the process of generating rules mining. In this case, the process just considers all transactions with the number of items in the transactions less than or equal to x . Formally, let D be a universal set of transactions. $m \subseteq D$ is considered as a subset of qualified transactions for generating rules mining that the number of items in its transactions is not greater than x as defined by:

$$M = \{T \mid card\ (T) \leq x,\ T \in D\},$$

Where card (T) is the number of items in transaction T.

Set k = 1, where k is used to store the level number being processed whereas $k \in \{1,2,3,4\}$ (as we consider up to 4-levels of hierarchies).

Set q=1, where q is an index variable to determine the number of combination of items in itemsets called qitemsets. $q \in \{1,2,3,4\}$ (as we consider up to 4-itemsets at each level of hierarchy).

Determine minimum support for q-itemsets at level k, denoted by $\beta_{items}^{k \in (0,|M|)}$ as a minimum threshold of a combination items appearing in the whole qualified transactions, where |M| is the number of qualified transactions. β_q^k May have same value for every q at level k. Group the items with the same first k digits in each transaction T_j, and add the occurrence of the items in the same groups in Ti. Denote the amount of the j-th group $I_j q$ for Ti as v_{ij}^q. Construct every candidate q-itemset, I^q as a fuzzy set on set of qualified transactions, M. A fuzzy membership function μ is a mapping:

$$M_{I'} : M \rightarrow [0,1]\, as\, defined\, by:$$

$$M_{I'}(T) = v_{ij}^q \cdot \inf_{i \in I'} \left\{ \frac{\eta,(i)}{Card(T)} \right\}, \forall T \in M$$

Where T be a qualified transaction in which T can be regarded also as a subset of items. $T \subseteq D$

A Boolean membership function, η, is a mapping $\eta_T : D \to \{0,1\}$ as defined by:

$$\eta_T(i) = \begin{Bmatrix} 1, i \in T \\ 0, otherwise \end{Bmatrix}$$

Such that if an item, *i*, is an element of T then $\eta_T(I)$ = 1, otherwise $\eta_T(i)$=0.

Calculate *support* for every (candidate) q-itemset using the following equations:

$$Support(I^q)^k = \sum_{T \in M} v_{ij}^q \,.\mu_{I'}^k$$

M is the set of qualified transactions; it can be proved that the previous equation satisfied the following property:

$$\sum_{i \in D} Support\ (i) = \mid M \mid$$

For q =1, I^q can be considered as a single item. if q>1 then generate candidate set C_2^k has to following steps for each newly from 2-itemsets.

I^q Will be stored in the set of frequent q-itemsets, N_q^k if and only if $support\ (I_j^q) \geq \beta_q^k$

Set q = q+1, for the same level k and if q >4, then find the confidence value Looking for possible/candidate q-itemsets from Lq-1 by the following rules:

A q-itemset I^q will be considered as a candidate q-itemset if I^q satisfies:

$$\forall F \subset I^p \mid F \mid = K - 1 \Rightarrow F \in N_{q-1}$$

If there is no candidate q-itemset then find the confidence of an association rule mining, $A \Rightarrow B$, can be calculated by the following equation:

$$conf(A \Rightarrow B) = P(B \mid A) = \frac{Support(A \cup B)}{Support(A)}$$

Where A, $B \in D$

The confidence value can also be represented as

$$conf\ (A \Rightarrow B) = \frac{\sum_{T \in M} \inf_{i \in A \cup B} (v_{ij}^q \,.\mu_{I'}^k\ (T))}{\sum_{T \in M} \inf_{i \in A} (v_{ij}^q \,.\mu_{I'}^k\ (T))}$$

Where A and B is any q-itemsets in Lq $[\mu_i(T) = \mu_{(i)}(T)]$

Therefore, support of an itemset can be expressed as following:

$$Support(I^q)^k = \sum_{T \in M} \inf_{i \in I'} (\mu_{(i)}(T))$$

Increment the value of k and perform repeat the whole processing for next level. This provides the best pairs of items for the user.

Results from the first two phases are combined to provide the user with the final attribute set that can be used on the Neural Networks.

3.3. Weight Incorporation

The process of selecting the appropriate weights for the input attributes is performed by the Relief-F algorithm. A default initial weight W(A$_x$) is set for all the attributes A$_1$, A$_2$, A$_3$… A$_n$. The Relief-F algorithm selects every attribute into consideration and it takes the current value of the attribute as the base pointer B. Two values from the same attribute are selected, one from the same class H and the other from a different class M. If H and B have different values, then the attribute does not influence the result, so weight of the attribute is decremented. If M and B have different values, then it means that the current attribute's value varies with class, so the weight of the attribute is incremented. This process is continued m times, where m is defined by the user as the maximum number of iterations.

Algorithm:

```
1. Initialize weights for all attributes W[A] := 0.0;
2. for i from 1 to m
        2.1 randomly select an instance Ri;
        2.2 find k nearest hits Hj;
        2.3 for each class C <> class(Ri) do
                2.3.1 from class C find k nearest misses Mj(C);
        2.4 for A from 1 to a
                2.4.1 W[A] :=W[A] – $(x + a)^n = \sum_{j=1}^{nk} \frac{\mathrm{diff(A,R_i,H_j)}}{\mathrm{(m.k)}} + \sum_{\mathrm{C \neq class(R_i)}} [\frac{\mathrm{P(C)}}{\mathrm{1 - P(class(R_i))}} \sum_{j=1}^{k} diff(A, R_i, M_j(C))]/(m.k)$
3. end;
```

Function diff(A; I_1; I_2) calculates the difference between the values of the attribute A for two instances $I1$ and $I2$. For nominal attributes it was originally defined as:

$$diff(A, I_1, I_2) = \begin{cases} 0, & value(A, I_1) = value(A, I_2) \\ x, & otherwise \end{cases}$$

and for numerical attributes as:

$$diff(A, I_1, I_2\ ) = |value(A, I_1) - value(A, I_2)\ |/(\max(A) - \min(A)\ )$$

The function diff is used also for calculating the distance between instances to find the nearest neighbors. The total distance is simply the sum of distances over all attributes (Manhattan distance).

The returned results can be directly used by the Neural Networks for further processing of the data.

4. Results and Discussion

The obtained results were compared with [14] and the results obtained shows considerable improvement in the performance of the system.

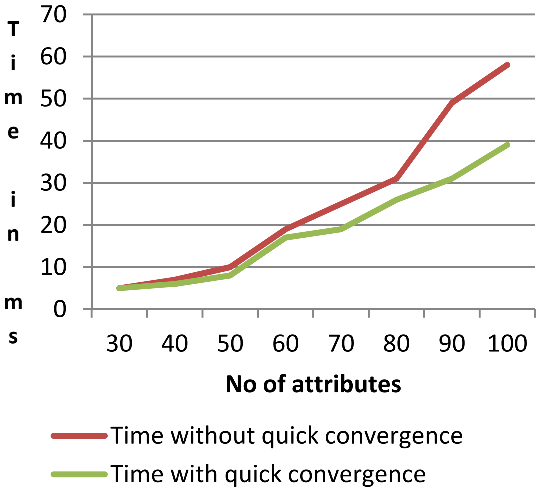

Fig 5: Time taken with and without quick convergence

The figure shows time taken for the attributes weights to be assigned with and without quick convergence. The x axis shows the number of attributes and y axis shows the time taken for assigning the weights in milliseconds. From the figure we can see that the time taken for assigning weights reduces comparatively when using the quick convergence technique.

5. Conclusion

The results obtained from this method can be directly used in a neural network system. This reduces the training time of neural networks, in turn increasing the accuracy of the system and provides enhanced generalization by reducing over fitting. The usage of quick convergence provides feature selection; hence we need not use all the attributes for analysis. This provides the user with multiple advantages. The processing time gets reduced due to the removal of redundant attributes, and the process provides maximum accuracy due to the fact that irrelevant attributes that have the probability of reducing the accuracy of the result are removed from the attribute list.

Further, using the fuzzy association rule mining provides relationship between various attributes, and the bonds with maximum strength (high confidence) are retained by the system. This combination assures an accurate result from the system.

References

[1] Ms.Smita.Nirkhi, “Potential Use of Artificial Neural Network In Data Mining”, 978-1-4244-5586-7/10, 2010 IEEE.
[2] Zhijun Chen, “Research of Data Mining Based On Neural Network”, 978-1-4244-7161-4/10, 2010 IEEE.
[3] M. Robnik-Sikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF”, Mach. Learn., 53(1- 2):23.69, 2003.
[4] H. Almuallim and T.G. Dietterich, Learning with many irrelevant features, in: Proceedings AAAI-91, Anaheim, CA (MIT Press, Cambridge, MA, 1991) 547-552, 1991.
[5] J.H. Gennari, P Langley and D. Fisher, “Models of Incremental Concept Formation”, Artl$cial Intelligence 40, 11-61, 1989.
[6] M.L. Ginsberg, “Essentials of Artijiciul Intelligence”, (Morgan Kaufmann, Los ALtos, CA), 1993
[7] S.J. Russell and P Norvig, “Artificial Intelligence: A Modern Approach”, (Prentice Hall, Englewood Cliffs, NJ), 1995.
[8] Jay Gholap, Anurag Ingole, Jayesh Gohil, Shailesh Gargade, Vahida Attar, “Soil Moisture Estimation Using Classification And Regression Trees And Neural Networks”.
[9] Ramesh Vamanan, K.Rama, Ramesh Vamanan et al, “Classification Of Agricultural Land Soils: A Datamining Approach”, International Journal on Computer Science and Engineering (IJCSE).
[10] Cheng-Lung Huang, Chieh-Jen Wang, “A GA-Based Feature Selection And Parameters Optimization For Support Vector Machines”, Expert Systems with Applications 31, 231–240, 2006.
[11] Ashish Mangalampalli, Vikram Pudi, “Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets”, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IIIT/TR/2009/173, 2009, IEEE.
[12] P. Ponce-Cruz, F.D. Ramirez-Figueroa, “Fuzzy Logic”, Intelligent Control Systems with LabVIEW™ , Springer 2010.
[13] P.Bhargavi1, Dr.S.Jyothi, “Fuzzy C-Means Classifier for Soil Data”, International Journal of Computer Applications (0975 – 8887), Volume 6– No.4, September 2010.
[14] Improved Techniques For Enhancing The Accuracy Level In A Feed-Forward Neural Networks.