Detecting Medical Conditions using Neural Network and a Genetic Algorithm for Neural Network Pruning

Hao Chen

The Australian National University,
Canberra ACT 2601
u6633251@anu.edu.au

Abstract. This paper investigates the performance of using a three-layer neural network and fuzzy k-mean cluster to predict whether the patient is normal, with SARS, high blood pressure, or with pneumonia. Then, this paper uses casual index and decision tree to generate some rules and try to explain the output from neural networks. The result shows decision tree method works better for linear separable dataset. In some rule extraction algorithm, the first step is to prune the neural network. Therefore this paper also explore the potential of using casual index or genetic algorithm to prune the original neural network. The result shows that both the casual index and genetic algorithm can prune the neural network efficiently for linear separable dataset.

Keywords: Predict medical conditions, Three-layer-neural network, Casual index, Characteristic input, Fuzzy-C-means, Genetic algorithm

1 Introduction

1.1 Dataset

The purpose of the experiment to obtain this data set is to distinguish normal people, people with SARS, with high blood pressure, and with pneumonia. [1] The data set used in the experiment composed of data collected from 1000 normal person, 1000 person with high blood pressure, 1000 person with SARS, and 1000 person with pneumonia.The fuzzified features in the dataset include a variety of observations and basic measurements about the patients, and table 1 illustrate them.

Table 1. Fuzzy labels in the dataset

Measurement	Associated fuzzy labels
Temperatures (measure four times a day)	'Slight', 'Moderate', 'High'
Nausea	'Slight', 'Medium', 'High'
Blood pressure (systolic and diastolic)	'Slight', 'Medium', 'High'
Abdominal pain	'No', 'Yes'

1.2 Problem description

Although the neural network has relatively high classification accuracy in wide varieties of application areas, the artificial neural network has one disadvantage: the artificial neural network like a black box, and how a network-making decision is not easily comprehensible. The lack of transparency reduces the acceptability of neural networks in many application areas. In order to solve this drawback to some extent, many researchers proposed many rule extraction algorithms. This paper solves predicting whether a person is normal or with

some disease using the attributes collected. First of all, this paper uses the traditional neural network method to train the three-layer neural network and uses this neuron network as a model to predict. Then this paper will use rule extraction algorithms based on casual index and decision tree to generate decision rules and then use the rule generated to predict and explain the neural network behavior. This paper will compare the performance of those if-else rules to the original neural network. In addition, some rule-extraction algorithm need to prune the neural network at first, this paper will also explore whether neural network pruning based on casual index or genetic algorithm is efficient for linear separable data.

1.3 Causal index

Many approaches and algorithms have been proposed to explain neural network prediction, which at minimum require the production of a set of decision rules [1]. One of the simplest solutions is to prune the network to a minimal size, forcing the internal representation to approximate a symbolic and readily extractable mode. However, it will reduce the robustness of the neural network [1]. Of course, rules can be derived from the existing neural network by encoding its activation values into rules. However, even for a single neural, the number of IF-THEN rules could grow exponentially with the number of inputs.[2] This paper uses another technique; instead of explaining neuron networks by each activation value, this paper uses a casual index concept. Since the neural network activation function used in this experiment is sigmoid, it is differentiable. We can determine the rate of change of output neural k with respect to input neural i by deriving the differentiability of the activation function. This rate is called causal index; it could be used to generate decision rules[3]. Another way of finding the effect an input has on the output of the network is sensitivity analysis; that is, we make a small change to input neuron i, and see how much the output neuron k will change. If a drastic change occurs, then we can conclude that the attribute fit into input neuron i is a critical factor in producing the current activation value of k.

1.4 Characteristic input

Since we may use the causal index method. To reduce computational complexity, we use a compressed representation of the training set to generate most explanations and only generate explanations for specific cases where necessary. This compressed representation is in the form of characteristic input patterns [1]. That is, we classify Input patterns in terms of their effect on each individual output. The set of patterns that activate that output neuron is used to produce a characteristic pattern [1]. Another set of patterns that do not activate that neuron can produce a characteristic OFF pattern. In this work, we will use the arithmetical mean of the vector components to calculate characteristic input patterns. As for the problem we want to solve, the data set divides all the patients into four labels. Take the label high blood pressure as an example, we can directly regard the person pattern makes the prediction as high blood pressure as output ON and others(classified as other three labels) as output OFF. Then we could calculate the arithmetical mean of these two groups to get characteristic ON and OFF patterns for high blood pressure. Similarly, we could derive characteristic patterns for the other three labels, and this method can be used to predict the output of the neural network according to the input.

1.5 Fuzzy-C-means

Also Known as “soft k means”, and as the name suggest, it is very similar to k means cluster. But in k means cluster, each data point is assigned to the cluster with the closest centroid, and each data point can belong to only one of the clusters.In another word, the membership in each cluster is 0 or 1.The k means clustering will first choose a number of clusters k and initialize cluster centers. In each iteration,for each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster. Then re-compute cluster centers.The algorithm will stop when there are no new re-assignments. The only difference between fuzzy-C-means and K means clustering is fuzzy-C-means uses soft clustering instead of hard partitions. Every object belongs to every cluster with a membership between 0 and 1 and clusters are treated as fuzzy sets

1.6 Genetic algorithm

Genetic Algorithm is a search-based optimization technique based on the principles of nature selection. It is frequently used to find optimal or near-optimal solutions. It is frequently used to solve optimization problems by finding optimal or near-optimal solutions. Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search, as they exploit historical information as well. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

2 Method

2.1 Data preprocess

The original dataset we got is composed of four tables corresponding to four labels. Each table composed of 1000 people belongs to this label. Since the table does not contain any heading, the first row in the table becomes the heading. So the first thing we did is add a heading to all the attributes and add the first row to the table. Then we merge these four tables into a single one and divide all the samples into a training set and a testing set. We randomly select 3200 samples as training data and 800 samples as testing data. The reason for not including validation data is that the neural network could quickly achieve 100% prediction accuracy without hyperparameter pruning, so we could use more samples to train and test. We considered using a technique like cross-validation to train the neural network. Finally, all the data are labeled according to the label they belong to. High blood pressure will be label as 0, ordinary people will be label as 1, SARS will be label as 2, and pneumonia will be label as 3. It seems all the attributes in the dataset already range from 0 to 1, so there is no need to normalize them(since the attributes has been fuzzified).

2.2 Neural network

This paper uses a three-layer neural network model, which includes an input layer, output layer, and a hidden layer. The dataset in this paper includes 23 attributes. Therefore the number of neurons in the input layer is set to 23. Similarly, there are four neurons in the output layer because the number of labels is four, and the hidden layer is composed of 8 neurons. This paper uses the sigmoid function as the activation function of the neurons. This function will map all real numbers to (0,1) intervals and normalize the data by the nonlinear method [5]. For the output layer, a softmax function will convert the output into a probability distribution. This paper use cross-entropy loss as the loss function. Based on the cross-entropy loss of the test data, the number of neurons in the hidden layer is set to 8, the learning rate is 1.0, and the model was trained 300 epochs. During the training process, all the loss values and partial accuracy are recorded and plotted. The network training and testing performance are determined by precision, cross-entropy loss, and confusion matrix. These three metrics show that the neural network performance is outstanding since it could assign all the samples in the test set to its correct label and reach 100% accuracy with extremely low cross-entropy loss.

2.3 Rule extraction using decision tree

One of the simplest ways to generate decision rules to explain the behavior of the neural network is to train another decision tree after the neural network finished the training and then extract decision rules from the decision tree. After the neural network finished the training, we use the label it predicted as the label we will pass into the decision tree and other attributes to train a decision tree. Since we use the label predicted by the neural network as the label of each sample, the decision tree will learn why the neural network predicts in such a way, so the decision rules generated by it could help to explain the neural network result.

After training a decision tree as described above, we then pass all the test data to the decision tree and compare the result with the neural network. What percentage of the predictions are the same could help us evaluate how much the decision tree has explained the neural network's behavior. Surprisingly, the decision tree makes the same prediction with the neural network. In other words, the accuracy is 100%, which means these decision rules could perfectly explain the behavior of the neural network.

2.4 Casual index and characteristic input

This paper uses the training set to determine the characteristic input. In this paper, the method we used is to group the data according to the prediction results of the neural network. For each label, those samples assigned to this label are into the ON group, and those assigned to other labels are into the OFF group. Then, the arithmetic mean of each group of data is used as the characteristic input for the corresponding output. Next, when we need to predict the test data, we use the euclidean distance between the sample and characteristic input to predict the neural network's output. The characteristic input, which has the smallest euclidean distance to the sample, becomes the most similar characteristic input. The characteristic input with the second smallest euclidean distance becomes the next most likely output. We compare the prediction of using characteristic input with the neural network's prediction and found they are the same.

For a three-layer neural with multiple output described in figure 1, we could use casual index to determine which characteristic inputs are important.

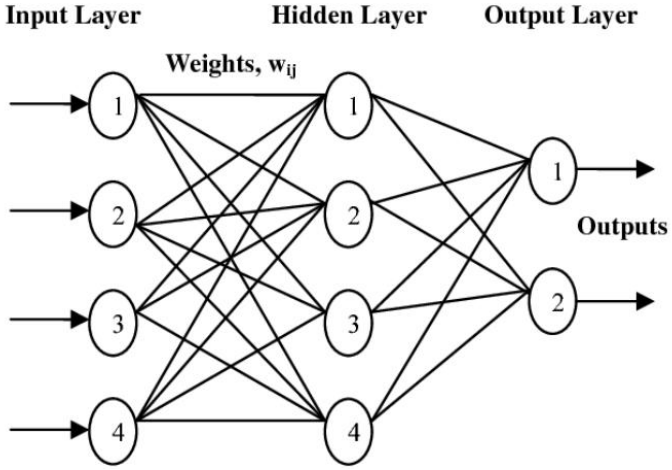

Fig. 1. Struture of a three layered ANN with multiple output

And based on the formula proposed in [3], causal index is the gradient of the output with respect to the input. Therefore we could calculate a full casual index matrix by the dot product of the weights connected first layer and hidden layer with the weights connect to hidden layers and output layers. The weights connected input layer and hidden layer is an 8*23 matrix, and the weights connect hidden layer and output layer is in the shape of 4*8. By multiplying these two matrix, we get a full casual index with shape 4*23. In this matrix, the i row j column element represents the derivative of label i with respect to attribute j. Therefore, for each output label, we could find some attributes with a large derivative, and they can be considered essential attributes for this label.In this paper, we use a naive way to generate rules from a full casual index matrix. Basically we pick the characteristic inputs and input variables with large absolute derivatives and use them to make decision rules. For each label, we will extract a certain number of rules and make a rule set. The value of the casual index indicated whether there is a positive or negative correlation between input and output signals, a positive and large casual index means if input i in large then output k is large, and a large negative casual index indicates if input i is large then output k is small.

For example, suppose a characteristic input C, the derivative of output neuron 2, and input attribute 1 is significant. In this case, we will make a rule like, if attribute 1 is greater than the corresponding attribute value in characteristic input, this sample belongs to output 2. We may find that several attributes all have large

derivatives. As a result, we will make several of these kinds of rules. They will be connected by AND for characteristic ON and connected by OR for characteristic OFF.

2.5 Casual index used in neural network prune

The first step of some rule extraction algorithms is to prune the original neural network to a smaller size. Inspired by the paper [5], it uses sensitivity analysis to prune the neural network. So this paper investigates whether we could use the casual index to prune the neural network. Again, we look at the full casual index matrix for each output and look at which input has a sizeable causal index with this output. We found that the most significant three casual indexes for all four labels all come from attributes 12,14,15,17. So we build another neural network and only include these four attributes, so this new neural network has four neurons in the input layer, six neurons in the hidden layer, and four neurons in output layers.

2.6 Genetic algorithm used in neural network prune

We want to prune the original neural network into a 3*2*4 simple neural network. The neural network will only be trained with 200 epochs and the learning rate set to 0.4.

Genetic algorithm is a search-based optimization technique based on the principles of natural selection. So we need to create the initial population at the first step. Since we want to reduce the number of attributes from 23 to 3, the chromosome is represented as a list and the list stores three attributes index. At the beginning of the algorithm, we randomly assign 23 attributes into eight chromosomes. For the fitness of the chromosomes, this paper directly uses the accuracy of the neural network to judge fitness. In each generation(iteration of the algorithm), the algorithm needs first to train eight neural networks using these chromosomes' attributes and then sort these chromosomes based on the accuracy. The chromosome with the best accuracy and the second-best accuracy will be selected to crossover. During the crossover, these two chromosomes will randomly select a gene(attribute) and exchange them. After crossover, there is a ten percent possibility to mutate, which means one of the chromosomes' genes (attribute) will change to a randomly selected attribute. If we find a chromosome that achieves 100% accuracy or the algorithm has reproduced ten generations, the algorithm will end.

3 Result and Explanation

Table 2. The accuracy of the method prediction and explain

Method	Predict accuracy	Explain accuracy
Original 3-layer Neural Network	100%	~
Decision Tree	100%	100%
Casual index and characteristic input	100%	75%
Pruned neural network using casual index	>97%	~
Fuzzy K mean cluster	100%	~
Pruned neural network using genetic algorithm	100%	~

The predicted accuracy of the neural network is compared against the true label. However, the predicted accuracy of row 2 to row 5 is compared with the neural network prediction. For the explain accuracy, it is calculated after the rules have been generated. We then assign all the test sample points based on these rules and then calculate the proportion of the samples assigned to the same label with the neural network prediction.

The first row of table 1 shows that even using a simple 3-layer neuron network; the disease can be perfectly predicted. The second row tells us that using a decision tree to explain the behavior of this neuron network is an excellent choice. 3-layer neuron network generally not too complex, the depth of the decision tree is relatively shallow, so the rule generated is concise and efficient. The third row also illustrates that using characteristic input to predict the neural network's behavior is worth considering. However, our naive way of generating rules not works so well compared with the decision tree method.

In addition, the rule generated in this naive way may not cover all the sample points in the test set, which means these rules can not identify some samples. The fourth row indicates that sometimes using the causal index to prune the neural network can be very efficient; the accuracy could remain relatively high even the complexity of the neural network decrease dramatically.

The accuracy of the pruned neural network using causal index indicates although the size of the neuron network is drastically decreasing compare with the original neural network,this new neuron could work surprisingly well. The accuracy on the test data set could reach 97%, which means most of the information of the original neural network has been reserved by this pruned neural network.

The pruned neural network's accuracy using a genetic algorithm illustrates that for a linearly separable dataset, most of the attributes may be redundant, and the neural network could make a very accurate prediction using a small proportion of the attributes. Our pruned neural network only has three input neurons, two hidden neurons, and four output neurons, but the accuracy of the neural network could reach 100%. If we want to extract rules from a neural network, it is much easier to get these rules from a much simpler neural network. Moreover, the result proved that the genetic algorithm could be considered the potential approach to prune the neural network.

4 Conclusion and Future Work

In this paper, Using a 3-layer neuron network on a linear separable dataset achieve good performance, and the behavior of the neuron network can be perfectly explained by the decision tree method. The boundary between each label in the input space seems well defined, so this dataset can be predicted and explain so well. However, it may not be the case for a more complex dataset. If the dataset is complex, the depth of the decision tree will grow, so the number of rules it generates could grow exponentially. It may be hard for a human to understand these rules without an algorithm to simplify them. Using the casual index to generate decision rules does not work well for this dataset, but it may be a better choice compared with the decision tree method for other datasets. Using characteristic input to predict neural network behavior works well for this dataset. However, this method also requires that the boundary between each class or label is not complex, so this method is not sophisticated, and same for fuzzy-c-means. And for both methods, we need extra space and time to build a decision tree or calculate a casual index matrix. If the dataset is large or the neural network is complex, then the cost of trying to explain is relatively high.

Although many researchers proposed many new methods to extract rules and try to explain the behavior of the neural network, we still do not have a perfect algorithm that could perfectly explain the prediction of the neural networks. The existing algorithm either requires us to take some extra steps during the neural network training or adapt some complex algorithms after the training to extract rules. For this dataset, using the casual index to prune the network works quite well. It could be considered an option to prune the neural network and to extract rules from a much simpler NN. However, this method is only possible for a relatively simple neural network.

For linearly separable data, the genetic algorithm could performance outstanding. The complexity of the neural network could reduce dramatically with this algorithm while still achieve good performance. However, the limitation of this algorithm is obvious, and it requires us to train many small neural networks. Although they are much simpler than the original neural network, it may still not be affordable for more complex cases.

The dataset used in this paper is linearly separable, and the performance of these methodologies has not been tested on the more complex dataset. It is unclear whether a genetic algorithm or causal index could achieve good performance when applying to a dataset that is not linearly separable. As a result, future work is recommended in exploring the applicability of these methods to an extended dataset containing more medical conditions.

References

1. Gedeon T.D. and Turner H,S,:Explaining student grades predicted by a neural network. In proceeding of 1993 International Joint Conference on Neural Networks(1993)
2. Gallant, SI"connectionist expert systems,"Communications of the ACM, vol. 31,no.2,pp.152-169,February 1988
3. Hora,N,Ebbutsu,I and Baba,K "Fuzzy rule extraction from a multilayer neural net,"Proc.IEEE,Turner, H and Gedeon, TD"Extracting Meaning from Neural Networks," Proceeding 13th Int, Con,on AI, Avignon,1993
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181--184. IEEE Press, New York (2001) Yoda,M,Baba,K and Enbutu,I "Explicit representation of knowledge aquired from plant historical data using neural networks," International Joint Conference on Neural Networks, San Diego, vol. 3,pp. 155-160,1991
5. AP Engelbrecht , HL Viktor : Rule Improvement Through Decision Boundary Detection Using Sensitivity Analysis. In Lecture Notes in Computer Science, 78-84,1999