Extracting Classification Rules for SARS-CoV-1 from A Neural Network with Feature Reduction Techniques

Pengbo Wei
Research School of Computer Science, Australian National University
u6547246@anu.edu.au
May 2021

Abstract. The goal of this study was to develop a feedforward neuron network to classify sars-cov-1 based on sars-cov-1 dataset [5], and extract classification rules from the trained neural network with feature reduction techniques. Two rule extraction methods were adopted: The first method was proposed by Gedeon and Turner (1993) on a back-propagation trained feed-forward neural network, where causal index technique was applied for selecting the essential features in a neural network. The second method involves extracting rules from a neural network by a decision tree with genetic algorithm which was used to reduce the number of features in a neural network. In addition to this, after comparing the two rule extraction methods, the decision tree with genetic algorithm method was proven to be better than Gedeon and Turner's approach. However, the generated rules by the two methods were not convincing. Therefore, the explanation procedure proposed by Gedeon and Turner (1993) was adopted to help make persuasive and accurate decisions. Finally, the rules generated by a decision tree with genetic algorithm from a trained neural network achieved high performance on classifying sars-cov-1 and the explanation procedure performs well on explaining the decision-making process.

Keywords: Classification of SARS-CoV-1, Rule extraction, Decision tree, Causal index, Feedforward neural network, Genetic algorithm

1 Introduction

Nowadays, COVID-19 has a huge negative impact on people's life whereas SARS-CoV-1 is a similar disease to COVID-19. By analysing SARS-CoV-1, the most relevant symptoms of SARS-CoV-1 can be found so that SARS-CoV-1 and common diseases can be easily differentiated, it can inspire medical workers to find an effective way to detect COVID-19.

Neural networks have a better performance than other machine learning classification models. With the help of neural networks, different diseases can be easily classified. In this paper, I designed a neural network with three layers to classify SARS-CoV-1 based on medical records and constructed a set of experiments to find good hyper-parameters of this neural network.

However, in the medical area, a single judgement without explanations is not enough as patients would not simply believe the results without any reasonings. What's more, neural networks are block-box models, which makes it difficult to understand the process of making decisions in a neural network. More specifically, the rules in a neuron network are represented as weights and bias of the network, and the network is always non-linear [1]. In previous work, Gedeon and Turner (1993) proposed a rule extraction method and achieved a good performance on predicting student grades. In this method, Gedeon and Turner utilized a full causal index technique to select important features in a neural network and then generated rules based on these features. Nowadays, decision tree is a famous model with a good explanatory ability and a set of if-then-else rules can be produced easily. However, the number of features in a dataset affects the performance of decision tree because too many features would cause many if-then-else rules. To avoid such drawback, I adopted genetic algorithm to reduce the number of input features in a neural network because genetic algorithm achieved high performance on selecting features [7]. Then I used a decision tree to extract rules from this network.

In order to find which method has better performance, I experimented with the two rule extraction methods and found decision tree with genetic algorithm works well than Gedeon and Turner's method. To further make a convincing and accurate judgement, the explanation procedure [2] was also adopted for explaining the trained neural network.

2 Methodology

2.1 Dataset information and data pre-processing

The sars-cov-1 dataset used for this paper contains four files: High BP, SARS, Normal, and Pneumonia. Files' names mean the label of data in each file. Each file has 1000 entries and contains the following features:

- Temp. at 8am - Patient's temperature at 8am. This feature is split into three columns: Slight, Mod, High. The values in each column are float type in range 0 to 1.
- Temp. at 12pm - Patient's temperature at 12pm. This feature is split into three columns: Slight, Mod, High. The values in each column are float type in range 0 to 1.
- Temp. at 4pm - Patient's temperature at 4pm. This feature is split into three columns: Slight, Mod, High. The values in each column are float type in range 0 to 1.
- Temp. at 8pm - Patient's temperature at 8pm. This feature is split into three columns: Slight, Mod, High. The values in each column are float type in range 0 to 1.
- BP Systolic - Patient's systolic blood pressure. This feature is split into three columns: Slight, Med, High. The values in each column are float type in range 0 to 1.
- BP Diastolic - Patient's diastolic blood pressure. This feature is split into three columns: Slight, Med, High. The values in each column are float type in range 0 to 1.
- Nausea - This feature is split into three columns: Slight, Med, High. The values in each column are float type in range 0 to 1.
- Abdominal Pain - This feature is split into two columns: No, Yes. The values in each column are float type in range 0 to 1.

The summary tables of the four files are shown as below:

HighBp file:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
count	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.0	1000.0	1000.0	1000.0
mean	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.350271	0.900005	0.0	0.351131	0.897459	1.0	0.0	0.0	1.0	0.0
std	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.059822	0.039767	0.0	0.059192	0.039328	0.0	0.0	0.0	0.0	0.0
min	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.200000	0.800000	0.0	0.200000	0.800000	1.0	0.0	0.0	1.0	0.0
25%	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.306875	0.872200	0.0	0.309475	0.870175	1.0	0.0	0.0	1.0	0.0
50%	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.352200	0.902050	0.0	0.349800	0.898100	1.0	0.0	0.0	1.0	0.0
75%	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.391925	0.928500	0.0	0.389125	0.926100	1.0	0.0	0.0	1.0	0.0
max	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.500000	1.000000	0.0	0.500000	1.000000	1.0	0.0	0.0	1.0	0.0

Table 1. Summary of HighBp file

SARS file:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.0	1000.000000
mean	0.101169	0.703283	0.751284	0.050087	0.748738	0.749452	0.0	0.749864	0.749064	0.0	0.752422	0.751402	0.0	0.751636	0.751094	0.101017	0.744870	0.746805	0.100987	0.748733	0.756530	0.0	0.748356
std	0.040363	0.121881	0.097093	0.020077	0.101534	0.099192	0.0	0.099513	0.096415	0.0	0.096808	0.097058	0.0	0.099488	0.102471	0.039901	0.097728	0.097907	0.039780	0.098159	0.096952	0.0	0.101780
min	0.000000	0.400000	0.500000	0.000000	0.500000	0.500000	0.0	0.500000	0.500000	0.0	0.500000	0.500000	0.0	0.500000	0.500000	0.000000	0.500000	0.500000	0.000000	0.500000	0.500000	0.0	0.500000
25%	0.073875	0.621775	0.685800	0.036800	0.679950	0.680900	0.0	0.682150	0.685975	0.0	0.686325	0.690475	0.0	0.682675	0.681875	0.072875	0.677475	0.680000	0.073275	0.679900	0.688800	0.0	0.679850
50%	0.101700	0.702000	0.750750	0.049300	0.749500	0.750150	0.0	0.747500	0.748300	0.0	0.750200	0.747650	0.0	0.757100	0.751600	0.100250	0.744650	0.750500	0.101500	0.750600	0.760550	0.0	0.752600
75%	0.127625	0.783025	0.815575	0.064400	0.817675	0.817300	0.0	0.818900	0.809675	0.0	0.817850	0.817250	0.0	0.821025	0.816800	0.126850	0.815400	0.814750	0.129550	0.820750	0.821750	0.0	0.814575
max	0.200000	1.000000	1.000000	0.100000	1.000000	1.000000	0.0	1.000000	1.000000	0.0	1.000000	1.000000	0.0	1.000000	1.000000	0.200000	1.000000	1.000000	0.200000	1.000000	1.000000	0.0	1.000000

Table 2. Summary of SARS file

Normal file:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
count	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.000000	1000.000000	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
mean	0.898268	0.101806	0.0	0.901062	0.099878	0.0	0.899694	0.100105	0.0	0.900173	0.101914	0.0	0.900296	0.100182	0.0	0.900844	0.100755	0.0	1.0	0.0	0.0	1.0	0.0
std	0.037645	0.040731	0.0	0.039242	0.040402	0.0	0.040058	0.038193	0.0	0.038887	0.039788	0.0	0.038706	0.039881	0.0	0.041217	0.039462	0.0	0.0	0.0	0.0	0.0	0.0
min	0.800000	0.000000	0.0	0.800000	0.000000	0.0	0.800000	0.000000	0.0	0.800000	0.000000	0.0	0.800000	0.000000	0.0	0.800000	0.000000	0.0	1.0	0.0	0.0	1.0	0.0
25%	0.872200	0.075050	0.0	0.874775	0.073200	0.0	0.872325	0.076075	0.0	0.874700	0.075375	0.0	0.876100	0.071200	0.0	0.873850	0.072975	0.0	1.0	0.0	0.0	1.0	0.0
50%	0.899500	0.102400	0.0	0.901850	0.098300	0.0	0.900000	0.100700	0.0	0.899950	0.102250	0.0	0.899000	0.101450	0.0	0.900600	0.099850	0.0	1.0	0.0	0.0	1.0	0.0
75%	0.924525	0.128700	0.0	0.927150	0.129000	0.0	0.926900	0.126150	0.0	0.925700	0.128400	0.0	0.926750	0.128000	0.0	0.927150	0.126100	0.0	1.0	0.0	0.0	1.0	0.0
max	1.000000	0.200000	0.0	1.000000	0.200000	0.0	1.000000	0.200000	0.0	1.000000	0.200000	0.0	1.000000	0.200000	0.0	1.000000	0.200000	0.0	1.0	0.0	0.0	1.0	0.0

Table 3. Summary of Normal file

Pneumonia file:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
mean	0.098268	0.803615	0.900744	0.101062	0.799758	0.899278	0.099694	0.800208	0.899286	0.100173	0.803826	0.900725	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
std	0.037645	0.081459	0.040306	0.039242	0.080808	0.040101	0.040058	0.076385	0.039826	0.038887	0.079577	0.038631	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	0.000000	0.600000	0.800000	0.000000	0.600000	0.800000	0.000000	0.600000	0.800000	0.000000	0.600000	0.800000	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
25%	0.072200	0.750150	0.873675	0.074775	0.746375	0.872075	0.072325	0.752050	0.871975	0.074700	0.750675	0.874400	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
50%	0.099500	0.804800	0.900500	0.101850	0.796550	0.899550	0.100000	0.801450	0.898600	0.099950	0.804550	0.899800	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
75%	0.124525	0.857325	0.926250	0.127150	0.857950	0.926025	0.126900	0.852375	0.926200	0.125700	0.856800	0.927000	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
max	0.200000	1.000000	1.000000	0.200000	1.000000	1.000000	0.200000	1.000000	1.000000	0.200000	1.000000	1.000000	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0

Table 4. Summary of Pneumonia file

There are no missing entries and outliers found in all of the four files. To generate a training dataset, I firstly added a new column named label in each file to record the label information and then combined the four files into one file. The dataset is then split into training set and testing set with the ratio of 8:2. To construct a general neural network, the number of each category's entries should be same in both training set and testing set. For example, training set should contain 800 entries from SARS, 800 entries from Normal, 800 entries from High BP, and 800 entries from Pneumonia. Normalization step is not required in this procedure because values in each column are already from 0 to 1.

2.2 Feedforward neural network model design

A feedforward neural network is constructed with three layers: the input layer has 23 input neurons as the dataset contains 23 columns, one hidden layer with the optimal number of hidden neurons, and the output layer with 4 output neurons that matches with four labels.

To construct a neural network with a generally good performance, the number of entries of each category should be the same in the training dataset. I used accuracy and confusion matrix methods to estimate the performance of the network because the given dataset is class-balanced [3]. This dataset contains four labels, and I used SoftMax function as output activation function and cross-entropy loss function to calculate loss value. MacLeod proves that the Adam optimizer is computation efficient and can handle sparse gradients well [4]. Therefore, I chose the Adam algorithm is chosen as then optimizer function.

With such trained neural network, a disease label is then predicted based on patient's medical record.

2.3 Optimal number of hidden neurons experiment

The number of hidden neurons is an important hyperparameter. Specifically, a small number of hidden neurons will make the network underfit while a large number of hidden neurons will make the network overfit. Moreover, a good number of hidden neurons has a positive impact on extracting rules. If the number of hidden neurons is too large, it is hard to extract rules and the extracted rules would be confusing [2]. Thus, the objective of this experiment is to find the smallest optimal number of hidden neurons with high accuracy.

In order to find the optimal number of hidden neurons, I experimented with a range of numbers of hidden neurons from 1 to 5 based on the neural network I have designed in section 2.2 with learning rate equals to 0.01 and number of epochs is 1000. In addition, each experiment was repeated by 10 times. Then, I plotted the mean result of each hidden neurons number as in Fig. 1. From Fig. 1, it shows that 2 is a good number of hidden neurons because it is the smallest number and with high test accuracy.

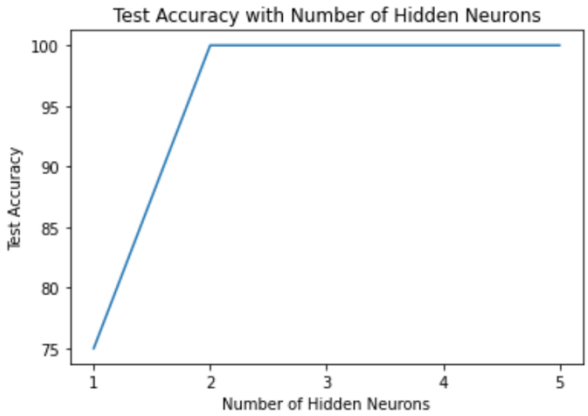

Fig. 1. Test accuracy with number of hidden neurons.

From Fig. 1, it shows the accuracy achieved 100% with two hidden neurons. Thus, I kept the setting of learning rate as 0.01 and number of epochs as 1000.

2.4 Extracting rules by causal index method

In [2], the author introduced a method called causal index method to find the important features and generate a set of rules. That is, each pair of causal indexes can be calculated by finding an output neuron gradient w.r.t an input neuron. And an output neuron gradient w.r.t an input neuron can be calculated by using the chain rule of differentiation. So, the

features with high causal value w.r.t an output neuron means are vital to this output neuron. By using the causal index on characteristic input patterns, the set of rules can be generated [2]. Here is an example of causal index on High BP characteristic ON input pattern w.r.t High BP output neuron.

	Slight	Mod	High
Temp 8am	3.81E-04	-4.74E-04	-3.10E-04
Temp 12pm	4.21E-04	-4.38E-04	-4.07E-04
Temp 4pm	3.16E-04	-4.81E-04	-3.81E-04
Temp 8pm	2.81E-04	-4.23E-04	-3.87E-04

	Slight	Med	High
BP Systolic	-6.20E-04	1.12E-03	6.65E-04
BP Diastolic	-5.65E-04	1.15E-03	5.69E-04
Nausea	4.07E-05	-2.21E-05	-1.26E-04

	No	Yes
Abdominal Pain	-5.57E-05	-7.95E-05

Table 5. Causal index on High BP characteristic ON input pattern

From Table 5, the negative values indicate that features have a negative impact on this output neuron. Similarly, the positive values indicate that features have a positive impact on this output neuron. It is clear that BP Systolic Med and BP Diastolic Med have the highest causal value, BP Systolic Slight and BP Diastolic Slight have the smallest causal value. This means that BP Systolic and BP Diastolic are important to this output neuron. In reality, BP Systolic and BP Diastolic are key factors to judge a person is within a High BP or not. The causal index values also prove this point of view. Therefore, in this situation, these features can be chosen as the important features to generate rules. For simplicity, I chose BP Systolic Med and BP Diastolic Med as the important features because they have the highest absolute causal value. Similarly, I also found the BP Systolic High and BP Diastolic High are important features based on the causal index on High BP characteristic OFF input pattern. Thus, BP Systolic Med, BP Diastolic Med, BP Systolic High, and BP Diastolic High can be used to generate rules for High BP label.

To extract rules based on the important features, the method in [6] can be adopted. For an important feature that has a positive impact on an output neuron, the boundary can be found when this feature value turns the output neuron on with a gradient equal to 0. This is because this feature would contribute to this output neuron when the gradient is larger than 0. Similarly, for an important feature has a negative impact on an output neuron, the boundary can also be found when this feature value turns the output neuron off with a gradient equal to 0. Find these boundaries directly is difficult because there is no support tool in calculating these values. So, to find a boundary, I assigned an important feature to a group of values in the range 0 to 1 with step of 0.01. Other features keep same as characteristic input pattern and recorded the gradient value in each iteration. Here is an example of BP Diastolic Med input feature with respect to High BP output neuron.

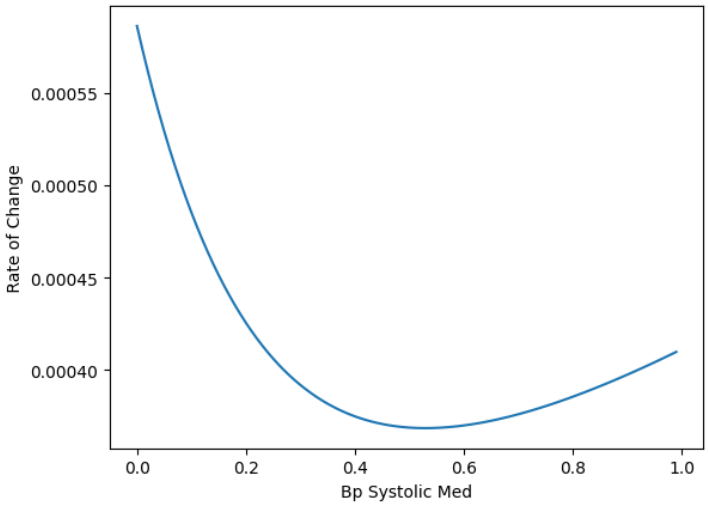

Fig. 2. BP Diastolic Med vs Rate of change w.r.t High BP output

From Fig. 2, as the rate of change approaches to 0, the value of BP Systolic Med is around 0.53. Therefore, 0.53 can be used as the boundary of BP Systolic High. Similarly, the boundary of other features can be found. After finding these boundaries, a rule can be generated. Here is an example of High BP ON rule.

Characteristic pattern	Rule Set
ON High BP	((BP Diastolic Med > 0.53) ∧ (BP Diastolic High > 0.28)) OR ((BP Systolic Med > 0.73) ∧ (BP Systolic High > 0.06))

Table 6. ON High BP pattern

The rules for other characteristic patterns can also be generated based on this method. Here is the result of generated rules.

Characteristic pattern	Rule Set
ON High BP	((BP Diastolic Med > 0.53) Λ (BP Systolic Med >0.73)) OR ((BP Systolic High > 0.06) Λ (BP Diastolic High > 0.28))
ON Normal	(Temp12pm High <=0) Λ (BP Systolic Med < 0.11) Λ (BP Diastolic Med < 0.1)
ON Pneumonia	(Temp 12pm High >0) Λ (Temp 4pm Mod >0) Λ (Temp 8pm High >0)
ON SARS	(Bp Diastolic Med < 0.61) Λ (BP Systolic Med < 0.58) Λ (Abdominal Pain Yes > 0.55)

Table 7. Rules extracted by causal index

2.5 Extracting rules by a decision tree with genetic algorithm

Decision tree is the state-of-the-art approach to generate understandable rules. However, with the high number of features, decision tree would generate too many if-then-else rules. To generate meaningful rules from a trained neural network, the number of input neurons in the neural network should be minimized. Genetic algorithm shows a good performance on optimizing NP-hard problems in practical times [8]. I applied genetic algorithm to build neural network with minimum input neurons. Here is the procedure of the genetic algorithm to find the optimal features.

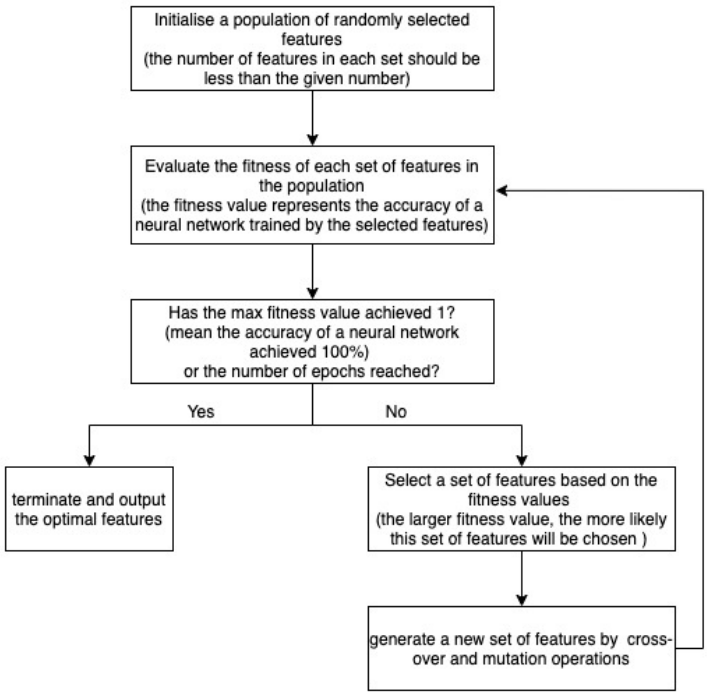

Fig. 3. Genetic algorithm procedure to find the optimal features

Basically, each DNA is a 23-dimension binary array as the dataset contains 23 features. The uniform random mutation method and the random cross-over method were used in this algorithm. Moreover, the big difference between this genetic algorithm and normal genetic algorithms is that the number of true bits in each DNA was restricted. This is because the

number of features in each DNA would increase during mutation process. Here is the hyperparameters setting in this genetic algorithm.

Parameters	Values
Number of bits in DNA	23
Number of individuals	30
Cross rate	0.8
Mutation rate	0.2
Generation size	50

Table 8. Genetic algorithm settings

It should be noted that the mutation rate plays an essential role in genetic algorithm. If the mutation rate is large, it is hard to converge to the optimal solution. Similarly, if the mutation rate is small, the optimal solution is hard to find. A good mutation rate is depending on the problem. I experimented with a range of mutation rates from 0.005 to 1 and found the small mutation rates performed worse than the large mutation rate. This is because the number of unique sets of features has a positive impact on reaching the termination criterion. Thus, I chose 0.2 as the mutation rate in this genetic algorithm. For other difficult tasks, the mutation rate would be changed during the training process to achieve a good performance. For instance, the mutation rate would be set as a large value in the initial stage and decreases during the training process.

The neural network structure used was same as section 2.2 with 2 hidden neurons. But I changed the number of epochs to 200 in order to reduce GA training time. To find the minimum features, I experimented with different number of features from 1 to 5 and the result showed in Fig.4.

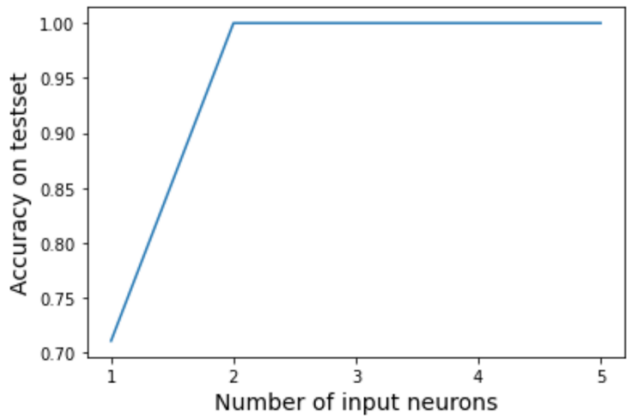

Fig.4 Accuracy vs. number of input neurons

From Fig.4, it shows a neural network with two input neurons achieved 100% accuracy on test dataset. Thus, I set two features as the upper bound of the number of features in genetic algorithm. The two important features found by genetic algorithm are Temp 8pm slight and BP Diastolic Med. After training a neural network with the two features, I trained a decision tree classifier based on the input values and predicted labels to extract rules from this neural network. Here are the generated rules.

Characteristic pattern	Rule Set
ON High BP	(BP Diastolic Med > 0.25) Λ (Temp 8pm Slight> 0.5)
ON Normal	(BP Diastolic Med <= 0.25) Λ (Temp 8pm Slight > 0.165)
ON Pneumonia	(BP Diastolic Med <= 0.25) Λ (Temp 8am Slight <= 0.165)
ON SARS	(Bp Diastolic Med > 0.25) Λ (Temp 8pm Slight <= 0.5)

Table 9. Rules generated by decision tree

2.6 Explanation procedure

The explanation procedure proposed by Gedeon and Turner [2] achieved high performance on explaining student grades. What's more, the generated rules by the two methods were not convincing (see section 3.2 and 3.3). Thus, I adopted this

explanation procedure to help make convincing and accurate decisions on classifying SARS-CoV-1. There are four steps in the explaining procedure. I modified the step 2 to make this procedure applicable to the decision tree with genetic algorithm method. In original procedure, the step 2 was to present the important features for the network output in the given input pattern. But in the decision tree with genetic algorithm method, the input features in a neural network were replaced with the important features. So, I changed the important features for the network output in the given input pattern to the important features for the network output in the dataset. Here are the four steps in this procedure.
Instead of taking important features from an input patter, now I am trying to obtain them directly from the dataset.

- Step 1: find the most similar characteristic input pattern to the given input pattern.
- Step 2: present the important features for the network output in the dataset.
- Step 3: produce a set of rules.
- Step 4: given the network's next most likely output.

A characteristic ON pattern is produced by the set of patterns which turns that output on. Similarly, a characteristic OFF pattern is produced by the set of patterns which turns that output off. I used the arithmetical mean of the vector components of those patterns to generate a characteristic pattern. To find the most similar characteristic input pattern, the given input pattern should be compared with all the characteristic patterns.

The set of rules can be generated by the two methods mentioned above. What's more, next most likely output can be calculated by finding the most similar characteristic ON input pattern.

3 Results and Discussion

3.1 Prediction performance of feedforward neural network

As shown in Fig. 1, a neural network with two or more hidden neurons achieved 100 % of test accuracy. Thus, neural network with two hidden neurons was chosen, as a small number of hidden neurons would help to extract rules from the network, whereas the large number of hidden neurons would cause overfitting and increases the training time.

The network with two hidden neurons can correctly predict a certain disease that a patient has by given this patient's medical record. However, the test accuracy would be lower in reality because the training data size is not large enough.

3.2 Prediction performance of causal index method

The rules showed in Table 3 was extracted by causal index, and it result in 57.6 % accuracy on predicting the outputs of this neural network. The reason of such low accuracy is that the values in this dataset are fuzzy values. For instance, choosing BP Diastolic Slight as an important feature to generate a rule of Normal ON characteristic pattern is not a good idea even if the causal index value of this pair is large. As the value of BP Diastolic Slight can be low while BP Diastolic is very high, which makes it hard to differentiate these two diseases.

To solve this issue, the explanation procedure can help to improve the accuracy of decision. And if these fuzzy values can be converted into original value, then the causal index would be a better method to extract rules.

3.3 Prediction performance of decision tree with genetic algorithm

These rules showed in Table 4 was extracted by the decision tree that achieved 100 % accuracy in predicting the outputs of this neural network. It also provides an invaluable insight into how the network makes its decision.

From these generated rules, BP Diastolic Med value is used to separate High BP, SARS from Normal, Pneumonia.
Temp 8 pm Slight as a factor to differentiate the High BP, SARS and Normal, Pneumonia. Commonly, a patient with SARS would have slightly higher BP diastolic. Similarly, a patient with High BP has high BP diastolic. However, using BP Diastolic Med as a factor to differentiate these diseases is not convincing because the BP Diastolic Med value of a patient with very high BP diastolic would be small. Similarly, the Temp 8 pm Slight feature is not convincing for the same reasons.

Even though these rules achieved high performance in classifying the diseases, these rules are still not convincing. To make these rules more persuasive, the other relevant features should be taken into account in the decision process. For instance, when using the ON High BP rule, the BP Diastolic High value of a patient should also be considered. In

explanation procedure, BP Diastolic High is also considered as an important feature when differentiating between High BP and other diseases. By using the explanation procedure, the final decision would be more convincing.

3.4 Example of explanation facility results

I used an input pattern to show the explanations of the two procedures. Here is a given input pattern.

	Slight	Mod	High
Temp 8am	1	0	0
Temp 12pm	1	0	0
Temp 4pm	1	0	0
Temp 8pm	1	0	0
	Slight	**Med**	**High**
BP Systolic	0	0.2411	0.8628
BP Diastolic	0	0.2595	0.8953
Nausea	1	0	0
	No	**Yes**	
Abdominal Pain	1	0	

Table 10. A given input pattern

The results of the explanation procedure with causal index method are showed as below.

Network Output	High BP
Most Similar Characteristic Input	High BP
Important Inputs	BP Systolic High [0.8629], BP Diastolic High [0.8953], BP Systolic Med [0.2411], BP Diastolic Med [0.2595].
Satisfied Rule Set	((BP Diastolic Med > 0.53) Λ (BP Systolic Med >0.73)) OR ((BP Systolic High > 0.06) Λ (BP Diastolic High > 0.28))
Next Most Likely Output	High BP

Table 11. An explanation results by using explanation procedure with causal index

The results of the explanation procedure with decision tree plus genetic algorithm method are showed as below.

Network Output	High BP
Most Similar Characteristic Input	High BP
Important Inputs	BP Diastolic Med [0.2595], BP Diastolic High [0.8953], Temp 8pm Med [0], Temp 8pm High [0].
Satisfied Rule Set	(BP Diastolic Med > 0.25) Λ (Temp 8pm Slight> 0.5)
Next Most Likely Output	High BP

Table 12. An explanation results by using explanation procedure with decision tree plus genetic algorithm

From Table 11, Table 12, the results present many aspects of information. In the medical filed, doctors can reduce risk of making a wrong judgement by just using generated rules. For example, in order to judge whether a patient has a High BP or not, the doctor would consider the most similar characteristic input at first. However, sometimes, a single feature would play an important role in judgement so these important features can help doctors to get a deep insight of patient's record to avoid wrong judgement and also help to check whether the satisfied rule is correct or not. After that, the next most likely output will show doctor another similar output in order to double check the correctness of such judgement.

4 Conclusion and Future work

Overall, the simple neural network with three layers shows a good capability to classify these diseases based on this dataset. Moreover, decision tree with genetic algorithm achieved high performance on explaining insight aspects of the trained neural network. However, the causal index method is not suitable to extract rules based on this dataset which would result in achieving a low accuracy. By using causal index method, the important features can be found, and a set of rules can be extracted based on characteristic input patterns. It provides a general explanation for the network's job. By using decision tree with genetic algorithm, the optimal features are found, and the minimum number of rules are generated based on the optimal features. Thus, each method has its own advantages and disadvantages. And a specific rule extraction method will be chosen depending on the given datasets.

Further work would be required to investigate the method to improve the performance of causal index. This could be done by either converting the fuzzy values into original values, compressing the features in the dataset, and test the performance of causal index, or modifying the way to find causal index value.

Lastly but not least, more research can be done on how to apply fuzzy logic to do classification and extract rules based on this dataset. This is because the values in this dataset are fuzzy values and fuzzy logic resembles human reasonings to generate decisions, therefore the rules generated by fuzzy logic would be easier to understand than other methods.

5 Reference

1. Krishnan, R., Sivakumar, G. and Bhattacharya, P., 1999. Extracting decision trees from trained neural networks. Pattern recognition, 32(12).
2. Gedeon, T.D. and Turner, H.S., 1993, October. Explaining student grades predicted by a neural network. In Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan) (Vol. 1, pp. 609-612). IEEE.
3. Bishop, C.M., 2006. Pattern recognition and machine learning. springer.
4. MacLeod, C., 2010. An introduction to practical neural networks and genetic algorithms for engineers and scientists. Scotland: The Robert Gordon University.
5. Mendis, B. S., Gedeon, T. D., & Koczy, L. T., 2005. Investigation of aggregation in fuzzy signatures, in Proceedings, 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore
6. Turner, H. and Gedeon, T.D., 1993. Extracting Meaning from Neural Networks. In Proceedings 13th International Conference on AI (Vol. 1, pp. 243-252).
7. Babatunde, O.H., Armstrong, L., Leng, J. and Diepeveen, D., 2014. A genetic algorithm-based feature selection.
8. Holland, J.H., 1992. Genetic algorithms. Scientific american, 267(1), pp.66-73.