Interpretable Piecewise Linear Classifier

Pitoyo Hartono

Department of Media Architecture,
Future University-Hakodate, Hakodate, Japan

Abstract. The objective of this study is to build a model of neural network classifier that is not only reliable but also, as opposed to most presently available neural networks, logically interpretable in a human-plausible manner. Presently, most of the studies of rule extraction from trained neural networks focus on extracting rule from existing neural network models that were designed without the consideration of rule extraction, hence after the training process they are meant to be used as a kind black box. Consequently, this makes rule extraction a hard task. In this study we construct a model of neural network ensemble with the consideration of rule extraction. The function of the ensemble can be easily interpreted to generate logical rules that are understandable to human. We believe that the interpretability of neural networks contributes to the improvement of the reliability and the usability of neural networks when applied critical real world problems.

1 Introduction

In the past decades, neural networks have been rigorously studied and applied in many fields. One of the most utilized models is Multilayered Perceptron (MLP) [1].The ability and flexibility of MLP to deal with vast kind of problems is the main reason for its unmatched success. Through the learning process, MLP is able to obtain knowledge to associate inputs and outputs, which is implicitly represented in the data set. However, in MLP this knowledge is represented as a set of connection weights values, which is not intuitively nor logically plausible (at least easily) for human. Hence, once trained, MLP is used as a kind of black box. Although, MLP is widely used for control, prediction, pattern recognition and so on, the lack of understanding in human side on the logical clarity on the decision making process inside MLP (and most of neural networks) is one of the drawback that hinders the usage of neural networks in more critical real world problems, for example problems that are crucial to human safety.

So far, several methods for extracting rules from a trained neural network were proposed [2,3,5]. The objective of most of these methods is to extract plausible rule from conventionally available neural networks, e.g. MLP. However, rule extractability is not considered in the design MLP, which naturally complicates the process of rule extraction. The nonlinearity of MLP complicates not only

M. Ishikawa et al. (Eds.): ICONIP 2007, Part II, LNCS 4985, pp. 434–443, 2008.
© Springer-Verlag Berlin Heidelberg 2008

the rule extraction process but sometimes also reduces the plausibility of the extracted rules.

The objective of our study is to propose a neural network model which structure and behavior significantly simplifies the rule extraction process without compromising the performance. The model is based on the previously proposed ensemble model [9]. As opposed to previously proposed ensemble models [6,7,8] whose objectives are to achieve better generalization performances compared to singular neural network models, our main objective is to build an ensemble model which behavior can be easily interpreted to generate rules that are logically comprehensible for human. Although we do not focus on the improvement of the generalization performance, the performance of the proposed ensemble is assured to be at least competitive to that of MLP.

The proposed ensemble is composed of several linear perceptrons (member hereafter). It is also equipped with a competitive training mechanism, which automatically and efficiently decomposes a given learning space into several learning sub-spaces and assigns a sub-space to a member that can deal with it best. Consequently, because each member is a perceptron that can only learn to form a linear function, the ensemble decomposes an arbitrary learning problem into several manageable linear problems, thus realizing a piecewise-linear classifier. The linearity of each member significantly lessens the complexity of rule extraction process, and the structure of the ensemble also contributes to the simplicity, thus plausibility of the extracted rules.

In the experiment the behavior of the proposed model is illustrated using an artificial logic problem, while the efficiency is tested on several benchmark problems.

2 Ensemble of Linear Experts

The proposed Ensemble of Linear Experts (ELE) is composed of several linear perceptrons. Each perceptron (member) has an additional neuron in its output layer (shown as a black circle in Fig.1) called confidence neuron(CN). CN is connected to the input neurons in the same way as the ordinary output neurons. The difference between CN and the ordinary output neuron is that, for a given input, CN generates a value that indicates the "confidence" of the member with regards to its ordinary output. A high confidence value is an indication that the output of the member is highly reliable while a low confidence value is an indication of the opposite.

In the running process, an input to the ensemble is processed independently by all members, so each of them produces a confidence value and an output. The ensemble then selects a winner, which is a member with the highest confidence value and adopts the output of the winner as the final output while disregarding other members' outputs. Based on the members' confidence the ensemble also executes a competitive training mechanism that will be elaborated in the latter part of this section.

2.1 Structure and Behavior of ELE

The structure of ELE is illustrated in Fig.1. It is composed of several independent linear perceptrons [10]. The activation of the ordinary output neurons is as follows.

$$
\begin{aligned}
O_k^i(t) &= f(I_k^i(t)) \\
I_k^i(t) &= \sum_{j=1}^{N_{in}} w_{jk}^i(t) x_j(t) + \theta_k^i(t) \\
f(x) &= \frac{1}{1+e^{-x}}
\end{aligned}
\tag{1}
$$

In Eq. 1, $O_k^i(t)$, $I_k^i(t)$ and $\theta_k^i(t)$ are the output, potential and the threshold of the k-th output neuron in the i-th member at time t, respectively. w_{jk}^i is the connection weight from the j-th input neuron leading to the k-th output neuron in the i-th member, while N_{in} and $x_j(t)$ are the number of the input neurons and the value of j-th input, respectively.

Similarly, the activation of the confidence neuron in the i-th member, $O_c^i(t)$ is as follows.

$$
\begin{aligned}
O_c^i(t) &= f(I_c^i(t)) \\
I_c^i(t) &= \sum_{j=1}^{N_{in}} v_j^i(t) x_j(t) + \theta_c^i(t)
\end{aligned}
\tag{2}
$$

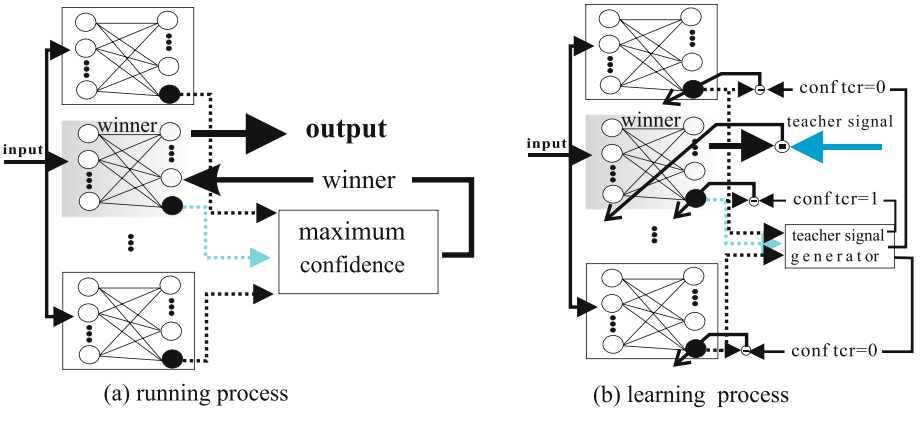


Fig. 1. Ensemble of Linear Experts

In Eq. 2, v_j^i and θ_c^i are the connection weights to from the j-th input neuron to the confidence neuron and the threshold of the confidence neuron in the i-th member, respectively.

The final output of ELE, $\mathbf{O}^{ens}$, given an input is formulated as follows, where the ensemble adopts the output of the winner and disregards others' outputs.

$$w = arg\max_i \{O_c^i(t)\} \tag{3}$$

$$\mathbf{O}^{ens}(t) = \mathbf{O}^w(t) \tag{4}$$

The running process of ELE is illustrated in Fig. 1 (a).

2.2 Competitive Learning of ELE

The competitive training of ELE is designed to enable the ensemble to decompose the learning space of a given problem into several sub-spaces and assign a sub-space to a member that is potentially the best to perform in that sub-space. Consequently, because each member is a linear perceptron, the ensemble behaves as a piecewise-linear classifier where a complex problem is efficiently decomposed into several more manageable linear sub-problems. The linearity of each of the member significantly simplifies the process for rule extraction.

In the training process, the ensemble chooses a winner in a similar manner as in the running process, and then calculates the performance of the winner, P^w as follows.

$$P^w(t) = 1 - E^w(t)$$

$$E^w = \frac{1}{N_{out}} \sum_{k=1}^{N_{out}} (O_k^w(t) - T_k(t))^2 \tag{5}$$

Where $T_k(t)$ is the teacher signal for the k-th output neuron at time t, and N_{out} is the number of the output neurons.

If the relative performance of the winner, R as shown in Eq.6 exceeds a threshold value, it is deemed to be potentially able to perform in the problem sub-space containing the given input, hence it is permitted to enhance its performance by applying Delta Rule to modify the connection weights leading to its ordinary output neurons as shown in Eq.7.

$$R(t) = \frac{P^w(t)}{\sum_{i=1}^{N} P^i(t)} \tag{6}$$

In Eq.6, N is the number of members.

$$\mathbf{W}^w(t+1) = \mathbf{W}^w(t) - \eta \frac{\partial E^{win}(t)}{\partial \mathbf{W}^w(t)} \tag{7}$$

In this equation, $\mathbf{W}^w$ is the weight vector of the winner and η is the learning rate.

In this case, consequently the confidence of the winner is enhanced by modifying the connection weight from input neurons to the confidence neuron, by setting the teacher for the confidence neuron, T_c, in Eq.8 as 1.

$$\mathbf{V}^w(t+1) = \mathbf{V}^w(t) - \eta \frac{\partial E^w(t)}{\partial \mathbf{V}^w(t)}$$
$$E_c^w(t) = (O^w - T_c)^2 \tag{8}$$

Furthermore, because the winner should dominate the rest of the members with regards to the given input, other members should suppress their confidence values by applying Eq. 8, by setting the teacher signal T_c to 0.

Oppositely, when the performance of the winner is below the threshold value, it is indication of the inability of the winner to perform, hence the winner should surrender the domination to other members. This is done by decreasing the confidence of the winner by setting the teacher signal for the confidence neuron of the winner to 0 and increasing the confidence values of the rest of the members by setting 1 as the teacher signals for their confidence neurons. Because, the confidence value and the actual performance have to be synchronized, in this case the losers are permitted to modify their weights leading to the ordinary output neurons according to Eq. 7.

The outline of the learning process is shown Fig.1(b) and Algorithm 1.

Algorithm 1. Competitive Learning Process of ELE

```
select a training example
run all members
select a winner
if performance(winner) ≥ threshold then
    train(winner)
    increase-confidence(winner)
    decrease-confidence(losers)
else
    decrease-confidence(winner)
    increase-confidence(losers)
    train(losers)
end if
```

The competitive learning process ensures the diversity of the members and at the same time guaranty the harmony between the confidence value and the actual performance of each member.

2.3 Rule Extraction from ELE

Because the activation of an output and a confidence neuron is sigmoidal and the neurons are trained to produce parity value of 0 or 1, we can assume that the

following intermediate rule is true (by setting a very large slope for the sigmoid function).

$$I_c^i(t) = \sum_{j=1}^{N_{in}} v_j^i(t)x_j(t) + \theta_c^i(t) > 0 \Rightarrow i : winner \tag{9}$$

Simillarly, when the proposed ensemble is applied to 1-of-M classification problems, the ordinary output neurons are also trained to produce 0 or 1, hence the following intermediate rules are also true.

$$I_k^i(t) = \sum_{j=1}^{N_{in}} w_{jk}^i(t)x_j(t) + \theta_k^i(t) > 0 \Rightarrow O_k^i(t) = 1 \tag{10}$$

$$I_k^i(t) = \sum_{j=1}^{N_{in}} w_{jk}^i(t)x_j(t) + \theta_k^i(t) < 0 \Rightarrow O_k^i(t) = 0 \tag{11}$$

From these intermediate rules we can easily generate plausible $if-then$ rules by applying any of rule extraction algorithm proposed in [2,3,4]. However, for simplicity we apply a simple rule extraction method explained in [2], where the range of inputs is divided into three parts based on their values, namely small(s), medium(m) and large(l), which are then quantized to 0, 0.5 and 1, respectively, and adopts logical propositions that satisfy Equations 9, 10, 11 as the rules.

It is obvious that the each of the member represents rules that are valid in a particular sub-problem space (in which the member has the highest confidence), and the winner-takes-all selection based on the members' confidences acts as a kind of "meta rule", which is a rule to select a rule, because the selection winner selection mechanism can be translated into the following rule.

Algorithm 2. Meta Rule

```
if winner = i then
  apply rule i
end if
```

The rule expression of ELE increases the plausibility of the general rule that governs the learning space. Because instead of a single complicated rule set it offers more understandable several partial rules that we consider helpful for human in understanding the knowledge of a neural network. The high plausibility of the rule expression is possible because of the structure and the competitive training algorithm of ELE.

3 Experiments

To illustrate the characteristics, we apply ELE to XOR problem, which is a non-linear classification problem that naturally cannot be dealt with any linear

classifier. Figure 2(a) shows the hyperspace of ELE with two members trained on this problem, in which areas that are classified as 1 are shown in black, areas that are classified as 0 are shown in white, and gray is for areas that are ambiguously classified in the vicinity of 0.5. For comparison Fig.2(b) shows the typical hyperspace of MLP. Figures 2 (c) and (d) show the hyperspace of member-1 and member-2 of ELE, respectively, where "low conf" indicates areas where the confidence of a member is lower than that of its counterpart. It is obvious that ELE decomposes this non-linear classification problem into two linear sub-problems and assigns each sub-problem to one of the member. After the learning process, the potentials of the confidence neurons of the members are as follows.

$$\begin{aligned} I_c^1 &= -5.2x_1 + 0.3x_2 + 2.5 \\ I_c^2 &= 5.3x_1 - 0.2x_2 - 2.2 \end{aligned} \tag{12}$$

From Eq. 12 it is clear that whenever $x_1 < medium$ then rule generated by member 1 is applied and rule generated by member 2 is otherwise applied.

Similarly, the potential of the output neurons of the members are as follows.

$$\begin{aligned} I_1^1 &= -1.2x_1 + 4.9x_2 - 2.3 \\ I_1^2 &= 0.3x_1 + -5.0x_2 + 2.2 \end{aligned} \tag{13}$$

From Eqs.12 and 13 the following rule can be extracted.

Algorithm 3. Extracted Rule: XOR

```
if x1 < medium then
   Apply Rule 1:
   if x2 > medium then
      classify as 1
   else
      classify as 0
   end if
else
   Apply Rule 2:
   if x2 < medium then
      classify as 1
   else
      classify as 0
   end if
end if
```

To test the efficiency of ELE, we apply ELE to several benchmark problems from UCI Repository [11]. The average generalization accuracies over 50 runs for each problem are listed in Table. 1. For comparison we also list the performances of MLP and Linear Perceptron. In every run, the number of learning

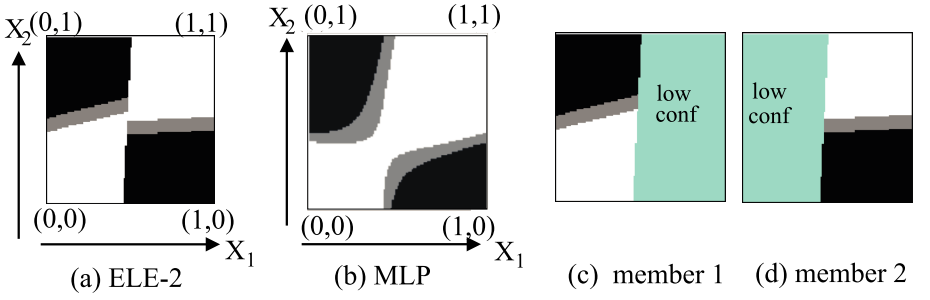


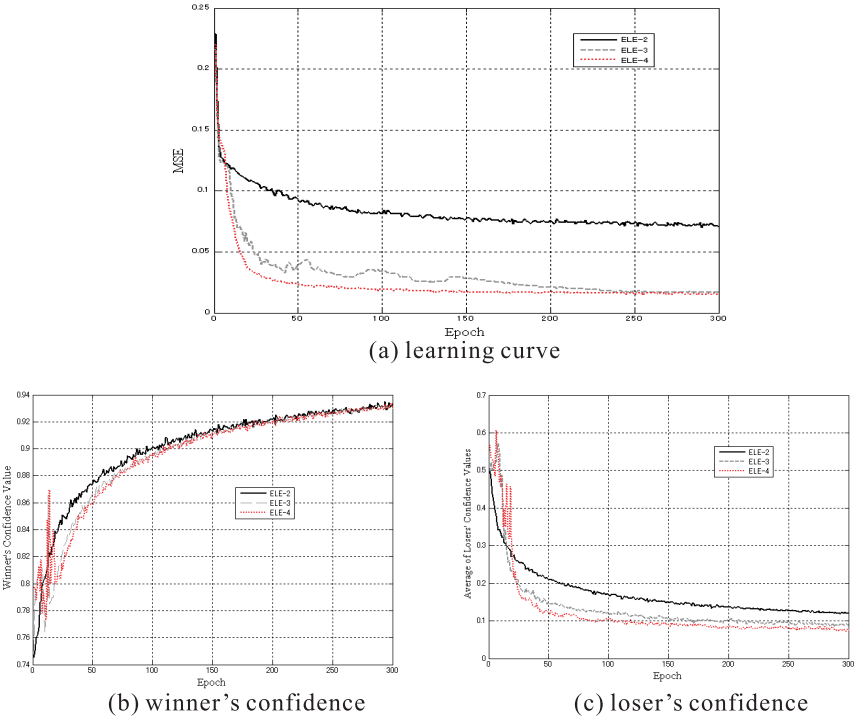
Fig. 2. Hyperspace (XOR)

Table 1. Generalization Accuracy (%)

	Perceptron	MLP	**ELE**
iris	72	100	**100**
cancer	97	97	**97**
liver	61	69	**70**
pima	75	76	**79**
balance	86	88	**88**
wine	90	97	**94**
ionos	91	94	**92**

iterations for the every classifier is the same. From Table 1, we can confirm that the performance of ELE over wide range of problems are competitive to the performance of MLP. In these experiments, the number of members in ELE is varied between 2 and 5, but we find that the difference in performance between ELEs with different number of members are not significant, because usually ELE is able to utilize a minimum number of members to deal with a given problem. The performance accuracies of ELE in Table 1 is the performance of the best ELE. For all the experiments, the learning rate η is set to 0.5, while the performance threshold, R is set to $\frac{1}{N}$, where N is the number of members.

To illustrate the characteristics of ELE, the learning process with regards to Iris Classification [12] problem is used as an example. This problem is a well known non-linear classification problem, where a four dimensional input (length and width of petal and sepal of an iris flower) has to be classified into one of the three classes of iris flower (setosa, versicolor and virginica). Figure 3(a) shows the learning curve of ELEs with two, three and four members, which clearly indicates that ELE can deal nicely with this non-linear problem. Figure 3(b) show the confidence of the winner during the training epoch. From Figures 3(a) and (b) we can draw a conclusion that the actual performance and the confidence of the winner are gradually synchronized by observing the fact that the decrease in the training error is always associated with the increase in the winner's confidence. Figure 3(c) shows the average of the losers' confidence. Figures 3(b) and (c) show that the increase of the winner's confidence is always associated with the decrease



(a) learning curve

(b) winner's confidence

(c) loser's confidence

Fig. 3. Learning Characteristics (Iris)

in the losers' confidences, which indicates that ELE diversifies the expertise of its members over the progress of the learning process. For this problem, ELE is able to choose two of its members to perform the classification. If ELE has more than two members, then the rest of the members have very low confidences in the whole problem space, thus they do not contribute to the classification process. From the two members the following rules can be extracted.

Algorithm 4. Extracted Rule: Iris Classification

if $x_3 : large \vee x_4 : large$ **then**
  Apply **Rule 2**:
  Classify as **Virginica**
else
  Apply **Rule** 1:
  if $x_3 : small \wedge x_4 : small$ **then**
    Classify as **Setosa**
  else
    Classify as **Versicolor**
  end if
end if

4 Conclusions

In this paper we propose a new of neural network ensemble model whose structure and learning algorithm support the extraction of plausible rules. The experiments confirm that the proposed ensemble acts as a piecewise linear classifier with a competitive accuracy compared with MLP and the generated rules are easily plausible for human. A thorough mathematical analysis of the behavior of ELE is one of the future plans of this research.

References

1. Rumelhart, D., McClelland, J.: Learning Internal Representation by Error Propagation. Parallel Distributed Processing I, 318–362 (1984)
2. Duch, W., Setiono, R., Zurada, J.: Computational Intelligence Methods for Rule-Based Data Understanding. Proceedings of The IEEE 92(5), 771–805 (2004)
3. Taha, A., Ghosh, J.: Symbolic Interpretation of Artificial Neural Networks. IEEE Trans. Knowledge and Data Engineering 11(3), 448–462 (1999)
4. Setiono, R.: Extracting M-of-N Rules from Trained Neural Networks. IEEE Trans. Neural Networks 11(2), 512–519 (2000)
5. Benitez, J.M., Castro, J.L., Requena, I.: Are Artificial Neural Networks Black Boxes? IEEE Trans. on Neural Networks 8(3), 1156–1164 (1997)
6. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive Mixture of Local Experts. Neural Computation 3, 79–87 (1991)
7. Freund, Y.: Boosting a Weak Learning Algorithm by Majority. Information and Computation 7 II, 256–285 (1995)
8. Hartono, P., Hashimoto, S.: Learning from Imperfect Data Applied Soft Computing Journal 7(1), 353–363 (2007)
9. Hartono, P., Hashimoto, S.: Analysis on the Performance of Ensemble of Perceptron. In: Proc. IJCNN 2006, pp. 10627–10632 (2006)
10. Widrow, B.: 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. Proceedings of IEEE 78(9), 1415–1441 (1990)
11. UCI Machine Learning Repository: `http://www.ics.uci.edu/~mlearn/MLRepository.html`
12. Fisher, R.: The Use of Multiple Measurements in Taxonomic Problems. Annual Eugenics 7(II), 179–188 (1936)