A Theoretical Interpretation for Layered Neural Network Classifier

階層型ニューラルネットワーク分類モデルの理論的意味解釈

Eihan SHIMIZU*

Abstract: Layered feed-forward neural networks (LNNs) have been broadly applied to classification, prediction and other modeling problems. There have been so far, however, few studies that have provided a theoretical interpretation for the application of LNN. Most of the conventional studies have been empirical and the LNNs have been applied just like “black box” machines. This paper discusses the application of LNN to image or remotely sensed data classification. It provides a theoretical interpretation for the LNN classifier in comparison with the conventional classification or discriminant methods. The most distinguished part is the derivation of a generalized form of LNN classifier based on the maximum entropy principle. According to the generalized form, this paper discusses the relationship between the familiar type of LNN classifier employing the sigmoidal activation function and the other types of discriminant models such as the Multinomial Logit Model.

1. Introduction

Layered feed-forward neural networks (LNN) have been broadly applied into prediction, simulation, classification, pattern recognition and other modeling problems. There have been so far, however, few studies that have provided a theoretical interpretation for the application of LNN except for comparisons with regression analysis (Hill et al., 1994). Most of the conventional studies have been empirical and LNNs have been applied just like “black box” estimation machines.

This paper discusses the applications of the LNN to classification and pattern recognition problems which have been often attempted in the fields of remote sensing and digital image analysis. It provides a theoretical interpretation for the LNN based classifier mainly in comparison with Bayesian classifier and Multinomial Logit Model.

2. Basic Formulation of LNN Classifier

Let \( \mathbf{x} \) represent a feature vector which is to be classified. Let the possible classes be denoted by \( \omega_f \), \( f = 1, 2, \ldots, J \). Consider the discriminant function \( d_f(\mathbf{x}) \), then decision rule is
\[ \mathbf{x} \in \omega_j \text{ if } d_j(\mathbf{x}) \geq d_i(\mathbf{x}) \text{ for all } j \neq i. \] (1)

A LNN is expected to be the I/O system which is similar to the discriminant function.

Let us make use of a typical LNN architecture which has been applied to a variety of classification problems. A feature vector is input to the input layer. In other words, the number of the neurons in the input layer is corresponding to the dimension of the feature vectors. The output layer has the number of neurons as same as the classes. The output signal from the \( j \)th neuron in the output layer is regarded as the discriminant value. Let the state of the \( j \)th output neuron be represented by

\[ u_j = g(\mathbf{x}, \mathbf{w}) \] (2)

where \( \mathbf{w} \) is the parameter vector included in the designed LNN. These parameters are mainly constituted by the connection weights (synaptic weights) between neurons. We are not concerned here with the formulation of \( g(\mathbf{x}, \mathbf{w}) \). The output of LNN, \( p_j(\mathbf{x}, \mathbf{w}) \), under presentation of \( \mathbf{x} \) is

\[ p_j(\mathbf{x}, \mathbf{w}) = f(u_j), \] (3)

where \( f(u_j) \) is the activation function. The following sigmoid function, which is a bounded, monotonic and non-increasing, is frequently used,

\[ f(u_j) = \frac{1}{1 + \exp(-u_j)} \] (4)

The feature vectors \( \mathbf{x}_k (k=1,2,\ldots,K) \) for training the LNN are prepared. The classes which these feature vectors belong to are all known. Training data (target data) are given as follows;

\[ d_j(\mathbf{x}_k) = \begin{cases} 1 \text{ if } \mathbf{x}_k \in \omega_j \\ 0 \text{ otherwise} \end{cases} \] (5)

The LNN is trained by minimizing a mean squared error, that is,

\[ \min \sum_{j=1}^{J} \sum_{k=1}^{K} [p_j(\mathbf{x}_k, \mathbf{w}) - d_j(\mathbf{x}_k)]^2 \] (6)

Training of the LNN is performed through the adjustment of connection weights. The most common method is so-called "back propagation" which is gradient descent in essence. After the completion of training, the LNN plays a role of the discriminant function. Let the output of trained LNN be denoted by \( p_j(\mathbf{x}, \mathbf{w}) \).

### 3. Interpretation for LNN Classifier

#### 3.1 Relationship between LNN and Bayesian classifier

The Bayesian optimal decision rule, in the sense of minimizing the probability of classification error, is to choose the class which maximizes the posterior probability;

\[ p(\omega_j | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_j) p(\omega_j)}{p(\mathbf{x})} \] (7)

If the prior probabilities \( p(\omega_j) \) are equal, then the conditional probability density function \( p(\mathbf{x} | \omega_j) \) corresponds to the optimal discriminant function. Maximum likelihood classifier is frequently applied under the condition of a multivariate normal distribution.

How the LNN classifier is related to the Bayesian optimal classifier? This question has been already discussed by Wan (1990) and Ruck et al. (1990). Its conclusion is that the output of the LNN, \( p_j(\mathbf{x}, \mathbf{w}) \), when trained by the criteria (6), approximates the Bayesian posterior probability. According to Wan (1990), we show a short proof. Consider the training data given in the form (5). Suppose that the training data are random variables and samples from the probability density function \( p(\mathbf{x}, d_j(\mathbf{x})) \), where

\[ d_j(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \in \omega_j \\ 0 \text{ otherwise} \end{cases} \] (8)

Since \( p_j(\mathbf{x}, \mathbf{w}) \) is the least squares estimate of \( d_j(\mathbf{x}) \), it is the conditional expectation of \( d_j(\mathbf{x}) \) given \( \mathbf{x} \). Therefore,
\[ p_j(x, w) = \mathbb{E}[d_j(x)|x] = \sum_{d_j(x) \in \{0,1\}} d_j(x) \cdot p(d_j(x)|x) = p(d_j(x) = 1|x) = p(w_j|x). \] (9)

This means that \( p_j(x, w) \), in the sense of minimizing a mean squared error, approximates the posterior probability \( p(w_j|x) \). This provides a theoretical interpretation for the application of the LNN into classification problems. It is proved that a three-layered neural network, when the appropriate number of neurons are set in the hidden layer and sigmoidal activation functions are used in the hidden layer, can approximate any continuous mapping (e.g. Gallant et al., 1988; Funahashi, 1989; Cybenko, 1989; Hornik et al., 1989). It is expected that LNN approximates very accurately the posterior probability.

Up to this point, however, the derivations have been for an arbitrary mapping trained by \( d_j(x_k) \in \{0,1\} \). Its result has been well-known in the field of statistics (Wan, 1990). The above proof is a theoretical justification for any non-parametric discriminant function trained by the least squares criteria. The following section will discuss the interpretation for the activation functions which are identical to the LNN.

3.2 Interpretation for activation functions

Let the activation function, \( f(u_j) \), be a monotonic increasing function. Then, the state of the output neuron, \( u_j \), and the posterior probability, \( p(w_j|x) \), have a one-to-one mapping, and consequently, \( u_j = g(x, w) \) becomes also an optimal discriminant function.

The activation function should be a probability distribution given a certain level of state. This is analogous to the probability distribution of a particle being in a certain state given the energy level of each state in the statistical mechanics. In the statistical mechanics different probability distributions are derived from so-called maximum entropy principle. Let us derive the activation forms from maximum entropy principle.

Consider the maximization of Kapur’s generalized measure of entropy under the expected discriminant value (Kapur, 1986):

\[ \max.H(p) = \sum_{j=1}^{J} p_j \cdot \log p_j + \frac{1}{a} \sum_{j=1}^{J} (1+a p_j) \cdot \log(1+a p_j), \quad a \geq 1 \quad (10) \]

subject to \( \sum_{j} p_j = 1 \) (11)

where \( H(p) \) is the Kapur’s generalized entropy in which the constant term is omitted, \( p_j(j=1,2,\ldots,J) \) is a probability distribution corresponding to \( p_j(x, w) \), \( a \) is a parameter prescribing the type of entropy, that is, the type of probability distribution, and \( U \) is an expected discriminant value. Here, we do not explicitly give the constraint;

\[ \sum_{j} p_j = 1 \]

to the maximization problem, because \( p_j \) approximates the posterior probability.

From (10) and (11), we get

\[ p_j = \frac{1}{-a + \exp(-\beta u_j)} \]

where \( \beta \) is a Lagrange multiplier associated with (11). The parameter \( \beta \) is the so-called temperature parameter by which the slope of the activation function. When \( a \) is fixed and the LNN with the activation function (13) is trained, \( \beta \) is estimated being included in the connection weights in a training process, since \( u_j \) is generally defined by the linear function of the connection weights between the output neuron concerned and the hidden neuron.

Now, assume that \( \beta \) is a constant. Then, the probability distribution (13) is equivalent to an optimal solution of the following maximization (Brotchie, 1979):

\[ \max.U = \sum_{j=1}^{J} p_j \cdot u_j + \frac{1}{\beta} H(p). \]

Therefore, the activation function form (13) is interpreted as the representation of the expected discriminant value maximization taking into account
the uncertainty shown as the Kapur’s entropy.

Now, let us return to the activation function form (13) and discuss the meaning of the parameter. For \( a = -1 \), (13) gives:

\[
p_j = \frac{1}{1 + \exp(-\beta u_j)}
\]  

(15)

This is truly the sigmoid function (i.e., (4)) most frequently used in the applications of LNNs. In addition, if \( a = -1 \), it is well-known that (10) subject to (11) and (12) gives Fermi-Dirac (F-D) distribution. Note that \( p_j(x, w) \) approximates the posterior probability; thus the familiar sigmoid function is interpreted as the representation of the expected discriminant value maximization under the F-D type entropy.

Similarly, for \( a = 1 \), (13) is

\[
p_j = \frac{1}{-1 + \exp(-\beta u_j)}
\]  

(16)

It is known that, for \( a = 1 \), (10) subject to (11) and (12) gives Bose-Einstein (B-E) distribution. Consequently, (16) approximates the B-E distribution.

Next, consider the case of \( a = 0 \), that is,

\[
p_j = \frac{1}{\exp(-\beta u_j)}
\]  

(17)

As the parameter \( a \) tends to zero, (10) approaches Shannon’s measure of entropy. It is well-known that the maximization of the Shannon’s entropy subject to (11) and (12) gives Maxwell-Boltzmann (M-B) probability distribution;

\[
p_j = \frac{\exp(\beta u_j)}{\sum_j \exp(\beta u_j)}
\]  

(18)

Accordingly, (17) approximates the M-B distribution. In addition, (18) gives the structural similarity with so-called Multinomial Logit Model which is familiar in the field of the discrete choice behavioral modeling (Anas, 1983). Hence the LNN classifier with the activation function (17) is interpreted as the approximate of the Multinomial Logit Model.

As mentioned above, the choice of \( a = -1, 0 \) and 1 leads to Fermi-Dirac (F-D), Maxwell-Boltzmann (M-B), and Bose-Einstein (B-E) probability distributions respectively in statistical mechanics. Let us compare the characteristics of the above representative distributions in statistical mechanics. These three distributions are all derived from Jaynes’s maximum entropy principle (Kapur, 1992). One distribution differs from another due to the constraints to Shannon’s measure of entropy. In the M-B distribution, the expected energy of a particle in the system is only prescribed. The F-D and B-E distributions are derived from the constraints with respects to the expected energy of the system and the expected number of the particles in the system. While in the F-D distribution the maximum number of the particles allowed in a certain state is assumed to be one, in the B-E distribution the maximum number is assumed to be infinite.

Thus, the parameter \( a \) is associated with the constraints to the maximization of the Shannon’s entropy. This gives us an implication that, for lying between -1 and 1, we can get the various types of probability distributions, though it may be difficult to provide the significant interpretation for the distributions in the framework of the statistical mechanics. We have a choice of infinite types of models corresponding to different values of \( a \). A possible method is to choose the parameter to get the best fit to the training data. Regardless as the selected parameter, we can provide the interpretation to the activation function as the representation of the expected discriminant value maximization under the Kapur’s generalized entropy.

4. Conclusion

This paper has provided an interpretation for the LNN classifier. The output of the LNN under the completion of training approximates the Bayesian posterior probability. Therefore, if we assume the activation function of the output neuron to be monotonic increasing, the state of the output neuron
is also Bayesian optimal discriminant function. From the maximum entropy principle, we can provide the interpretation for the activation function. The familiar sigmoid function is approximate to the Fermi-Dirac distribution. The LNN classifier using the activation function of the Maxwell-Boltzmann distribution approximates the Multinomial Logit Model. The maximization of Kapur’s generalized measure of entropy gives the generalized form of the probability distributions including the Maxwell-Boltzmann, Fermi-Dirac, and Bose-Einstein distributions. In the practical sense, it is proposed to apply the Kapur’s generalized distribution into the generalized activation function and to fix the function form in the process of training. Regardless as the selected function form, we can provide an interpretation for that as the representation of the maximization of the expected discriminant value under the Kapur’s generalized entropy.

(受付日1996.2.22，受理日1996.7.2)

REFERENCES


