Neural Network Approach for Multiple-Class Classification

To cite this article: Dengyi Wu et al 2019 J. Phys.: Conf. Ser. 1302 032045

View the article online for updates and enhancements.

You may also like
- Optical design and extended multi-objective optimization of miniature L-type optics
  Jung-Hung Sun, Bo-Ren Hsueh, Yi-Chin Fang et al.
- The Combined Multi-objective Optimization Design for a Light Guide Rod
  Yu-Sen Yang, Chun-Yao Shih, Hong-Yao Chen et al.
- Magnetite nanoparticle interactions with insulin amyloid fibrils
  Yun-Wen Chen, Chiung-Wen Chang, Huey-Shan Hung et al.
Neural Network Approach for Multiple-Class Classification

Dengyi Wu¹, Chingsheng Tu¹, Shiejue Lee², Chihhung Wu³

¹Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan.
²Department of Electrical Engineering, Intelligent Electronic Commerce Research Center, National Sun Yat-Sen University, Kaohsiung, Taiwan.
³Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung, Taiwan.

Abstract. Classification is a very important problem-solving technique. Given a set of attribute values about an object, a classifier determines which category the object belongs to. A classification task can be a binary or multiple classification; it can also be a single-label or multiple-label classification. Neural networks are an effective technique in the field of artificial intelligence. In this paper, we develop different schemes of neural networks for solving binary, multiple, single-label, and multiple-label classification problems. The ideas behind them are described in detail.

1. Introduction

Classification is one of the major tasks attended to in the artificial intelligence field [1], [2], [3], [4]. In daily life, we encounter classification problems from time to time, and here and there. When we see a person, we’d like to know if the person is a male or a female. A policeman needs to know whether a captured man is a criminal or not. A bank wants to determine if an applicant is permitted for having a credit card based on the information provided in the application form.

A classification problem can be a binary classification if the resulting category is one of two categories. For instance, determining a person’s gender is a binary classification problem. The result can only be male or female, and there are only two categories involved [5], [6]. A classification problem can be a multiple classification if the resulting category is one of two or more categories. Determining which disease a patient may get is such a case. There are quite a few diseases, e.g., flu, toothache, fever, etc., a person can get. If the number of categories involved in multiple classification is restricted to two, the task is reduced to binary classification. In this regard, binary classification is a special case of multiple classification.

On the other hand, a classification problem can be a single-label classification if the object being considered can belong to only one category. Determining the permission of issuing a credit card to an applicant is a single-label problem. Recognizing the identity of a person is also a single-label problem. However, a patient may have one or more diseases at the same time. Therefore, based on the symptoms of a patient, a doctor may conclude that the patient is suffering from two or more diseases. This shows that it is a multiple-label classification problem [9]. Clearly, single-label classification is a special case of multiple-label classification.

Neural networks have been proposed and are widely used for solving classification problems [7], [8]. A neural network has an input layer, an output layer, and a certain number of hidden layers although one hidden layer is adequate in most cases. Binary classification can be solved by a neural network with only one output node in the output layer. Multiple classification can be solved based on a
set of binary classifiers, but can also be solved with a network on its own. A similar relationship also exists between single-label classification and multiple-label classification.

In this paper, we develop different schemes of neural networks for implementing a classifier and describe how they work in detail. Section 2 states the problem to be solved. Section 3 describes the proposed neural network schemes. Section 4 illustrates how our ideas work. Finally, Section 5 gives a conclusion.

2. Problem Statement
The classification problem we are considering in this paper is concerned with a training set of \( N \) labeled instances, \( \{x_1, y_1\}, \{x_2, y_2\}, ..., \{x_N, y_N\} \), where

- \( x_i = (x_{1,i}, x_{2,i}, ..., x_{n,i}) \), \( 1 \leq i \leq N \), is the input vector of the \( i \)th instance, containing \( n \) attribute values, with \( x_{j,i} \) being a real number, \( 1 \leq j \leq n \), and
- \( y_i = (y_{1,i}, y_{2,i}, ..., y_{m,i}) \), \( 1 \leq i \leq M \), is the target vector of the \( i \)th instance, indicating the categories the \( i \)th instance belongs to, with \( y_{j,i} = +1 \) if the instance belonging to category \( j \) and \( y_{j,i} = -1 \) if the instance not belonging to category \( j \), \( 1 \leq j \leq m \).

Note that \( N \) is the number of training instances, \( n \) is the number of attributes involved in each instance, and \( m \) is the number of categories an instance can belong to. Note also that \( (a_1, a_2, ..., a_k) \) indicates the \( k \)-vector \( [a_1, a_2, ..., a_k]^T \).

For the problem, if \( m = 2 \), it is a binary classification problem, and it is a multiple classification problem if \( m \geq 2 \). On the other hand, if only one entry in any target vector is plus one, it is a single-label classification problem, and it is a multiple-label problem if two or more entries in any target vector are plus one. Our goal is to develop a neural network from the training set, so that given an input vector the network can decide the classifications of the input vector.

3. Neural Networks For Classification
In this section, we present neural network schemes for different types of classification. First of all, we give a basis structure as shown in Figure 1(a).

The input layer has \( n \) nodes. The output layer has \( m \) nodes. The hidden layer is determined by applying a self-constructing clustering algorithm [10] to the given training dataset. After clustering, \( J \) clusters are obtained, each having center \( c_i = (c_{1,i}, c_{2,i}, ..., c_{n,i}) \) and deviation \( v_i = (v_{1,i}, v_{2,i}, ..., v_{n,i}) \), \( 1 \leq i \leq J \). Then we have \( J \) hidden nodes in the hidden layer, and the output of the \( i \)th hidden node is

\[
o_i^1(p) = \prod_{j=1}^{n} \exp \left[ -\left( \frac{p_j^c - c_{1,i}}{v_{j,i}} \right)^2 \right]
\]

Where \( p = (p_1, p_2, ..., p_n) \) is any input vector. The output of the output node \( k \), \( 1 \leq k \leq m \), is

\[
o_i^2(p) = \sum_{j=1}^{n} w_{0,k} o_i^1(p) + w_{1,k} o_i^1(p) + ... + w_{J,k} o_i^1(p)
\]

Where \( w_k = (w_{0,k}, w_{1,k}, ..., w_{J,k}) \) is the weight vector between the hidden layer and the \( k \)th output node.

The weight vector is derived as follows. For training instance \( x_i, 1 \leq i \leq N \), we let

\[
y_{1,i} = w_{0,1} + (w_{1,1} \cdot w_{j,1}) \cdot o_i^1(x_i), \ldots, o_i^1(x_i))
\]

\[
y_{2,i} = w_{0,2} + (w_{1,2} \cdot w_{j,2}) \cdot o_i^1(x_i), \ldots, o_i^1(x_i))
\]

\[
\vdots \ \
\vdots \ \
\vdots \ \
\vdots \ 
\]

\[
y_{m,i} = w_{0,m} + (w_{1,m} \cdot w_{j,m}) \cdot o_i^1(x_i), \ldots, o_i^1(x_i))
\]

In this way, we have \( mN \) equations. Then the least squares method [11] is applied to find an optimal solution which are desired weight vectors \( w_1, \ldots, w_m \).

Figure 1(b) is the simplified diagram of Figure 1(a).
3.1. Multiple-Label Classification
We add the transfer function hardlims(x) in each output node of the output layer, as shown in Figure 2(a):

\[
\text{hardlims}(x) = \begin{cases} 
  +1, & \text{if } x \geq 0 \\
  -1, & \text{if } x < 0 
\end{cases} \quad (7)
\]

Which is the symmetrical hard limit function. For an input vector \( p \), the network output \( \hat{y}_i \) is
\[
\hat{y}_i = \text{hardlims}(o_{i}^{2}(p)) \quad (8)
\]

For \( 1 \leq i \leq m \). The input vector \( p \) is predicted belonging to class \( j \) if \( \hat{y}_j \) is +1. In this way, \( p \) can belong to two or more categories if two or more network outputs are +1.

3.2. Single-Label Classification
We add the transfer function compet(x) for all the output nodes of the output layer, as shown in Figure 2(b):

\[
\text{compet}(x) = \begin{cases} 
  +1, & \text{for the neuron with max input} \\
  -1, & \text{for all other neurons} 
\end{cases} \quad (9)
\]

Which is the competitive function. The result is that only one output node will have +1 and the others have -1. For an input vector \( p \), the network output \( \hat{y}_i \) is +1 if \( o_{i}^{2}(p) \) is the maximum of the peer values, and \( \hat{y}_i \) is -1 otherwise. In this way, \( p \) can belong to only one category.

4. Illustration
An example is given here to illustrate how our ideas work. Suppose we have a single-label application with 3 categories, having 12 training instances:
\[
\mathbf{x}_1 = (0.30, 0.60), \quad \mathbf{y}_1 = (+1, -1, -1);
\]
\[ \mathbf{x}_2 = (0.70, 0.35), \quad \mathbf{y}_2 = (-1, +1, -1); \]
\[ \mathbf{x}_3 = (0.50, 0.52), \quad \mathbf{y}_3 = (+1, -1, -1); \]
\[ \mathbf{x}_4 = (0.35, 0.38), \quad \mathbf{y}_4 = (+1, -1, -1); \]
\[ \mathbf{x}_5 = (0.19, 0.89), \quad \mathbf{y}_5 = (-1, -1, +1); \]
\[ \mathbf{x}_6 = (0.78, 0.20), \quad \mathbf{y}_6 = (-1, +1, -1); \]
\[ \mathbf{x}_7 = (0.62, 0.25), \quad \mathbf{y}_7 = (-1, +1, -1); \]
\[ \mathbf{x}_8 = (0.24, 0.81), \quad \mathbf{y}_8 = (-1, -1, +1); \]
\[ \mathbf{x}_9 = (0.29, 0.89), \quad \mathbf{y}_9 = (-1, -1, +1); \]
\[ \mathbf{x}_{10} = (0.40, 0.65), \quad \mathbf{y}_{10} = (+1, -1, -1); \]
\[ \mathbf{x}_{11} = (0.28, 0.48), \quad \mathbf{y}_{11} = (+1, -1, -1); \]
\[ \mathbf{x}_{12} = (0.24, 0.89), \quad \mathbf{y}_{12} = (-1, -1, +1). \]

Note that \( n = 2 \) and \( m = 3 \). Figure 3(a) shows these instances as circles. After clustering, three
clusters are obtained, as shown in Figure 3(b), with
\[ \mathbf{c}_1 = (0.366, 0.526), \quad \mathbf{v}_1 = (0.0882, 0.1053); \]
\[ \mathbf{c}_2 = (0.70, 0.2667), \quad \mathbf{v}_2 = (0.08, 0.0764); \]
\[ \mathbf{c}_3 = (0.24, 0.87), \quad \mathbf{v}_3 = (0.0408, 0.04). \]

Note that \( \mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_{10}, \) and \( \mathbf{x}_{11} \) belong to the first cluster,
\( \mathbf{x}_2, \mathbf{x}_6, \) and \( \mathbf{x}_7 \) belong to the second cluster, and \( \mathbf{x}_5, \mathbf{x}_8, \mathbf{x}_9, \) and \( \mathbf{x}_{12} \) belong to the third cluster.

Based on the obtained clusters, the neural network is constructed as shown in Figure 4. Note that
we have

![Figure 3](image1.png)

**Figure 3.** (a) Training instances. (b) Training instances.

![Figure 4](image2.png)

**Figure 4.** Single-label classification.

\( \mathbf{x}_2, \mathbf{x}_6, \) and \( \mathbf{x}_7 \) belong to the second cluster, and \( \mathbf{x}_5, \mathbf{x}_8, \mathbf{x}_9, \) and \( \mathbf{x}_{12} \) belong to the third cluster.

Based on the obtained clusters, the neural network is constructed as shown in Figure 4. Note that
\[ o_1^1(p) = \prod_{j=1}^{p} \exp \left( -\frac{(p_{j,c,1})^2}{v_{j,1}} \right), \]
\[ o_2^2(p) = \prod_{j=1}^{p} \exp \left( -\frac{(p_{j,c,2})^2}{v_{j,2}} \right), \]
\[ o_3^3(p) = \prod_{j=1}^{p} \exp \left( -\frac{(p_{j,c,3})^2}{v_{j,3}} \right) \]

And
\[ o_1^2(p) = w_{0,1} + \sum_{j=1}^{3} w_{j,1} o_j^1(p), \]
\[ o_2^2(p) = w_{0,2} + \sum_{j=1}^{3} w_{j,2} o_j^1(p), \]
\[ o_3^2(p) = w_{0,3} + \sum_{j=1}^{3} w_{j,3} o_j^1(p) \]

And the transfer function compet is added to the output layer to ensure that only one output node has +1 and the others have -1.

5. Conclusion
Neural networks are an effective technique for solving classification problems. We have developed different schemes of neural networks for solving binary, multiple, single-label, and multiple-label classification problems. The ideas behind them have been described in detail and an illustration has been given.

6. Acknowledgments
This paper was financially supported in part by the NSYSU-NUK Joint Research Project #NSYSUNUK-107-P006, by the “Intelligent Electronic Commerce Research Center”, and by grant B106002, STSP AI Robot Project, MOST, TAIWAN.

7. References
[1] D. W. Aha. Lazy learning: Special issue editorial. Artificial Intelligence Review, 11(1–5):7–10, 1997.
[2] M. N. Murty and V. S. Devi. Pattern Recognition: An Algorithmic Approach. Springer, 2011.
[3] D. Zissis. A cloud based architecture capable of perceiving and predicting multiple vessel behaviour. Applied Soft Computing, 35:652-661, 2015.
[4] V. K. Ojha, A. Abraham, and V. Snášel. Metaheuristic design of feedforward neural networks: A review of two decades of research. Engineering Applications of Artificial Intelligence. 60:97-116, 2017.
[5] K. Fukunaga. Introduction to Statistical Pattern Recognition. Academic Press, San Diego, CA, 1990.
[6] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2nd ed. Wiley & Sons, New York, NY, 2001.
[7] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK, 1995.
[8] B. D. Ripley. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge, UK, 1996.
[9] S.-J. Lee and J.-Y. Jiang. Multi-Label Text Categorization Based on Fuzzy Relevance Clustering. IEEE Transactions on Fuzzy Systems, 22(6):1457-1471, 2014.
[10] Z.-Y. Wang. Some Variants of Self-Constructing Clustering. National Sun Yat-Sen University, 2017.
[11] G. H. Golub and C. F. Van Loan. Matrix computations. JHU Press, 2012.