



An oblique elliptical basis function network approach for supervised learning applications



Hung-Wen Peng^a, Shie-Jue Lee^{a,*}, Chie-Hong Lee^b

^a Department of Electrical Engineering, Electronic Commerce Research Center, and Information Technologies Research Center, National Sun Yat-Sen University, Kaohsiung 804, Taiwan

^b Department of Digital Content Application and Management, Wenzao Ursuline University of Languages, Kaohsiung 807, Taiwan

ARTICLE INFO

Article history:

Received 29 June 2015

Received in revised form 18 February 2017

Accepted 7 July 2017

Available online 19 July 2017

Keywords:

Supervised learning

Rotational clustering

Oblique elliptical functions

Supervised classification

Least squares estimation

Regression estimation

ABSTRACT

We propose a neural network architecture based on the oblique elliptical basis function for supervised learning problems. In classification, a category can be a disconnected or non-convex region involving several overlapping or disjoint sub-regions of the feature space. Other existing supervised learning methods may have the restriction that only allows decision regions to be convex. Our proposed method overcomes this restriction by employing a rotational self-constructing clustering algorithm to decompose the feature space into a collection of sub-regions which can then be combined to make up individual categories. An unseen instance is classified to a certain category if its similarity to the category exceeds a threshold. The whole framework fits in a five-layer network consisting of input, component-similarity, cluster-similarity, aggregation, and output layers. A similar idea also applies to solving regression problems. A parameter learning algorithm based on least squares estimation is used to derive the weights of the underlying network. Our approach can offer some advantages in practicality. Through the incorporation of rotation, data can be clustered more appropriately than by standard elliptical basis functions. Also, our approach is applicable to single-label classification, multi-label classification, as well as regression problems. A number of experiments are conducted to show the effectiveness of the proposed approach.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Supervised learning, including classification and regression, plays an important role in many fields [1–9]. Two kinds of classification tasks are encountered. One is single-label classification in which each instance belongs to only one category. The other one is multi-label classification in which an instance is allowed to belong to more than one category. For regression, one is usually interested in estimating the numerical relationship between a dependent variable and one or more independent variables.

1.1. Single-label classification

Many methods, based on machine learning techniques such as k -nearest neighbors (KNN) [10], multilayered perceptrons (MLP) [11], and radial basis function (RBF) neural networks [12], have been proposed for single-label classification. KNN is a type of lazy learning. For an unseen pattern, its k nearest neighbors in the feature

space are found. The unseen pattern is classified by being assigned to the most common class among these k nearest neighbors. KNN is simple, but no training is done and all computation is deferred until classification. A MLP is a feedforward artificial neural network model consisting of multiple layers of nodes, with each layer fully connected to the next one. Each node in the hidden layers is a neuron with a nonlinear activation function. MLP usually utilizes a supervised learning technique, called backpropagation, for training the network. A RBF network uses radial basis functions as activation functions. The output of the network is a linear combination of the outputs from radial basis functions. In [13], basis functions are interpreted as probability density functions. The weights are seen as prior probabilities. Models that output class conditional densities or mixture densities are proposed. A training algorithm based on the expectation-maximization (EM) algorithm [14], is developed. A hybrid training of this type of RBF networks is addressed in [15]. Some extensions of RBF neural networks, e.g., elliptical basis function (EBF) [16,17] and versatile EBF (VEBF) [18] neural networks, have also been proposed. Fast training algorithms are used to learn a data set in only one pass. The network structure is flexible and can be adjusted during the training process. However, the Euclidean or

* Corresponding author.

E-mail address: leesj@mail.ee.nsysu.edu.tw (S.-J. Lee).

Mahalanobis distance is used, and the learning of output weights is not taken into consideration.

1.2. Multi-label classification

To deal with multi-label text classification, two approaches are mainly adopted [19,20]. One approach transforms a multi-label classification task into several single-label classification tasks to which single-label classification methods can be applied [21,22]. The other extends the capability of a specific single-label classification algorithm to handle multi-label data directly. One popular problem transformation method, called binary relevance, was proposed in [21]. The method transforms the original dataset into p datasets, where p is the number of categories associated with the original dataset. Each resulting dataset contains all instances of the original dataset with only two labels, “belonging to” or “not belonging to” a particular category. Since the resulting datasets are single-labeled, all single-label classification techniques are applicable to them. Several neural-network approaches adaptively designed for multi-label classification have been proposed [27,19,24–26,23]. A kernel method proposed for multi-label classification is proposed in [21]. The back-propagation multi-label learning (BP-MLL) [28] algorithm is a multi-label version of the back-propagation neural network. To take care of multi-labels, label co-occurrence is incorporated into the pairwise ranking loss function. However, it has a complex global error function to be minimized. The multi-label k -nearest neighbors (ML-KNN) [29] is a lazy learning algorithm which requires a big run-time search. ML-RBF [30] is a multi-label RBF neural network which is an extension of the traditional RBF learning algorithm. Multi-label with fuzzy relevance clustering (ML-FRC) [31] is a fuzzy relevance clustering based method for multi-label text categorization. Nam et al. [32] report that binary cross entropy can outperform the pairwise ranking loss by leveraging rectified linear units for nonlinearity. Kurata et al. [33] propose a neural network initialization method to treat some of the neurons in the final hidden layer as dedicated neurons for each pattern of label co-occurrence. These dedicated neurons are initialized to connect to the corresponding co-occurring labels with stronger weights than to others.

1.3. Regression

Various approaches based on KNN, MLP, and RBF neural networks have been proposed for regression problems [10–12]. The KNN regression method is a nonparametric method which bases its prediction on the k nearest neighbors of the target to be forecasted. However, it lacks the capability of adaptation and the testing time is long. For the MLP and RBF networks used in regression, the outputs are numerical values instead of categories. Network parameters are learned from the training data. However, determining the number of hidden nodes is a challenging issue. Fuzzy theory has been incorporated for prediction [34]. However, membership functions need to be determined which is often a challenging task. Also, no learning is offered by fuzzy theory. A neuro-fuzzy scheme which combines the fuzzy theory and a hybrid learning method is proposed for regression problems [35]. A set of fuzzy IF-THEN rules is extracted from the training examples. Then a fuzzy neural network is constructed accordingly and parameters are refined to increase the precision of the fuzzy rule-base. Extreme learning machines (ELMs) [36,37] have also been proposed to deal with regression problems. ELM is a single-hidden layer feedforward network. The input weights connecting the input layer to the hidden layer, as well as the biases of the hidden neurons, are assigned to random values and no training is done for them. The output weights, con-

necting the hidden layer and the output layer, are determined by learning from training patterns.

1.4. Paper overview

In this paper, we present a neural network architecture based on the oblique elliptical basis function for classification and regression problems. In classification, a category can involve several overlapping or disjoint sub-regions. We decompose the feature space into a collection of sub-regions which can then be combined to make up individual categories. An unseen instance is classified to a certain category if its similarity to the category exceeds a threshold. The whole framework fits in a five-layer network consisting of input, component-similarity, cluster-similarity, aggregation, and output layers. A similar idea also applies to solving regression problems. Three steps are proposed for training such networks in the training phase: network initialization, parameter learning, and output activation. Firstly, a group of clusters is located for the given training data by a rotational self-constructing clustering algorithm. Secondly, a parameter learning algorithm based on least squares estimation is used to set the weights of the network. Finally, transfer functions are applied to derive the output response of the network. Other existing supervised learning methods may have the restriction that only allows decision regions to be convex. For instance, the decision regions KNN [10] provides are convex. For a given input, the decision region can be regarded as the surrounding convex space containing its k nearest neighbors. The perceptron algorithm [38,39] is another example where the decision boundary separates the decision regions which are convex. Support vector machines (SVM) [40,41] are yet another example. In a SVM model, a hyperplane is derived to divide the space into two convex decision regions. We have added these comments in this revision. However, our proposed method allows the decision region associated with an output node to be disconnected or non-convex.

This work differs from [35,31] in that the clustering algorithm allows the given data to be grouped into clusters expressed in the form of oblique elliptical basis functions. In [35,31], standard elliptical basis functions are used. No rotations are applied to the basis function and its axes are parallel to the coordinates. In this work, rotations are allowed and the axes may not be parallel to the coordinates. As a result, data can be clustered more appropriately by oblique elliptical basis functions than by standard elliptical basis functions. Consider a simple example in Fig. 1(a) containing 9 two-dimensional data points. A cluster with standard elliptical basis function embracing these points is shown by the dotted curve in Fig. 1(b), while a cluster with oblique elliptical basis function for the points is shown in Fig. 1(c).

The cluster of Fig. 1(c) is more fit to the 9 data points than the cluster of Fig. 1(b), covering less extraneous space. Moreover, the work in [35,31] can only apply to either regression or classification problems, while the approach proposed in this paper is applicable to single-label classification, multi-label classification, as well as regression problems.

The rest of this paper is organized as follows. The use of neural networks for supervised learning is addressed in Section 2. Our proposed neural network approach is described in Section 3. A simple example for illustration is given in Section 4. Experimental results are presented in Section 5. Finally, concluding remarks are given in Section 6.

2. Supervised learning with neural networks

Let $\mathbf{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) | 1 \leq i \leq N\}$ be a finite set of N training instances, where $\mathbf{x}^{(i)} \in \mathbb{R}^s$ is an input vector with s features, i.e., $\mathbf{x}^{(i)} = \langle x_1^{(i)}, x_2^{(i)}, \dots, x_s^{(i)} \rangle^T$, and $\mathbf{y}^{(i)}$ is the corresponding target vector with

Download English Version:

<https://daneshyari.com/en/article/4963011>

Download Persian Version:

<https://daneshyari.com/article/4963011>

[Daneshyari.com](https://daneshyari.com)