Neural Networks 63 (2015) 10–17

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Estimates on compressed neural networks regression

Yongquan Zhang [a,*], Youmei Li [a], Jianyong Sun [b], Jiabing Ji [a]

[a] Department of Information and Mathematics Sciences, China Jiliang University, Hangzhou 310018, Zhejiang Province, PR China

[b] School of Engineering, University of Greenwich, Central Avenue, Chatham Maritime, Kent ME4 4TB, UK

ARTICLE INFO

Article history:
Received 20 February 2014
Received in revised form 10 October 2014
Accepted 24 October 2014
Available online 10 November 2014

Keywords:
Regression learning
Neural networks
Compressed projection

ABSTRACT

When the neural element number n of neural networks is larger than the sample size m, the overfitting problem arises since there are more parameters than actual data (more variable than constraints). In order to overcome the overfitting problem, we propose to reduce the number of neural elements by using compressed projection A which does not need to satisfy the condition of Restricted Isometric Property (RIP). By applying probability inequalities and approximation properties of the feedforward neural networks (FNNs), we prove that solving the FNNs regression learning algorithm in the compressed domain instead of the original domain reduces the sample error at the price of an increased (but controlled) approximation error, where the covering number theory is used to estimate the excess error, and an upper bound of the excess error is given.

1. Introduction

In machine learning, feedforward neural networks (FNNs) and radial basis function networks (RBFNs) are usually considered as a hypothesis space for the study of the convergence performance of learning algorithms. For example, Barron (1993) gave the convergence rate of least square regression learning algorithm by using the approximation property of FNNs. RBFNs have become one of the most popular feedforward neural networks with applications in regression, classification and function approximation problems (see Bishop, 1997, Chen, Cowan, & Grant, 1991 and Haykin, 1994).

In 2006, Hamers and Kohler (2006) obtained the non-asymptotic bounds on the least square regression estimates by minimizing the empirical risk over suitable set of FNNs. Recently, Kohler and Mehnert (2011) presented an analysis on the convergence rate of least squares learning algorithms in set of FNNs for smooth regression function. All these mentioned analysis on regression learning algorithm are based on the assumption that the sample size m is higher than the neural element number n. However, in many real situations, m is less than n. It will lead to the overfitting problem. In other words, many minimizers of the empirical risk exist.

To overcome the overfitting problem, several approaches have been proposed in the literature. These approaches can be categorized as follows:

(1) **Regularization**. That is, the empirical error is combined with a penalty term, for examples, ℓ_1 norm (see Lasso (Tibshirani, 1994)), ℓ_2 norm (see ridge-regression (Tikhonov, 1963)), $\ell_{1/2}$ norm (e.g. Xu, Chang, & Xu, 2012), group Lasso (e.g. Mairal, Jenatton, Obozinski, & Bach, 2010 and Yuan & Lin, 2006) or overlapping group Lasso (e.g. Yuan, Yin, & Ye, 2011) and many others.

(2) **Minimizing norm**. That is, to find the minimizers of the empirical error with minimal norm (ℓ_1 or ℓ_2) (e.g. Tsaig & Donoho, 2006). However, the regularization parameter in the regularization term has not been addressed theoretically. On the other hand, for large n, finding solutions of minimal norm (for ℓ_1 or ℓ_2-norm problem) is numerically expensive.

In the paper, we propose to study the minimizer of the empirical error in the compressed hypothesis space instead of the original hypothesis space. That is, we propose to find solutions in the compressed hypothesis space. In recent years, dimension reduction and random projections in various learning areas has received considerable interests. Zhou, Lafferty, and Wasserman (2007) proposed to use compressed linear regression, in which the data set Y is compressed by the multiplication of a matrix A which satisfies the "Restricted Isometric Property" in a linear regression model $Y = X\beta + \epsilon$ where β is the coefficient and ϵ is noise. For the purpose of classification, Calderbank, Jafarpour, and Schapire (2010) studied an SVM algorithm in a compressed space and showed that their algorithm has good generalization properties. They also gave

* Corresponding author.
E-mail address: zyqmath@163.com (Y.Q. Zhang).

http://dx.doi.org/10.1016/j.neunet.2014.10.008
0893-6080/

some analysis on the Lasso estimator which built in these compressed data.

Davenport, Wakin, and Baraniuk (2006) discussed how compressed measurements may be useful to solve many detection, classification and estimation problems without having to reconstruct the signal. Interestingly, they made no assumption about the signal being sparse. Blum (2006) and Rahimi and Recht (2007) showed how to map a kernel $k(x, y) = \Phi(x) \times \Phi(y)$ into a low-dimensional space, while they still approximately preserved the inner products. Maillard and Munos (2009) studied the compressed least squares regression and gave the upper bound of the excess risk, using compressed projections. Motivated by those mentioned jobs, we aim to study the regression estimate in neural networks by the approximation property of neural networks and compressed projection in the paper.

The main contributions of the paper include that (1) we prove that the FNNs regression learning algorithm in the compressed domain reduces the sample error but at the price of an increased (but controlled) approximation error; (2) we give an estimation on the excess error and an upper bound of the excess error for the first time in literature for the compressed neural network regression. The new results provide a profound understanding of the overfitting problem and a mathematical estimation on the accuracy that the compressed neural network regression can reach. Moreover, the analysis applied in this paper also provides a mathematical framework for analysing the error bounds in the new network model, which has been studied little.

The rest of the paper is organized as follows. In Section 2, we present a brief introduction of regression learning and neural networks. In Section 3, we give the compressed projection of regression learning algorithm and give the convergence rate of the compressed regression learning algorithm. Section 4 concludes the paper.

2. Preliminaries on neural networks and regression learning

In the paper, we use FNNs set as the hypothesis space. That is, FNNs with one hidden layer and n hidden neurons. These FNNs can be formulated as a real-valued function on $\mathcal{R}^d$ of the form

$$N(x) = \sum_{j=1}^{n} c_j \sigma\left(\alpha_j^T x + \beta_j\right),$$

where $\sigma : \mathcal{R} \to [0, 1]$ is called a sigmoidal function and $\alpha_j \in \mathcal{R}^d$, $\beta_j, c_j \in \mathcal{R}$ $(j = 1, 2, \ldots, n)$ are the parameters that determine the neural networks.

Let $\phi_j : \mathcal{R}^d \to \mathcal{R}$ $(j = 0, 1, \ldots, n)$ be a family of real functions, then we define

$$N(x) = \sum_{j=1}^{n} c_j \phi_j(x), \quad c_j \in \mathcal{R},$$

and

$$\mathcal{N}_{n,\phi}^d = \left\{ N(x) : N(x) = \sum_{j=1}^{n} c_j \phi_j(x),\ c_j \in \mathcal{R} \right\}.$$

Clearly, $N(x)$ can be understood as a model of FNNs. In form, it looks quite similar to RBFNs (see Leonardis & Bischof, 1998 and Musavi, Ahmed, Chan, Farms, & Hummels, 1992).

Neural computation research has developed powerful methods for approximating continuous or integrable functions on compact subsets of $\mathcal{R}^d$ since 1980s. Most approximation schemes using FNNs and RBFNs have been studied (e.g. Cybenko, 1989, Funahashi, 1989 and Musavi et al., 1992). In such schemes, function approximation capabilities critically depend on the activation function nature of the hidden layer.

In the following, we introduce a class of activation function $\phi_j : \mathcal{R}^d \to \mathcal{R}$, defined by

$$\phi_j(x) = \phi_j(x, B) = \frac{e^{-B\rho(x,a_j)}}{\sum_{i=1}^{n} e^{-B\rho(x,a_i)}}, \quad j = 1, 2, \ldots, n,$$

where $a_1, \ldots, a_n$ are the data in $\mathcal{R}^d$, $\rho(a, b)$ denotes the Euclidean distance between two points a and b in $\mathcal{R}^d$, and $B > 0$ is a parameter. Furthermore, we define the linear combination of $\phi_j(x, B)$ as

$$N(x) = \sum_{j=1}^{n} c_j \phi_j(x, B).$$

Obviously, $N(x)$ can be understood to be a FNN with four layers: the first layer is the input layer, the input is $x \in \mathcal{R}^d$; the second layer is the processing layer for computing values $\rho(x, a_j)$ $(j = 0, 1, \ldots, n)$, between the input x and the prototypical input points a_j, and it is the input of the third layer that contains $n + 1$ neurons; $\phi_j(x, B)$ is an activation function of the jth neuron; the fourth layer is the output layer, and the output is $N(x)$.

It is well known that the sigmoidal function $\sigma(x) = \frac{1}{1+e^{-x}}$ is a logistic model. This model is important and has been widely used in biology, demography and so on (see Brauer & Castillo-Chavez, 2001 and Hritonenko & Yatsenko, 2006). Naturally, the functions

$$\phi_j(x) = \frac{e^{-B\rho(x,a_j)}}{\sum_{i=1}^{n} e^{-B\rho(x,a_i)}}, \quad j = 1, 2, \ldots, n$$

can be regarded as a multi-class generalization of the logistic model (see Section 10.6 in Hastie, Tibshirani, & Friedman, 2001), which was also used in a regression model for the case of multi-class in the classification problems. Although the functions $\phi_j(x)$ are not sigmoidal, they possess some properties that common sigmoidal functions do not have, for example

$$0 < \phi_j(x) \le 1,\ j = 1, 2, \ldots, n, \quad \sum_{j=1}^{n} \phi_j(x) = 1.$$

On the other hand, it follows from their structures that $\phi_j(x)$ contain the information of the interpolation samples. The second layer of the network composed of $\phi_j(x)$ can be regarded as the processing layer and the input of the third layer, which is more convenient for the study of network interpolations. Motivated by those properties of $\phi_j(x)$, we introduce functions $\phi_j(x)$ as activation functions in the hidden layer of networks. In Cao, Zhang, and He (2009), we studied the convergence rate of neural networks $N(x)$ approximating continuous function by continuous modulus.

Let (X, d) be a compact metric space, $Y = \mathcal{R}$ and ρ be a probability distribution on $Z = X \times Y$. Denote by $\mathbf{z} = \{z_i\}_{i=1}^m = \{(x_i, y_i)\}_{i=1}^m \in Z^m$ a set of random samples, which are independently drawn according to ρ. Let ρ_X, $\rho(y|x)$ be margin probability measure and condition probability measure of ρ respectively. In the paper, we define the set $\mathcal{F}_{m,n}$ as the hypothesis space according to the neural networks $N(x)$:

$$\mathcal{F}_{m,n} = \left\{ N(x) = \sum_{j=1}^{n} c_j \phi_j(x) : c_j \in \mathcal{R}, \sum_{j=1}^{n} |c_j| \le M \ln m \right\},$$

where M is a positive number.

Since every ϕ_j is bounded in absolute value by 1, the functions in $\mathcal{F}_{m,n}$ are bounded in absolute value by $M \ln m$. For $f \in \mathcal{F}_{m,n}$, we define the empirical square error

$$\mathcal{E}_{\mathbf{z}}(f) = \frac{1}{m} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$