Neural Networks 63 (2015) 31–47

Contents lists available at ScienceDirect

Neural Networks

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

Robust sequential learning of feedforward neural networks in the presence of heavy-tailed noise

Najdan Vuković[a,*], Zoran Miljković[b,1]

[a] University of Belgrade - Faculty of Mechanical Engineering, Innovation Center, Kraljice Marije 16 ; 11120 Belgrade 35, Serbia
[b] University of Belgrade - Faculty of Mechanical Engineering, Production Engineering Department, Kraljice Marije 16 ; 11120 Belgrade 35, Serbia

ARTICLE INFO

Article history:
Received 18 February 2014
Received in revised form 18 September 2014
Accepted 4 November 2014
Available online 15 November 2014

Keywords:
Feedforward neural networks
Sequential learning
Robust extended Kalman filter
Structured variational approximation
Heavy-tailed noise
Inverse Wishart distribution

ABSTRACT

Feedforward neural networks (FFNN) are among the most used neural networks for modeling of various nonlinear problems in engineering. In sequential and especially real time processing all neural networks models fail when faced with outliers. Outliers are found across a wide range of engineering problems. Recent research results in the field have shown that to avoid overfitting or divergence of the model, new approach is needed especially if FFNN is to run sequentially or in real time. To accommodate limitations of FFNN when training data contains a certain number of outliers, this paper presents new learning algorithm based on improvement of conventional extended Kalman filter (EKF). Extended Kalman filter robust to outliers (EKF-OR) is probabilistic generative model in which measurement noise covariance is not constant; the sequence of noise measurement covariance is modeled as stochastic process over the set of symmetric positive-definite matrices in which prior is modeled as inverse Wishart distribution. In each iteration EKF-OR simultaneously estimates noise estimates and current best estimate of FFNN parameters. Bayesian framework enables one to mathematically derive expressions, while analytical intractability of the Bayes' update step is solved by using structured variational approximation. All mathematical expressions in the paper are derived using the first principles. Extensive experimental study shows that FFNN trained with developed learning algorithm, achieves low prediction error and good generalization quality regardless of outliers' presence in training data.

1. Introduction

Any real world application of neural network based model of the system is subjected to the high/moderate noise and existence of outliers in data. Outliers have enormous practical significance because these data points occur relatively often in engineering. Outlier may be defined as an observation that numerically significantly differs from the rest of the data (Agamennoni, Nieto, & Nebot, 2012) that it raises suspicion in phenomena or mechanism we believe that actually generated all data. Typically, outliers fall outside of an overall pattern of distribution (Agamennoni, Nieto, & Nebot, 2011; Ting, Theodorou, & Schaal, 2007). In engineering, especially in applications with real time data processing ability, outliers are a common phenomenon that needs to be analyzed and their influence has to be integrated into the model analysis and validation. Failing to recognize their influence may significantly jeopardize performance of the model, especially if our model is to perform sequential processing of the data or run in real time (Miljković, Vuković, Mitić, & Babić, 2013; Vuković & Miljković, 2013). Outliers may occur by chance, but more often, they may originate from temporary sensor failures, some unknown system anomalies or unmodeled reactions from the environment or some other disturbances; all of these may cause data points to fall far away from expected pattern of data distribution, and as an overall result they may cause our model of the system to diverge from designed performance.

In this paper, we develop an original approach for neural network sequential learning that does not require preprocessing of the data to model and process outliers. Our model is based on a standard extended Kalman filter (EKF), which is modified to process outliers as if these were a "normal" data points. Performance of EKF is based on an assumption that system and measurement equations are corrupted with additive white

* Corresponding author. Tel.: +381 63 363 858.
E-mail addresses: nvukovic@mas.bg.ac.rs (N. Vuković), zmiljkovic@mas.bg.ac.rs (Z. Miljković).
[1] Tel.: +381 11 3302 468.

http://dx.doi.org/10.1016/j.neunet.2014.11.001
0893-6080/© 2014 Elsevier Ltd. All rights reserved.

Gaussian noise. The noise level is constant, defined with covariance matrix. Gaussian assumption is backed up with Central Limit Theorem—which states that as sample goes to infinity, arithmetic mean of a set of random variables with finite mean and variance having arbitrary distribution, in limit tends to the Gaussian distribution; furthermore, Gaussians are popular due to their simple mathematical form which (in most cases) results in straightforward closed mathematical calculations. However, in nature and in engineering, not much of processes obey Gaussian assumption. Similarly, Gaussian has tin tails, which suggests that there is a zero chance for misreading or fake measurements. Failing to recognize and process non-Gaussian noises can seriously damage model's performance and cause divergence. To provide more flexibility with respect to exogenous noise, in our model we assume additive noise as well, but in contrast to standard EKF we do not assume Gaussian probability distribution of noise and allow observation noise covariance matrix to change over time. These two assumptions have following ramifications: firstly, we acknowledge that real world does not obey Gaussian distribution and that is why we introduce probability distribution of the noise with heavier and longer tail. Secondly, we estimate noise covariance in each iteration, which helps us to introduce possibly unmodeled environmental disturbances in the model, where new information is encoded into estimated noise covariance matrix. Flexibility of this approach is obvious when it comes to explaining outliers in data, especially if it is sequentially processed.

The learning algorithm is developed in sequential form (Vuković, 2012), which means that whenever new data is available, the sequential learning continues learning process by updating the existing neural network, instead of going through entire learning process from the beginning (Vuković & Miljković, 2013). This is why sequential algorithms are preferred, especially in engineering and applications where fast development of neural network based models are needed (Miljković & Aleksendrić, 2009). Sequential learning has the following characteristics (Huang, Saratchandran, & Sundararajan, 2005):

(1) Learning system uses one and only one training example in iteration. The examples are presented to the learner sequentially, one following the other.
(2) Training example is erased from memory after learning procedure finishes update of network parameters.
(3) Learning system has no prior knowledge of the total number of examples.

These features of sequential learning are important for modeling of engineering problems, where new data might be available after neural network model was built. Sequential learning enables learning system to continue learning process if new data is received, without need to memorize and use old data (Vuković & Miljković, 2013). Furthermore, when sequential learning is used the need to have learning algorithm robust to outliers able to sequentially process them is even more emphasized.

This paper is organized as follows. The second part of the paper provides analysis of research results and compares features of proposed sequential learning algorithm with ability to treat outliers with the current state in the field. In Section 3 we provide basic intuition, foundations and mathematical derivation of the learning algorithm. Through various experimental studies using real world and synthetic data, in Section 4 we demonstrate and discuss the potential of the developed learning algorithm for training of two types of FFNN when faced with outliers in the data. Eventually, final conclusions and assessments of learning algorithm's performance are given in Section 5.

2. Related work and contributions of the paper

Robust statistics is a broad field of research and in this section of the paper we wish to concentrate solely on soft computing approaches, especially neural networks. For wider prospective the reader is kindly referred to Agamennoni et al. (2011, 2012), Chandola, Banerjee, and Kumar (2009), Đurović and Kovačević (1999), Hodge and Austin (2004), Markou and Singh (2003a, 2003b) Stanković and Kovačević (1986), Schick and Mitter (1994) and references therein.

If model minimizes L_2 norm, than it emphasizes outliers more than it should; this situation leads to over-fitting and poor generalization of the model when outliers are present. On the other hand, when L_1 norm is minimized, model puts emphasis on data points close to the prediction, which is yet another undesired situation which neglects update step; certain data point may not be an outlier but it may generate large error between prediction of the model and the actual value, which will make the learning algorithm to classify it as an outlier. To solve this issue, research community has proposed a great diversity of robust cost functions called M-estimators (Huber, 2011) for developing of robust statistical/neural models. The main attractiveness of M-estimators is their influence function; it is bounded which guarantees bounded response given arbitrary query point, unlike non robust cost function whose influence function is unbounded. This feature makes M-Estimator popular approach for robust estimation/learning.

In this paragraph we provide information related to the usage of robust cost function in neural network/support vector machine community. To achieve robustness of their model, authors in Lee, Chung, Tsai, and Chang (1999) propose Hample M-estimator to accommodate large errors in data. Instead of Gaussian activation function, authors in their radial basis function (RBF) neural network propose composite of sigmoid functions and introduce growing and pruning of neurons. The robustness in Chuang, Su, Jeng, and Hsiao (2002) is introduced in terms of traditional concept of robustness in statistics; authors make use of robust cost function such as hyperbolic tangent estimator and build their Support Vector Regression (SVR) Network in two phases. In the first phase a classical approach towards SVR optimization is taken. In the second phase, weights are being adjusted using robust learning based on robust cost function. Similarly, in Chuang, Su, and Chen (2001) authors develop fuzzy based model and enable robustness by using different cost function (Tukey's biweight cost function). Authors in Lee, Chiang, Shih, and Tsai (2009) build their model based on RBF and use Welsch M-estimator as cost function. The Welsch is chosen as cost function because of its smoothness and reduction of effects of large errors. Furthermore, their robust RBF model allows growing and pruning of the neurons based on the concept of neuron significance (Huang, Saratchandran, & Sundararajan, 2004; Huang et al., 2005; Vuković & Miljković, 2013). In Zhu, Hoi, and Lyu (2008) authors point out that robustness in the model may be introduced with regularization term in cost function or to use robust cost function. To solve these issues and achieve robustness into their Regularized Kernel regression authors use Huber robust function because of its ability to use both L_2 and L_1 norms. Instead of solving optimization problem in dual, their algorithm is run in primal optimization form. A modified Huber robust cost function is applied in Pernía-Espinoza, Ordieres-Meré, Martínez-de-Pisón, and González-Marcos (2005); authors argue that scalability of the Huber function is not appropriate, hence they develop a τ based approach (Yohai & Zamar, 1988) which uses scale estimator. Similarly to previous result (Chuang et al., 2002), in Chuang and Jeng (2007) and Chuang, Jeng, and Lin (2004) one may see the two phases in learning, (i) determination of initial structure and (ii) robust learning. However, the logistic cost function is used in