Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

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

Adding reliability to ELM forecasts by confidence intervals

Anton Akusok [a,d,*], Andrey Gritsenko [a], Yoan Miche [b,c], Kaj-Mikael Björk [e], Rui Nian [f], Paula Lauren [g], Amaury Lendasse [a]

[a] Department of Mechanical and Industrial Engineering and the Iowa Informatics Initiative, The University of Iowa, Iowa City, USA
[b] Bell Labs, Nokia, Espoo, Finland
[c] Department of Information and Computer Science, Aalto University School of Science, FI-00076, Finland
[d] Department of Business Management and Analytics, Arcada University of Applied Sciences, Helsinki, Finland
[e] Risklab at Arcada University of Applied Sciences, Helsinki, Finland
[f] School of Information Science and Engineering, Ocean University of China, Qingdao, China
[g] School of Engineering and Computer Science, Oakland University, Rochester, USA

ARTICLE INFO

Article history:
Received 22 February 2016
Received in revised form
25 August 2016
Accepted 10 September 2016
Communicated by G.-B. Huang

Keywords:
Extreme learning machines
Confidence
Confidence interval
Regression
Skin segmentation
Big data

ABSTRACT

This paper proposes a way of providing transparent and interpretable results for ELM models by adding confidence intervals to the predicted outputs. In supervised learning, outputs are often random variables because they may depend on information that is unavailable, due to the presence of noise, or the projection function itself may be stochastic. Probability distribution of outputs is input dependent, and the observed output values are samples from that distribution. However, ELM predicts deterministic outputs. The proposed method addresses that problem by estimating predictive Confidence Intervals (CIs) at a confidence level α, such that random output values fall between these intervals with probability α.

Assuming that the outputs are normally distributed, only a standard deviation is needed to compute CIs of a predicted output (the predicted output itself is a mean). Our method provides CIs for ELM predictions by estimating standard deviation of a random output for a particular input sample. It shows good results on both toy and real skin segmentation datasets, and compares well with the existing Confidence-weighted ELM methods. On a toy dataset, the predicted CIs accurately represent the variable variance of outputs. On a real dataset, CIs improve the precision of a classification task at a cost of recall.

1. Introduction

Extreme Learning Machines [1–3] (ELM) are fast and robust [4,5] methods for training feed-forward neural networks that have the universal approximation property [6] and obtained numerous applications in regression [7–9] and classification [10] problems. They are an active research topic with multiple extensions and improvements proposed over the last decade.[1]

ELMs are powerful non-linear methods, but they share one common drawback of non-linear methods in practical applications, which is a non-transparency of results (predictions). A prediction made by a linear model from input data is easily explained and interpreted in terms of the coefficients of the input data features. Results with an explanation are easier to trust and apply for people outside the Machine Learning field. Non-linear models lack such transparency, so their results are less trusted, and thus non-linear methods (including ELM) are sometimes rejected despite supreme performance compared to linear methods.

More reliability and intuition can be added to outputs predicted by ELM model by computing confidence intervals (CIs) [11,12] for them. Here, an $\alpha_*100\%$ CIs are the upper and lower boundaries on a random variable, such that samples from that variable fall between the intervals with probability $\alpha_*100\%$. Outputs of a supervised prediction task are often random variables because the training data is corrupted by noise, the outputs may depend on information not present in inputs, or the underlying projection function itself may be stochastic. The observed output values are samples from the actual random outputs. The Ordinary Least Squares method used for learning the output weights in ELM assumes that an output for a given input is normally distributed, and ELM predicts mean values of the outputs. Variance of outputs is also needed for an estimation of CIs, and with an additional assumption of equal variance over the whole dataset its maximum likelihood estimator is given by the Mean Squared Error (MSE) [13].

* Corresponding author at: Department of Mechanical and Industrial Engineering and the Iowa Informatics Initiative, The University of Iowa, Iowa City, USA.
E-mail addresses: anton-akusok@uiowa.edu (A. Akusok), amaury-lendasse@uiowa.edu (A. Lendasse).

[1] This paper is an extension of a publication accepted at the ELM'15 conference.

http://dx.doi.org/10.1016/j.neucom.2016.09.021
0925-2312/

However, an assumption of equal variance over the whole dataset is a limitation in practical applications that require the highest precision at a cost of recall [10], in ensembling the results of multiple methods, and in exploring ELM performance over the available data. An input-dependent variance of outputs allows for variable CIs that separate part of data with reliable and stable predictions from other parts where an ELM model is unstable and inaccurate for some reason (like insufficient amount of training data in that region, a complex form of underlying transformation function, or being at model boundaries). This paper estimates the input-dependent variance of outputs by sampling predicted outputs at the same input point from different ELM models.

Unfortunately, a simple sampling from ELM predictions is insufficient for an output variance estimation, because an ELM model with correct hyper-parameters and enough training data provides precise estimation of the mean value of outputs disregard their variance. An original idea of this paper is to deliberately put ELM in sub-optimal conditions by training it on a small subset of the training data, keeping hyper-parameters selected on the full training set. In this setup, an ELM model starts to overfit due to insufficient amount of data and excessive model complexity. Predictions of an overfitted ELM model differ more from the true mean of random outputs, and that difference depends on the variance of outputs corresponding to a particular input region. An input-dependent variance of outputs is computed by analyzing such behavior of ELM models. Using the input-specific variance and an ELM prediction as a mean, CIs are constructed to cover $\alpha_*100\%$ of probability in output distribution of a given ELM output.

Another approach to estimation of CIs is to assume a probability distribution on the model weights. Then predicted outputs and their CIs are obtained simultaneously from the parameters of weights distribution. An example of such methods is Gaussian Processes [14]. For ELM model, such methods are known in literature as a family of Bayesian ELM methods [15–17]. A Confidence Weighted ELM [11] (CW-ELM) is another recent approach that computes input-dependent CIs. It outperforms Bayesian ELM, and in fact can be applied on top of any other ELM model just like the proposed method. CW-ELM is explained in the next Section 2, and the comparison results are given in the experimental Section 4.

The next Section 3 introduces the method of computing input-specific CIs. The experimental Section 4 presents the examples of confidence intervals on an artificially made toy dataset, a comparison of CIs on a benchmark dataset to the Bayesian ELM family of methods, and a visualization of results (as well as a performance on large data) for the real image segmentation task. In the conclusion, Section 7, the method is summarized and further research directions are discussed.

2. Summary of confidence-weighted ELM

A confidence-weighted ELM is a method that provides confidence intervals for particular forecasts of an Extreme Learning Machine. As noted by the authors, CW-ELM can be applied on top of any improved ELM method ([11], Section 4.2) such as Optimally Pruned ELM [4] (OP-ELM). The latter is used as a base ELM model stand-alone, in combination with CW-ELM and the proposed method for a fair comparison of the results.

The CW-ELM assumes that the output weight vector is normally distributed $\mathbf{w} \sim \mathcal{N}_p(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ where $\boldsymbol{\beta} \in \mathbb{R}^p$ is an estimate of the mean value of output weights (or the output weighs themselves for deterministic algorithms) obtained from some other ELM training method, such as OP-ELM. Matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$ is a positive definite covariance matrix. Parameter p is the number of hidden neurons in the base ELM model, like the number of selected neurons in OP-ELM.

As a predicted output $y_i = f(\mathbf{x}_i)^T\mathbf{w}$, $i \in [1, N]$ for N data samples, it follows the distribution

$$y_i \sim \mathcal{N}(f(\mathbf{x}_i)^T\boldsymbol{\beta},\ f(\mathbf{x}_i)^T\boldsymbol{\Sigma}f(\mathbf{x}_i)). \tag{1}$$

The goal of CW-ELM is to keep the targets t_i within the predicted confidence intervals

$$f(\mathbf{x}_i)^T\boldsymbol{\beta} - \eta\sqrt{f(\mathbf{x}_i)^T\boldsymbol{\Sigma}f(\mathbf{x}_i)} \le t_i f(\mathbf{x}_i)^T\boldsymbol{\beta} + \eta\sqrt{f(\mathbf{x}_i)^T\boldsymbol{\Sigma}f(\mathbf{x}_i)} \ge t_i \tag{2}$$

Parameters of the least informative distribution that keeps the targets within the aforementioned CIs are found by the solution of the following problem:

$$\begin{aligned}
&\min_{\boldsymbol{\beta},\boldsymbol{\Sigma}} -\frac{1}{2}\ln\det\boldsymbol{\Sigma} + \frac{1}{2a}\boldsymbol{\beta}^T\boldsymbol{\beta} + \frac{1}{2a}\operatorname{tr}\boldsymbol{\Sigma}\\
&\text{s. t. } f(\mathbf{x}_i)^T\boldsymbol{\beta} - \eta\sqrt{f(\mathbf{x}_i)^T\boldsymbol{\Sigma}f(\mathbf{x}_i)} \le t_i\\
&\qquad f(\mathbf{x}_i)^T\boldsymbol{\beta} + \eta\sqrt{f(\mathbf{x}_i)^T\boldsymbol{\Sigma}f(\mathbf{x}_i)} \ge t_i, \quad i \in [1, N]
\end{aligned} \tag{3}$$

Eq. (3) is hard to solve directly, thus a simplified version is used. The covariance matrix $\boldsymbol{\Sigma}$ is assumed to be diagonal $\boldsymbol{\Sigma} = \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$, $\lambda_i > 0$. Then a simplified problem (ignoring the constant term $\boldsymbol{\beta}^T\boldsymbol{\beta}$) is formulated as

$$\begin{aligned}
&\min_{\boldsymbol{\Lambda}} -\frac{1}{2}\sum_{i=1}^{p}\ln\lambda_i + \frac{1}{2a}\sum_{i=1}^{p}\lambda_i\\
&\text{s. t. } f(\mathbf{x}_i)^T\boldsymbol{\Lambda}f(\mathbf{x}_i) \le \frac{1}{\eta^2}|t_i - f(\mathbf{x}_i)^T\boldsymbol{\beta}|^2, \quad i \in [1, N].
\end{aligned} \tag{4}$$

Let us construct matrix $\mathbf{G}$ by taking element-wise square of all elements in the hidden layer output matrix $\mathbf{H}$ of an ELM, and define vectors $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]^T$, $\xi_i = |t_i - f(\mathbf{x}_i)^T\boldsymbol{\beta}|^2$, $i \in [1, N]$ and $\boldsymbol{\lambda} = [\lambda_i, \ldots, \lambda_p]^T$. Then a re-arranged optimization task is written as

$$\begin{aligned}
&\min_{\boldsymbol{\Lambda}} \sum_{i=1}^{p}\lambda_i - a\sum_{i=1}^{p}\ln\lambda_i\\
&\text{s. t. } \mathbf{G}\boldsymbol{\lambda} \ge \boldsymbol{\xi}.
\end{aligned} \tag{5}$$

The solution to Eq. (5) can be computed by various approaches. The experimental section uses a fast conic optimization method from [18], that has an efficient implementation with an interface to Python.[2] Once the optimal value of $\boldsymbol{\lambda}$ is known, the CIs for a test sample $\mathbf{x}^*$ are computed as

$$\begin{aligned}
&\mathrm{CI}(\mathbf{x}^*) = [f(\mathbf{x}^*)^T\boldsymbol{\beta} - \eta\sigma(\mathbf{x}^*), f(\mathbf{x}^*)^T\boldsymbol{\beta} - \eta\sigma(\mathbf{x}^*)]\\
&\sigma(\mathbf{x}^*) = \sqrt{f(\mathbf{x}^*)^T\boldsymbol{\Lambda}f(\mathbf{x}^*)}.
\end{aligned} \tag{6}$$

3. Confidence intervals method for extreme learning machines

3.1. Intuition

In supervised prediction, true outputs are often random variables, and the training outputs are samples from them. Random outputs can be approximated by the normal distribution, mean value of that is easy to predict (i.e. by the Ordinary Least Squares method). Variance of the outputs distribution is estimated by sampling multiple points from it.

If variance is assumed to be equal over the whole dataset, then all dataset outputs are samples from the same distribution. Its variance is easily computed after subtracting the mean, effectively what MSE does. However, if the output variance is input-dependent, then the dataset outputs are samples of different distributions, and only one sample is available for variance estimation in distribution of each particular output, that is not enough.

[2] http://www.cvxpy.org.

Please cite this article as: A. Akusok, et al., Adding reliability to ELM forecasts by confidence intervals, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.09.021