ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Model Variance for Extreme Learning Machine

Fabian Guignard[1], Mohamed Laib[2] and Mikhail Kanevski[1] *

1- University of Lausanne - Institute of Earth Surface Dynamics (IDYST)
UNIL-Mouline, 1015 Lausanne - Switzerland

2- Luxembourg Institute for Science and Technology (LIST)
IT for Innovative Services, L-4362 Esch-sur-Alzette - Luxembourg

Abstract. We derived theoretical formulas for the variance of extreme learning machine ensemble in a general case of a heteroskedastic noise. They provide a decomposition of the variance, which helps in the understanding of how the different sources of randomness contribute. The application of the proposed method to simulated datasets shows the effectiveness of the newly introduced estimations in replicating the expected variance behaviours.

1 Introduction

Model uncertainty quantification is important to provide assessment of the model quality. In particular, it allows the development of active learning procedures and is essential to derive confidence intervals. Moreover, uncertainty of the regression estimate is necessary to build accurate prediction intervals.

Extreme Learning Machine (ELM) [1] is a single-layer feed-forward neural network, for which the input weights and biases are randomly drawn. This enables the optimisation of the output weights, with respect to the L_2 criterion, by solving N linear equations, where N denotes the number of neurons of the hidden layer. From a statistical point of view, the inputs are projected in a $N-$dimensional random feature space, where a Multivariate Linear Regression (MLR) with a null intercept is performed.

Several methods were proposed to obtain confidence/prediction intervals with ELM, such as Bayesian ELM [2], bootstrap-based ELM [3] and weighted Jackknife [4]. This research proposes a model variance estimation based on an analytical computation within the frequentist framework, which avoids the use of prior knowledge, resampling procedures, or hypothesis on data distribution. The main formulas are derived in section 2 and estimators are proposed in section 3. Simulated experiments are conducted in section 4. Section 5 concludes the paper.

2 Analytical developments

2.1 Extreme Learning Machine

Let the training set $\mathcal{D} = \{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}_{i=1}^n$ be a realization of independent and identically distributed random variables $\{(X_i, Y_i)\}_{i=1}^n$. Input

*This research is partly supported by the Swiss PNR75 ”Big Data”, project No. 167285 ”HyEnergy”.

weights $\mathbf{w}_j \in \mathbb{R}^d$ and biases $b_j \in \mathbb{R}$ are randomly drawn, independent and identically distributed. Then, we can compute the hidden matrix $H \in \mathbb{R}^{n\times N}$, defined by $(H)_{ij} = g(\mathbf{x}_i^T \mathbf{w}_j + b_j)$, where g is any infinitely differentiable activation function. The output weights $\beta \in \mathbb{R}^N$ are related to the hidden matrix by $H\beta = \mathbf{y}$, where $\mathbf{y} = (y_1, \dots, y_n)^T \in \mathbb{R}^n$. They are optimized regarding the L_2 criterion $E = ||\mathbf{y} - H\beta||^2$, which is exactly the least squares (LS) procedure for a fixed design matrix H [5]. If the matrix $H^T H$ is of full rank and then invertible, the output weights are directly estimated by $\hat{\beta} = H^\dagger \mathbf{y}$, where $H^\dagger = (H^T H)^{-1} H^T$. With small datasets, the model can be retrained several times and averaged in order to reduce the randomness induced by the input weight initialization.

2.2 Model variance for ELM

Assume that $\mathcal{D}$ is generated by $y = f(\mathbf{x}) + \varepsilon(\mathbf{x})$, where f is the true function we want to approximate and $\varepsilon(\mathbf{x})$ denotes an independent centered noise depending on the input. At the training points we note $\mathbf{f}$, respectively $\boldsymbol{\varepsilon}$, the vector defined by $(\mathbf{f})_i = f(\mathbf{x}_i)$, respectively $(\boldsymbol{\varepsilon})_i = \varepsilon(\mathbf{x}_i)$, and Σ the covariance matrix of $\boldsymbol{\varepsilon}$.

At a new point $\mathbf{x}_0 \in \mathbb{R}^d$, the prediction is given by $\hat{f}(\mathbf{x}_0) = \mathbf{h}_0^T \hat{\beta}$, where $\mathbf{h}_0 \in \mathbb{R}^N$ is the vector defined by $(\mathbf{h}_0)_j = g(\mathbf{x}_0^T \mathbf{w}_j + b_j)$. Note that $\hat{f}(\mathbf{x}_0)$ is a random variable depending on $\boldsymbol{\varepsilon}$, but also on the stochastic quantities used in H and $\mathbf{h}_0$. Let us denote $\mathbf{X} = (X_1, \dots, X_n)^T$ and $\mathbf{W}$ the random vector of all input weights and biases. Using the law of total conditional variance, one can compute the variance of the model at $\mathbf{x}_0$, conditioned on the input data,

$$\begin{aligned} \mathrm{Var}\left[\hat{f}(\mathbf{x}_0)|\mathbf{X}\right] &= \mathbb{E}\left[\mathrm{Var}\left[\hat{f}(\mathbf{x}_0)|\mathbf{W},\mathbf{X}\right]|\mathbf{X}\right] + \mathrm{Var}\left[\mathbb{E}\left[\hat{f}(\mathbf{x}_0)|\mathbf{W},\mathbf{X}\right]|\mathbf{X}\right] \\ &= \mathbb{E}\left[\mathbf{h}_0^T H^\dagger \Sigma H^{\dagger T} \mathbf{h}_0 |\mathbf{X}\right] + \mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{f}|\mathbf{X}\right]. \end{aligned} \tag{1}$$

The first term of the right-hand side (RHS) is the variance of the LS step averaged on all the random feature spaces. In the MLR statistical framework, assumptions on data generation force the model to be unbiased. Here, this is not the case and the variation of $\mathbb{E}\left[\hat{f}(\mathbf{x}_0)|\mathbf{W},\mathbf{X}\right]$ — which is equivalent to the bias variation of the LS step — has to be considered and yields the second term.

2.3 Model variance for ELM ensemble

As mentioned before, the training could be done several times, yielding M models $\hat{f}_m, m = 1, \dots, M$. Then, the final prediction $\hat{f}$ is the average of the M predictions. An analogous direct and long calculation can be done for the variance of the mean predictor yielding the following three-terms formula,

$$\begin{aligned} \mathrm{Var}\left[\hat{f}(\mathbf{x}_0)|\mathbf{X}\right] = &\frac{1}{M}\mathbb{E}\left[\mathbf{h}_0^T H^\dagger \Sigma H^{\dagger T} \mathbf{h}_0 |\mathbf{X}\right] \\ &+ \frac{M-1}{M}\mathbb{E}\left[\mathbf{h}_{0,1}^T H_1^\dagger \Sigma H_2^{\dagger T} \mathbf{h}_{0,2} |\mathbf{X}\right] + \frac{1}{M}\mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{f}|\mathbf{X}\right], \end{aligned} \tag{2}$$

where indices 1 and 2 are used to distinguish interaction between two different models. The RHS first and third terms are the equation (1) divided by the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

number of models. The bias variation of the LS step is reduced by a $1/M$ factor. Although the average variance of the LS step represented by the RHS first term seems to decrease by a $1/M$ factor, models are pairwise strongly dependent due to the fact that the same training dataset is used to train all models, which yields the RHS second term. Notice that if $M = 1$, formula (1) is recovered.

If the noise variance is assumed to be homoskedastic, we can write $\Sigma = \sigma_\varepsilon^2 I$. Then, the model variance becomes

$$\begin{aligned}\mathrm{Var}\left[\hat{f}(\mathbf{x}_0)|\mathbf{X}\right] &= \frac{\sigma_\varepsilon^2}{M}\,\mathbb{E}\left[\mathbf{h}_0^T(H^TH)^{-1}\mathbf{h}_0|\mathbf{X}\right] \\ &+ \frac{(M-1)\sigma_\varepsilon^2}{M}\,\mathbb{E}\left[\mathbf{h}_{0,1}^T H_1^\dagger H_2^{\dagger T}\mathbf{h}_{0,2}|\mathbf{X}\right] + \frac{1}{M}\,\mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{f}|\mathbf{X}\right].\end{aligned} \tag{3}$$

Notice that if $\mathbf{W}$ and $\mathbf{X}$ are deterministic and $M = 1$, we recover the classical MLR formula for the model variance at a prediction point, see [5].

3 Model variance estimation for ELM ensemble

First, one want to estimate $\mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{f}\right|\mathbf{X}]$, i.e. the variance induced by the randomness of $\mathbf{W}$ knowing the true f at the training points. In particular, this quantity is the same for the homoskedastic and heteroskedastic case and noise is not involved in it. As $\mathbf{f}$ is not accessible, one can approximate it by the model predictions at training points, $\hat{\mathbf{f}}$, defined by $(\hat{\mathbf{f}})_i = \hat{f}(\mathbf{x}_i)$. Using the fact that $\hat{\mathbf{f}} = HH^\dagger\mathbf{y}$, one has for each model

$$\mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{f}|\mathbf{X}\right] \approx \mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \hat{\mathbf{f}}|\boldsymbol{\varepsilon},\mathbf{X}\right] = \mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{y}|\boldsymbol{\varepsilon},\mathbf{X}\right] = \mathrm{Var}\left[\hat{f}(\mathbf{x}_0)|\boldsymbol{\varepsilon},\mathbf{X}\right].$$

This motivates the following estimator for $\mathrm{Var}\left[\mathbf{h}_0^T H^\dagger \mathbf{f}\right|\mathbf{X}]$,

$$\hat{\sigma}_{\hat{f}}^2(\mathbf{x}_0) = \frac{1}{M-1}\sum_{m=1}^{M}\left(\hat{f}_m(\mathbf{x}_0) - \frac{1}{M}\sum_{l=1}^{M}\hat{f}_l(\mathbf{x}_0)\right)^2.$$

3.1 Estimator for the homoskedastic case

Here, σ_ε^2 is estimated by

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{n-N}\sum_{i=1}^{n} r_{i,m}^2\right),$$

where $r_{i,m} = y_i - \hat{f}_m(\mathbf{x}_i)$ is the residual at the i^{th} training data for the m^{th} model. Although this is the mean of all σ_ε^2 MLR estimates, it is biased.

By approximating expectations by means in equation (3), the model variance at a new point $\mathbf{x}_0$ can be estimated by

$$\hat{\sigma}^2(\mathbf{x}_0) = \frac{\hat{\sigma}_\varepsilon^2}{M^2}\left(\sum_{m=1}^{M}\mathbf{h}_{0,m}^T(H_m^TH_m)^{-1}\mathbf{h}_{0,m} + 2\sum_{m<l}\mathbf{h}_{0,m}^T H_m^\dagger H_l^{\dagger T}\mathbf{h}_{0,l}\right) + \frac{\hat{\sigma}_{\hat{f}}^2(\mathbf{x}_0)}{M},$$

where H_m and $\mathbf{h}_{0,m}$ are the analogous quantity to H and $\mathbf{h}_0$ for the m^{th} model.

3.2 Estimator for the heteroskedastic case

For non-constant noise variance, the diagonal matrix Σ need to be estimated. In MLR, this quantity is sometimes estimated by the diagonal matrix $\hat{\Sigma}_m$ defined by $(\hat{\Sigma}_m)_{ii} = r^2_{i,m}/(1-(H_m H_m^\dagger)_{ii})^2$, where H_m is the fixed design matrix [5]. In our case, each model provides an estimate $\hat{\Sigma}_m$ which is injected in the expectation estimation of the first term of equation (2). For the expectation estimation of the second term of equation (2) which involves pairs of models, Σ is estimated with $\hat{\Sigma}_{m,l} = (\hat{\Sigma}_m + \hat{\Sigma}_l)/2$. Rearranging this using the transpose operator on scalars, the proposed model variance estimation becomes

$$\hat{\sigma}^2(\mathbf{x}_0) = \frac{1}{M^2} \sum_{m,l=1}^{M} \mathbf{h}_{0,m}^T H_m^\dagger \hat{\Sigma}_m H_l^{\dagger T} \mathbf{h}_{0,l} \ + \frac{\hat{\sigma}^2_{\hat{f}}(\mathbf{x}_0)}{M}.$$

4 Synthetic experiments

The sigmoid function will be used as an activation function for all experiments. Experiment A is a simple one-dimensional simulated case study of $n = 60$ training points. Input probability density is the trapeze shape defined by

$$\rho(x) = -\frac{x}{4\pi^2} + \frac{3}{4\pi}, \quad \text{if } x \in [0, 2\pi],$$

$\rho(x) = 0$ otherwise. Outputs are generated with a Gaussian noise according to

$$y = \sin(x) + \varepsilon(x), \quad \text{with} \quad \sigma^2_\varepsilon(x) = 0.1.$$

The homoskedastic estimate is computed with $N = 6$ and $M = 20$. This is repeated $1'000$ times with fixed input. An example of prediction with twice standard deviation estimate is displayed in Figure 1a). Notice that the true $f(x)$ lies within $]\hat{f}(x) \pm 2\hat{\sigma}(x)[$, although this does not define a confidence interval — as the distribution of $\hat{f}(x) - f(x)$ is unknown.

In order to evaluate our method, $10'000$ ensembles with $M = 20$ and $N = 6$ are trained with new outputs. The variance of the $10'000$ ensembles provides a reliable baseline and will be used as a ground truth for the model variance. Figure 1b) shows twice standard deviation around the mean of model variance estimates and compares it with the ground truth. In average, the proposed method recovers effectively the variance from the $10'000$ simulations base line. The increasing variance in the borders due to the side effect of the modelling is fairly replicated. The uncertainty due to the trapezoidal shape of the input data distribution is also captured. Qualitatively, all aspects of the expected variance behaviour are globally reproduced. To assess quantitatively each estimation, we follow [6]. Let us define $se_k = \text{median}_i \left(\hat{\sigma}_k(\mathbf{x}_i)\right)$, $e_k = \text{median}_i |\hat{\sigma}_k(\mathbf{x}_i) - \sigma(\mathbf{x}_i)|$ and $re_k = \text{median}_i \frac{|\hat{\sigma}_k(\mathbf{x}_i) - \sigma(\mathbf{x}_i)|}{\sigma(\mathbf{x}_i)}$, which are respectively the median, the absolute error and the relative error of the k^{th} standard deviation estimate over the training set, for $k = 1, \ldots, 1'000$. Similar measures are defined on a random testing

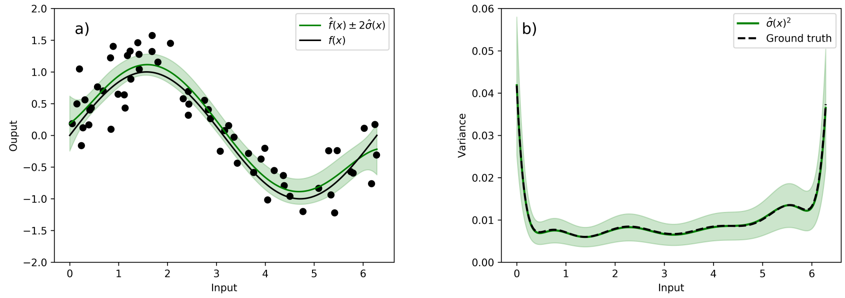

Fig. 1: Experiment A : (a) A single estimation; (b) Model variance estimates.

	Exp. A			Exp. B			Exp. C		
Training set	Mean	Std dev.	Grnd tr.	Mean	Std dev.	Grnd tr.	Mean	Std dev.	Grnd tr.
se_k	0.088	0.009	0.089	0.265	0.008	0.244	0.428	0.016	0.386
e_k	0.007	0.005	–	0.021	0.008	–	0.049	0.009	–
re_k	0.077	0.058	–	0.087	0.034	–	0.130	0.026	–
Testing set	Mean	Std dev.	Grnd tr.	Mean	Std dev.	Grnd tr.	Mean	Std dev.	Grnd tr.
se_k	0.087	0.008	0.088	0.284	0.009	0.261	0.458	0.017	0.403
e_k	0.007	0.005	–	0.023	0.009	–	0.053	0.012	–
re_k	0.077	0.059	–	0.088	0.034	–	0.136	0.033	–

Table 1: Results of the synthetic experiments.

set of $5'000$ points. In order to compute these quantities, $\sigma(\mathbf{x}_i)$ is estimated with the ground truth. The means and standard deviations of se_k, e_k and re_k over the 1'000 experiment repetitions are presented in Table 1. For the training set, the ground truth is recovered by se_k and the mean and standard deviation of e_k appear quite small. The mean of re_k shows that, on average, the median error at training points represents 7.7% of the true standard deviation. This percentage is inflated by the fact that we look at the standard deviation estimate and not at the variance estimate. The results on the 5000 testing points are similar, which shows that the estimation is good both at testing and training points.

Experiment B used the multivariate dataset described by Friedman in [7], with fixed inputs $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ drawn independently from uniform distribution on the interval $[0, 1]$ and outputs generated with a Gaussian noise according to

$$y(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon(\mathbf{x}), \quad \sigma_\varepsilon^2(\mathbf{x}) = 0.5.$$

We draw $n = 500$ training points. The homoskedastic estimate with $M = 20$ and $N = 92$ is repeated $1'000$ times while ground truth is recomputed. Results are reported in Table 1. The training se_k tends to slightly overestimate the true standard deviation median over the training points. The testing se_k has an analogous behaviour. Although the testing se_k tends to be greater than the training se_k, the testing e_k and re_k are similar to the training e_k and re_k, showing again that the estimation works at testing points as well as at the training points. The training (testing) re_k is 8.7% (8.8%), which is consistent with the previous univariate case, considering the re_k standard deviations.

Experiment C is the same as experiment B with a non-constant noise variance inspired from [8] , $\sigma_\varepsilon^2(\mathbf{x}) = 0.5\left(1 + \sin(0.8\pi \cdot ||\mathbf{x}|| - 0.6\pi)\right)^2$, where $||\cdot||$ denotes the Euclidean norm. The heteroskedastic estimate is computed $1'000$ times with $n = 500$, $M = 20$, $N = 73$ and the results are shown in Table 1. Again, the training (testing) se_k tends to somewhat overestimate the true training (testing) standard deviation median, and the training absolute (relative) errors are similar to the testing ones. Although the testing re_k reach 13.6%, it is still satisfying considering the use of the heteroskedasic variance estimate.

5 Conclusion

As ELM can be seen as a linear regression in a random feature space, it was possible to derive analytical results by conditioning model uncertainty quantities on the random input weights and biases, yielding probabilistic formulas. In particular, the model variance, knowing input data, has been decomposed into three terms, supporting the identification and the interpretation of the contribution of the different variability sources. Based on these formulas, estimations for the model variance were provided for homoskedastic and heteroskedasic cases, partly inspired from MLR theory. These results were confirmed by numerical experiments.

References

[1] G.-B. Huang, Q.-Y. Zhu, and C-K Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 985–990 vol.2, July 2004.

[2] E. Soria-Olivas, J. Gomez-Sanchis, J. D. Martin, J. Vila-Frances, M. Martinez, J. R. Magdalena, and A. J. Serrano. Belm: Bayesian extreme learning machine. *IEEE Transactions on Neural Networks*, 22(3):505–509, March 2011.

[3] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong. Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Transactions on Power Systems*, 29(3):1033–1044, May 2014.

[4] Anton Akusok, Yoan Miche, Kaj-Mikael Björk, and Amaury Lendasse. Per-sample prediction intervals for extreme learning machines. *International Journal of Machine Learning and Cybernetics*, 10(5):991–1001, May 2019.

[5] A. C. Davison. *Statistical Models*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2003.

[6] R. Tibshirani. A comparison of some error estimates for neural network models. *Neural Computation*, 8(1):152–163, 1996.

[7] J. H. Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19(1):1–67, 03 1991.

[8] D. A. Nix and A. S. Weigend. Learning local error bars for nonlinear regression. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 489–496. MIT Press, 1995.