The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/112420

Please be advised that this information was generated on 2021-06-06 and may be subject to change.
Response to “Comment on a recent sensitivity analysis of radial base function and multi-layer feed-forward neural network models”

E.P.P.A. Derks a, *, M.S. Sánchez Pastor b, L.M.C. Buydens a

a Laboratory for Analytical Chemistry, Faculty of Science, Catholic University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

b Department of Mathematics and Computation, Faculty of Science and Food Technology and Chemistry, University of Burgos, Apdo. 231, 09080 Burgos, Spain

Abstract

In our paper [1], the modeling capabilities of multi-layered feed-forward (MLF) and radial base function (RBF) networks were investigated on simulated data and well described experimental data from chemical industry [4]. Since both networks are based on a different concept (that is, RBF in contrast to MLF shows more local modeling behaviour) both modeling capability and robustness to input errors have been examined. The ‘robustness’ was expressed in terms of sensitivity of the network output units to random input perturbations by means of controlled pseudo-random noise. In this response paper, the comment of Faber et al., i.e., applying theoretical error propagation on artificial neural networks, and the consequences for the conclusions drawn in the original paper [1], are addressed.

Keywords: Response to Comment; Sensitivity analysis; Radial base function; Multi-layered feed-forward; Neural networks

In the comment of Faber et al. [5], theoretical error propagation is proposed by linearizing the network model around the training data by means of a truncated Taylor series expansion. As is suggested, this approach has been successfully undertaken for various methods [2,3] in cases where the experimental errors are sufficiently small.

Faber et al. compare the univariate error propagation formula directly to the Monte Carlo approach undertaken in Ref. [1] and focus on the interpretation of the ordinary least squares (OLS) parameters (β0 and β1).

In Ref. [1], it is outlined that the β1 parameter yields information about sensitivity. Next to that, the offset parameter β0 indicates a strong resemblance to the ‘pure errors’ obtained from replicates [4]. In the conventional form (without the logarithmic transformation), it is allowed to describe the sensitivity by means of a nonlinear regression of the output variance against the noise control factor Ω. The logarithmic transform of the output variance has been applied in order to linearize the regression problem.

However, as correctly noticed by Faber et al., the logarithmic transformation changes the meaning of the offset and slope parameters β0 and β1, respectively. In the comment, it is pointed out that in this particular case, the offset parameter contains the sensitivity information where the slope parameter re-
mains constant. Unfortunately, the comment of Faber et al. restricts to the univariate case where a multivariate problem is considered. We regret that the comment only implies validity of error propagation on neural networks and actually lacks all proof, since no extensions of the univariate case are provided.

In order to emphasize some of the problems we encountered, some simple extensions of error propagation to neural networks are given. For convenience, the same structure and nomenclature of the first paper [1] are used.

For a feed-forward neural network containing three layers with \( K \) linear input units (flow-through units), \( L \) hidden units containing a tangens hyperbolicus activation function and \( M \) linear output units, the general network Equation can be used

\[
y_m = \sum_{l=1}^{L} V_{lm} \Psi(\text{net}_l)
\]

where the tangens hyperbolicus activation function is denoted by \( \Psi \) and the input of the \( l \)-th hidden unit is given by

\[
\text{net}_l = \sum_{k=1}^{K} W_{kl} \cdot x_k
\]

Linearizing Eq. (1) by means of a first-order Taylor expansion around \( (x_o^o, y_o^o) \) yields

\[
\Delta y_m = \hat{y}_m - y_m^o
\approx \sum_{l=1}^{L} \left( \nabla \Psi(\text{net}_l) \sum_{k=1}^{K} (W_{kl} \cdot \Delta x_k) \right) \cdot \hat{x}_o^o
\]

where \( \hat{y}_k \) is the estimate of the linearized network and \( \nabla \Psi(\text{net}_l) \) represents the first derivative of hidden unit \( l \) for input object \( x_o^o \). Consequently, the variance of network output \( y_m \) can be derived by squaring Eq. (3), yielding

\[
\sigma_{\Delta y_m}^2 = \left( \Delta y_m \right)^2
\approx \sum_{l=1}^{L} \left( \nabla \Psi(\text{net}_l) \hat{x}_o^o \sum_{k=1}^{K} \left( \left( V_{lm} \cdot W_{kl} \right)^2 \cdot \Delta x_k^2 \right) \right)
\]

Note that Eq. (4) is only valid when no correlations in input space and no or weak correlations in the activated hidden units exist.

Given the simulated experimental uncertainty, \( \Delta x_k = \Omega \sigma \) and the weight contribution per hidden unit \( A_{lm} = \sum_k V_{lm} \cdot W_{kl} \), one finds

\[
\sigma_{\Delta y_m}^2 = \sum_{l=1}^{L} \left( \left( \nabla \Psi(\text{net}_l) \right)^2 \cdot \hat{x}_o^o A_{lm} \right) \cdot \Omega^2 \sigma^2
\approx \sum_{l=1}^{L} \left( (1 - \tanh^2(\text{net}_l)) \hat{x}_o^o \cdot A_{lm} \right) \cdot \Omega^2 \sigma^2
\]

In [1] the robustness of MLF and RBF models to random input perturbations has been studied by Monte Carlo simulations. The experimental data were perturbed with artificial noise, controlled by the parameter \( \Omega (\hat{\sigma}_k = \Omega \sigma_k) \). The sensitivity of the output units with respect to \( \Omega \) was determined by means of ordinary least squares regression of the logarithm of the output variance against the logarithm of \( \Omega \),

\[
\log(\sigma_y^2) = \beta_0 + \beta_1 \cdot \log(\Omega)
\]

Using the square root and the logarithm of both sides of Eq. (5)

\[
\log(\sigma_y) = \log\left( \sum_l \left(1 - \tanh^2(\text{net}_l)\right) \cdot \Omega |A_{lm}| \sigma \right) \hat{x}_o^o
\]

Comparing this to Eq. (6), it follows that

\[
\beta_0 = \log\left( \sum_l \left(1 - \tanh^2(\text{net}_l)\right) \cdot |A_{lm}| \sigma \right) \hat{x}_o^o
\]

\[
\beta_1 = 1
\]

As is discussed above, this derivation is totally based on the rather unrealistic assumptions of independency and normality of (sufficiently small) errors. Since neural networks trained by the generalized delta learning rule do not take these properties into account, the validity of Eq. (8) remains questionable.

However, if the assumptions are met, the conclusion based on the results summarized in Table 3 of Ref. [1] must be interpreted from the proposed point of view (i.e. \( \beta_0 \) comprises the sensitivity information), and it must be concluded that in contrast to the first conclusion, the MLF model outperforms the RBF.
model, considering the required number of parameters and the sensitivity properties of the networks. Naturally, these conclusions strongly depend on the type of problem and data at hand.

The authors object to the suggestion that the standard error propagation theory can generally be applied to neural networks in the suggested straightforward way. In the comment of Faber et al., some critical aspects of applying error propagation (i.e. normality, independence of error sources, size of errors, etc.) remain unaddressed. Our experience indicates that error propagation in neural networks is a delicate issue. Especially, in presence of large errors, strong violations on the error propagation constraints are frequently encountered, making the method proposed invalid. Moreover, it is also assumed that the first derivative of the nonlinear function remains constant over a sufficiently large neighborhood of $x'_0$. It needs to be stressed here, that also the location in the input space needs to be considered, since heteroscedasticity is not uncommon in practice.

Furthermore, hesitation towards the first order approach remains due to the strong assumptions which still have to be considered. No attention has been paid to the common situation that both the network input as the hidden units are correlated causing serious problems for error propagation. Additionally, the strong assumptions of normality of the (sufficiently small) errors causes extra violations on the theory used. An interesting approach to tackle this problem is outlined in Section 21.9 of the book of Efron et al. [2], where nonparametric error propagation is described.

The authors agree that error propagation is a promising method for the quantification of the confidence intervals of neural network predictions and progressing work at our laboratory within this field of chemometrical research, is going on.

**Acknowledgements**

The authors want to thank Dr. W.J. Melssen, for the elaborations and discussions about theoretical error propagation theory for neural network models.

**References**