Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-2 (2020) 1108–1113

Asymptotic Prediction Error Variance for Feedforward Neural Networks

Magnus Malmström * **Isaac Skog** * **Daniel Axehill** *
Fredrik Gustafsson *

* *Dept. of Electrical Engineering, Linköping University, Sweden,*
(firstname.lastname@liu.se)

Abstract: The prediction uncertainty of a neural network is considered from a classical system identification point of view. To know this uncertainty is extremely important when using a network in decision and feedback applications. The asymptotic covariance of the internal parameters in the network due to noise in the observed dependent variables (output) and model class mismatch, i.e., the true system cannot be exactly described by the model class, is first surveyed. This is then applied to the prediction step of the network to get a closed form expression for the asymptotic, in training data information, prediction variance. Another interpretation of this expression is as the non-asymptotic Cramér-Rao Lower Bound. To approximate this expression, only the gradients and residuals, already computed in the gradient descent algorithms commonly used to train neural networks, are needed. Using a toy example, it is illustrated how the uncertainty in the output of a neural network can be estimated.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/4.0)

Keywords: Neural Networks, Feedforward Networks, Uncertainty, System Identification, Estimation Theory, Cramér-Rao Bound, Identification for Control, Machine Learning

1. INTRODUCTION

Despite the huge success of neural networks (NNs) to solve very hard regression and classification problems, their applications in safety critical applications, such as autonomous vehicles, are limited due to their lack of ability to assess the *uncertainty* of the computed predictions, see e.g., NTSB (2018). Modern cars and early demonstrated autonomous cars have an abundance of sensor information. Even if one sensor modality is uncertain about what it observes, there are often alternative ones to support the decisions or to close the feedback loops. This holds, given that the system knows that the currently used modality is uncertain about what it observes. The knowledge of the relative uncertainty of different sensor modalities is also the cornerstone of sensor fusion theory, Gustafsson (2018).

The subject of assessing uncertainty in NN has recently received increased attention Ghahramani (2015); Kendall and Gal (2017); Garnelo et al. (2018); Kendall and Cipolla (2016). Bayesian neutral networks (BNN), which by design provide uncertainty measures, were proposed already in the 1990's Neal (1996). However, they were known to be computational intensive and did not scale well with network size. Recently, some more efficient implementations have been proposed Gustafsson et al. (2019); Blundell et al. (2015); Kendall and Cipolla (2016). Along with these implementation methods, the notion of aleatoric and epistemic uncertainty has been coined to separate the effects of uncertainty in the training data[1] (aleatoric) and in the model (epistemic).

Most implementations of the BNN rely on creating ensembles of NNs, from which the network parameters can be sampled. A recent trend is to use dropouts or batchnorm to create those ensembles, e.g., Gal and Ghahramani (2016); Teye et al. (2018). In simple terms, dropouts involve randomly disabling nodes in a NN, which provides an ensemble of models. In this way, Monte Carlo (MC) like generation of outputs can be obtained, from which the uncertainty in the predictions can be assessed. The advantage of these methods is that they rely on existing NN structures and are easy to implement. The downside is that multiple forward passes are required to create the MC like simulations needed to obtain the uncertainty. Another suggested approach is to include the variance as a component in the loss function for the NN, and hence, the NN learns its own uncertainty, e.g., Eldesokey et al. (2018); Kendall and Gal (2017). However, this increases the size of the NN and thus the computational complexity.

Yet another approach is to estimate the uncertainty in the parameters of the NN from the training phase, and then let this uncertainty be propagated to the output uncertainty in the prediction step. Several publications, He and Li (2011); Papadopoulos et al. (2001); Hwang and Ding (1997) have proposed this as a means to get a variance expression for the prediction. Recently, similar ideas have been used to investigate what impact a single training example might have on the outcome in the testing phase, e.g., Koh and Liang (2017).

The contribution in this work continues along similar lines, though our analysis is asymptotic in the number of training data and non-asymptotic in the network size. The latter assumption implies that there will be a model mismatch, even when the number of training data tends to infinity, since the true system is typically not in the model class. The contribution to the uncertainty then has two sources: the stochastic uncertainty in the training data and the distance between the true system and the selected model class. Using a toy example, it is illustrated how the different sources contribute to the uncertainty in the predictions of the network in regions both with and without training data, and how the proposed method in

[1] Generally, when referring to uncertainties in the training data only uncertainties in the dependent variables are considered and the independent variables are assumed to be perfectly known.

2405-8963 Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2020.12.1310

an efficient way can be used to calculate the prediction uncertainty.

2. PROBLEM FORMULATION

Consider the following static nonlinear signal model

$$z_n = s_n + e_n \tag{1a}$$

$$s_n = f^*(\mathbf{x}_n), \tag{1b}$$

where the function $f^*(\mathbf{x}_n)$ describes the relation between the input (independent variable) $\mathbf{x}_n$ and output (dependent variable) s_n of the true system under consideration. Further, e_n is the observation noise which is identically and independently distributed according to some distribution. Given a set of training data consisting of the observations $z_{1:N} \triangleq \{z_n\}_{n=1}^N$ and inputs $\mathbf{x}_{1:N} \triangleq \{\mathbf{x}_n\}_{n=1}^N$, a parametric model of the form

$$s_n = f(\mathbf{x}_n, \boldsymbol{\theta}), \tag{2}$$

with parameter vector $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d$ containing d parameters is fitted to the data by minimizing a least squares cost function, a.k.a. learning the parameters. That is, the parameters $\hat{\boldsymbol{\theta}}_N$ are estimated as

$$\hat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta}), \tag{3}$$

where

$$V_N(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_{n=1}^{N} ||z_n - f(\mathbf{x}_n, \boldsymbol{\theta})||^2. \tag{4}$$

The parametric model together with the estimated parameters are then used to predict new outputs $\hat{s} = f(\mathbf{x}, \hat{\boldsymbol{\theta}}_N)$ of the system given some input $\mathbf{x}$; from hereon the sample index n will be omitted for brevity. The aim of this paper is to analyze how the prediction error

$$\varepsilon(\mathbf{x}, \hat{\boldsymbol{\theta}}_N) \triangleq z - \hat{s} = f^*(\mathbf{x}) + e - f(\mathbf{x}, \hat{\boldsymbol{\theta}}_N), \tag{5}$$

depends on the uncertainty in the training data (aleatoric uncertainties) and in the model (epistemic uncertainties), as well as to provide an analytic expression that can be used to calculate a lower bound on the prediction uncertainty. Though the presented analysis is valid for a generic parametric model, specific focus will be directed towards the commonly used feedforward NN model.

In a feedforward network, the model $s = f(\mathbf{x}, \boldsymbol{\theta})$ can be described by the recursions

$$\mathbf{h}^{(0)} = \mathbf{x} \tag{6a}$$

$$\mathbf{h}^{(l+1)} = \sigma\big(\mathbf{W}^{(l)}\left[\mathbf{h}^{(l)}\ 1\right]^\top\big), \quad l = 0, \cdots, L-1 \tag{6b}$$

$$s = \mathbf{W}^{(L)}\left[\mathbf{h}^{(L)}\ 1\right]^\top. \tag{6c}$$

Here L is the number of layers in the network and $\mathbf{W}^{(l)}$ is the weights of the l:th layer. Hence, the model parameters

$$\boldsymbol{\theta} = \left[\mathrm{vec}(\mathbf{W}^{(0)})^\top \cdots \mathrm{vec}(\mathbf{W}^{(L)})^\top\right]^\top. \tag{7}$$

Furthermore $\sigma(\cdot)$ is the activation function operating element-wise. Two of the most commonly used activation functions are the sigmoid function $\sigma(u) = 1/(1+e^{-u})$ and the rectified linear unit $\sigma(u) = \max\{u, 0\}$.

Due to the symmetries in the NN model structure and the activation function, as well as possible overparameterizations, in terms of nodes needed to describe the true input-output relationship, the choice of parameters in the model is not unique; see Hwang and Ding (1997) for details. Hence, the parameter estimate $\hat{\boldsymbol{\theta}}_N$ is non-unique and will depend on factors such as the training data realization, as well as the initial condition and choice of optimization algorithm and its implementation. Let the set of parameter vectors that minimize the cost function (4) be defined as

$$\mathcal{S}_{\hat{\boldsymbol{\theta}}_N} \triangleq \{\hat{\boldsymbol{\theta}}_N \in \Theta : \hat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta})\}. \tag{8}$$

Next, it will be studied which sources that contribute to the uncertainties in the estimated parameters and how they affect the prediction accuracy. To pursue our analysis some assumptions on the smoothness of $f(\mathbf{x}, \boldsymbol{\theta})$ is required. To that end, it will be assumed that the function $f(\mathbf{x}, \boldsymbol{\theta})$ is at least three times continuously differentiable.

3. PREDICTION UNCERTAINTY

The purpose of this section is to quantify different sources of uncertainty and to get an explicit expression for the contribution of each source to the total uncertainty. Depending on if the true model belongs to the chosen model class $\mathcal{M}$ or not, a number of different sources contributing to the prediction uncertainty, will be considered. If $f^* \in \mathcal{M}$ there exists a non-empty parameter set

$$\mathcal{S}_{\boldsymbol{\theta}^o} \triangleq \{\boldsymbol{\theta}^o \in \Theta : f(\mathbf{x}, \boldsymbol{\theta}^o) = f^*(\mathbf{x})\ \forall \mathbf{x}\} \tag{9}$$

of parameter vectors where $\boldsymbol{\theta}^{io} \in \mathcal{S}_{\boldsymbol{\theta}^o}$ is the ith parameter vector such that the NN model describes the input-output relationship perfectly. Otherwise, if $f^* \notin \mathcal{M}$, we know from Ljung (1999) that the parameter estimate converges to a vector $\boldsymbol{\theta}^{i*}$ that gives the best fit of the training data in the least squares sense. That is, $\boldsymbol{\theta}^{i*} \in \mathcal{S}_{\boldsymbol{\theta}^*}$, where

$$\mathcal{S}_{\boldsymbol{\theta}^*} \triangleq \{\boldsymbol{\theta}^* \in \Theta : \boldsymbol{\theta}^* = \lim_{N\to\infty} \arg\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta})\}. \tag{10}$$

If $f^* \in \mathcal{M}$ then $\mathcal{S}_{\boldsymbol{\theta}^*} \equiv \mathcal{S}_{\boldsymbol{\theta}^o}$ and the parameter estimate converges to one vector in this set $\mathcal{S}_{\boldsymbol{\theta}^o}$. This is under the assumption that the data used to estimate the model is informative enough to make the model observable.

To be able to proceed with the uncertainty analysis and handle the fact that the choice of parameters in the NN is in general non-unique, a canonical parametric model $f_c : \mathbb{R} \times \mathbb{R}^{d_c} \to \mathbb{R}$ with a corresponding canonical, i.e., unique and irreducible, parameter vector $\boldsymbol{\theta}^c \in \mathbb{R}^{d_c}$, where $d_c \leq d$, is introduced. For the canonical parametric model it holds that given any $\boldsymbol{\theta}$ there exists a unique $\boldsymbol{\theta}^c$ such that

$$f(\mathbf{x}_n, \boldsymbol{\theta}) = f_c(\mathbf{x}_n, \boldsymbol{\theta}^c), \quad \forall \mathbf{x}_n. \tag{11}$$

Hence, $f_c \in \mathcal{M}$ is able to represent any input-output relation that $f \in \mathcal{M}$ is able to, but the parameterization is assumed unique and potentially of lower dimension. Furthermore, assume that there exists k differentiable mappings T_i , $i = 1, .., k$, relating any parameter vector $\boldsymbol{\theta}^i$ in the original model and the corresponding one $\boldsymbol{\theta}^c$ in the canonical model such that $\boldsymbol{\theta}^i = T_i(\boldsymbol{\theta}^c)$. See Hwang and Ding (1997) for an example of how such a mapping can be constructed in a two-layer NN with sigmoid activation functions. Noteworthy is that the, somewhat abstract, function T_i is only needed for the forthcoming analysis and not, as shown in later parts of the paper, for the application of the analytic results.

3.1 Different Sources of Uncertainty

Assume that the estimated (learned) model parameter vector is $\hat{\boldsymbol{\theta}}_N^i \in \mathcal{S}_{\hat{\boldsymbol{\theta}}_N}$ and the corresponding asymptotic (in the number of training data samples N) parameter estimate is $\boldsymbol{\theta}^{i*} \in \mathcal{S}_{\boldsymbol{\theta}^*}$. The prediction error can then be decomposed as

$$\varepsilon(\mathbf{x},\hat{\boldsymbol{\theta}}^i_N) = e + \underbrace{f^*(\mathbf{x}) - f(\mathbf{x},\boldsymbol{\theta}^{i*})}_{\text{Model error}} + \underbrace{f(\mathbf{x},\boldsymbol{\theta}^{i*}) - f(\mathbf{x},\hat{\boldsymbol{\theta}}^i_N)}_{\text{Estimation error}}. \tag{12}$$

Next, let $\hat{\boldsymbol{\theta}}^c_N$ and $\boldsymbol{\theta}^{c*}$ denote the canonical representation of $\hat{\boldsymbol{\theta}}^i_N$ and $\boldsymbol{\theta}^{i*}$, respectively. Then the parameter estimate can be further decomposed as

$$\begin{aligned}\hat{\boldsymbol{\theta}}^i_N &= T_i(\hat{\boldsymbol{\theta}}^c_N)\\ &= T_i(\boldsymbol{\theta}^{c*}) + \underbrace{T_i(\hat{\boldsymbol{\theta}}^c_N) - T_i(\boldsymbol{\theta}^{c*}) - T_i(\boldsymbol{\theta}^{cb})}_{\text{Stochastic error}} + \underbrace{T_i(\boldsymbol{\theta}^{cb})}_{\text{Bias error}}\end{aligned} \tag{13}$$

where $\boldsymbol{\theta}^{cb}$ is the bias in the parameters due to the model error. Thus, the total uncertainty in the prediction has several components with different origins, and the following observations can be made:

- If $f^* \in \mathcal{M}$. Even if we were given one of the feasible parameter vectors $\boldsymbol{\theta}^{io} \in \mathcal{S}_{\boldsymbol{\theta}^o}$ of the true system by an oracle we would still have an estimation error $\varepsilon(\mathbf{x},\boldsymbol{\theta}^{io}) = e$. The observation noise e can of course not be predicted.
- If the parameter estimation error is sufficiently small, the prediction uncertainty that originates from the error in the parameter estimate can be found via first order Taylor expansion

$$\begin{aligned}\hat{s} &= f(\mathbf{x},\hat{\boldsymbol{\theta}}^i_N) = f(\mathbf{x},T_i(\hat{\boldsymbol{\theta}}^c_N))\\ &\approx f(\mathbf{x},T_i(\boldsymbol{\theta}^{co})) + f'_{\boldsymbol{\theta}}(\mathbf{x},T_i(\boldsymbol{\theta}^{co}))\big(T_i(\hat{\boldsymbol{\theta}}^c_N) - T_i(\boldsymbol{\theta}^{co})\big),\end{aligned} \tag{14}$$

  from which it follows that

$$\operatorname{Var}(\hat{s}) \approx f'_{\boldsymbol{\theta}}(\mathbf{x},T_i(\boldsymbol{\theta}^{co}))^\top \operatorname{Cov}(\hat{\boldsymbol{\theta}}^i_N) f'_{\boldsymbol{\theta}}(\mathbf{x},T_i(\boldsymbol{\theta}^{co})). \tag{15}$$

  Here $f'_{\boldsymbol{\theta}}(\mathbf{x},T_i(\boldsymbol{\theta}^{co})) \triangleq \frac{\partial}{\partial\boldsymbol{\theta}} f(\mathbf{x},\boldsymbol{\theta})|_{\boldsymbol{\theta}=T_i(\boldsymbol{\theta}^{co})}$ is the Jacobian of the parametric model w.r.t. the parameter vector $\boldsymbol{\theta}$ evaluated some at $\boldsymbol{\theta}^{io} = T_i(\boldsymbol{\theta}^{co})$. Furthermore, $\operatorname{Cov}(\hat{\boldsymbol{\theta}}^i_N)$ is the covariance matrix of the parameter estimate.
- If $f^* \notin \mathcal{M}$, which is the typical case in practice, we have to replace $\boldsymbol{\theta}^{co}$ in (14) and (15) with $\boldsymbol{\theta}^{c*}$. This will introduce a systematic deterministic uncertainty in the estimate. In contrast to the estimation error, that decays with the size of the training data set, this model error is independent of the training data size.

3.2 True System Not in Model Class

This section reviews the results from Ljung (1999); Ljung and Caines (1980), that asymptotically the random variable $\sqrt{N}(\hat{\boldsymbol{\theta}}^c_N - \boldsymbol{\theta}^{c*})$ will be Gaussian distributed under weak assumptions. We will begin with the general case where the true system is not necessarily in the model class, $f^* \notin \mathcal{M}$, and then derive the other one as a special case.

First, assume that $\boldsymbol{\theta}^{i*}$ is an interior point of Θ. Define the gradient of the prediction error as

$$\begin{aligned}\psi(\mathbf{x},\boldsymbol{\theta}^{c*}) &\triangleq -\frac{\partial}{\partial\boldsymbol{\theta}}\varepsilon(\mathbf{x},T_i(\boldsymbol{\theta}))\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{c*}}\\ &= \frac{\partial}{\partial\boldsymbol{\theta}} f(\mathbf{x},T_i(\boldsymbol{\theta}))\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{c*}}.\end{aligned} \tag{16}$$

Then, we have

$$-V'_N(\boldsymbol{\theta}^{c*}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \psi(\mathbf{x}_n,\boldsymbol{\theta}^{c*})\varepsilon(\mathbf{x}_n,T_i(\boldsymbol{\theta}^{c*})) \tag{17}$$

For stochastic variables $g(t)$, the asymptotic mean is defined as

$$\bar{E}[g(t)] \triangleq \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[g(t)]$$

where $E[\mathbf{x}]$ is the mathematical expectation operator. Define

$$\bar{V}(\boldsymbol{\theta}) \triangleq \bar{E}[\varepsilon^2(\mathbf{x},T_i(\boldsymbol{\theta}))]. \tag{18}$$

In Ljung and Caines (1980) it is shown that (17) can be written as a sum of an asymptotically normal distributed random variable and

$$\begin{aligned}D_N = E\Bigg[\frac{1}{N}\sum_{n=1}^{N}&\psi(\mathbf{x}_n,\boldsymbol{\theta}^{c*})\varepsilon(\mathbf{x}_n,T_i(\boldsymbol{\theta}^{c*}))\\ &- \bar{E}[\psi(\mathbf{x},\boldsymbol{\theta}^{c*})\varepsilon(\mathbf{x},T_i(\boldsymbol{\theta}^{c*}))]\Bigg].\end{aligned} \tag{19}$$

From the definition of $\boldsymbol{\theta}^{c*}$ we have

$$\bar{V}'(\boldsymbol{\theta}^{c*}) = -\bar{E}[\psi(\mathbf{x},\boldsymbol{\theta}^{c*})\varepsilon(\mathbf{x},T_i(\boldsymbol{\theta}^{c*}))] = 0, \tag{20}$$

which gives us that

$$\sqrt{N}D_N \to 0, \text{ as } N \to \infty. \tag{21}$$

Hence, we can conclude that (17) is asymptotically normal distributed.

As a consequence of the definition of the canonical parametrization, there exists a unique $T_i(\boldsymbol{\theta}^{c*}) \in \mathcal{S}_{\boldsymbol{\theta}^*}$ and $\hat{\boldsymbol{\theta}}^c_N \to \boldsymbol{\theta}^{c*}$ with probability 1 as $N \to \infty$. From assumptions on the function $f(\mathbf{x},\boldsymbol{\theta})$, the second derivative of the asymptotic cost function

$$\bar{V}''(\boldsymbol{\theta}^{c*}) \triangleq \frac{\partial^2}{\partial\boldsymbol{\theta}^2}\bar{V}(T_i(\boldsymbol{\theta}))\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{c*}} \tag{22}$$

is positive definite. Then, we have the final result

$$\sqrt{N}(\hat{\boldsymbol{\theta}}^c_N - \boldsymbol{\theta}^{c*}) \to \mathcal{N}(0, P_{\boldsymbol{\theta}^c}(\boldsymbol{\theta}^{c*})) \quad \text{as } N \to \infty \tag{23}$$

where

$$P_{\boldsymbol{\theta}^c}(\boldsymbol{\theta}^{c*}) = [\bar{V}''(\boldsymbol{\theta}^{c*})]^{-1} Q [\bar{V}''(\boldsymbol{\theta}^{c*})]^{-1} \tag{24a}$$

$$Q = \lim_{N\to\infty} N E\{[V'_N(\boldsymbol{\theta}^{c*})][V'_N(\boldsymbol{\theta}^{c*})]^\top\}. \tag{24b}$$

3.3 True System in Model Class

If $f^* \in \mathcal{M}$, and $T_i(\boldsymbol{\theta}^{co}) \in \mathcal{S}_{\boldsymbol{\theta}^o}$, then we have that the prediction error $\varepsilon(\mathbf{x},T_i(\boldsymbol{\theta}^{co})) = e$ has zero mean and variance λ_0 by assumption. A first consequence is that $D_N = 0$. Furthermore, the covariance in (24a) can be greatly simplified by noting that $Q = \lambda_0\bar{V}''(\boldsymbol{\theta}^{co})$, and as a result, (24a) simplifies to

$$P_{\boldsymbol{\theta}^c}(\boldsymbol{\theta}^{co}) = \lambda_0\mathcal{I}^{-1}_{\boldsymbol{\theta}^c} \tag{25a}$$

$$\mathcal{I}_{\boldsymbol{\theta}^c} = \bar{E}[\psi(\mathbf{x},\boldsymbol{\theta}^{co})\psi^\top(\mathbf{x},\boldsymbol{\theta}^{co})] \tag{25b}$$

where if e is Gaussian distributed, $\mathcal{I}_{\boldsymbol{\theta}^c}$ is the Fisher information matrix for $\boldsymbol{\theta}^c$.

In practice, one way to make it highly likely that the true system is included in the model class is to choose a very rich model class, then use regularization to avoid overfitting the model to the data. L^2-regularisation is one example of a regularisation that introduces an explicit cost for using models with many parameters, i.e., adding the l_2-norm of the parameters to the loss function. This, would imply that a scaled identity matrix is added to the Fisher information $\mathcal{I}_{\boldsymbol{\theta}^c}$. It is in particular necessary to use regularization if the number of parameters exceeds the number of training data samples. One downside with regularisation is that it will introduce a bias in the estimated parameters, which in turn will propagate to a bias of the model. This in turn would be equivalent to adding additional knowledge about the true system in terms of model order and smoothness of the function $f^*(\mathbf{x})$.

3.4 Prediction Variance

As stated before in (15), the covariance of the model parameters can be propagated to the covariance of the model output. This is done using the first moment of a Taylor expansion of the model around the true network parameters. That is, the prediction error of the NN model is given as

$$\operatorname{Var}(\hat{s}) \approx \frac{1}{N}\psi^\top(\mathbf{x}, \boldsymbol{\theta}^{co})P_{\boldsymbol{\theta}^c}(\boldsymbol{\theta}^{co})\psi(\mathbf{x}, \boldsymbol{\theta}^{co}) \tag{26}$$

Using that

$$\psi(\mathbf{x}, \boldsymbol{\theta}^{co}) = -T_i'(\boldsymbol{\theta}^{co})f_{\boldsymbol{\theta}}'(\mathbf{x}, T_i(\boldsymbol{\theta}^{co})), \tag{27a}$$

$$P_{\boldsymbol{\theta}^c}(\boldsymbol{\theta}^{co}) = \lambda_0\left[T_i'(\boldsymbol{\theta}^{co})\mathcal{I}_{\boldsymbol{\theta}}(T_i'(\boldsymbol{\theta}^{co}))^\top\right]^{-1}, \tag{27b}$$

$$\mathcal{I}_{\boldsymbol{\theta}} = \bar{E}[f_{\boldsymbol{\theta}}'(\mathbf{x}, T_i(\boldsymbol{\theta}^{co}))(f_{\boldsymbol{\theta}}'(\mathbf{x}, T_i(\boldsymbol{\theta}^{co})))^\top] \tag{27c}$$

where

$$T_i'(\boldsymbol{\theta}^{co}) \triangleq \left.\frac{\partial T_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{co}}. \tag{27d}$$

In order make (27b) well-defined in general, the inverse in (27b) can be replaced by the Moore- Penrose inverse. Then the prediction error in (26) can be rewritten as

$$\operatorname{Var}(\hat{s}) \approx \frac{1}{N}(f_{\boldsymbol{\theta}}'(\mathbf{x}, T_i(\boldsymbol{\theta}^{co})))^\top P_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{io})f_{\boldsymbol{\theta}}'(\mathbf{x}, T_i(\boldsymbol{\theta}^{co})) \tag{27e}$$

$$P_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{io}) = \lambda_0\mathcal{I}_{\boldsymbol{\theta}}^+ \tag{27f}$$

where $+$ denotes the Moore-Penrose inverse defined by the singular value decomposition. By assumption $\operatorname{rank}(\mathcal{I}_{\boldsymbol{\theta}}) = d_c$, hence usage of the Moore-Penrose inverse makes it possible to omit the dependences from the derivative of the mapping $T_i(\boldsymbol{\theta})$ with respect of the parameters $\boldsymbol{\theta}$ in (27e). The Moore-Penrose inverse is still well-defined even though there exist linear combinations of parameters without or with low excitation in $\mathcal{I}_{\boldsymbol{\theta}}$, i.e., with zero singular values. This to prevent infinite variance in (27e), when the parameter uncertainty is propagated to uncertainty in the output.

3.5 Relation to previous work

The expression (26) has been presented in related publications Hwang and Ding (1997); Rivals and Personnaz (2000); He and Li (2011); Papadopoulos et al. (2001); Chryssoloiuris et al. (1996), but we would like to point out some important remarks in this work. Hwang and Ding (1997) generate $n+1$ data points where n points are used for training the model and predict the $n+1$'th sample with a confidence interval. By repeating the experiment multiple times and counting the number of times the confidence interval covers the true point, the analytic expression of the variance is motivated. This can be compared to our experiments where the confidence interval is calculated once followed by multiple realisations of NNs were prediction is done over a grid. Rivals and Personnaz (2000) compare the analytic expression to variance calculated from multiple realisations of NNs, but use the true parameters $\boldsymbol{\theta}^{co}$ compared to the estimate $\hat{\boldsymbol{\theta}}_N^i$, used in our simulation in Sec. 5. He and Li (2011); Papadopoulos et al. (2001), use simulated data and results are presented over multiple realisations of NNs, but the analytic expressions are calculated over an ensemble of NNs, while Chryssoloiuris et al. (1996) evaluate the estimated variance on real data. Furthermore, how the estimate of the variance is effected in the case where the NN extrapolate to areas were no data was available during training is not considered.

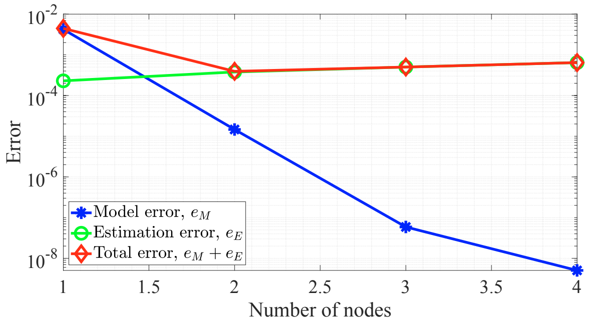


Fig. 1. Total error, model error, and estimation error, as a function of the numbers of nodes in the NN.

Another difference is that these papers state that the output is Student-t distributed. This would have been true, if the model would have been linear. Furthermore, the degrees of freedom in the Student-t distribution equals the number of training data minus the number of parameters. If the number of parameters exceeds the available number of training data, which is the case in some applications of NN, then the degrees of freedoms in the student-t distribution is negative, which we find hard to interpret. So far, our analysis has been asymptotically in number of training data. Since, asymptotically when the number of training data N increases, the Student-t distribution converges to a Gaussian distribution and hence we expect similar results as the cited papers in regions where training data is available.

3.6 Approximating the Variance Expressions

If the parameters $\boldsymbol{\theta}^{co}$ and λ_0 are unknown, $P_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{io})$ and λ_0 in (27f) can be approximated with data by

$$\hat{P}_N = \hat{\lambda}_N\left[\frac{1}{N}\sum_{n=1}^{N} f_{\boldsymbol{\theta}}'(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N^i)(f_{\boldsymbol{\theta}}'(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N^i))^\top\right]^+, \tag{28a}$$

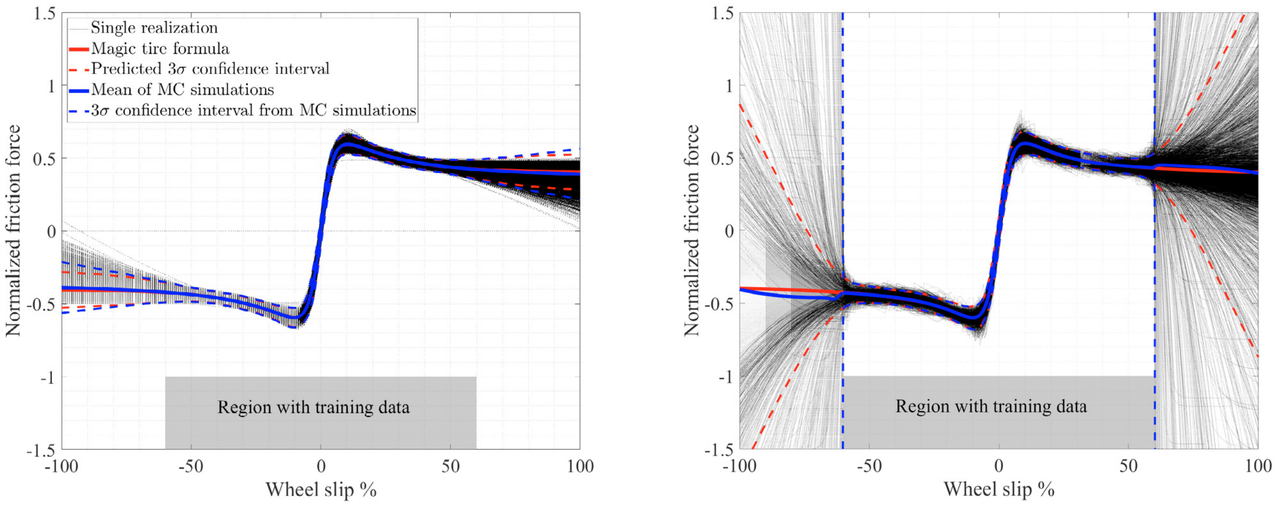
$$\hat{\lambda}_N = \frac{1}{N}\sum_{n=1}^{N}\varepsilon^2(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N^i) \tag{28b}$$

where $T_i(\boldsymbol{\theta}^{co})$ is approximated by $\hat{\boldsymbol{\theta}}_N^i$. This gives us that $\operatorname{Cov}(\hat{\boldsymbol{\theta}}_N^i)$ in (15) can be approximated by $\hat{P}_N/N$. Asymptotically, these expressions converge to the true values.

The central components to compute for the variance of the NN model is the derivative of the model $f_{\boldsymbol{\theta}}'(\mathbf{x}, \hat{\boldsymbol{\theta}}_N^i)$ given in (16) and the error of the predictor $\varepsilon(\mathbf{x}, \hat{\boldsymbol{\theta}}_N^i)$ given in (5). If a gradient-based optimization method is used to minimize the cost function $V_N(\boldsymbol{\theta})$, the covariance of the parameter estimate can be obtained for more or less free during the training of the NN model.

4. CRAMÉR-RAO BOUND

The previous results presented in this paper are asymptotic in the number of points in the training data. In practice, there is nothing as infinite number of data points, hence, results from finite number of data points are required.



(a) Prediction error of NN with 2 nodes.

(b) Prediction error of NN with 3 nodes.

Fig. 2. Analytically and empirically calculated prediction uncertainty for a two layer NN with 4 and 6 nodes, respectively.

From classic statistic theory and system identification literature, e.g., Kay (1993); Ljung (1999); Liero and Zwanzig (2011), we know that if $f^* \in \mathcal{M}$, any unbiased estimator $\hat{\boldsymbol{\theta}}_N^i$ has a lower bound on the stochastic uncertainty given by

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}_N^i) \succeq \frac{1}{N} P_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{io}), \tag{29}$$

where $\succeq$ denotes that the difference between the left-hand side and the right-hand side is positive semidefinite. This bound is referred to as the Cramér-Rao lower bound (CRLB). In particular, the parameter estimate for a NN has a lower bound on the stochastic uncertainty that can be approximated by (29) with the right hand side approximated with $\hat{P}_N$ given in (28) for a large but finite N.

If the noise in (1a) is Gaussian distributed, the central limit theorem can be used to show that the parameters estimated by (3) is the maximum likelihood (ML) estimate with its variance given by CRLB in (25a).

5. NUMERICAL ILLUSTRATION

The simulation example is inspired by autonomous car applications, where the drive-line is a part of the model for the vehicular dynamics control systems. We focus on the tire-road friction, which is a critical component in any advanced driver-assistance system (ADAS). For instance, the friction level influences the safety distance in an adaptive cruse controller (ACC), which avoidance manoeuvres that are most effective, and vehicular dynamics control systems such as the ABS. If the NN is intended to be used in any of these cases, the NN should also be able to approximate the friction level from wheel slip measurements. The simulation data is generated from a parametric model called the *magic tire formula* [2] presented in Pacejka and Besselink (1997). The model describing how the normalized friction force s depends on the wheel slip $\mathbf{x}$ of a tire

Using the simulated data from the magic tire formula, i.e., the true output model $f^*(\mathbf{x}_n)$, four different two-layer $\mathrm{NN}_{2,l}$s using sigmoids as activation functions were trained with $l = 1, ..., 4$ number of nodes in the hidden layer. In order to separate the estimation error from the model error, the simulated output from the trained $\mathrm{NN}_{2,l}$, $l = 1, ..., 4$, referred to as $f(\mathbf{x}_n, \boldsymbol{\theta}^{i*})$, with added noise was used to train 10 000 networks, $f(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N^{i(m)})$, $m = 1, ..., 10^4$, in an MC-like procedure. In the simulations, the signal to noise ratio, (SNR) is of 20 dB, where the observation noise e is Gaussian distributed with known variance $\lambda = 0.01$. For training of the NN, $N = 200$ samples were used; both for the reference network and the MC-simulation.

In Fig. 1, the model error e_M and the estimation error e_E calculated as

$$e_M = \frac{1}{N}\sum_{n=1}^{N} \left\| f^*(\mathbf{x}_n) - f(\mathbf{x}_n, \boldsymbol{\theta}^{i*}) \right\|^2 \tag{30a}$$

$$e_E = \frac{1}{NM}\sum_{m=1}^{M}\sum_{n=1}^{N} \| f(\mathbf{x}_n, \boldsymbol{\theta}^{i*}) - f(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N^{i(m)}) \|^2 \tag{30b}$$

are shown. As expected, and as seen from the figure, the model error decreases with the model order and the estimation error increases. Already in a NN with 2 nodes the model error is a magnitude smaller than the estimation error, and increasing the model order further will not significantly contribute in decreasing the prediction error.

The prediction and the MC realisations for a $\mathrm{NN}_{2,2}$ and $\mathrm{NN}_{2,3}$ are plotted with a 3σ confidence interval, see Fig. 2a and Fig. 2b, respectively. The interval is calculated by taking the variance of the MC realisations (in dashed blue) and using (27e) with $P_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{io}) = \hat{P}_N$ (in dashed red), i.e., a lower bound on the variance. The region with training data is indicated by a grey box.

As one would expect, the uncertainty is relatively small in the region with training data and grows in regions away from where training data was collected. The mean of the MC simulations and the simulated output from the magic tire formula coincide in the region containing training data, but they start to diverge from each other the further away from the region with training data we get. This is also true for the 3σ confidence interval; in the region with training data the confidence interval calculated from the MC realisations and the one calculated using (26) coincide while they do not in the region without training data.

[2] $s = D\sin\{C\arctan[(1-E)\mathbf{x} + E/B\arctan(B\mathbf{x})]\}$, in this article $B = 14.00$, $C = 1.60$, $D = 0.60$, and $E=$ -0.20.

Since already NN$_{2,2}$ can represent the output from the magic tire formula almost perfectly one can conclude that the larger NN$_{2,3}$ is an overparameterized model. In the region without training data, this is the reason why the variance of the MC simulations grows much faster in Fig. 2a compared to Fig. 2b. This is also true for the variance estimate computed using (27e), but for the overparameterized model the linearization, (27e), fails to fully capture the flexibility given by the extra parameter in regions without training data. In reality, one seldom knows true model structure, hence overparameterised models such as NN$_{2,3}$ are more common. Then, a large confidence interval in regions without training data is essential to indicate uncertainty in the model there. Overparametrization can thus be an instrument to reveal when the model extrapolates in regions that lack training data.

6. CONCLUSION

In this work, a method to calculate a confidence interval of the prediction of a feedforward NN using classical statistical theory and system identification theory has been outlined. As a result, the work strengthens the link between machine learning and system identification, and shows how results from system identification can enable the use of machine learning methods in safety-critical applications. It is shown in numerical experiments that in a region where there is training data available, the suggested approach of calculating a confidence interval for a NN coincides with the variance obtained from MC simulations. This is illustrated by a toy example, in which it can be ensured that the true system is accurately approximated within to the model class. Furthermore, the toy example also illustrates that in regions with no training data available, the uncertainty of the predicted output increases as expected. Moreover, the 3σ confidence interval from the MC simulations diverges from the lower bound calculated by the suggested approach, i.e., the computed variance for the predicted output for an overparameterized NN cannot be accurately computed with the suggested approach in regions with no training data available. Our future research will focus on extending the analysis to the case when different types of regularizations are used, and create bounds for the prediction error in regions without training data given assumptions regarding the smoothness of the true system-input-output function.

ACKNOWLEDGEMENTS

This work has been performed with the support from the Swedish Governmental Agency for Innovation Systems within the framework of the project *iQDeep - Machine learning for Autonomous Heavy Vehicles* (2018-02700).

REFERENCES

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. In *Proc. of the 32Nd Int. Conf. on Mach. Learn. (ICML).*, 1613–1622. Lille, France. 6–11 Jul.

Chryssoloiuris, G., Lee, M., and Ramsey, A. (1996). Confidence interval prediction for neural network models. *IEEE Trans. Neural Netw.*

Eldesokey, A., Felsberg, M., and Khan, F.S. (2018). Propagating confidences through cnns for sparse data regression. In *British Mach. Vision Conf. (BMVC)*, 14. Newcastle, UK, Sep 3-6.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proc. of the 33td Int. Conf. on Mach. Learn. (ICML).*, 1050–1059. New York, NY, USA. 20–22 Jun.

Garnelo, M., Schwatz, J., Rosenbaum, D., Viola, F., Rezende, D., Eslami, S.M.A., and Teh, Y.W. (2018). Neural processes. In *Proc. of the 35th Int. Conf. on Mach. Learn. (ICML) Workshop on Theo. Founda. and Appl. of Deep Generative Models*. Stockholm, Sweden. URL `https://arxiv.org/abs/1807.01622`. 10–15 Jul.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553), 452 – 459.

Gustafsson, F. (2018). *Statistical sensor fusion.* Studentlitteratur: Lund Sweden.

Gustafsson, F.K., Danelljan, M., and Schön, T.B. (2019). Evaluating scalable bayesian deep learning methods for robust computer vision. In *Adv. in Neural Inf. Process. Syst. (NIPS) 33.* Vancouver, Canada.

He, S. and Li, J. (2011). Confidence intervals for neural networks and applications to modeling engineering materials. In *Artificial Neural Netw.*, chapter 16. IntechOpen.

Hwang, J.T.G. and Ding, A.A. (1997). Prediction intervals for artificial neural networks. *J. Am. Stat. Assoc. (JSTOR).*

Kay, S.M. (1993). *Fundamentals of statistical signal processing Estimation theory.* Prentice Hall PTR, cop. 1993: Upper Saddle River, NJ, USA.

Kendall, A. and Cipolla, R. (2016). Modelling uncertainty in deep learning for camera relocalization. In *IEEE Int. Conf. on Robot. and Autom. (ICRA)*, 4762–4769. Stockholm, Sweden. 16–21, May.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Adv. in Neural Inf. Process. Syst. (NIPS) 30*, 5574–5584. Curran Associates, Inc. Long Beach, CA, USA, 4–9 Dec.

Koh, P.W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proc. of the 34th Int. Conf. on Mach. Learn. (ICML).*, 1885–1894. Sydney, Australia. 06–11 Aug.

Liero, H. and Zwanzig, S. (2011). Introduction to the theory of statistical inference. In *Chapman and Hall CRC Texts in Statistical Science.*

Ljung, L. (1999). *System identification: theory for the user.* PTR Prentice Hall: Upper Saddle River, NJ, USA.

Ljung, L. and Caines, P.E. (1980). Asymptotic normality of prediction error estimators for approximate system models. *Stochastics*, 3.

Neal, R.M. (1996). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media: New York, NY, USA.

NTSB (2018). Preliminary Report Highway HWY18MH010. Technical specification (ts), National Transportation Safty Board (NTSB). URL `https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf`.

Pacejka, H. and Besselink, I. (1997). Magic formula tyre model with transient properties. *Veh. syst. dynamics-Int. J. of Veh. Mechanics and Mobility*, 27(S1), 234–249.

Papadopoulos, G., Edwards, P., and Murray, A. (2001). Confidence estimation methods for neural networks: a practical comparison. *IEEE Trans. Neural Netw.*

Rivals, I. and Personnaz, L. (2000). Construction of confidence intervals for neural networks based on least squares estimation. *Elsevier J. Neural Netw.*, 13.

Teye, M., Azizpour, H., and Smith, K. (2018). Bayesian uncertainty estimation for batch normalized deep networks. In *Proc. of the 35th Int. Conf. on Mach. Learn. (ICML).* Stockholm, Sweden,6–11 Jul.