Developing Optimal Neural Network Metamodels Based on Prediction Intervals
Abbas Khosravi, IEEE member, Saeid Nahavandi, IEEE Senior member, Doug Creighton

Abstract—Finding optimal structures for neural networks remains an open problem, despite the rich array of literature on the application of neural networks in different areas of science and engineering. The stochastic nature of operations common in complex systems makes point prediction performance of neural network metamodels an additional challenge. We propose a method for selecting the best structure of a neural network metamodel. For selecting the network structure, the new method uses interval prediction capability of neural networks and chooses a topology that yields the narrowest prediction band for targets. This is an improvement on traditional criteria, such as mean square error or mean absolute percentage error.

As a case study, the interval prediction method is applied to a metamodel of a complex system composed of many inextricably interconnected entities and stochastic processes. The demonstrated results expressively show that selecting the network structure based on the proposed method yields more reliable estimates.

I. INTRODUCTION

As systems grow larger and subsystem processes more interdependent their modelling and analysis is no longer straightforward. Due to the high level of complexity, traditional pen and paper-based approaches for modelling these systems are inefficient. To satisfy industry demand for accurate performance estimates, simulation models have been developed for describing the operation and behaviour of systems such as manufacturing enterprises, airports, and material handling systems. Discrete event simulation is an appropriate tool in such instances. However, despite the value of this technique and similar alternatives, they suffer from a couple of vexing issues in different stages of their life cycle. A major shortcoming of these models is the expense of development. Also, their computational requirements significantly restrict their real time application for operational planning and optimisation.

Metamodelling has been suggested for addressing these drawbacks in the literature. By definition, a metamodel is an approximate model describing the relationships between a set of inputs and outputs [1]. Metamodelling, coupled with detailed simulation models, may be developed to partially represent behaviour of an underlying system. At the cost of reduced accuracy, metamodelling significantly reduces computation mass required for generating outputs. The time required for running a simulation metamodel is almost negligible in comparison to a discrete event simulation replicates. Such a reduction in requirements paves the way for developing and implementing optimisation procedures for complex systems.

The most frequently used methodologies for metamodelling are regression techniques [1]. They are simple to develop and exploit, but suffer from lack of accuracy in cases of high system complexity and nonlinearity. Splines, radial basis functions, Kriging, and support vector machine are other methods proposed in literature as more powerful metamodelling. Clark et al. [2] compared applications of these methods for metamodelling of selected engineering problems.

Another metamodelling approach is Neural Networks (NN). The last two decades have witnessed proliferation of NN applications in various fields, including science and engineering. One of the areas that NNs have been widely used is metamodelling of complex systems, such as commercial and industrial enterprises. While NNs are sometimes used for prediction purposes [3], [4], [5], [6], [7], they can also be adopted for optimisation [8], [9], [10]. Studies completed in recent years have shown that they often outperform their traditional counterparts (e.g., regression methods) in terms of both accuracy and capability of learning nonlinear mechanisms governing the underlying systems [7] [11].

In spite of the widespread application of NNs for metamodelling, there are two issues that require further investigation. The first one, which is indeed a general problem in all NN applications, concerns the network structure. The number of hidden layers and the number of neurons per layer has a dramatic influence on the performance of NNs. Therefore, ample care must be exercised in their selection. The second issue pertains to the suitability of NN metamodels for point predictions of targets. Like other metamodelling, performance of NNs may drop due to the stochastic nature of many processes within a complex system. As this degradation is high for some outputs, potential usability of NNs as metamodelling becomes questionable.

This study addresses these two issues through adopting a technique for identifying NN metamodels with an optimal structure. A NN metamodel is used to construct prediction
Numerical results and experiments are demonstrated and two aforementioned problematic issues in more detail.

This paper is organised as follows. Section 2 discusses the two aforementioned problematic issues in more detail. Techniques for calculating prediction intervals for NN estimation are discussed in Section 3. Section 4 presents the proposed method for developing optimal metamodels. Numerical results and experiments are demonstrated and discussed in Section 5. Finally, Section 6 lists the conclusion.

II. NEURAL NETWORK FOR METAMODELLING

In this section, we discuss in more detail some problematic issues related to the application of NN metamodels for point prediction of complex system outputs. The presented discussion explains the background and motivations for conducting this research.

A. NN Structure Selection

Although NN training and weight adjustment has an established theoretical foundation [12], no well developed theory or widely accepted rule of thumb for selection of NN structure exists. The selection of an appropriate NN structure is the first step in constructing a NN model. The structure of a feedforward NN is characterised by the number of hidden layers and the number of neurons in those layers. The definition directly affects the NN learning capability and prediction performance.

Several techniques for finding the optimum of these quantities have been proposed in literature. In the late 1980s and early 1990s, researchers tried to find upper and lower bounds for the number of neurons [13]. Their results are applicable to networks with one hidden layer and cannot be widely applied to real world problems. Generally, one can divide the associated techniques into two broad categories [14]:

1. Construction: starting with either a small or a large network, neurons or layers are removed or added until some prespecified criteria are met. Pruning or weight decay approaches fall into this category. The most frequently used criteria are Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE).

2. Evolutionary Techniques: through optimisation, quantities related to network topology are found. Genetic algorithms are often used for this purpose.

Attik et al. [15] provided another classification of structure selection, which is similar to the one described. As none of these techniques can guarantee satisfactory results, determining a suitable or possible optimal structure for a NN has been labeled a ‘black art’ [16]. This is why the time consuming ‘trial and error’ method plays an important role in identifying the appropriate topology.

A common practice employed by NN developers is to vary the number of neurons and hidden layers and pick up the network that yields the most accurate results, e.g., minimum MSE or MAPE for training/test data, to reduce computational load.

B. NN Metamodel Suitability for Point Prediction

NN implementation for point prediction compromises the majority of studies completed in the area of metamodelling. Researchers often adopt NNs to estimate point values of outputs (targets) given a set of inputs [3], [4], [5], [6], [7], [8], [9], [10], [11]. Although the obtained results in these studies are acceptable and at times promising, it does not follow that using NNs for point prediction always yield acceptable results.

A real threat for performance of NN metamodels is uncertainties prevailing in the systems. Many operations and processes are stochastic in complex systems, such as manufacturing enterprises. Events may or may not occur, even if operational conditions remain constant during system operation. As there can be more than one reality for system outputs given fixed inputs, achieving good performance in point prediction can be problematic.

The mechanisms governing the operation of complex systems are often dynamic. Internal dependencies amongst constituent entities may change based on conditions. This dynamic nature and also the presence of noise in measurements degrades the performance of NN metamodels (and in general any other type of metamodel) for point prediction. Due to all these issues, the validity of point predictions and decisions made based on them are in doubt.

Some researchers have observed the weakness of NNs for deterministic modelling of stochastic systems. Aware of the translation defect, they have tried to avoid problems by finding the likelihood that a stochastic system will produce outputs that lie within an a priori specified interval. They have actually turned their focus from point prediction to prediction intervals. Work by Kilmer et al. [17] is an early contribution in this area. This problem is also discussed in [11] without any practical solution provided. More recently, Zobel et al. [18] suggested construction of confidence intervals (based on the method introduced in [19]) for NN predictions. Calculated for each test sample, confidence intervals quantify precisely the uncertainty associated with the underlying system and network prediction.

The literature is not complete in this realm and prediction interval performance requires more attention from both academia and industry. Nevertheless, metamodellers of stochastic systems can potentially derive much benefit from use of prediction and confidence intervals, an area with many issues unarticulated.

III. PREDICTION INTERVALS FOR NN OUTPUTS

Constructing prediction intervals for NN outputs is the main thesis of the proposed technique discussed in the next section. Therefore, it is worth to concisely review common techniques for calculating them.

Firstly, there is a need to distinguish between confidence intervals and prediction intervals. When one trains a NN, two types of uncertainty affect the final outcome: uncertainty in model parameters (due to error in specification of network parameter values), and uncertainty in the training data (mainly due to measurement noise). We
refer to these as \( \sigma_m^2 \) and \( \sigma_v^2 \) respectively. With the assumption of independence of these two types of uncertainty, the total prediction variance will be: 
\[
\sigma_{total}^2 = \sigma_m^2 + \sigma_v^2.
\]
While confidence intervals define how well a model describes the true regression (\( \sigma_m^2 \) is a matter of importance), prediction intervals are about network capability for generating real outputs (therefore, \( \sigma_{total} \) becomes the main player). Based on this discussion, prediction intervals are wider than confidence intervals and cover them. As approaches for calculating these two quantities are similar, their mathematical discussion is often presented together. We refer interested readers to [20] and [21] for further discussion.

There are three schools of thought for creating prediction/confidence intervals for NNs [22]: (a) the delta method, which is based on a Taylor expansion of the regression function [23] [24]; (b) the bootstrap technique, which is essentially a resampling method [25]; and (c) the Bayesian approach, based on Bayesian statistics [26]. Each method has its own pros and cons. Generally, there is no strong evidence that one method outperforms another in terms of accuracy of calculated boundaries. Amongst influential factors that must be considered when selecting any of these techniques are the problem domain, computation burden, number of available samples, and analysis purpose.

The focus of this paper is on the delta method for interval construction. The method is based on representing and interpreting feedforward neural networks as nonlinear regression models. This allows the application of standard asymptotic theory for constructing prediction intervals. For the first order Taylor expansion of regression model of NNs, an approximate prediction interval with 100(1 - \( \alpha \))% confidence can be calculated as follows:
\[
\hat{y}_i \pm t_{n-p}^{1-\alpha/2} s \sqrt{1 + f_0^T (FF')^{-1} f_0}
\]  
(1)

where \( \hat{y}_i \), \( s \), \( f_0^T \), and \( F \) are the network point prediction, the standard deviation estimate, the gradient of the i-th output against the network parameters (weights and biases), and the Jacobian matrix of neural network outputs with respect to its parameters, respectively. Mathematically, these terms are calculated as follows:
\[
f_0^T = \begin{bmatrix}
\frac{\partial f(X_i, \hat{\theta})}{\partial \theta_1} & \frac{\partial f(X_i, \hat{\theta})}{\partial \theta_2} & \cdots & \frac{\partial f(X_i, \hat{\theta})}{\partial \theta_p}
\end{bmatrix}
\]
\[
F = \begin{bmatrix}
\frac{\partial f(X_1, \hat{\theta})}{\partial \theta_1} & \frac{\partial f(X_1, \hat{\theta})}{\partial \theta_2} & \cdots & \frac{\partial f(X_1, \hat{\theta})}{\partial \theta_p} \\
\frac{\partial f(X_2, \hat{\theta})}{\partial \theta_1} & \frac{\partial f(X_2, \hat{\theta})}{\partial \theta_2} & \cdots & \frac{\partial f(X_2, \hat{\theta})}{\partial \theta_p} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f(X_n, \hat{\theta})}{\partial \theta_1} & \frac{\partial f(X_n, \hat{\theta})}{\partial \theta_2} & \cdots & \frac{\partial f(X_n, \hat{\theta})}{\partial \theta_p}
\end{bmatrix}
\]
\[
s = \frac{1}{n-p} \sum_{i=1}^{n} (y_i - f(X_i, \hat{\theta}))^2
\]
\[
t_{n-p}^{1-\alpha/2}
\]  is also the inverse of the Student t-distribution function with \( n - p \) degrees of freedom, evaluated at \( 1 - \frac{\alpha}{2} \).

IV. PROPOSED TECHNIQUE

As discussed in Section 2, there is no general rule or reliable criterion for determining the number of hidden layers and the number of neurons per layer. In this section, a method for determining these two critical quantities is proposed based on prediction intervals constructed for test data.

Essentially, the proposed method is a type of construction approach for finding optimal quantities of network layers and neurons. While in the literature, the optimal network is found using a trial and error approach, the proposed method wisely uses specific features of t-distribution for confining...
the search space. Prediction intervals calculated using (1) depend on the t-distribution and its critical values. The number of network parameters (weights and biases) has a direct effect on degrees of freedom of the t-distribution (degrees of freedom are differences between number of samples and number of network parameters), and therefore contributes to the critical values of t-distribution. Evolution of the t-distribution along its two parameters (degrees of freedom and \( \alpha \)) has specific features.

Fig. 1 displays the inverse of t-cumulative distribution function versus its degrees of freedom for different levels of confidence. This plot indicates that as degrees of freedom decreases, the critical values of t-distribution rise quickly. Such a growth significantly widens prediction intervals computed using (1). Therefore, any action that may decrease degrees of freedom will yield poor results in term of narrowness of prediction intervals. This interesting feature significantly restricts the number of neurons and hidden layers for exploration. This rule of thumb is used here to find the optimal structure for NN metamodels.

Using this rule, lower and upper bounds for the number of neurons in each layer are computed. This results in an \( L \) dimensional grid (\( L \) is the number of layers), where the points carry information regarding the networks’ structure. Grid point values determine number of neurons in each corresponding layer. For each point within this grid, a neural network is constructed and trained using training samples. Test samples are then projected to these networks for generating outputs. For each test sample and each network, prediction intervals are calculated based on (1). Traditionally researchers at this point use MSE or MAPE criteria for selecting the best neural network (and consequently determining the optimal number of layers and neurons per layer). Instead of using these measures, a visual inspection of computed intervals is employed for selecting the optimal network. The criteria for selecting the best networks are:

- the computed interval must cover the corresponding target;
- the prediction tube must be sufficiently narrow; and
- there must be no jump in the upper and lower bounds of each sample prediction interval. A wide prediction interval means that the test sample has been interpreted oddly by the network. Therefore, there is no reliability in the network prediction, even if error prediction is small and negligible.

In summary, the proposed method can be implemented in three steps: (a) determine potential networks based on number of samples and curves shown in Fig. 2; (b) construct and train neural network metamodels; and (c) choose the best network structure based on narrowness and coverage of constructed prediction intervals. Prediction intervals in the last step can be constructed in any level of significance that analysts wish.

A potentially narrow prediction interval with a high significance level is more desirable in practice than a point prediction without any indication of its accuracy. Such a preference arises from two interesting features of the proposed method. Firstly, it better addresses issues related to the accuracy of point prediction yielded by metamodels. Secondly, it solves the problem of finding the optimal structure for NN metamodels.

V. EXPERIMENTS AND SIMULATION RESULTS

In this section, the application of the proposed method for metamodeling of a complex material handling system is explained and results are demonstrated and discussed.

The underlying system in this study is a Baggage Handling System (BHS). Within a BHS, many machines and human operators can work both independently or collaboratively to accomplish some assigned tasks. Due to the security and safety issues, associated administrations have been charged to deploy 100% screening of all checked baggage for explosives. The rapid growth in the number of passengers in recent years has put excessive pressure on airports and their facilities, including BHSs. Therefore, precise modelling and analysis of the BHS is a vital task for optimisation and planning purposes. A highly detailed discrete event simulation model was developed based on real data from the underlying BHS. The Quest® software package from Delmia was used for 3D representation of the real BHS. As the final model is composed of many entities, including conveyors, buffers, screening machines, diverters, merges, laterals, makeup loops, check-in counters, controller and operators, its replicate for simulating one day operation of BHS with 272 flights takes more than 90 minutes. Such a computational burden makes if-then analysis time consuming and expensive. NN metamodeling is a potential alternative to alleviate the long time necessary to run this simulation model.

BHSs are characterised by their baggage screening capacity, which directly affects travel time of a bag from its origin (check-in counter) to its destination (lateral or makeup loop). The most relevant performance metric is time required for processing specific travel time percentiles of flight bag (e.g., 50%, 80%, or 90%). In this study, the 50% percentile has been considered as the important target. Hereafter and for the sake of ease in reference, this target is called P50. The experiment here involves constructing an optimal NN metamodel for constructing prediction intervals for P50. Variable selection for the NN was completed through preliminary replicates of the simulation model and consultation with experts in this domain. Flight type (economy or economy-business), check-in counter (six piers), exit lateral or makeup loop (in total 40), and Work-In-Progress (WIP) are variables recognised with significant impacts on P50.

To only describe and model steady state behaviour of the underlying BHS, some early and late flights were removed from data sets (220 samples in total). 75% (165) and 25% (55) of samples were considered as training and test datasets respectively. This means that \( n \) in (1) is equal to 165. To avoid overweighting of some variables over others in the training stage, a normalisation method is applied to all samples. As a practical requirement, samples are autoscaled...
using minimum and maximum of the training dataset (not overall minimum and maximum). For the purpose of comparison, MSE and MAPE are used as network assessment criteria. In all cases, prediction intervals for P50 are constructed with 90% confidence.

According to the $n$ value and curves plotted in Fig. 1, networks with more than around 130 parameters will not yield good results. Furthermore, as number of parameters rises quickly with networks with three layers, only networks with maximum two layers are considered in the analysis. First we start with a network with only one hidden layer. The maximum number of neurons for this type of network is 21 (a NN metamodel with 21 neurons in its hidden layer has 127 parameters). The number of neurons is varied from 1 to 21 and the performance of trained networks was determined using MSE, MAPE and a visual inspection of the computed prediction interval for each test sample. Obtained results showed that this type of network is not appropriate for either point prediction or prediction interval computation.

To remedy the weakness of NN metamodels with only one hidden layer, another hidden layer was considered in the NN structure. If we consider four inputs and one output for the NN metamodel, the number of metamodel parameters with $n_1$ and $n_2$ neurons in its first and second hidden layers is as follows:

$$5n_1 + n_1 n_2 + 2n_2 + 1$$ (3)

An upper bound for this value is 130. NN metamodels are constructed for those values of $n_1$ and $n_2$ that results in a network with less than 130 parameters. The total number of networks with such a feature is 194. Again MSE and MAPE performance criteria are used for assessing prediction capability of these networks. The results for different networks show that networks with 5 neurons in their second layer yield more acceptable predictions. According to (3), if the second layer has 5 neurons, the maximum number of neurons in the first layer will be 12. Fig. 2 displays MSE and MAPE for networks with 5 neurons in their second layer and between 1 to 12 neurons in their first layer. The minimum values for these two criteria have been determined by a circle and square in Fig. 2. NN metamodels with 1 and 4 neurons in their first layer are the best ones based on MAPE and MSE criteria respectively. Therefore, structures 1-5-1 and 4-5-1 are optimal based on the considered criteria.

Now the proposed method is employed for selecting the best network. For each NN metamodel and each test sample, prediction intervals are constructed based on (1) and (2). A network is called optimal if it best satisfies criteria explained in the previous section. Visualisation of prediction intervals shows that a network with three and five neurons in its first and second hidden layers yields the best results. The upper prediction bound, lower prediction bound, point prediction, and target value have been shown in Fig 3 for NN metamodels with the following structure: 1-5-1 (the best in term of MAPE), 4-5-1 (the best in term of MSE), and 2-5-1. Constructed prediction intervals using the last NN metamodel are narrower and better cover the outputs. Prediction intervals for the network with 1-5-1 structure are wide, which means that network reliability is low. Conversely, although prediction intervals for the network with 4-5-1 structure are narrow and appropriately cover targets, they are wide for some samples. Such a phenomena means that the underlying metamodel has not been able to recognise the test sample, and consequently, has yielded large upper and lower prediction bounds. In contrast, prediction intervals constructed using point prediction of the third metamodel are narrow. For the majority of cases, test targets lie within the intervals. This is an evidence of the metamodel prediction reliability.

It is also worthy to mention that this network has a good rank among other networks in term of MSE and MAPE criteria. According to the results shown in Fig. 2, its rank is 3 in both cases. This explicitly means that not only does this metamodel yield good point prediction (Fig.2), but also computed prediction intervals are the most reliable ones (Fig.3, bottom). Therefore, it is reasonable to select the NN metamodel with 2 neurons in the first layer and 5 neurons in the second layer optimal in structure.
The results demonstrated here can be obtained the time required for processing any percentile of flight bag travel time through the system for a given flight schedule. Through this method, both problems relating to accurate point prediction using NN metamodels are well handled, but also concerns about network structure and its effect on point predictions are addressed.

For cases studied in this paper, there was a considerable difference in range of prediction intervals. Therefore, comparison of prediction intervals through visualisation was straightforward and reliable. For those cases that prediction intervals constructed using different NN models are similar in size, a quantitative evaluation procedure must be developed. The mean of prediction interval lengths normalised by variation of targets is a quantitative measure of quality of prediction intervals.

VI. CONCLUSION AND FUTURE WORK

Although NN metamodeling is not a new idea, there are many issues that remain unarticulated. While some of these issues arise from difficulties in neural network construction (topology selection), others pertain to suitability of NN metamodels for point prediction (reliability problem). In this paper, we proposed a technique for addressing these issues through constructing prediction intervals for outputs of the underlying system. First, potentially suitable NN metamodels were determined based on degrees of freedom of the t-distribution. They were then constructed and assessed based on quality of prediction intervals for their point prediction. In the BHS case study, a highly detailed discrete event simulation model of a realistic complex system was used for conducting experiments and data collection. Within this system, many processes and operations are stochastic, which make point prediction a challenge. Demonstrated results showed that prediction intervals computed for each not-yet-seen sample are narrow and cover the corresponding target well. The optimal network structure was identified through assessing prediction intervals for all test samples. Furthermore, it was found that performance of the optimal NN metamodel is again acceptable based on traditional assessment criteria.

Further study may be conducted in two main area: (i) improving the quality of prediction intervals through better designing experiments for logging data or through taking into account density of samples in input space; and (ii) utilising the proposed technique for operational planning and scheduling.

In future, usefulness of the proposed technique will be rigorously justified through applying it to some datasets, which have been used in literature as benchmarks. Also, prediction interval quality will be quantified through adopting some techniques for measuring their length against the range of targets.

REFERENCES


