Pergamon

Neural Networks, Vol. 10, No. 4, pp. 599–614, 1997

Printed in Great Britain
0893–6080/97 $17.00+.00

PII: S0893-6080(96)00098-6

CONTRIBUTED ARTICLE

Optimal Linear Combinations of Neural Networks

SHERIF HASHEM

Pacific Northwest National Laboratory

(*Received* 20 *November* 1994; *revised and accepted* 23 *August* 1996)

Abstract—*Neural network-based modeling often involves trying multiple networks with different architectures and training parameters in order to achieve acceptable model accuracy. Typically, one of the trained networks is chosen as best, while the rest are discarded. Hashem and Schmeiser (1995) proposed using optimal linear combinations of a number of trained neural networks instead of using a single best network. Combining the trained networks may help integrate the knowledge acquired by the components networks and thus improve model accuracy. In this paper, we extend the idea of optimal linear combinations (OLCs) of neural networks and discuss issues related to the generalization ability of the combined model. We then present two algorithms for selecting the component networks for the combination to improve the generalization ability of OLCs. Our experimental results demonstrate significant improvements in model accuracy, as a result of using OLCs, compared to using the apparent best network. © 1997 Elsevier Science Ltd.*

Keywords—Optimal linear combination, Model selection, Function approximation, Collinearity, Robust estimation, Mixture of experts.

1. INTRODUCTION

In neural network (NN) based modeling, there are many network design and training parameters that need to be selected by practitioners. Design parameters include the number of layers, number of hidden units, type of activation functions, and type of connectivity. Training parameters include the initial connection-weights, the learning rates,the momentum term, and the weight decay term. The choice of the topological and training parameters significantly affect the accuracy of the resultant model as well as training time (Haykin, 1994; Kolen & Pollack, 1991; Lari-Najafi et al., 1989; Drago & Ridella, 1992). Parameter selection is often the result of rules of thumb combined with trial and error (Haykin, 1994; Zurada, 1992; Hansen & Salamon, 1990; Mani, 1991). At the end of the training process, a number of trained networks is produced, and then typically one of them is chosen as best, based on some optimality criterion, while the rest are discarded.

Hashem and Schmeiser (1995) proposed forming a linear combination of the corresponding outputs of the trained NNs, instead of just using the apparent best network. Combining the trained networks may help integrate the knowledge acquired by the component networks and often produces superior model accuracy compared to the single best-trained network (Hashem, 1993; Hashem & Schmeiser, 1993; Hashem et al., 1993, 1994). Optimal linear combinations (OLCs) of neural networks are constructed by forming weighted sums of the corresponding outputs of the networks. The combination-weights are selected to minimize the mean squared error (MSE) with respect to the distribution of the model inputs, a criterion commonly used by both the statistics and the neural network communities. The resultant optimal linear combinations are referred to as MSE-OLCs. Constructing MSE-OLCs is straightforward. Expressions for the (MSE-) optimal combination-weights are obtained in closed-form and require modest computational effort, mainly simple matrix manipulations. In practice, the optimal combination-weights need to be estimated from observed data. Because the component networks are trained to approximate the same physical quantity (or quantities), collinearity (linear dependency) among the corresponding outputs

Acknowledgements: This research was supported in part by PRF research grant 6901627 from Purdue University, the Associated Western Universities Inc. Northwest Division under grant DE-FG06-89ER-75522 with the U.S. Department of Energy, and by the Environmental Molecular Sciences Laboratory construction project at the Pacific Northwest National Laboratory

Requests for reprints should be sent to (present address), Department of Engineering Mathematics and Physics Faculty of Engineering, Cairo University, Egypt; Internet: shashem@idsc.gov.eg

and/or approximation errors of the component networks can sometimes undermine the robustness (generalization ability) of the estimated MSE-OLCs. In this paper. We introduce two algorithms for selecting the component networks in order to enhance the generalization ability (robustness) of the MSE-OLCs. We conduct an experimental study to examine the effectiveness of MSE-OLCs, guided by the selection algorithms, in improving model accuracy. Some important remarks regarding the discussions in this paper are:

- The class of neural networks investigated here is the class of multilayer feedforward networks. No further assumptions regarding the network architecture or the learning method are needed.
- This paper focuses mainly on function approximation or regression problems. However, MSE-OLCs are also applicable to supervised classification problems.
- To evaluate the effectiveness of an MSE-OLC, its performance is compared to the two most popular alternatives: the simple average (equal combination-weights) and the (apparent) best NN. While the former method, the simple average, does not require any data for estimation, the latter method, best NN, does. The NN that best fits the training data is not necessarily the "true" best among a number of trained networks. A common practice is to test the trained networks on a data set separate from the training data set in order to select the best performer.
- In the examples treated here, the true function is known. Thus, we measure accuracy in terms of the MSE with respect to the known function by integrating the (squared) error over the range of inputs using Monte Carlo integration (Sobol', 1974). The resultant MSE is referred to as the "true MSE".

The paper is divided into two parts:

In the first part, we briefly discuss related work in Section 2. In Sections 3 and 4, we formulate the optimal linear combination problem and present closed form expressions for the optimal combination-weights in four cases of MSE-OLCs. The estimation of the optimal combination-weights is discussed in Section 5.

In the second part, we study the robustness of the estimated MSE-OLCs and investigate the ill effects of collinearity in Section 6. We then propose two algorithms for improving the robustness of the MSE-OLC by the proper selection of the component networks in Section 7. In Section 8, we present experimental results to explore the merits of the MSE-OLCs and the effectiveness of the proposed algorithms in improving robustness. The conclusions are summarized in Section 9.

2. RELATED WORK

The literature on combining estimators is rich and diverse. Clemen (1989) cites more than 200 studies in his review of the literature related to combining forecasts, including contributions from the forecasting, psychology, statistics, and management science literature. He traces the idea of combining estimators back to Laplace (1818). Linear combinations of estimators have been studied and used by the statistics community for a long time (Granger, 1989). The simple average of a number of estimators has been frequently compared to the individual estimators, and in many cases improves the resultant model accuracy (Clemen, 1989; Granger, 1989).

More recently, the idea of combining multiple neural networks has been investigated by a number of researchers. Application areas included both regression and classification.

In the context of regression, Hashem and Schmeiser (1993, 1995) investigated combining a number of trained neural networks by forming weighted sums of the corresponding outputs of the component networks. Hashem and Schmeiser (1995) illustrated that using optimal linear combinations (OLCs) may significantly improve model accuracy. They studied two cases of OLCs, depending on whether or not the combination-weights are constrained to sum to one. Perrone (1993) and Perrone and Cooper (1993) independently developed the general ensemble method (GEM) for constructing improved regression estimates, which is equivalent to the constrained MSE-optimal linear combination in (Hashem, 1993; Hashem & Schmeiser, 1995). Jacobs and Jordan (1991, 1994) developed the hierarchical mixtures of experts (HME) architecture, which uses a divide-and-conquer approach for dividing a complex problem into simple problems that can be solved by separate expert networks. The final solution is obtained by combining the outputs of the expert networks using "gating networks". Unlike Jacobs and Jordan's mixtures of experts, the MSE-OLC approach assumes that the component networks are solving the same problem. Wolpert (1992) introduced "stacked generalization," a scheme for minimizing the generalization error of a number of generalizers by combining them. Breiman (1992) extended Wolpert's work to stacking regressions, which is a method for forming linear combinations of different predictors to improve prediction accuracy. The combination-weights are computed using a cross-validation approach that involves constructing an auxiliary set of predictors that are of the same structure as the original predictors. However, the auxiliary predictors are trained using leave-one-out cross-validation data. While stacking may improve prediction accuracy, obtaining the combination-weights can be very expensive computationally, especially when the predictors are neural networks that need to be re-trained on the cross-validation data.

In the context of classification problems, the idea of using an ensemble of trained networks instead of simply using the best NN has been investigated by a number of researchers. Hansen and Salamon (1990) suggested training a group of networks of the same architecture but

initialized with different connection-weights. Then, a screened subset of the trained networks is used for making the final classification decision by some voting scheme. Similar approaches have been introduced by Alpaydin (1993), and Battiti and Colla (1994). Cooper (1991) suggested constructing a multi-neural network system in which a number of networks independently compete to accomplish the required classification task. The multi-network system learns from experience which networks have the most effective separators and those networks determine the final classification. Mani (1991) suggested training a portfolio of networks, of possibly different topologies, using a variety of learning techniques. He also sketched an approach for lowering the variance of the decision of the networks using portfolio theory (Markowitz, 1952). Pearlmutter and Rosenfeld (1991) replicated networks, in which a number of identical networks are independently trained on the same data and their results are averaged. They concluded that replication almost always results in a decrease in the expected complexity of the network and increases expected generalization. The MSE-OLCs presented in this paper are compared to the simple average of the corresponding outputs of the trained networks. Benediktsson et al. (1993) proposed an architecture called the parallel consensual neural network that is based on statistical consensus theory. The architecture consists of several-stage networks whose outputs are combined to make a decision. In the field of medical diagnosis, Baxt (1992), trained two networks separately on data sampled from populations of different likelihoods in order to simplify the training and improve model accuracy. Parmanto et al. (1994), Rogova (1994), and Ohno-Machado and Musen (1994) used multiple neural networks in their models in order to improve classification performance. The rationale behind using multiple networks for classification is that, in the absence of the "true" model, one may improve on the apparent best NN by employing several networks to solve the classification task independently, and then construct a final vote by making use of the individual votes. The MSE-OLC approach presented in this paper is similar to these approaches in that it assumes that the component networks are trained to approximate the same physical quantity (or quantities). However, the combined output of an MSE-OLC is a weighted sum of the corresponding outputs of the component networks, where the combination-weights are estimated based on observed data.

3. LINEAR COMBINATIONS OF NEURAL NETWORKS

From a NN perspective, combining the corresponding outputs of a number of trained networks is similar to creating a large network in which the trained NNs are subnetworks operating in parallel, and the combination-

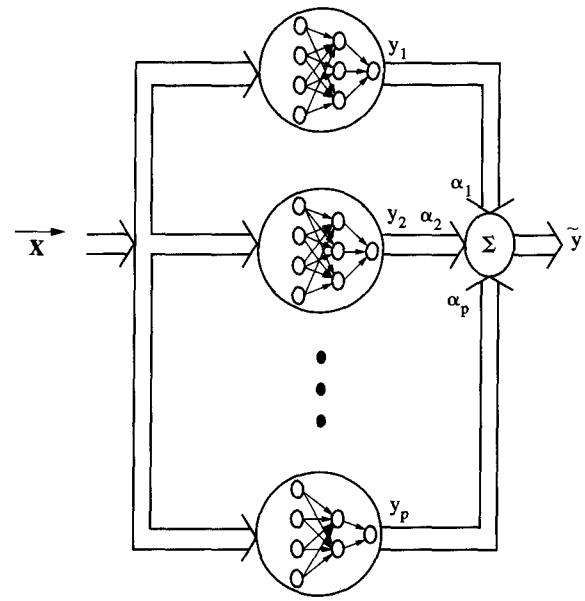

FIGURE 1. Linear combination of the outputs of *p* trained neural networks.

weights are the connection-weights of the output layer (Fig. 1). For a given input, $\mathbf{x}$, the output of the combined model, $\tilde{y}$, is the weighted sum of the corresponding outputs of the component NNs, y_j, $j = 1,\cdots,p$, and the α_j's are the associated combination-weights. The main difference between the two situations is that when combining NNs, the connection-weights of the trained NNs are fixed and the combination-weights are computed by performing simple (fast) matrix manipulations, as discussed in Sections 4 and 5. However, when training one large NN, there is a large number of parameters (weights) that need to be simultaneously estimated (trained). Thus, the training time for a large NN may be longer, and also the risk of over-fitting to the data may increase as the number of parameters in the model becomes large compared to the cardinality of the data set used to estimate these parameters.

We here consider approximating multi-input single-output mappings. One approach for the multi-output case is to compute an optimal combination-weight vector for each output separately. Such independent treatment is straightforward (Hashem et al., 1993) and minimizes the total MSE for multi-input multi-output mappings.

Consider a multi-input single-output mapping approximated by a trained NN. A trained NN accepts a vector-valued input $\mathbf{x}$ and returns a scalar output (response) $y(\mathbf{x})$. The approximation error is $\delta(\mathbf{x}) = r(\mathbf{x}) - y(\mathbf{x})$, where $r(\mathbf{x})$ is the response of the real system (true response) for $\mathbf{x}$.

According to (Hashem & Schmeiser, 1995), a linear combination of the outputs of p NNs returns the scalar output $\tilde{y}(\mathbf{x};\ \boldsymbol{\alpha}) = \Sigma_{j=1}^{p} \alpha_j y_j(\mathbf{x})$, with the corresponding approximation error $\tilde{\delta}(\mathbf{x};\boldsymbol{\alpha}) = r(\mathbf{x}) - \tilde{y}(\mathbf{x};\boldsymbol{\alpha})$; where $y_j(\mathbf{x})$ is the output of the jth network and α_j is the associated

combination-weight; $j = 1,\cdots,p$. Hashem et al. (1994) extended this definition of $\tilde{y}(\mathbf{x};\boldsymbol{\alpha})$ to include a constant term, $\alpha_0 y_0(\mathbf{x})$, where $y_0(\mathbf{x}) = 1$. This term allows for correction of any bias in the $y_j(\mathbf{x})'s$; $j = 1, \ldots, p$. Thus, $\tilde{y}(\mathbf{x};\boldsymbol{\alpha})$ is given by

$$\tilde{y}(\mathbf{x};\boldsymbol{\alpha}) = \sum_{j=0}^{p} \alpha_j y_j(\mathbf{x})$$

$$= \boldsymbol{\alpha}^t \mathbf{y}(\mathbf{x}), \tag{1}$$

where $\boldsymbol{\alpha}$ and $\mathbf{y}(\mathbf{x})$ are $(p + 1) \times 1$ vectors. The problem is to find good values for the combination-weights $\alpha_0,\alpha_1,\cdots,\alpha_p$. One approach is to select one of the p networks as best, say NN_b, set $\alpha_b = 1$, and set the other combination-weights to zero. Using a single network has the advantage of simplicity, but the disadvantage of ignoring the (possibly) useful information in the other $p - 1$ networks. Another approach, which is widely used by the forecasting community (Clemen, 1989) and also by the NN community (Pearlmutter & Rosenfeld, 1991) is to use equal combination-weights (simple average), The simple average is straightforward but assumes that all the component networks are equally good.

4. MSE-OLC OF NEURAL NETWORKS

Think of the input $\mathbf{x}$ as an observation of a random variable $\mathbf{X}$ from a (usually unknown) multivariate distribution function $F_{\mathbf{X}}$. Then, the real response is the random; variable $r(\mathbf{X})$, the output of the jth network is the random variable $y_j(\mathbf{X})$, and the associated approximation error is the random variable $\delta_j(\mathbf{X})$; $j = 1,\cdots,p$. The linear-combination output is the random variable $\tilde{y}(\mathbf{X};\boldsymbol{\alpha}) = \Sigma_{j=0}^{p} \alpha_j y_j(\mathbf{X})$, and the linear-combination error is the random variable $\tilde{\delta}(\mathbf{X};\boldsymbol{\alpha}) = r(\mathbf{X}) - \tilde{y}(\mathbf{X};\boldsymbol{\alpha})$.

The optimal linear combination (OLC) is defined by the optimal combination-weight vector $\boldsymbol{\alpha}^* = (\alpha_0^*, \alpha_1^*, \ldots, \alpha_p^*)$ that minimizes the expected loss

$$\int_S \ell(\tilde{\delta}(\mathbf{x};\boldsymbol{\alpha}))dF_{\mathbf{X}}(\mathbf{x}),$$

where S is the support of $F_{\mathbf{X}}$ and ℓ is a loss-function. Although various loss functions could be pursued, we focus on squared-error loss, $\ell(\tilde{\delta}) = \tilde{\delta}^2$. The objective is then to minimize the mean squared error (MSE),

$$\mathrm{MSE}(\tilde{y}(\mathbf{X};\boldsymbol{\alpha})) = \mathrm{E}[(\tilde{\delta}(\mathbf{X};\boldsymbol{\alpha}))^2] \tag{2}$$

where E denotes expected value with respect to $F_{\mathbf{X}}$. The resultant linear combination is referred to as the MSE-optimal linear combination (MSE-OLC). Thus, the MSE-OLC is defined by the optimal combination-weight vector $\boldsymbol{\alpha}^* = (\alpha_0^*, \alpha_1^*, \ldots, \alpha_p^*)$ that minimizes $\mathrm{MSE}(\tilde{y}(\mathbf{X};\boldsymbol{\alpha}))$.

4.1. MSE-OLC Combination-weights

From the MSE-OLC problem given by eqn (2), three other MSE-OLCs problems are derived and considered. The variations among the four MSE-OLC problems are in the inclusion (or exclusion) of the constant term, $\alpha_0 y_0(\mathbf{X})$, defined in Section 3, and/or constraining the combination-weights, $\alpha_j, j = 1, \ldots, p$, to sum to one. The inclusion of the constant term helps to correct for (possible) biases in the component NNs. On the other hand, constraining the combination-weights to sum to one may sometimes be used in improving the generalization ability of a combined model, as described in Section 7.

Among these four cases of MSE-OLCs, the unconstrained MSE-OLC with a constant term, discussed in Section 4.1.1, (theoretically) yields the smallest MSE. In this section, we present closed-form expression for the optimal combination-weights as well as expressions for the increase in MSE as a result of constraining the combination-weights to sum to one and/or not allowing the constant term.

4.1.1. *Unconstrained MSE-OLC with a Constant Term.* Consider the problem

$$\mathrm{P1}: \min_{\boldsymbol{\alpha}} \mathrm{MSE}(\tilde{y}(\mathbf{X};\boldsymbol{\alpha})).$$

Differentiating the MSE with respect to $\boldsymbol{\alpha}$ leads to the optimal combination-weight vector,

$$\boldsymbol{\alpha}_{(1)} = \boldsymbol{\Psi}^{-1}\mathbf{U}, \tag{3}$$

Where $\boldsymbol{\Psi} = [\psi_{ij}] = [\mathrm{E}\langle y_i(\mathbf{X})\, y_j(\mathbf{X})\rangle]$ is a $(p + 1) \times (p + 1)$ matrix, and $\mathbf{U} = [u_i] = [\mathrm{E}\langle r(\mathbf{X})\, y_i(\mathbf{X})\rangle]$ is a $(p + 1) \times 1$ vector. The corresponding (minimal) MSE is

$$\mathrm{MSE}^{(1)} = \mathrm{E}(r^2(\mathbf{X})) - \mathbf{U}^t\boldsymbol{\Psi}^{-1}\mathbf{U}. \tag{4}$$

P1 is equivalent to regressing $r(\mathbf{X})$ against $y_j(\mathbf{X}); j = 1, \ldots, p$, with an intercept term. Thus, the optimal combination-weights in eqn (3) are equal to the ordinary least squares (OLS) regression coefficients.

4.1.2. *Constrained MSE-OLC with a Constant Term.* Consider the case of constraining the combination-weights,$\alpha_1\cdots,\alpha_p$ to sum to one

$$\mathrm{P2}: \min_{\boldsymbol{\alpha}} \mathrm{MSE}(\tilde{y}(\mathbf{X};\boldsymbol{\alpha})), \text{ such that } \boldsymbol{\alpha}^t\mathbf{1}_z = 1,$$

where $\mathbf{1}_z$ is a vector of proper dimension with the first components equal to zero and the remaining components equal to 1.

Solving the Lagrangian equivalent of P2 leads to the optimal combination-weight vector

$$\boldsymbol{\alpha}_{(2)} = \boldsymbol{\alpha}_{(1)} - \beta_{(2)}\boldsymbol{\Psi}^{-1}\mathbf{1}_z, \tag{5}$$

where

$$\beta_{(2)} = \frac{-1 + \mathbf{1}_z^t\boldsymbol{\Psi}^{-1}\mathbf{U}}{\mathbf{1}_z^t\boldsymbol{\Psi}^{-1}\mathbf{1}_z}$$

The corresponding (minimal) MSE is

$$\mathrm{MSE}^{(2)} = \mathrm{MSE}^{(1)} + \beta_{(2)}^2\mathbf{1}_z^t\boldsymbol{\Psi}^{-1}\mathbf{1}_z. \tag{6}$$

The second term in the expression of $MSE^{(2)}$ can be easily shown to be non-negative. It reflects the cost (increase in MSE) of constraining the sum of the combination-weights to unity.

P2 is equivalent to regressing $(r(\mathbf{X}) - y_c(\mathbf{X}))$ against $(y_j(\mathbf{X}) - y_c(\mathbf{X}))$, for some $c \in \{1, \ldots p\}; j = 1, \ldots, p, j \neq c$; with an intercept term. Thus, the associated optimal-weight vector may be (alternatively) obtained using

$$\alpha_{(2)} = \acute{\Psi}^{-1}\acute{\mathrm{U}} \tag{7}$$

where $\acute{\Psi} = [\acute{\psi}_{ij}] = [\mathrm{E}\langle(y_i(\mathbf{X}) - I_{(I>0)}y_c(\mathbf{X}))(y_j(\mathbf{X}) - I_{(j>0)}y_c(\mathbf{X}))\rangle]$ is a $p \times p$ matrix, $I_{(i>0)}$ is an indicator variable that is equal to unity for $(i > 0)$ and to zero otherwise, and $\acute{\mathrm{U}} = [\acute{u}_i] = [E\langle(r(\mathbf{X}) - y_c(\mathbf{X}))(y_i(\mathbf{X}) - I_{(i>0)}y_c(\mathbf{X}))\rangle]$ is a $p \times 1$ vector.

4.1.3. *Unconstrained MSE-OLC without a Constant Term.* Consider the problem

$$\mathrm{P3}: \min_{\alpha} \mathrm{MSE}(\tilde{y}(\mathbf{X};\boldsymbol{\alpha})), \text{ such that } \boldsymbol{\alpha}^t\vartheta_z = 0,$$

where ϑ_z is a vector of proper dimension with the first component equal to 1 and the remaining components equal to zero.

Solving the Lagrangian equivalent of P3 leads to the optimal combination weight vector

$$\boldsymbol{\alpha}_{(3)} = \boldsymbol{\alpha}_{(1)} - \beta_{(3)}\Psi^{-1}\vartheta_z, \tag{8}$$

where

$$\beta_{(3)} = \frac{\vartheta_z^t\Psi^{-1}\mathrm{U}}{\vartheta_z^t\Psi^{-1}\vartheta_z}$$

The corresponding (minimal) MSE is

$$\mathrm{MSE}^{(3)} = \mathrm{MSE}^{(1)} + \beta_{(3)}^2\vartheta_z^t\Psi^{-1}\vartheta_z. \tag{9}$$

The second term in the expression of $MSE^{(3)}$ can be easily shown to be non-negative. It reflects the cost (increase in MSE) of not using the constant term α_0, in the combination.

P3 is equivalent to the same OLS problem in Section 4.1.1, but with no intercept term. Hence, the optimal combination-weights may be (alternatively) obtained using

$$\boldsymbol{\alpha}_{(3)} = \acute{\Psi}^{-1}\acute{\mathrm{U}} \tag{10}$$

where $\acute{\Psi} = [\acute{\psi}_{ij}] = [\mathrm{E}\langle y_i(\mathbf{X})\, y_j(\mathbf{X})\rangle]$, $(i,j > 0)$, is a $p \times p$ matrix, and $\acute{\mathrm{U}} = [\acute{u}_i] = [\mathrm{E}\langle r(\mathbf{X})\, y_i(\mathbf{X})\rangle]$, $(i > 0)$ is a $p \times 1$ vector.

4.1.4. *Constrained MSE-OLC without a Constant Term.* In this case, α_0 is set to zero and the sum of the remaining combination-weights is constrained to unity. Restricting the sum of the combination-weights to unity makes the MSE-OLC a weighted average of the outputs of the component networks. Thus, if the y_j's are unbiased (in a statistical sense), then $\tilde{y}$ will also be unbiased. This MSE-OLC is equivalent to the General Ensemble Method developed independently by Perrone and Cooper (1993).

$$\mathrm{P4}: \min_{\alpha} \mathrm{MSE}(\tilde{y}(\mathbf{X};\boldsymbol{\alpha})), \text{ such that } \boldsymbol{\alpha}^t\vartheta_z = 0, \text{ and } \boldsymbol{\alpha}^t\mathbf{1}_z = 1.$$

Solving the Lagrangian equivalent of P4 leads to the optimal combination-weight vector

$$\boldsymbol{\alpha}_{(4)} = \boldsymbol{\alpha}_{(1)} - \beta_{(4a)}\Psi^{-1}\vartheta_z - \beta_{(4b)}\Psi^{-1}\mathbf{1}_z, \tag{11}$$

where

$$\beta_{(4a)} = \frac{\vartheta_z^t\Psi^{-1}\mathrm{U} - \beta_{(4b)}\vartheta_z^t\Psi^{-1}\mathbf{1}_z}{\vartheta_z^t\Psi^{-1}\vartheta_z}$$

and

$$\beta_{(4b)} = \frac{(\mathbf{1}_z^t\Psi^{-1}\mathrm{U} - 1)(\vartheta_z^t\Psi^{-1}\vartheta_z) - (\vartheta_z^t\Psi^{-1}\mathrm{U})(\mathbf{1}_z^t\Psi^{-1}\vartheta_z)}{(\mathbf{1}_z^t\Psi^{-1}\mathbf{1}_z)(\vartheta_z^t\Psi^{-1}\vartheta_z) - (\vartheta_z^t\Psi^{-1}\mathbf{1}_z)^2}$$

The corresponding (minimal) MSE is

$$\mathrm{MSE}^{(4)} = \mathrm{MSE}^{(1)} + \beta_{(4a)}^2(\vartheta_z^t\Psi^{-1}\vartheta_z) + \beta_{(4b)}^2(\mathbf{1}_z^t\Psi^{-1}\mathbf{1}_z) + 2\beta_{(4a)}\beta_{(4b)}(\vartheta_z^t\Psi^{-1}\mathbf{1}_z).$$

Each of the three last terms in the expression of $MSE^{(4)}$ can be easily shown to be non-negative. Thus, their sum reflects the cost (increase in MSE) of not using the constant term, α_0, and at the same time restricting the remaining combination-weights to sum to one.

P4 is equivalent to the same OLS problem in Section 4.1.2, but with no intercept term. Hence, the optimal combination weights may be (alternatively) obtained using

$$\boldsymbol{\alpha}_{(4)} = \acute{\Psi}^{-1}\acute{\mathrm{U}}, \tag{13}$$

where $\acute{\Psi} = [\acute{\psi}_{ij}] = [\mathrm{E}\langle(y_i(\mathbf{X}) - y_c(\mathbf{X}))(y_j(\mathbf{X}) - y_c(\mathbf{X}))\rangle]$, $(i,j > 0)$, is a $(p-1) \times (p-1)$ matrix, *and* $\acute{\mathrm{U}} = [\acute{u}_i] = [\mathrm{E}\langle(r(\mathbf{X}) - y_c(\mathbf{X}))(y_i(\mathbf{X}) - y_c(\mathbf{X}))\rangle]$, $(i > 0)$ is a $(p-1) \times 1$ vector.

4.2. Alternate Expressions for the Constrained MSE-OLC Combination-weights

In Sections 4.1.2 and 4.1.4, the constrained optimal combination-weights are expressed as functions of the outputs of the NNs, $y_j(\mathbf{X}); j = 1, \ldots, p$. Alternatively, for Problems P2 and P4, the optimal combination-weights may be expressed in terms of the NNs approximation errors, $\delta_j(\mathbf{X}); j = 1, \ldots, p$.

4.2.1. *Constrained MSE-OLC with a Constant Term.* From eqn (2), the constrained MSE-OLC, P2, is equivalent to:

$$\mathrm{P2}': \min_{\alpha} \mathrm{E}\left[(\tilde{\delta}(\mathbf{X};\boldsymbol{\alpha}))^2\right], \text{ such that } \boldsymbol{\alpha}^t\mathbf{1}_z = 1$$

Solving the Lagrangian equivalent of P2 leads to the optimal combination-weight vector

$$\alpha_{(2)} = \frac{\Omega^{-1}\mathbf{1}_z}{\mathbf{1}_z^t\Omega^{-1}\mathbf{1}_z} \tag{14}$$

Where $\Omega = [\omega_{ij}] = [E\langle\delta_i(\mathbf{X})\delta_j(\mathbf{X})\rangle]$ is a $(p+1) \times (p+1)$ matrix.

4.2.2. *Constrained MSE-OLC without a Constant Term.* From eqn (2), the constrained MSE-OLC problem P4 is equivalent to

$$\text{P4}': \min_{\alpha} E\left[(\tilde{\delta}(\mathbf{X};\alpha))^2\right], \text{ such that } \alpha^t \vartheta_z = 0 \text{ and } \alpha^t \mathbf{1}_z = 1.$$

Solving the Lagrangian equivalent of P4′ leads to the optimal combination-weight vector

$$\alpha_{(4)} = (0, \alpha''^t_{(4)})^t, \tag{15}$$

where

$$\alpha''_{(4)} = \frac{\Omega''^{-1}\mathbf{1}}{\mathbf{1}^t\Omega''^{-1}\mathbf{1}}.$$

$\mathbf{1}$ is a vector of proper dimension with all components equal to 1, and $\Omega'' = [\omega''_{ij}] = [E\langle\delta_i(\mathbf{X})\delta_j(\mathbf{X})\rangle]$, $(i,j > 0)$, is a $p \times p$ matrix.

5. MSE-OLC COMBINATION-WEIGHTS ESTIMATION PROBLEM

In Section 4, closed-form expressions for MSE-OLC combination-weights are presented. These expressions are based on expected values taken with respect to the multivariate distribution function $F_{\mathbf{X}}$ of the model inputs. In practice, one seldom knows $F_{\mathbf{X}}$. Thus, $\Psi, \mathbf{U}, \hat{\Psi}, \hat{\mathbf{U}}, \Omega$, and Ω'' in eqns (3)–(15) need to be estimated. In this section, we study the problem of estimating the MSE-OLC combination-weights, and present an example to illustrate significant improvement in approximation accuracy as a result of using MSE-OLCs.

5.1. Problem Definition

Given a set K of observed data, estimate the MSE-OLC combination-weights, where $K = \{k_j : k_j = (\mathbf{x}_j, r(\mathbf{x}_j), \mathbf{y}(\mathbf{x}_j)), j = 1, \ldots, \kappa\}$ and $\mathbf{x}_j$'s are independently sampled from $F_{\mathbf{x}}$.

5.2. Ordinary Least Squares Estimators

From Section 4, the equivalance relation between the MSE-OLC combination-weights and the OLS regression coefficients allows the use of the OLS estimators in estimating the MSE-OLC combination-weights. OLS estimators are also used by Granger and Ramanathan (1984) in estimating the optimal combination-weights for combining forecasts. The analysis of the OLS estimators is straightforward and may provide measures of the quality of the estimated MSE-OLC, as discussed in Section 5.4.

For the MSE-OLC Problem P1 in Section 4.1.1, the equivalent regression model is

$$r(\mathbf{X}) = \alpha_0 + \sum_{j=1}^{p} \alpha_j y_j(\mathbf{X}) + \varepsilon. \tag{16}$$

where ε is a random error with zero mean and variance σ^2.

The OLS estimator of $\alpha_{(1)}$ is

$$\hat{\alpha}_{(1)} = \hat{\Psi}^{-1}\hat{\mathbf{U}} \tag{17}$$

where

$$\hat{\Psi} = [\hat{\psi}_{ij}] = [\sum_{k=1}^{\kappa} (y_i(\mathbf{X}_k)y_j(\mathbf{X}_k))/\kappa]$$

and

$$\hat{\mathbf{U}} = [\hat{u}_i] = [\sum_{k=1}^{\kappa} (r(\mathbf{X}_k)y_i(\mathbf{X}_k))/\kappa].$$

Similar OLS estimators may be used for $\alpha_{(2)}, \alpha_{(3)}$, and $\alpha_{(4)}$ based on eqn (7), 10 and 13, respectively.

5.3. Alternate Estimators for the Constrained MSE-OLC Combination-weights

Based on the expression for MSE-OLC combination-weights presented in Section 4.2.1., one may estimate the MSE-OLC combination-weights for the constrained MSE-OLC Problem P2 using

$$\hat{\alpha}_{(2)} = \frac{\hat{\Omega}^{-1}\mathbf{1}_z}{\mathbf{1}_z^t\hat{\Omega}^{-1}\mathbf{1}_z} \tag{18}$$

where

$$\hat{\Omega} = [\hat{\omega}_{ij}] = [\sum_{k=1}^{\kappa} (\delta_i(\mathbf{X}_k)\delta_j(\mathbf{X}_k))/\kappa].$$

For the constrained MSE-OLC Problem P4 in Section 4.1.4, and based on the expression for MSE-OLC combination-weights presented in Section 4.2.2, one may estimate the MSE-OLC combination-weights using

$$\hat{\alpha}_{(4)} = (0, \hat{\alpha}''^t_{(4)})^t \tag{19}$$

where

$$\hat{\alpha}''_{(4)} = \frac{\hat{\Omega}''^{-1}\mathbf{1}}{\mathbf{1}^t\hat{\Omega}''^{-1}\mathbf{1}},$$

and

$$\hat{\Omega}'' = [\hat{\omega}''_{ij}] = [\sum_{k=1}^{\kappa} (\delta_i(\mathbf{X}_k)\delta_j(\mathbf{X}_k))/\kappa].$$

5.4. Example 1

Consider the problem of approximating the function

$$r_1(x) = 0.02(12 + 3x - 3.5x^2 + 7.2x^3) \times (1 + \cos 4\pi x)(1 + 0.8 \sin 3\pi x),$$

over the interval [0,1], adopted from (Namatame & Kimata, 1989). The range of $r_1(x)$ is [0,0.9].

Six networks (NN1, NN2, NN3, NN4, NN5, NN6) were trained using error backpropagation. Each of

the first three networks were 1-5-5-1 feedforward NNs, i.e., had two hidden layers with five hidden units in each hidden layer. The other three networks were 1-10-1 networks. The activation function for the hidden units as well as the output units was the logistic sigmoid function $g(s) = (1 + e^{-s})^{-1}$. Linear data scaling was used for both model input and output. The networks were initialized with independent random connection weights uniformly distributed in [−0.3,0.3]. A set of 200 independent uniformly distributed points was used in training all the networks and in estimating the optimal combination-weights as well. Except for the structural differences and the different initial connection-weights, the six networks were trained in the same manner.

Eqn (17) yielded an estimated unconstrained optimal combination-weight vector (0.0003,0.125,−0.195, 0.639,0.781, −0.665,0.315)^t. The constant term did not appear to be statistically significant (at a 0.10 level of significance) with an associated two-sided P-value (Neter et al., 1985, p. 12), of 0.15. However, the other combination-weights were statistically significant with associated two-sided P-values less than 0.001. Dropping the constant from the combination, the estimated unconstrained optimal combination-weight vector became (0.,0.126, −0.195,0.639,0.779, −0.660,0.312)^t. The estimated unconstrained MSE-OLC resulted in a *true*[2] MSE of 0.000018; 87% less than that produced by NN4, the true best NN to approximate $r_1(x)$; and 95% less than that of the simple average of the corresponding outputs of the six NNs. These results demonstrate that MSE-OLCs can dramatically improve the accuracy of a neural network based model. This improvement in model accuracy is due to the fact that the six NNs provide approximations which are (slightly) different, so, by combining them, one many improve model accuracy.

An interesting observation is that the sum of the estimated optimal combination-weights was approximately one and the constant term was not statistically significant. In our experience (Hashem, 1993; Hashem et al., 1994), this is quite common in the cases where the component NNs are well-trained (accurate). In this example, the root mean squared (RMS) error associated with the best NN was 0.011 and that associated with the worst NN was 0.030. The fact that the RMS errors are low shows that the component NNs are well-trained. Also, the constant term was not statistically significant, which illustrates that since well-trained NNs may have insignificant biases, there may be no need for the constant term in the MSE-OLC.

6. THE ROBUSTNESS OF MSC-OLCS

By construction, the unconstrained MSE-OLC with a constant term (theoretically) yields the minimal MSE compared to the best network, the simple average, and to the other three MSE-OLCs as shown in Section 4. Also by construction, the estimated unconstrained MSE-OLC with a constant term has superior accuracy – in the MSE sense – on the combination data set K (defined in Section 5.1). However, the more important performance measure is the accuracy measured on a separate data set sampled from the same multivariate distribution $F_{\mathbf{x}}$. This performance measure is referred to as the *out-of-sample* performance or the *generalization ability*. This measure determines the *robustness* of the MSE-OLC. We here use the terms *out-of-sample performance, generalization ability*, and *robustness* interchangeably. In Example 1, the true function $r_1(x)$ is known. In practice, however, the robustness of the OLC model may be evaluated by testing performance of the model on a data set other than that used to construct the model. Thus, the accuracy of the MSE-OLC may be compared to that of the apparent best NN and that of the simple average in order to determine the robustness of the MSE-OLC.

A problem that sometimes affects the estimation of the optimal combination-weights, as well as the robustness of the MSE-OLC, is the collinearity among the predictor variables $y_j, j = 1,\cdots,p$, in the regression model described in Section 5.2. Since the y_j's are the outputs of NNs that are trained to approximate the same response variable, $r(\mathbf{x})$, we would expect them to be highly (positively) correlated. Thus, the matrix $\hat{\Psi}$ in eqn (17) may be ill-conditioned, making its inversion sensitive to round-off errors and also sensitive to small variations in the data, as in the case of noisy data. In the forecasting literature, the computational and statistical ill effects of collinearity are blamed for undermining the robustness of OLCs (Bunn, 1989; Clemen & Winkler, 1986; Guerard & Clemen, 1989; Menezes & Bunn, 1991; Winkler & Clemen, 1992). Likewise, in the literature on combining NNs, Perrone and Cooper (1993) pointed to the potential problems of ill-conditioned correlation matrices. In this section, we discuss the effect of collinearity on the estimation of the optimal combination-weights.

6.1. Collinearity and Correlation

Collinearity and correlation are related concepts. According to Belsley (1991), p. 19, k variates are collinear, or linearly dependent, if one of the vectors that represents them is in an exact linear combination of the others, that is, if the k vectors lie in a subspace of dimension less than k. On the other hand, the correlation between two variables (variates) is defined as the expected value of the normalized product of the variables centered around their corresponding means (i.e., mean-centered), where "normalized" stands for normalization with respect to the standard deviations of the two

[2] True means computed relative to the true (known) response function.

variables, respectively. Belsley (1991), pp. 26–27, indicates that while a high correlation coefficient between two explanatory (regressor) variables can indeed point to possible collinearity problems, the absence of high correlations cannot be viewed as evidence of the absence of collinearity problems, and that a high correlations implies collinearity, but the converse is not true. At this point in the discussion, we would like to draw attention to two fundamental points:

- Collinearity and correlation are not the same thing (Belsley, 1991, pp. 20, 26). Hence, special diagnosis needs to be applied to detect the presence of collinearity, (possibly) in addition to estimating the pairwise correlations among the variables being studied.
- Linear dependencies do not always degrade the estimates of the combination-weights (Belsley, 1991, pp. 72–74), nor the robustness of the MSE-OLC. Thus, beside looking for a diagnostic tool to detect the presence of collinearity, one also needs to look for an appropriate measure of the ill effects of such collinearity.

In Section 4, the four MSE-OLC problems P1–P4 are shown to be equivalent to OLS regression, where the regressor variables are $y_j(\mathbf{X})'$s, the outputs of the p trained NNs, or a function of these outputs. Since the NNs are individually trained to approximate the same response, $r(\mathbf{X})$, one can expect the correlation between the $y_j(\mathbf{X})'$s to be fairly (positively) high. The inherent high (positive) correlations between the y_j's[3] make estimating the MSE-OLC, as well as any other method for combining estimators, prone to collinearity problems. As a result, the robustness of the MSE-OLC may be affected (Bunn, 1989; Guerard & Clemen, 1989).

The alternate expressions presented in Section 4.2 may (appear to) be less vulnerable to the ill effects of collinearity than those in Section 4, since the former expressions rely on the approximation errors of the component NNs, δ_j's, instead of the y_j's (Bunn, 1989). However, the correlations between the δ_j's may also be high (and positive). In Example 1, the pairwise correlations between the outputs of the six trained NNs ranged from 0.997 to above 0.999, and hence were fairly high. Also, the pairwise correlations between the δ_j's of the six trained NNs ranged from -0.021 to 0.986, with most of them (10 out of 15) above 0.5. However, there was sufficient difference in the approximations provided by the six NNs, to yield an unconstrained MSE-OLC that reduced the *true*[4] MSE by 87% and 95% compared to the best NN and the simple average, respectively (Section 5.4). Thus, high positive correlations between the y_j's, which are associated with mostly positive correlations between the δ_j's, do not necessarily eliminate the benefit of combining nor result in harmful collinearity. From the above discussion, it is evident that for the successful deployment of the MSE-OLC, one needs to answer the following three questions:

1. How can one detect the presence of collinearity and identify the (regressor) variables associated with it?
2. How can one determine that an existing collinearity is harmful?
3. How can one deal with harmful collinearity in order to improve the robustness of the estimated MSE-OLC?

In the regression literature, several procedures have been developed for collinearity detection. Belsley (1991), pp. 26–37, discusses the main classes of these procedures and points to their strengths and weaknesses. Belsley et al. (1980) and Belsley (1991) develop diagnostics for explicit measurement of the severity of collinearity. These diagnostics are capable of determining the existence of multiple linear dependencies and identifying the variables involved in each collinearity as well. We thus adopt these diagnostics here and refer to them as the BKW diagnostics. We now proceed to investigate the remaining two questions.

6.2. Detecting Harmful Collinearity

Example 1, discussed in Section 5.4, demonstrates the benefit of using MSE-OLC to significantly reduce the MSE for approximating the function. However, as mentioned in Section 6.1, the pairwise correlation among the y_j's in Example 1 were fairly high. Moreover, the correlations among the δ_j's were mostly positive. High positive correlations, in themselves, cannot be considered as conclusive evidence of the ill effects of existing collinearity, but merely as a "warning" that collinearity exists and may harm the robustness of the estimated MSE-OLC. Before discussing methods for detecting the harmful effects of existing collinearity, we present an example to demonstrate that such harmful effects exist, and that collinearity can severely undermine the robustness of the estimated MSE-OLC.

6.2.1. *Example 2.* Consider approximating the function $r_2(x) = \sin[2\pi(1 - x)^2]$, where $x\epsilon[0,1]$, adapted from Cherkassky and Lari-Najafi (1991). Two 1-3-1 networks (NN1 and NN2), two 1-2-2-1 networks (NN3 and NN4), and two 1-4-1 networks (NN5 and NN6) were initialized with independent random connection-weights uniformly distributed in $[-0.3,0.3]$. The six NNs were trained using backpropagation on a training data set which consisted of 10 uniformly distributed independent points.

NN3, the true best NN, yielded an MSE of 0.09 on the training data and a *true* MSE of 0.46. The simple average of the outputs of the six NNs yielded an MSE of 0.10 on the training data and a *true* MSE of 0.68. By using the training data to estimate the optimal combination-

[3] We write y_j instead of $y_j(\mathbf{X})$ for simplicity. The same applies to δj in subsequent discussions.

[4] True refers to the MSE with respect to the true function, since the true function is know in this example.

weights, the unconstrained MSE-OLC with a constant term reduced the MSE on the training data set to almost zero (up to six decimal places). However, it yielded a true MSE of 91, which was larger than the true MSEs produced by NN3 and by simple average by orders of magnitude. This indicated that the robustness of the estimated MSE-OLC can be seriously undermined by existing collinearity. The MSE on the training data is listed only for completeness, since the true measure of performance and robustness is true MSE obtained relative to the true (known) function, $r_2(x)$.

An interesting observation regarding the above MSE OLC is that the two-sided P values of all the regression coefficients (including the constant term) were less than 0.035. In fact, the two-sided P-values of six out of the seven regression coefficients were less than 0.001. Thus, all the individual regression coefficients were statistically significant at a level of significance of 0.05. Hence, the statistical significance of the optimal combination weights may not be an adequate measure of the robustness of the MSE-OLC.

6.2.2. *A Cross-validation Approach for Detecting Harmful Collinearity.* By construction, the estimated MSE-OLC results in the smallest MSE on the data set that is used to estimate the optimal combination-weights (referred to as the estimation data set) compared to the best NN among the component NNs, and to the simple average of the corresponding outputs of the NNs in the combination. The robustness of the resultant MSE-OLC may be tested by comparing its performance to that of the best NN and the simple average on a separate validation data set, which is disjoint from the estimation data set but sampled from the same distribution, $F_{\mathbf{X}}$. If the MSE-OLC is still the best performer on the validation test, then one may conclude that it is robust. Otherwise, one may look for corrective measures to improve the robustness of the MSE-OLC. Asymptotically, as the size of the validation set increase, this test measures the true robustness of the MSE-OLC. This testing strategy is straightforward and is similar to the strategy often adopted in testing the generalization ability of a (single) NN (Drucker & Le Cun, 1991; Levin et al., 1990; Morgan & Bourlard, 1990; Weigend et al., 1991). Likewise, in the literature on combining forecasts, many advocate the use of out-of-sample testing of the combination (Gardner & Makridakis, 1988; Holden & Peel, 1989).

Some important issues concerning the application of this validation approach follow.

- According to the definition of robustness in Section 6, the data in the validation set need to be randomly sampled from $F_{\mathbf{X}}$. In practice, $F_{\mathbf{X}}$ is often unknown, and only a set of observed data, K, is available for constructing the MSE-OLC. in such cases, assuming that the data are independent and equally likely, the observed data set may be split into an estimation data set, K_1, and a validation data set, K_2. Such splitting is at the expense of reducing the data used in estimating the optimal combination-weights. Meanwhile, K_2 needs to be sufficiently large in order to accurately test the robustness of the MSE-OLC.
- The notion of "best" NN needs a more precise definition. In practice, one does not know which of the trained NNs is the "true" best. There is no reason to believe that the best NN on the training data set will be the true best NN. In fact, a NN that overfits the most to the training data will have the lowest MSE among a number of trained NNs. A consistent estimator of the true best is the best performer among the trained NNs on the validation set, K_2. We refer to such a network as the "apparent best" NN. Asymptotically, as the number of data points in K_2 increases, this estimate yields the true best NN.

7. IMPROVING THE ROBUSTNESS OF MSE-OLC

There are a variety of methods to deal with harmful collinearity (Hashem, 1993). Among these methods are: introducing new data to break up the collinearity pattern (Neter et al., 1985; Belsley, 1991; Guerard & Clemen, 1989; Hines & Montgomery, 1990); using biased estimation techniques, such as ridge regression (Neter et al., 1985, pp. 394–400) or latent root regression (Webster et al., 1974), to improve the efficiency of estimating the regression coefficients (the optimal combination-weights); and dropping one or several regressor variables (component networks) from the model (Neter et al., 1985; Moskowitz & Wright, 1985; Scheaffer & McClave, 1990). In the literature on combining forecasts, Guerard and Clemen (1989) indicate that, while the use of latent roots produces more efficient estimates of the combination-weights compared to the OLS estimates, their out-of-sample forecasting performances are comparable. In Section 7.2, we introduce two algorithms for selecting NNs for MSE-OLCs.

Before presenting the selection algorithms, we discuss another common approach for improving the robustness of the OLC. Clemen (1986) suggested that it may indeed be appropriate to restrict the combination-weights, or to require no constant term, or both, if the restricted combination results in a more efficient (robust) forecast. Bunn (1989) argued that while the unconstrained model should give the better fit to past data, constraints can improve the robustness of the combination in forecasting. To examine the effectiveness of restricting the combination-weights to sum to one and/or removing the constant term from the combination, we re-examine the MSE-OLC in Example 2.

7.1. Example 2 (Continued)

The estimated unconstrained MSE-OLC with a constant term (P1), constructed in Section 6.2.1, yielded a true

MSE of 91. The three other MSE-OLCs (P2,P3,P4) yielded smaller true MSE's of 8.2, 20.4 and 4.1, respectively. However, they were still larger than those of the best NN and the simple average of the trained networks, which were 0.46 and 0.68, respectively.

Analysis of the collinearity structure, using BKW's diagnostics, revealed that the constant term in the MSE-OLC was involved in the strongest collinearity among the regressor variables. This may explain why dropping the constant term from the combination improves its robustness. In Section 7.2, we present an approach for selecting the component networks for the MSE-OLCs based on BKW's diagnostics.

7.2. Algorithms for Selecting the Component Neural Networks

In this section, we introduce an approach for improving the robustness of the MSE-OLC through the proper selection of the NNs, guided by diagnostics of the collinearity among the y_j's (the outputs of the NNs) and/or among the δ_j's (the approximation errors of the NNs). We present two algorithms, algorithm (A) and algorithm (B) based on this approach. The inputs to the algorithms are the p trained NNs, an estimation data set K_1, and a validation data set K_2. Both algorithms use BKW collinearity diagnostics (Belsley, 1991) to analyze the collinearity structure among the given networks, determine the relative strength of existing linear dependencies, and identify the networks involved in each collinearity. While algorithm (A) relies on diagnosing collinearity among the y_j's, algorithm (B) relies on diagnosing collinearity among the δ_j's. Both algorithms use the cross-validation approach, outlined in Section 6.2.2, to test robustness. The algorithms are greedy in the sense that they target the strongest collinearity. Once the networks involved in the strongest collinearity are identified, the algorithms attempt to break up this collinearity by dropping the "worst performer" among these networks from the combination. The worst performer is defined as the NN that yields the largest MSE on K_2. Thus, the algorithms never drop the (apparent) best NN from the combination.

In both algorithms, the performance of the best NN and of the simple average of the outputs of the component networks, measured on a validation set K_2, are taken as yard-sticks for measuring the robustness of the resultant combination. If the best combination produced by an algorithm yields an inferior performance on K_2 compared to either the best NN or the simple average, then the algorithm selects its final outcome to be the best performer of the latter two.

Algorithms (A) and (B) are conservative. They only drop networks from the combination where the current MSE-OLC is deemed inferior to the best NN or the simple average, as determined by their relative performance on the K_2. A more aggressive approach may allow dropping more networks as long as the performance on K_2 keeps improving. When employing the selection algorithms, one needs to keep in mind that the component networks may carry different information. Hence, the more NNs that can be salvaged and included in the final combination, the better. The only reason for excluding some networks (or likewise the constant term) from the MSE-OLC is the presence of harmful collinearity. Algorithms (A) and (B) are listed in Appendix A.

7.2.1. *Modification to the Algorithms.* In Section 6.2.2, we discussed the need for splitting the data set K into K_1 and K_2 in order to detect harmful collinearity. However, including extra data in the estimation of the optimal combination-weights helps break up collinearity among the networks. A compromise may be achieved by using all the available data (i.e., K) in the final estimation step, i.e. after the algorithms decide upon the networks to be included in the final combination. To illustrate the effectiveness of the algorithms in handling harmful collinearity problem, we re-examine Example 2.

7.2.2. *Example 2 (Continued).* In Section 6.2.1 and Section 7.1, we examined several approaches for constructing MSE-OLCs of the six trained networks generated in this example. None of these approaches yielded performance comparable to that of NN3, the best NN, or that of the simple average. On using algorithms (A) and (B), algorithm (A) recommended using NN3 alone, while algorithm (B) recommended combining NN3, NN4, and NN6, in addition to using a constant term. The estimated C-OLC constructed by algorithm (B) yielded a true MSE of 0.037, which is 92% less than the true MSE of NN3, and 95% less than the true MSE of the simple average of the six networks. This illustrates that both algorithms are capable of handling harmful collinearity and improving the robustness of the estimated MSE-OLCs. In Section 8, we present experimental results to further examine the effectiveness of the algorithms.

8. EXPERIMENTAL RESULTS

In this section, we test the MSE-OLC approach on three function approximation problems. For each problem, a number of replications, with independent data sets, were generated and used for creating NN based models. In each replication, MSE-OLCs were formed using algorithms (A) and (B), and their resultant true MSE's were compared with those of the single best NN and the simple average. A total of 94 replications were tested.

8.1. Problem I

Consider approximating $f_1(x) = \sin[2\pi(1-x)^2]$, where $x \in [0, 1]$, adapted from (Cherkassky & Lari-Najafi,

TABLE 1
Results of Function Approximation Problem I

	Mean % reduction in true MSE with respect to best NN		Mean % reduction in true MSE with respect to simple average		Number of wins				Best MSE
	Algorithm A	Algorithm B	Algorithm A	Algorithm B	Algorithm A	Algorithm B	Best NN	Simple average	
Small number of samples									
No noise	81	87	89	94	9	9	0	0	0.0002
With noise	44	52	57	62	7	9	1	1	0.0025
Large number of samples									
No noise	67	68	91	91	9	10	1	0	0.0016
With noise	11	11	29	28	7	7	4	2	0.0045
Over all replications	51	55	67	69	32	35	6	3	N/A

1991). The range of $f_1(x)$ is $[-1,1]$. We used two sizes of training data sets (10 points and 30 points) with associated validation data sets (of 5 points and 20 points respectively). Beside using noise-free data, we corrupted the data with additive Gaussian noise, $N[0,(0.2)^2]$, corresponding to a signal-to-noise ratio (S/N) of approximately 3. Thus, a total of four different cases (combination of data size and noise level) were tested.

For each case, ten independent replications, with independent data sets, were carried out. In each replication, we used six networks (NN1, NN2, NN3, NN4, NN5, and NN6) that were initialized with independent connection-weights uniformly distributed in $[-0.3,0.3]$. NN1 and NN2 were 1-3-1 networks, NN3 and NN4 were 1-2-2-1 networks, NN5 and NN6 were 1-4-1 networks. Since a separate $K^{(j)}$ was used for each replication j, 40 different sets of the six networks were produced at the end of the training process; a total of 240 trained networks. (Actually, the six networks in Example 2 result from one of the replications examined here).

The resultant *true* – computed relative to the true function $f_1(x)$ –MSEs from algorithms (A) and (B), the best NN, and the simple average were computed and compared in every replication, and the approach that

TABLE 2
Results of Function Approximation Problem II

	Mean % reduction in true MSE with respect to best NN		Mean % reduction in true MSE with respect to simple average		Number of wins				Best MSE
	Algorithm A	Algorithm B	Algorithm A	Algorithm B	Algorithm A	Algorithm B	Best NN	Simple average	
Small number of samples									
No noise	24	22	42	41	6	5	0	0	0.032
Medium noise	22	19	38	35	6	5	2	0	0.048
High noise	6	3	6	4	5	4	3	3	0.087
Large number of samples									
No noise	39	39	62	62	6	6	0	0	0.014
Medium noise	32	32	54	54	6	6	0	0	0.026
High noise	23	19	37	35	5	4	2	0	0.062
Over all replications	24	22	40	39	34	30	7	3	N/A

yielded the minimal true MSE (or within 5% from the achieved minimal true MSE) in a given replication, was considered the "winner" of that replication (multiple winners were allowed).

For each of the four cases, the mean percentage reduction in true MSE and the number of "wins" achieved by each approach were computed over the ten replications and given in Table 1. In each case, the smallest true MSE over all replications (denoted by "best MSE") was also given to provide a measure of the accuracy of the resultant models.

From Table 1, algorithms (A) and (B) outperformed the approaches of using the best NN and the simple average in most of the 40 replications. Over all replications, their corresponding mean percentage reductions in the *true* MSE, compared to the best NN and the simple average, were between 51 and 69%.

8.2. Problem II

Consider approximating $f_2(x_1, x_2) = 1{\cdot}9(1{\cdot}35 + e^{x_1}\sin(13(x_1-.6)^2) * e^{-x_2^2}\sin(7x_2))$, where $x_1, x_2 \in [0,1]^2$, adopted in (Maechler et al., 1990; Hwang et al., 1994; Cherkassky et al., 1995).

We used two sizes of training data sets (150 points and 225 points) with associated validation data sets of (75 points and 175 points) respectively. Beside using noise-free data, we corrupted the data with two levels of additive Gaussian noise: $N[0,(0.25)^2]$ and $N[0,(0.5)^2]$, which correspond to S/N ratios of 4 and 2, respectively. Thus, a total of six different cases (combination of data size and noise level) were tested.

For each case (combination of data size and noise level), six independent replications were carried out. In each replication, we used six networks (NN1,NN2,NN3, NN4, NN5, and NN6) that were initialized with independent connection-weights uniformly distributed in $[-0.3,0.3]$. NN1 and NN2 were 2-10-1 networks, NN3 and NN4 were 2-5-1 networks, NN5 and NN6 were 2-5-5-1 networks. A total of 36 replications, involving 216 different networks, were tested. The results are shown in Table 2.

From Table 2, algorithms (A) and (B) outperformed the approaches of using the best NN and the simple average in almost all the 36 replications. Over all replications, their corresponding mean percentage reductions in the *true* MSE, compared to the best NN and the simple average, were between 22 and 40%

8.3. Problem III

Consider the problem of approximating the function

$$f_3(x_1,x_2,x_3,x_4,x_5) = 0{\cdot}0647(12 + 3x_1 - 3{\cdot}5x_2^2 + 7{\cdot}2x_3^3) \times (1 + \cos 4\pi x_4)(1 + 0{\cdot}8\sin 3\pi x_5),$$

where $x_i \in [0,1]$, $= 1,\cdots, 5$.

We used two sizes of training data sets (342 points and 1024 points) with associated validation data sets of (108 points and 342 points) respectively. Besides using noise-free data, we corrupted the data with two levels of additive Gaussian noise: $N[0,(0.25)^2]$ and $N[0,(0.5)^2]$, which correspond to S/N ratios of 4 and 2, respectively. Thus, a total of six different cases (combination of data size and noise level) were tested.

For each case, three independent replications were

TABLE 3
Results of Function Approximation Problem III

	Mean % reduction in true MSE with respect to best NN		Mean % reduction in true MSE with respect to simple average		Number of wins				Best MSE
	Algorithm A	Algorithm B	Algorithm A	Algorithm B	Algorithm A	Algorithm B	Best NN	Simple average	
Small number of samples									
No noise	0	0	72	72	3	3	3	0	0.065
Medium noise	9	2	25	18	3	2	1	1	0.054
High noise	13	15	34	36	2	3	1	1	0.150
Large number of samples									
No noise	16	16	86	86	3	3	0	0	0.014
Medium noise	17	17	40	40	3	3	0	0	0.048
High noise	16	16	20	20	3	3	1	1	0.073
Over all replications	12	11	46	45	17	17	6	3	N/A

carried out. In each replication, we used four networks (NN1, NN2, NN3, and NN4) that were initialized with independent connection-weights uniformly distributed in [− 0.3,0.3]. NN1 and NN2 are (5-15-1) networks and NN3 and NN4 are (5-10-1) networks. A total of 18 replications, involving 72 different networks, were tested. The results are shown in Table 3.

From Table 3, algorithms (A) and (B) outperformed the approaches of using the best NN and the simple average in almost all the 18 replications. Over all replications, their corresponding mean percentage reductions in the *true* MSE, compared to the best NN and the simple average, were between 11 and 46%.

8.4. Discussion of the Experimental Results

The results, shown in Tables 1–3, demonstrate that the MSE-OLC algorithms yield significant improvement in modeling accuracy, compared to choosing the best NN or using the simple average of the trained networks in almost all the cases examined here. The improvement in accuracy, achieved in a given replication, depends on the following factors:

- Degree of redundancy in the information obtained from the component networks. If all the component networks carry the same information, no benefits may be expected from combining them. To increase the benefits of combining, the component networks may be constructed using different topologies, different initial connection-weights, different learning algorithms, different activation function, etc.
- Superiority of the best network, where one network is much better than the rest, while the remaining networks have no additional knowledge to contribute. In this case, the MSE-OLC algorithms will tend to favor using the best network by itself.
- Adequacy of the combination data (data set K). If the combination data are not adequate, the ill effects of collinearity may be so severe that the MSE-OLC algorithms can only selvage the best network, or alternatively they may recommend using the simple average.

The MSE-OLC algorithms exhibit robustness both for small and large sample sizes and at different noise levels. The algorithms not only yield significant mean percentage reduction in the *true* MSE over the best network and the simple average, but they also consistently outperformed the latter approaches in most replications. As the amount of data available for combining increase (quantity and quality), the MSE-OLC algorithms become clearly superior to the other two approaches, which is expected based on the theoretical results in Section 4.1.

The extra computational time required for combining the trained networks, in a given replication, is a function of the number of data points and the number of networks in the combination. For the replications discussed here, this extra computational time is of the order of a few CPU seconds on a SUN Sparc 2 workstation, which is fairly modest compared to the training time, which is typically on the order of several CPU minutes.

In our experimental study, the mean performances of algorithms (A) and (B) are comparable, which indicates that analyzing the collinearity among the networks outputs and the collinearity among the network errors are equally valuable in improving the robustness of MSE-OLCs. We recommend that, in practice, one may try both algorithms, then select the algorithm which yields better accuracy on the validation data set.

The resultant model accuracy from the combined models in Problem II was somewhat better than Hwang et al. (1994) and close to the accuracy reported by Cherkassky et al. (1995)[5]. Meanwhile, using the MSE-OLC approach, the total computational time for training and combining the networks, in any replication, is of the order of several CPU minutes on a SUN Sparc 2 workstation, compared to much longer training times in excess of several hours on a Sun 4 workstation, reported by Cherkassky et al. (1995). Thus, combining a number of trained networks may be used as an alternative to extensive training time required to achieve a given accuracy. By testing the MSE from the MSE-OLC of a number trained networks, one may choose to terminate training much earlier, without sacrificing model accuracy.

9. CONCLUSIONS

The use of MSE-OLCs of a number of trained neural networks can substantially improve model accuracy compared to the single best network, or to the simple average of the corresponding outputs of the component networks.

Since multiple trained networks are often available as a by-product of the modeling process, the additional computational effort required to create an MSE-OLC is essentially that of estimating the optimal combination-weights, which is mainly a matrix inverse.

In theory, the unconstrained MSE-OLC with a constant term yields the minimal MSE among all the combination methods considered in this paper. However, in practice, the robustness of any *estimated* MSE-OLC may be undermined due to the presence of harmful collinearity. Proper selection of the component networks, by utilizing collinearity diagnostics and cross-validation methods, such as in the algorithms presented above, can significantly improve the robustness of the estimated MSE-OLC, and hence yields more accurate models.

The MSE-OLC approach can also be used to combine neural networks with other non NN-based models. No

[5] The standard deviation of our test data is unity, which makes the MSE presented here equal to fraction of variance unexplained (FVU) in Hwang et al. (1994) and equal to the square of the normalized root mean squared error (NRMS) in (Cherkassky et al., 1995).

special extensions are required for this step, since the framework presented in this paper is readily applicable to such situations, and the expressions derived for the optimal combination-weights may be directly applied.

REFERENCES

Alpaydin, E. (1993). Multiple networks for function learning. In *Proceedings of the 1993 IEEE International Conference on Neural Networks: Vol. 1.* (pp. 9–14). New Jersey: IEEE Press.

Battiti, R., & Colla, A.M. (1994). Democracy in neural nets: Voting schemes for classification. *Neural Networks*, **7 (4),** 691–707.

Baxt, W.G. (1992). Improving the accuracy of an artificial neural network using multiple differently trained networks. *Neural Computation*, **4,** 772–780.

Belsley, D. A. (1991). *Conditioning Diagnostics: Collinearity and Weak Data in Regression.* New York: John Wiley and Sons.

Belsley, D. A., Kuth, E., & Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity.* New York: John Wiley.

Benediktsson, J. A., Sveinsson, J. R., Ersoy, O. K., & Swain, P. H. (1993). Parallel consensual neural networks. In *Proceedings of the 1993 IEEE International Conference on Neural Networks: Vol. 1.* (pp. 27–32). New Jersey: IEEE Press.

Breiman, L. (1992) *Stacked regressions.* Technical Report 367, Department of Statistics, University of California, Berkeley, California 94720, USA. Revised June 1994.

Bunn, D.W. (1989). Forecasting with more than one model. *Journal of Forecasting*, **8,** 161–166.

Cherkassky, V., Gehring, D., & Mulier, F. (1995). Pragmatic comparison of statistical and neural network methods for function estimation. In *Proceedings of the 1995 World Congress on Neural Networks, Vol. 2.* (pp. 917–926). New Jersey: Lawrence Erlbaum Assoc.

Cherkassky, V., & Lari-Najafi, H. (1991). Constrained topological mapping for nonparametric regression analysis. *Neural Networks*, **4,** 27–40.

Clemen, R.T. (1986). Linear constraints and the efficiency of combined forecasts. *Journal of Forecasting*, **5,** 31–38.

Clemen, R.T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, **5,** 559–583.

Clemen, R.T., & Winkler, R.L. (1986). Combining economic forecasts. *Journal of Business and Economic Statistics*, **4 (1),** 39–46.

Cooper, L. (1991). Hybrid neural network architectures. Equilibrium systems that pay attention. In R.J. Mammone & Y.Y. Zeevi (Eds.), *Neural Networks: Theory and Applications*, (pp. 81–96). New York: Academic Press.

Drago, G.P., & Ridella, S. (1992). Statistically controlled activation weight initialization SCAWI. *IEEE Transactions on Neural Networks*, **3 (4),** 627–631.

Drucker, H., & Le Cun, Y. (1991). Double backpropagation increasing generalization performance. In *Proceedings of the 1991 International Joint Conference on Neural Networks in Seattle: Vol. 2.* (pp. 145–150). New Jersey: IEEE Press.

Gardner, E.S. Jr., & Makridakis, S. (1988). The future of forecasting. *International Journal of Forecasting*, **4,** 325–330.

Granger, C.W.J. (1989). Combining forecasts – twenty years later. *Journal of Forecasting*, **8,** 167–173.

Granger, C.W.J., & Ramanathan, R. (1984). Improved methods of combining forecasts. *Journal of Forecasting*, **3,** 197–204.

Guerard, J.B. Jr., & Clemen, R.T. (1989). Collinearity and the use of latent root regression for combining GNP forecasts. *Journal of Forecasting*, **8,** 231–238.

Hansen, L.K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12 (10),** 993–1001.

Hashem, S. (1993). *Optimal Linear Combinations of Neural Networks.* Unpublished PhD thesis, School of Industrial Engineering, Purdue University, West Lafayette, Indiana.

Hashem, S., & Schmeiser, B. (1993). Approximating a function and its derivatives using MSE-optimal linear combinations of trained feedforward neural networks, In *Proceedings of the 1993 World Congress on Neural Networks, Vol. 1.* (pp. 617–620). New Jersey: Lawrence Erlbaum Associates.

Hashem, S., & Schmeiser, B. (1995). Improving model accuracy using optimal linear combinations of trained neural networks. *IEEE Transactions on Neural Networks*, **6 (3),** 792–794.

Hashem, S., Schmeiser, B., & Yih, Y. (1994). Optimal linear combinations of neural networks: An overview. In *Proceedings of the 1994 IEEE International Conference on Neural Networks: Vol. 3.* (pp. 1507–1512). New Jersey: IEEE Press.

Hashem, S., Yih, Y., & Schmeiser, B. (1993). An efficient model for product allocation using optimal combinations of neural networks. In C. Dagli, L. I. Burke, B. R. Fernández, & J. Ghosh (Eds.), *Intelligent Engineering Systems through Artificial Neural Networks Vol. 3.* (pp. 669–674). ASME Press.

Haykin, S. (1994). *Neural networks: a comprehensive foundation.* New Jersey: IEEE Press.

Hines, W. W., & Montgomery, D. C. (1990) *Probability and Statistics in Engineering and Management Science.* New York: John Wiley.

Holden, K., & Peel, D.A. (1989). Unbiasedness, efficiency and the combination of economic forecasts. *Journal of Forecasting*, **8,** 175–188.

Hwang, J.N., Lay, S.R., Maechler, M., Martin, R.D., & Schimert, J. (1994). Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks*, **5 (3),** 342–353.

Jacobs, R. A., & Jordan, M. (1991). A competitive modular connectionist architecture. In R. Lippmann, J. Moody, & D. Touretzky (Eds.), *Advances in Neural Information Processing System , Vol. 3* (pp. 767–773). California: Morgan Kaufman.

Jacobs, R.A., & Jordan, M. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, **6,** 181–214.

Kolen, J. F., & Pollack, J. B. (1991). Back propagation is sensitive to initial conditions. In R. Lippmann, J. Moody, & D. Touretzky (Eds.), *Advances in Neural Information Processing Systems, Vol. 3* (pp. 860–867). California: Morgan Kaufman.

Laplace, P.D. (1818) *Deuxième Supplément a la Théorie Analytique des Probabilités.* Courcier, Paris. Reprinted 1847 in *Oeuvers Complètes de Laplace*, (Paris, Gauthier-Villars), 7, pp. 531–580.

Lari-Najafi, H., Nasiruddin, M., & Samad, T. (1989). Effect of initial weights on back propagation and its variations. In *Proceedings of the 1989 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 218–219). New Jersey: IEEE Press.

Levin, E., Tishby, N., & Solla, S.A. (1990). A statistical approach to learning and generalization in layered neural networks. *Proceedings of the IEEE*, **78 (10),** 1568–1574.

Maechler, M., Martin, D., Schimert, J., Csoppenszky, M., & Hwang, J. (1990). Projection pursuit learning networks for regression. In *Proceedings of the 2nd International Conference on Tools for Artificial Intelligence* (pp. 350–358). New Jersey: IEEE Press.

Mani, G. (1991). Lowering variance of decisions by using artificial neural networks portfolios. *Neural Computation*, **3,** 484–486.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, **7 (1),** 77–91.

Menezes, L., & Bunn, D. (1991). Specification of predictive distribution from a combination of forecasts. *Methods of Operations Research*, **64,** 397–405.

Morgan, N., & Bourlard, H. (1990). Generalization and parameter estimation in feedforward nets: Some experiments. In D.S. Touretzky (Ed.), *Advances in neural information processing systems, Vol. 2* (pp. 630–637). California: Morgan Kaufman.

Moskowitz, H., & Wright, G. P. (1985). *Statistics for management and economics.* Ohio: Charles Merrill Publishing Company.

Namatame, A., & Kimata, Y. (1989). Improving the generalising capabilities of a back-propagation network. *The International Journal of Neural Networks Research and Applications*, **1 (2),** 86–94.

Neter, J., Wasserman, W., & Kutner, M. H. (1985). *Applied linear statistical models*. Homewood, IL: Irwim.

Ohno-Machado, L., & Musen, M.A. (1994). Hierarchical neural networks for partial diagnosis in medicine. In *Proceedings of the 1994 World Congress on Neural Networks: Vol. 1* (pp. 291–296). New Jersey: Lawrence Erlbaum Associates.

Parmanto, B., Munro, P.W., Doyle, H.R., Doria, C., Aldrighetti, L., Marino, I.R., Mitchel, S., & Fung, J.J. (1994). Neural network classifier for hepatoma detection. In *Proceedings of the 1994 World Congress on Neural Networks, Vol. 1* (pp. 285–290). New Jersey: Lawrence Erlbaum Associates.

Pearlmutter, B.A., & Rosenfeld, R. (1991). Chaitin-Kolmogorov complexity and generalization in neural networks. In *Advances in neural information processing systems, Vol. 3* (pp. 925–931). California: Morgan Kaufman.

Perrone, M.P. (1993). *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization*. Unpublished PhD thesis, Department of Physics, Brown University. Providence, RI, U.S.A.

Perrone, M.P., & Cooper, L.N. (1993). When networks disagree: Ensemble methods for hybrid neural networks. In R.J. Mammone (Ed.), *Neural networks for speech and image processing*. Chapman and Hall.

Rogova, G. (1994). Combining the results of several neural network classifiers. *Neural Networks*, **7 (5),** 777–781.

Scheaffer, R.L., & McClave, J.T. (1990). *Probability and statistics for engineers*. Boston: PWS-KENT Publishing Company.

Sobol', I.M. (1974). *The Monte Carlo method*. University of Chicago Press, Illinois, U.S.A. Translated and adapted from the 2nd Russian edition by R. Messer, J. Stone, & P. Fortini.

Webster, J.T., Gunst, R.F., & Mason, R.L. (1974). Latent root regression analysis. *Technometrics*, **16 (4),** 513–522.

Weigend, A. S., Rumelhart, D.E., & Huberman, B.A. (1991). Generalization by weight-elimination with application to forecasting. In R. Lippmann, J. Moody, & D. Touretzky (Eds.). *Advances in neural information processing systems, Vol. 3* (pp. 875–882). California: Morgan Kaufman.

Winkler, R.L., & Clemen, R.T. (1992). Sensitivity of weights in combining forecasts. *Operations Research*, **40 (3),** 609–614.

Wolpert, D.H. (1992). Stacked generalization. *Neural Networks*, **5,** 241–259.

Zurada, J.M. (1992). *Introduction to artificial neural systems*. St Paul, MN: West Publishing Company.

APPENDIX

1. Algorithm (A)

Algorithm (A) employs unconstrained MSE-OLCs (U-OLCs) and relies solely on the information provided by the BKW diagnostics of collinearity among the y_j's. Algorithm (A) proceeds as follows:

1. Determine the MSE of the best NN and of the simple average of the p networks on K_2.
2. Consider all the p networks for the combination.
3. Form the U-OLC of all the considered networks, including a constant term (unless a decision to exclude the constant has been taken earlier), using K_1 to estimate the optimal combination-weights.
4. Determine the MSE of the U-OLC (from step 3) on K_2.
5. If the U-OLC yields the lowest MSE on K_2 compared to the best NN and the simple average, then STOP and return the current U-OLC.
6. Construct G_1, the set of networks involved in the strongest collinearity among the y_j's in the current combination, using BKW diagnostics.
7. If G_1 has two or more elements, then
   - If there are more than two networks in the current combination or if the constant term is not involved in the strongest collinearity, then drop the worst performer in G_1 from the combination. Go to Step 8.
   - Else if the constant is involved in the strongest collinearity, then drop it from the combination.
   - Else STOP and return the best performer between the best NN and the simple average.

   Else (G_1 has one element, which means that the constant term is involved in the strongest collinearity.)
   - If this NN is the best NN, then drop the constant term from the combination.
   - Else drop the NN in G_1 from the combination.
8. If there are more than one NN left in the combination, then go to Step 3.

   Else STOP and return the best performer between the best NN and the simple average.

2. Algorithm (B)

Algorithm (B) employs constrained MSE-OLCs (C-OLCs) and relies solely on the information provided by the BKW diagnostic of collinearity among the δ_j's. As discussed in Section 7, C-OLCs may be more robust than U-OLCs, especially for small samples. That is why algorithm (B) uses C-OLCs to improve robustness in the cases where the robustness of the U-OLC with a constant term is deemed unsatisfactory. Algorithm (B) is identical to algorithm (A) except for two features:

- When the robustness of the U-OLC of all the p networks is deemed unsatisfactory, algorithm (B) adopts C-OLC instead of U-OLC in the subsequent steps.
- Instead of relying on a collinearity diagnosis for the y_js, algorithm (B) relies on diagnosing the collinearity among the δ_j's. Thus, instead of the set G_1, a set G_2 of all the networks involved in the strongest collinearity among the δ_j's is used.

Algorithm (B) proceeds as follows:

1. Determine the MSE of the best NN and of the simple average of the p networks on K_2
2. Consider all the p networks for the combination.
3. If this is the first execution (of this step), then
   - Form the U-OLC of all the networks, including a constant term, using K_1 to estimate the optimal combination-weights.
   - Determine the MSE of the U-OLC on K_2.
   - If the U-OLC yields the lowest MSE on K_2 compared to the best NN and the simple average, then STOP and return the current U-OLC.

   Else
   - Form the C-OLC of all the networks, including a constant term (unless a decision to exclude the constant has been taken earlier), using K_1 to estimate the optimal combination-weights.
   - Determine the MSE of the C-OLC on K_2.
   - If the C-OLC yields the lowest MSE on K_2 compared to the best NN and the simple average, then STOP and return the current C-OLC.
4. Construct G_2, the set of networks involved in the strongest collinearity among the δ_j's in the current combination, using BKW diagnostics.
5. If G_2 has two or more elements, then
   - If there are more than two networks in the current combination or if the constant term is not involved in the strongest collinearity, then drop the worst performer in G_2 from the combination. Go to Step 6.
   - Else if the constant is involved in the strongest collinearity, then drop it from the combination.
   - Else STOP and return the best performer between the best NN and the simple average.

Else (G_2 has one element which means that the constant term is involved in the strongest collinearity).
- If this NN is the best NN, then drop the constant term from the combination.
- Else drop the NN in G_2 from the combination.

6. If there are more than one NN left in the combination, then go to Step 3.
   Else STOP and return the best performer between the best NN and the simple average.