Selection of Weights for Sequential Feed-forward Neural Networks: An Experimental Study

Enrique Romero

Dept. de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
eromero@lsi.upc.edu

Abstract. The selection of the frequencies of the new hidden units for sequential Feed-forward Neural Networks (FNNs) usually involves a non-linear optimization problem that cannot be solved analytically. Most models found in the literature choose the new frequency so that it matches the previous residue as best as possible. Several exceptions to the idea of matching the residue perform an (implicit or explicit) orthogonalization of the output vectors of the hidden units. An experimental study of the aforementioned approaches to select the frequencies in sequential FNNs is presented. Our experimental results indicate that the orthogonalization of the hidden vectors outperforms the strategy of matching the residue, both for approximation and generalization purposes.

1 Introduction

The selection of a proper number of hidden units for Feed-forward Neural Networks (FNNs) is a very important issue in practical applications, and it has been widely discussed through the literature. In terms of the Bias/Variance decomposition, as far as the number of hidden units of an FNN grows, bias decreases and variance increases [1]. This happens because the flexibility of the model grows with the number of hidden units [2]. Sequential approximations (also named constructive or incremental) with FNNs allow to dynamically construct the network, starting from scratch, without setting *a priori* the architecture [3]. Hidden units are sequentially added until a satisfactory solution is found.

The well-known architecture of a fully connected FNN with one output linear unit and one hidden layer of units computes a function $f : \mathbb{R}^I \to \mathbb{R}$ defined as

$$f(x) = b_0 + \sum_{k=1}^{N} \lambda_k \varphi_k \left(\omega_k, x, b_k\right) \quad \omega_k \in \mathbb{R}^I \quad \lambda_k, b_0, b_k \in \mathbb{R},$$

where N is the number of units in the hidden layer and φ_k is the activation functions of the k-th hidden unit. For convenience, we refer to the weights in the first layer $\{\omega_k\}_{k=1}^N$ (the non-linear weights) as *frequencies* and to the weights in the second layer $\{\lambda_k\}_{k=1}^N$ (the linear weights) as *coefficients*. Regarding the *biases*, $\{b_k\}_{k=1}^N$ can be considered as part of the frequencies and b_0 as a coefficient.

Algorithm
  repeat
    Increase by 1 the number of hidden units N
    Pick an activation function for the new hidden unit
    Obtain the frequency of the new hidden unit
    Fix the frequency in the network
    Compute the coefficients
  until a certain stopping criterion is satisfied
end Algorithm

Fig. 1. A general algorithm for sequential FNNs.

Sequential algorithms have several shortcomings. First, it is not clear when to stop the addition of hidden units. Second, obtaining the frequencies of the new hidden units, although in theory is easier to solve than for a non-sequential procedure, is still a difficult non-linear optimization problem [4]. The coefficients can usually be computed analytically.

This work deals with the selection of weights of the new hidden units for sequential FNNs. In particular, we will focus on sequential approximations that obtain the new frequencies while keeping the previously selected frequencies fixed. That is, the previously obtained frequencies in the network are frozen in order to obtain the new one [5]. Most models found in the literature choose the new frequency so that it matches the previous residue as best as possible. Several exceptions to the idea of matching the residue perform an (implicit or explicit) orthogonalization of the output vectors of the hidden units, so that the new frequency is selected taking into account the interactions with the previously selected ones in order to minimize the global error. Theoretical results show that these two approaches may construct a sequence of output functions convergent to the target. However, the optimization problems posed at every step cannot be solved analytically in the general case. The difficulty lies on the selection of the frequencies, since the coefficients can usually be computed analytically.

An experimental study of the aforementioned approaches to select the frequencies in sequential FNNs is presented. The results in our experiments indicate that the orthogonalization of the hidden vectors outperforms the strategy of matching the residue, both for approximation and generalization purposes.

The rest of the paper is organized as follows. Several approaches to obtain the weights in sequential FNNs are discussed in section 2. The experiments can be found in section 3. Finally, section 4 outlines some directions for further research.

2 Selection of Weights in Sequential FNNs

Most of the sequential FNN models found in the literature keep the previously selected frequencies fixed and search only for the new frequency and the coefficients. Figure 1 shows the general structure of these sequential FNN schemes.

Their output can be defined as ($X_0 = 0$):

$$X_N = \sum_{k=1}^{N-1} \lambda_k^N h_{\omega_k} + \lambda_N^N h_{\omega_N}.$$

where X_N is a vector in $\mathbb{R}^L$, N is the number of hidden units, L is the number of patterns in a data set $D = \{x_1, \cdots, x_L\}$, h_{ω_k} is the output vector of the k-th hidden unit (with frequency ω_k) and λ_k^N is the coefficient of that hidden unit at step N. Output units are linear. The objective is to minimize the sum-of-squares error function $\|f - X_N\|_2^2$, where f is the target vector ($f = \{f_1, \cdots, f_L\}$).

The frequencies are obtained by optimizing a certain objective function, which somehow depends on the previously obtained frequencies:

1. Most of them choose the new frequency ω_N so that it matches the previous residue as best as possible [5]. By inner product properties, it is equivalent to say that
$$\omega_N = \arg\max_{\omega} |\langle f - X_{N-1}, h_\omega \rangle|^2 / \|h_\omega\|_2^2 \tag{1}$$
where $\langle \cdot, \cdot \rangle$ is the dot product in $\mathbb{R}^L$ and h_ω is the output vector of a hidden unit with frequency ω. After the selection of every frequency, the coefficients of the whole network can be either optimized or not. By inner product properties, the optimal coefficients can be obtained solving a linear equations system [6]. When the whole set of coefficients are not optimized, the coefficient of the new hidden unit with frequency ω_N is
$$\lambda_N^N = \langle f - X_{N-1}, h_{\omega_N} \rangle / \|h_{\omega_N}\|_2^2. \tag{2}$$
Some models sharing these underlying ideas are Projection Pursuit Regression [7] originally described in the Statistics field, Matching Pursuit [8] in the context of Signal Processing, or Projection Pursuit Learning Network [9] in the Neural Networks framework. Sometimes, convex approximations are constructed, as in the Incremental Linear Quasi-Parallel algorithm [10].
2. Several exceptions to the idea of matching the residue are the Orthogonal Least Squares Learning algorithm [11], Kernel Matching Pursuit with *pre-fitting* [12] or the Sequential Approximation with Optimal Coefficients and Interacting Frequencies [13], where an (implicit or explicit) orthogonalization of the output vectors of the hidden units is performed. The frequency ω_N is selected taking into account the interactions of h_{ω_N} with $h_{\omega_1}, \cdots, h_{\omega_{N-1}}$ in order to minimize $\|f - X_N\|_2^2$. There is no explicit intention to match the residue. In other words, a frequency ω_1 is considered better than ω_2 if h_{ω_1} allows, together with the previously selected frequencies (and after computing the optimal coefficients of the whole network), a better approximation of f than h_{ω_2}. The coefficients $\{\lambda_k^N\}_{k=1}^N$ are always optimal.

In the following, we will refer to *MPR*, *OCMPR* and *SAOCIF* as:

– *MPR* (Matching the Previous Residue): The new frequency ω_N is selected according to (1). The coefficients of the whole network are not optimized. That is, $X_N = X_{N-1} + \lambda_N^N h_{\omega_N}$, with λ_N^N computed according to (2).

Algorithm
  repeat
    Increase by 1 the number of hidden units N
    Pick an activation function for the new hidden unit
    repeat
      Assign a candidate frequency ω to the new hidden unit
      MPR/OCMPR: Set $\omega_N := \omega$ if $|\langle f - X_{N-1}, h_\omega \rangle|^2 / \|h_\omega\|_2^2$ is maximized
      SAOCIF: Compute the optimal coefficients $\{\lambda_k^N\}_{1 \leqslant k \leqslant N}$
      SAOCIF: Set $\omega_N := \omega$ if $\|f - X_N\|_2^2$ is minimized
    until there are no more candidate frequencies
    Fix the frequency ω_N in the network
    MPR: Compute the coefficient $\lambda_N^N = \langle f - X_{N-1}, h_{\omega_N} \rangle / \|h_{\omega_N}\|_2^2$
    OCMPR/SAOCIF: Compute the optimal coefficients $\{\lambda_k^N\}_{1 \leqslant k \leqslant N}$
  until a certain stopping criterion is satisfied
end Algorithm

Fig. 2. Algorithms for *MPR*, *OCMPR* and *SAOCIF*.

– *OCMPR* (Optimal Coefficients after Matching the Previous Residue): The new frequency ω_N is selected according to (1). Subsequently, the coefficients of the whole network are optimized by solving the associated linear equations system.
– *SAOCIF* (Sequential Approximation with Optimal Coefficients and Interacting Frequencies): The new frequency ω_N is selected taking into account the interactions of h_{ω_N} with $h_{\omega_1}, \cdots, h_{\omega_{N-1}}$ in order to minimize $\|f - X_N\|_2^2$. The coefficients are always optimal.

As it is well-known, an approximation scheme is said to have the convergence property if it is able to produce a sequence of output functions convergent to the target. *MPR*, *OCMPR* and *SAOCIF* have been proved to have the convergence property (see [14, 8] for *MPR*, [10, 5] for *OCMPR* or [13] for *SAOCIF*, for example). Unfortunately, these theoretical results cannot be directly applied in practice. The optimization problem posed in the general case cannot be solved analytically, and a suboptimal solution is searched heuristically. Note that the difficulty lies on the selection of the frequencies, since the coefficients can always be analytically computed.

3 Experiments

The experiments were performed on several artificial data sets. The main objectives of these experiments were to compare *MPR*, *OCMPR* and *SAOCIF* (see section 2). In particular, we studied the effect of their respective criteria of selection of frequencies both for approximation and generalization.

To that end, the algorithms in figure 2 were designed, as particular cases of that in figure 1. In the proposed algorithms, a number of candidate frequencies are selected at every step. The frequency finally chosen depends on the different objective functions of every algorithm.

$5,000$ random frequencies in $[-0.5,+0.5]\times[-0.5,+0.5]$ were assigned as candidate frequencies for every model, and a maximum of 50 hidden units were added to the initial architecture. The rest of parameters (activation functions, gain factor and other internal parameters) were chosen in a preprocessing procedure (see [15] for details).

3.1 Data Sets

We used the same data sets described in [9], where five non-linear functions $g_i : [0,1]^2 \to \mathbb{R}$ were defined as follows:

- $g_1(x_1,x_2) = 10.391\left((x_1-0.4)(x_2-0.6)+0.36\right)$.
- $g_2(x_1,x_2) = 24.234\left(r^2(0.75-r^2)\right)$, where $r^2=(x_1-0.5)^2+(x_2-0.5)^2$.
- $g_3(x_1,x_2) = 42.659\left((2+x_1)/20+Re(z^5)\right)$, where $z=x_1+ix_2-0.5(1+i)$.
- $g_4(x_1,x_2) = 1.3356(1.5(1-x_1)+e^{2x_1-1}\sin(3\pi(x_1-0.6)^2)+ e^{3(x_2-0.5)}\sin(4\pi(x_2-0.9)^2)\ )$.
- $g_5(x_1,x_2) = 1.9\left(1.35+e^{x_1}\sin(13(x_1-0.6)^2)e^{-x_2}\sin(7x_2)\right)$.

225 pairs (x_1,x_2) of values were generated from the uniform distribution in $[0,1]^2$. These data were used for all five functions in order to generate five noise-free training sets:

$$\text{HEA}n\text{-NF} = \{(x_1^i, x_2^i, g_n(x_1^i,x_2^i))\}_{i=1,\cdots,255}$$

where $n\in\{1,2,3,4,5\}$. In addition, another five training data sets were generated adding independent and identically distributed Gaussian noise:

$$\text{HEA}n\text{-WN} = \{(x_1^i, x_2^i, g_n(x_1^i,x_2^i)+0.25\varepsilon^i)\}_{i=1,\cdots,255}$$

where $\varepsilon^i \sim \mathcal{N}(0,1)$. The test set was built sampling every function on a regularly spaced grid on $[0,1]^2$ with $10,000$ points. In summary, 10 training sets (5 noise-free and 5 noisy versions) and 5 test sets were generated in [9] for the 5 aforementioned functions. These data sets have been widely used in the literature (see [5, 16, 17], for example).

In our experiments, we constructed 10 training sets for every function, each containing 225 points, changing the initial seed of the random function for the uniform distribution (the noise-free data sets). Similar to [9], 10 noisy training data sets were generated in the same way. For every function, the test set in [9] was used as a validation set for the adjustment of the parameters in the preprocessing procedure. For the final results, a new test set was constructed, with an offset of 0.0025 with respect to the input points in the original test set.

In summary, 100 training sets (50 noise-free and 50 noisy versions), 5 validation sets and 5 test sets were generated for the five aforementioned functions. For every function and version (noise-free and noisy), every model was trained with every one of the 10 different training sets, and tested on the test set.

	Test Error			Num. Hidden Units		
Data Set	*MPR*	*OCMPR*	*SAOCIF*	*MPR*	*OCMPR*	*SAOCIF*
HEA1-NF	0.06 (0.1)	**0.00 (0.0)**	**0.00 (0.0)**	49.3	20.3	9.6
HEA2-NF	29.81 (2.6)	1.18 (0.5)	**0.18 (0.1)**	49.1	47.1	44.1
HEA3-NF	694.11 (27.2)	31.41 (6.9)	**5.57 (2.2)**	49.2	45.0	49.1
HEA4-NF	69.29 (6.9)	24.32 (1.8)	**9.80 (3.7)**	49.4	45.8	47.3
HEA5-NF	49.22 (7.8)	14.79 (3.4)	**7.76 (1.0)**	49.8	49.4	46.2
HEA1-WN	14.37 (3.1)	**10.70 (1.6)**	15.37 (2.9)	39.1	4.5	4.4
HEA2-WN	110.85 (6.8)	99.13 (7.1)	**89.47 (6.3)**	30.9	19.1	15.4
HEA3-WN	704.35 (50.5)	303.65 (18.9)	**256.01 (32.6)**	49.9	33.6	31.6
HEA4-WN	213.67 (8.0)	202.91 (14.7)	**120.58 (11.5)**	47.0	27.5	17.4
HEA5-WN	174.63 (9.0)	211.82 (18.0)	**173.35 (5.7)**	35.0	23.7	19.4

Table 1. HEA data sets: Squared test set error and number of hidden units for *MPR*, *OCMPR* and *SAOCIF* in the tested data sets.

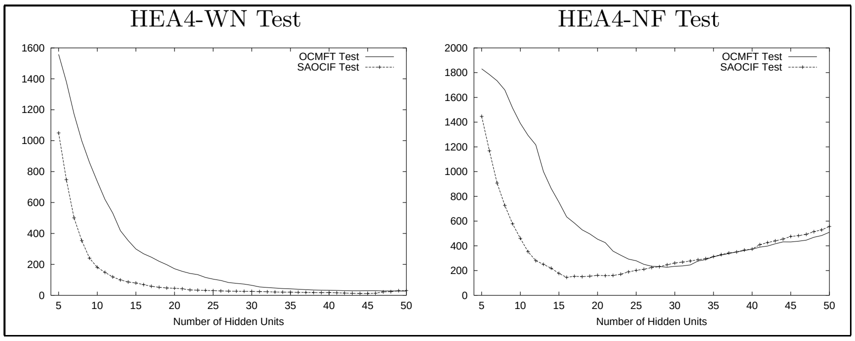

Fig. 3. Evolution of the average test error of *OCMPR* and *SAOCIF* with respect to the number of hidden units for the HEA4-NF (left) and HEA4-WN (right) data sets.

3.2 Results

Results are shown in table 1 as the average of the minimum squared test set errors. Figures in boldface indicate the best results. Numbers in brackets are $\hat{\sigma}_n/\sqrt{n}$, the standard errors[1] estimated from the sample standard deviation $\hat{\sigma}_n$. The average number of hidden units where these minima are achieved is also shown. For the HEA1-NF data set, *SAOCIF* stopped with a fewer number of hidden units because the training error was less than 0.000001, so that it made no sense adding new hidden units. As an example, figure 3 shows a comparison of the evolution of the average test errors of *OCMPR* and *SAOCIF* with respect to the number of hidden units for the HEA4 data sets. The training errors were similar, although in a different scale, to left plot in figure 3.

[1] Under normality assumptions, the confidence interval can be computed from this value. For example, the deviation of the true value from the observed mean $\overline{x_n}$ will be less than $1.96\hat{\sigma}_n/\sqrt{n}$ with a probability of 0.95.

Regarding the overall behavior, *SAOCIF* obtains better results than *OCMPR*, which in turn compares favorably with *MPR*. This fact can be understood by looking at the number of hidden units of the obtained results:

1. For noise-free data sets the number of hidden units is around 50, the maximum number of allowed hidden units. This is due to the fact that overfitting was not observed during the learning process with these data sets (see left plot in figure 3). Therefore, the best results are obtained by those models that are able to fit more accurately the data. The scheme of selection of frequencies of *SAOCIF* allows to find better approximations with the same number of hidden units as *OCMPR* or *MPR*. The same happens when *OCMPR* is compared to *MPR*.
2. For noisy data sets, there is a high correlation between the number of hidden units and the goodness of the model: those models that attain their minima with less hidden units usually obtain better results. *SAOCIF* obtains simpler models (in terms of the number of hidden units), with the same training error, than *OCMPR*. According to the Bias/Variance trade-off, the minimum test set error should be smaller for *SAOCIF* than for *OCMPR*, as observed. The same happens with respect to *MPR*.

Therefore, the interacting frequencies selected by *SAOCIF* show a better behavior than frequencies selected so as to match the previous residue, both for approximation and generalization purposes. It can be observed that the differences among methods are greater for noise-free data sets than for noisy ones.

4 Future Work

New experiments can be done by changing the heuristic to select the new frequencies in the tested models. It would be interesting to test if other heuristics also maintain the observed differences. Some variation of the Breeder Genetic Algorithm [18], for example, could be useful to that end.

A different point of view can be introduced in sequential FNNs if we reconsider the goodness of the previously selected frequencies. Comparing the addition of hidden units in a sequential FNN with the selection of features in the Sequential Forward Selection procedure for Feature Selection [19], we can see that they share the same general ideas, although applied to different objects. Whereas sequential FNNs applies the forward selection to the hidden units, Sequential Forward Selection applies it to the features. Therefore, other Feature Selection search procedures can be applied to the construction of FNNs.

Acknowledgements

This work was supported by Consejo Interministerial de Ciencia y Tecnología (CICYT), under projects DPI2002-03225 and CGL2004-04702-C02-02.

References

1. Geman, S., Bienenstock, E., Doursat, R.: Neural Networks and the Bias/Variance Dilemma. Neural Computation **4** (1992) 1–58
2. Barron, A.R.: Approximation and Estimation Bounds for Artificial Neural Networks. Machine Learning **14** (1994) 115–133
3. Kwok, T.Y., Yeung, D.Y.: Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems. IEEE Transactions on Neural Networks **8** (1997) 630–645
4. Kůrková, V.: Incremental Approximation by Neural Networks. In Karny, M., Warwick, K., Kůrková, V., eds.: Dealing With Complexity: A Neural Network Approach. Springer-Verlag, London (1998) 177–188
5. Kwok, T.Y., Yeung, D.Y.: Objective Functions for Training New Hidden Units in Constructive Neural Networks. IEEE Transactions on Neural Networks **8** (1997) 1131–1148
6. Achieser, N.I.: Theory of Approximation. Frederick Ungar Pub. Co., New York (1956)
7. Friedman, J.H., Stuetzle, W.: Projection Pursuit Regression. Journal of the American Statistical Association **76** (1981) 817–823
8. Mallat, S.G., Zhang, Z.: Matching Pursuits with Time-Frequency Dictionaries. IEEE Transactions on Signal Processing **41** (1993) 3397–3415
9. Hwang, J.N., Ray, S.R., Maechler, M., Martin, D., Schimert, J.: Regression Modelling in Back-Propagation and Projection Pursuit Learning. IEEE Transactions on Neural Networks **5** (1994) 342–353
10. Kůrková, V., Beliczyński, B.: Incremental Approximation by One-Hidden-Layer Neural Networks. In: International Conference on Artificial Neural Networks. Volume 1. (1995) 505–510
11. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Transactions on Neural Networks **2** (1991) 302–309
12. Vincent, P., Bengio, Y.: Kernel Matching Pursuit. Machine Learning **48** (2002) 165–187 Special Issue on New Methods for Model Combination and Model Selection.
13. Romero, E., Alquézar, R.: A New Incremental Method for Function Approximation using Feed-forward Neural Networks. In: International Joint Conference on Neural Networks. Volume 2. (2002) 1968–1973
14. Jones, L.K.: On a Conjecture of Huber Concerning the Convergence of Projection Pursuit Regression. The Annals of Statistics **15** (1987) 880–882
15. Romero, E.: Learning with Feed-forward Neural Networks: Three Schemes to Deal with the Bias/Variance Trade-off. PhD thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Spain (2004)
16. Treadgold, N.K., Gedeon, T.D.: Exploring Constructive Cascade Networks. IEEE Transactions on Neural Networks **10** (1999) 1335–1350
17. Ma, L., Khorasani, K.: New Training Strategies for Constructive Neural Networks with Application to Regression Problems. Neural Networks **17** (2004) 589–609
18. Mühlenbein, H., Schlierkamp-Voosen, D.: Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization. Evolutionary Computation **1** (1993) 25–49
19. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers (1998)