Neural Networks with Orthogonalised Transfer Functions

P. Strumillo[1], W. Kaminski[2]

[1]Institute of Electronics, Technical University of Lodz, POLAND
e-mail: pstrumil@ck-sg.p.lodz.pl

[2]Faculty of Process and Environmental Engineering,
Technical University of Lodz, POLAND
e-mail: kaminski@ck-sg.p.lodz.pl

Approximation capabilities of single non-linear layer networks, that feature a single global minimum of the error function are addressed. Bases of different transfer functions are tested (Gaussian, sigmoidal, multiquadratics). These functions are orthogonalised in an incremental manner for training and restored back to the original basis for network deployment. Approximation results are given for a benchmark ECG signal. Results of incremental training with basis orthogonalisation are also shown for 2D approximations.

1. Introduction

Multilayer perceptron (MLP) networks, of feed-forward type, are the most popular artificial neural networks structures for generating many input – many output nonlinear mappings. In particular, it has been proved that networks containing at least one layer of continuous discriminatory functions (e.g., sigmoidal functions) are capable of approximating any continuous mappings. Nevertheless, networks with more than one hidden layer are of frequent use that, in some applications, outperform single-hidden layer networks, i.e., yield more precise mappings for a lower overall number of nonlinear nodes. For such networks, however, the training problem becomes a complicated nonlinear optimisation task, defined in a multidimensional space, since the cost function assumes an equation of nested nonlinear expressions:

$$\varepsilon(\boldsymbol{w}) = \left\| y - f(\boldsymbol{x},\boldsymbol{w}) \right\|^2 = \left\| y - \varphi\left(\sum_{H_i} w^i \varphi\left(\sum_{H_{i-1}} w^{i-1} \varphi\left(\sum_{H_{i-2}} \ldots s\left(\ldots w^1 \boldsymbol{x} \right) \right) \right) \right) \right\|^2 \tag{1}$$

where: y – is the approximated function, $f(\cdot)$ – is the network generated approximation of y, w^i – are network connection weights in H_i-th hidden layer (counting from the input), and $\varphi(\cdot)$ – is the transfer function of an individual neural node (usually of sigmoidal type). Search for the optimum weight set $\boldsymbol{w}^*$ that minimises $\varepsilon(\boldsymbol{w})$ can only be performed in an iterative manner, e.g., of gradient decent type or other procedures like evolutionary algorithms. These algorithms, however, can be trapped in local minima of the error function (1). Thus, it is recommended to repeat the training runs many times (each time starting from different weight space location $\boldsymbol{w}_0$) and choosing best ***w**** that minimises (1). Such training schemes significantly increase demand for computing power and are not practical in embedded systems or real time processing tasks.

In this study we concentrate our interest on networks with a single nonlinear layer, that have strong theoretical background in linear approximation theory, thus are simpler to design, analyse and optimise for applications at hand.

2. $1\frac{1}{2}$ layer networks

Feed-forward networks with a single non-linear hidden layer and a single linear output layer are frequently called "$1\frac{1}{2}$ layer networks". In the language of function approximation theory such an approximation scheme is termed discrete linear approximation, since an unknown function $y: \boldsymbol{R}^N \rightarrow \boldsymbol{R}$ is approximated by a function f_m being a linear span of a set of m basis functions $\varphi_k(\cdot): \boldsymbol{R}^N \rightarrow \boldsymbol{R}$, $k=1, 2, ..., m$. This type of mapping is given by:

$$f_m(x) = \sum_{k=1}^{m} w_k \varphi_k(x) + w_0 \tag{2}$$

For such a network the approximation task consists of the following steps:

i. provide samples $\left\{(\mathbf{x}_i, y_i) \in R^N \times R\right\}_{i=1}^{P}$, i.e., P training pairs of an unknown function y belonging to some normed space of functions Y,
ii. define a family Φ of basis functions $\varphi_k \in \Phi$,
iii. use appropriate number of m basis functions from family Φ and find a corresponding set of weights w_k, $k=1, 2, ..., m$ so that the error $\varepsilon = \left\| f - f_m \right\|$, i.e., the distance ε, according to a norm $\|\cdot\|$, (e.g., Euclidean) between an unknown function f and its estimate f_m falls within an acceptable margin.

Searching for network weights that minimise the error is a linear optimisation problem with m unknowns and P linear equations (so called normal equations). For an approximation problem $P>m$ (for $P=m$ we have an interpolation) the task of finding optimum $\boldsymbol{w}$ is overdetermined and can be solved either by calculation of matrix pseudoinverse or by an iterative error-correction learning rule such as the LMS algorithm. Note, that for linear optimisation, the problem of local minima does not exist.

In step (ii) of the algorithm, the family Φ of basis functions φ_k is predetermined (i.e., the type of basis functions and their number). Special families of basis functions, termed wavelets, have been devised, mainly for signal processing applications [1]. A wavelet is a function $\psi \in L^2(\boldsymbol{R})$ with zero average that satisfies certain admissibility conditions (e.g., sufficient decay of its modulus). A family of functions $\psi_{t,s} = \frac{1}{\sqrt{s}}\psi\left(\frac{x-t}{s}\right)$ can be obtained by translating and dilating a prototype wavelet

$\psi_{0,1}$. Approximations using discrete wavelet decomposition (that use dyadic scaling for discrete values of the dilation factor) are suitable predominantly for signal processing applications, and generate redundant representations. They can be considered as a special case of $1\frac{1}{2}$ layer neural networks. The latter, however, as opposed to the wavelet basis, do not impose strong conditions on the basis transfer functions and can find applications in data classification applications [2].

3. Orthogonalisation of network basis functions

Suitability of the employed basis functions, can only be verified after determination of an optimum weight set in (2), that requires $O(Pm^2 + 5m^3)$ summations and multiplications (matrix pseudoinverse). Search for best basis involves multiple computation of weights. This computing cost can be dramatically reduced if the set of basis functions is orthogonal. Then, the optimum weight vector can be calculated from a simple formula [3]:

$$\boldsymbol{w}^* = \begin{bmatrix} \frac{\langle f, u_1 \rangle}{\|u_1\|^2} & \frac{\langle f, u_2 \rangle}{\|u_2\|^2} & \cdots & \frac{\langle f, u_m \rangle}{\|u_m\|^2} \end{bmatrix}^T \tag{3}$$

where $\langle \cdot,\cdot \rangle$ is the inner product operator defined for discrete samples of approximated function f and orthogonal functions u_k. Orthogonality means that $\langle u_i, u_j \rangle \neq 0$ for $i=j$ and zero otherwise. More importantly, for a constructive training scheme, each new orthogonal basis and its corresponding weight is computed independently from other weights that were determined earlier (see (3)).

The Fourier series, with complex exponential functions $u_n = \left\{ \frac{1}{\sqrt{2\pi}} e^{jnx} \right\}$ ($n \in \boldsymbol{Z}$ and $\boldsymbol{Z}$ is a set of integers) is an example of a set of orthogonal basis functions defined in $L^2(-\pi, \pi)$. Thus, Fourier series can be seen as a special case of $1\frac{1}{2}$ layer neural network. Fourier basis have excellent concentration is spectrum domain and poor concentration in spatial (or time) domain. At the other extreme, there is an orthogonal basis of Dirac distributions $\{\delta(x-n)\}$, $n \in \boldsymbol{Z}$ that feature excellent space concentration and poor frequency concentration. Between these two extremes there is a special basis, i.e., the Gaussian basis. Interestingly, Gaussian kernel posses best joint location-frequency resolution and meets with equality the uncertainty bound $\Delta x \Delta \omega \geq \frac{1}{2}$ set by Heisenberg, i.e., the area of the space-frequency box $\Delta x \Delta \omega$, defining function joint resolution reaches minimum. Gaussian kernel is the most frequently used transfer function in radial basis functions (RBF) neural networks.

Unfortunately, Gaussian basis and other families of transfer functions like general multiquadratics and thin-plate splines, or sigmoidal functions are not orthogonal. The following theorem provides functional equivalence between linearly independent basis and orthogonal basis defined in a normed space of functions [3].

Theorem (due to Schmidt): *For a set of linearly independent functions $\{\varphi_i\}_{i=1}^{m}$ defined in a Hilbert space, there exists a set of orthogonal functions $\{u_i\}_{i=1}^{m}$, such that any function u_i is a linear combination of $\{\varphi_i\}_{i=1}^{m}$ and vice versa, any φ_i is a linear combination of orthogonal functions $\{u_i\}_{i=1}^{m}$.*

This theorem, has an important practical application for $1\frac{1}{2}$ layer neural network constructive training methods. Earlier, we have proposed a procedure for training RBF networks in which training is carried out on orthogonal basis, that afterwards is restored back to the original basis [4, 5]. Considerable savings in computing time have been gained. Here, we propose and test the following procedure for training $1\frac{1}{2}$ layer networks comprising various transfer functions (i.e., that are not limited to basis of radial symmetry):

1. *select the family of basis functions (e.g., sigmoids) and start from the network containing a single basis function,*
2. *orthogonalise the current basis function (using the Gram-Schmidt orthonormalisation procedure),*
3. *compute the corresponding weight (using (3)),*
4. *check the network error,*
5. *if the error is too large add new basis and go to step (2), otherwise go to step (6),*
6. *use the Schmidt's theorem to recompute weights for the original basis and stop.*

In fact, a modified version of the standard Gram-Schmidt procedure giving better numerical stability, in which orthogonal rather than orthonormal basis has been used in the proposed procedure. More details are given in [5].

We provide a set of computing examples that illustrate shapes of transfer functions obtained by means of orthogonalising linearly independent basis for $1\frac{1}{2}$ layer neural networks. For one dimensional approximation examples, three families of basis functions are considered: the sigmoidal function $\varphi(x) = \frac{1}{1+e^{-\beta(x-\theta)}}$ and two radial functions, namely the Gaussian $\varphi(r) = e^{-\frac{r^2}{2\sigma^2}}$ and multiquadratics function $\varphi(r) = (b^2 + r^2)^{\alpha}$, $0 < \alpha < 1$, where $r = \|x - t\|$ and t is the centre of radial symmetry. For a 2D approximation example the Gaussian is used, that is a separable function, i.e., it can be expressed as a product 1D Gaussians defined separately for each of the dimension. An overview of different basis functions is given in [2].

3.1 ECG signal approximation

A single cycle of an ECG signal is used as a benchmark function for showing approximation capabilities of $1\frac{1}{2}$ layer neural networks containing different basis. The proposed orthogonalisation procedure is used for network training in which the number of basis is incremented one at a time. Fig. 1 illustrates example shapes of the orthogonal functions obtained for ECG signal approximation task.

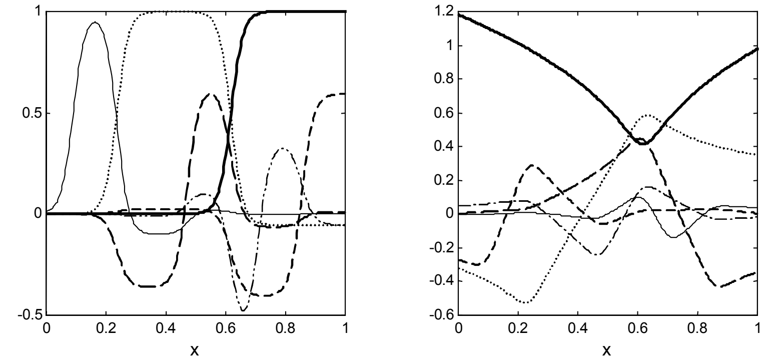

Fig. 1. Families of orthogonal functions obtained from two different basis functions: sigmoidal functions (left panel) and multiquadratics functions (right panel). Thick solid lines correspond to the first (non-orthogonalised) basis used in the incremental approximation scheme.

In Fig. 2 graphical representation of the approximation precision obtained by means of orthogonalised basis is shown. Table 1 collects approximation results for the tested basis functions. Note, that all three bases yield very good and similar approximation accuracy of which the best is obtained for multiquadratics basis. Parameters of the defined basis functions, that were used are: $\beta=50$ for the sigmoid, $\sigma=0.05$ for the Gaussian, and $b=0.1$, $\alpha=0.2$ for the multiquadratics. Centres of the radial functions and thresholds for the sigmoidal functions were selected by means of the golden ratio partition of the interval (*0, 1*), i.e., $t_k = k \cdot z - trunc(k \cdot z)$, where $z = \frac{\sqrt{5}-1}{2}$ and $k=1, 2, ..., m$.

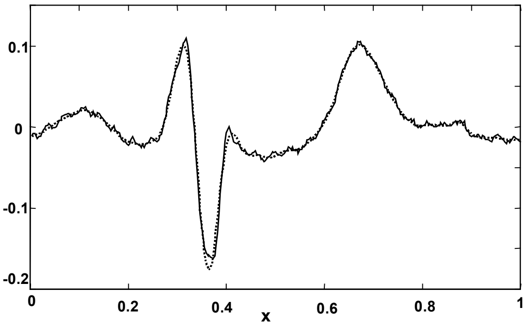

Fig. 2. ECG signal (solid line) and its approximation (dotted line) obtained by means of one hidden layer network of 30 multiquadratic basis functions.

Table. 1 Approximation results

Basis (m=30)	Mean square error
mutiquadratics	$2.15\cdot10^{-4}$
gaussian	$2.25\cdot10^{-4}$
sigmoidal	$4.21\cdot10^{-4}$

3.2 Approximation in two dimensions

As a benchmark function a complicated interaction two dimensional function is used [6]: $f(x_1,x_2)=1.9\left(1.35+e^{x_1}\, sin\left(13(x_1-0.6)^2\right)e^{-x_2}\, sin(7x_2)\right)$. Its plot is depicted in the left panel of Fig. 3.

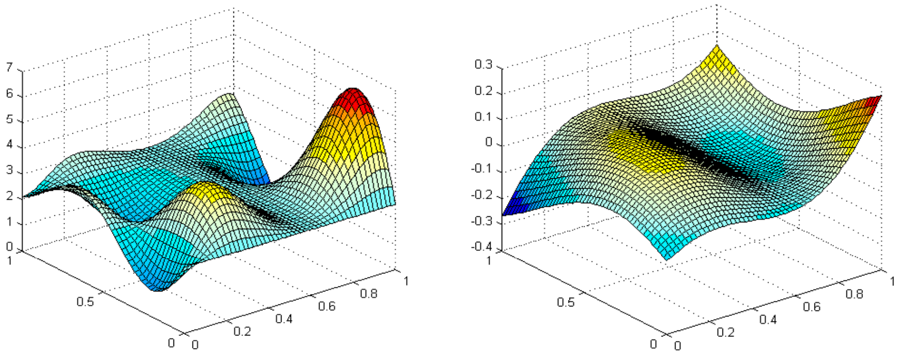

Fig. 3. An approximated two dimensional function (left panel) and one of the orthogonal functions obtained from 2D Gaussian basis.

In the applied constructive function approximation scheme, 2D Gaussians were added one by one with centres selected randomly on $[0, 1]\times[0, 1]$ domain. Approximation accuracy of $9.91\cdot10^{-2}$ has been obtained for 30 orthogonalised basis. An example of a 2D function from the orthogonal set obtained from the Gaussian basis is shown in the right panel of Fig. 3.

4. Conclusions

Universal approximation properties of $1\frac{1}{2}$ layer neural networks with non-polynomial basis in the hidden layer have been demonstrated. In theory, networks with more than one nonlinear layer can yield better performance for a lower overall number of weights. On the other hand, incremental training scheme that includes orthogonalisation of a single layer basis, enables fast, one shot computations of network weights. Moreover, by using, the Schmidt's theorem a functional equivalence between network transfer functions and orthogonal basis is guaranteed. The orthogonal basis set can be used for fast training and traditionally used transfer functions (e.g., sigmoidal) can be used for network deployment.

References

1. S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1998.
2. W. Duch, N. Jankowski, Survey of neural transfer functions*, Neural Computing Surveys*, 2, pp. 63-212.
3. W. Rudin, *Real and complex analysis*, McGraw Hill, 1974.
4. W. Kaminski, P. Strumillo, Kernel orthonormalisation in Radial Basis Function neural networks, *IEEE Trans. Neural Networks*, vol.8, no.5, 1177-1183, 1997.
5. P. Strumillo, W. Kaminski, Ortho-system of kernels in RBF Neural Networks, *Proc. of the International ICSC/IFAC Symposium on Neural Computation*, pp. 891-895, Vienna, 1998.
6. T-Y. Kwok, D-Y. Yeung, Objective functions for training new hidden units in constructive neural networks, *IEEE Trans. Neural Networks*, vol.8, no.5, pp.1131-1148, 1997.