ADAPTIVE MULTIDIMENSIONAL SPLINE NEURAL NETWORK FOR DIGITAL EQUALIZATION

Mirko Solazzi (*), Aurelio Uncini (**)

(*) Dipartimento di Elettronica e Automatica - University of Ancona
Via Brecce Bianche, 60131 Ancona-Italy.
email: mrksol@eealab.unian.it

(**) Dipartimento INFOCOM - University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome - Italy.
email: aurel@infocom.uniroma1.it

ABSTRACT

This paper presents a new neural architecture suitable for digital signal processing applications. The architecture, based on adaptable multi-dimensional activation functions, allows to collect information from the previous network layer in aggregate form. In other words the number of network connections (structural complexity) can be very low respect to the problem complexity. This fact, as experimentally demonstrated in the paper, improve the network generalization capabilities and speed up the convergence of the learning process.
A specific learning algorithm is derived and experimental results, on channel equalization, demonstrate the effectiveness of the proposed architecture.

1. INTRODUCTION

The classical feedforward neural architectures with sigmoidal activation functions is able to approximate any continuous function of several variables. Anyway, Vitushkin in [1] proved that not all the functions with a given degree of complexity can be represented in simple way by means of functions with a lower degree of complexity. Recently a new interest in adaptive activation functions has arisen. In [2] the authors have involved polynomial functions, which allow to increase the neuron complexity and reduce the size of the network. This solution implies some drawbacks, principally with the learning phase coefficients adaptation, due to spurious minima and maxima. Later adaptive spline activation function was introduced [3]. With this approach each neuron is characterized by a different activation function whose shape can be modified through some control points. Further works [4], [5] demonstrated that such neuron architecture can improve approximation and generalization abilities of the entire network.
In this paper more general activation function is introduced; in particular we first define multidimensional neuron and the related network architecture. Next we implement such activation function using cubic spline interpolation of some

control points. Usually the introduction of a too large number of adaptive coefficients leads to a poor generalization capacity. But, this is not true in practical situation when only few control points for each neuron are locally updated to obtain a good approximation and, at the same time, the number of hidden units and connection weights are drastically reduced.
The proposed neural architecture is then applied for the digital equalization task to reduce the effects of intersymbol interference (ISI) and nonlinear channel distortions [6]-[10]. Experimental trials on different channel models and symbol alphabets confirmed the validity of this architecture, from the points of view of both the average symbol error and the convergence speed.

2. MULTIDIMENSIONAL SPLINE ACTIVATION FUNCTIONS

In a feed-forward neural network the output $y_j^{(l)}$ of neuron *j* in the layer *l*, is computed by passing the net input through a non-linear activation function $\Phi^{(l)}(x)$ that is generally squashing and sigmoidal.

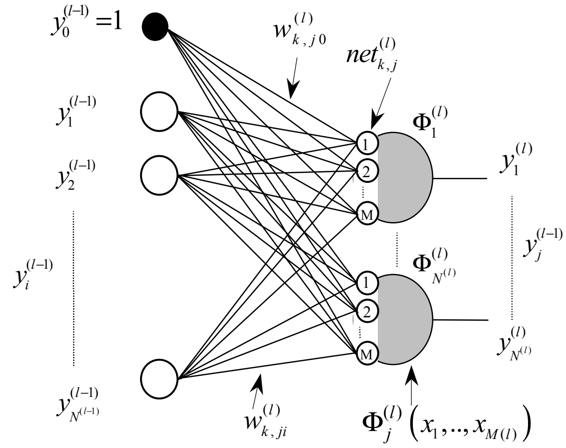

Figure 1 Architecture of a neural network with multi-dimensional neurons.

We can think a generalization of this structure, introducing more general activation functions. In particular multi dimensional functions is now considered. Let $\Phi^{(l)}(x_1, x_2, ..., x_M)$ a generic *M*-variables function, following the notation of Figure 1, we define multiple net inputs as:

$$net_{k,j}^{(l)} = \sum_{i=0}^{N^{(l-1)}} w_{k,ji}^{(l)} y_i^{(l-1)} = \mathbf{w}_{k,j}^{(l)} \mathbf{y}^{(l-1)} \qquad (1)$$

where $w_{k,ji}^{(l)}$ is the connection weight from unit *i* in the upper layer to unit *j* for the *k*-th input (*k*=1, …, *M*) of function $\Phi^{(l)}(.)$ and $N^{(l)}$ represents the number of neurons in the layer *l*. The activation $y_j^{(l)}$ of neuron *j* in the layer *l*, is computed as:

$$y_j^{(l)} = \Phi_j^{(l)}\left(net_{1,j}^{(l)}, net_{2,j}^{(l)}, ..., net_{M,j}^{(l)}\right) = \Phi_j^{(l)}\left(\mathbf{w}_{1,j}^{(l)}\mathbf{y}^{(l-1)}, \mathbf{w}_{2,j}^{(l)}\mathbf{y}^{(l-1)}, .., \mathbf{w}_{M,j}^{(l)}\mathbf{y}^{(l-1)}\right); \qquad (2)$$

in other words, the activation function's inputs are collected as *M*-dimensional subset of the linear combiner outputs $net_{k,j}^{(l)}$.

Adaptive activation functions allow to increase the neuron complexity and reduce the size of the network. In recent works definition and implementation of adaptive spline activation functions is made [3]-[5]. Our idea consists in realizing multidimensional activation functions as hyper-surface interpolation of some control points using higher order interpolants. In particular piecewise cubic spline are here employed in order to render the hyper-surface continuous in its first and second partial derivatives. The *M*-dimensional formulation is developed by first considering 1D, 2D, …, *M*-1D splines in the special case where the control points lie on a regular grid in the space of M dimension. The entire approximation is represented through the concatenation of local functions $h_{n;j}(u_1, u_2, .., u_M)$, each related to centers $\alpha_{n;k,j}$ and controlled by 4^M control points. General formulation of basis functions $h_{n;j}(u_1, u_2, .., u_M)$ is given by:

$$h_{n;j}(u_1, u_2, .., u_M) = \sum_{s_1=0}^{3}\sum_{s_2=0}^{3}\cdots\sum_{s_M=0}^{3} \rho_{s_1,s_2,...,s_M;j} u_1^{s_1} u_2^{s_2} \cdot u_M^{s_M} \qquad (3)$$

where $\rho_{s_1,s_2,...,s_M;j}$ are coefficients depending on control points of the *j*-th basis function.

Synthetic formulation for 1D and 2D functions can be derived in matrix form :

$$h_{n;j}(u_1) = \mathbf{T}_1 \cdot \mathbf{M} \cdot \mathbf{P}_{j[1]}^{(n)};$$

$$h_{n;j}(u_1, u_2) = \mathbf{T}_2 \cdot \mathbf{M} \cdot \left(\mathbf{T}_1 \cdot \mathbf{M} \cdot \mathbf{P}_{j[2]}^{(n)}\right)^T; \qquad (4)$$

where : $\mathbf{T}_k = \begin{bmatrix} u_k^3 & u_k^2 & u_k & 1 \end{bmatrix}$ with $0 \le u_k \le 1, \ \forall k : 1 \le k \le M$

and $\mathbf{P}_{j[M]}^{(n)}$ is a *M*-dim structure that collect the control points of the *j*-th basis function (see Figure 2):

$$\mathbf{P}_{j[1]}^{(n)} = \begin{bmatrix} P_j^{(n_1-1)} & P_j^{(n_1)} & P_j^{(n_1+1)} & P_j^{(n_1+2)} \end{bmatrix}^T;$$

$$\mathbf{P}_{j[2]}^{(n)} = \begin{bmatrix} P_j^{(n_1-1,n_2-1)} & P_j^{(n_1-1,n_2)} & P_j^{(n_1-1,n_2+1)} & P_j^{(n_1-1,n_2+2)} \\ P_j^{(n_1,n_2-1)} & P_j^{(n_1,n_2)} & P_j^{(n_1,n_2+1)} & P_j^{(n_1,n_2+2)} \\ P_j^{(n_1+1,n_2-1)} & P_j^{(n_1+1,n_2)} & P_j^{(n_1+1,n_2+1)} & P_j^{(n_1+1,n_2+2)} \\ P_j^{(n_1+2,n_2-1)} & P_j^{(n_1+2,n_2)} & P_j^{(n_1+2,n_2+1)} & P_j^{(n_1+2,n_2+2)} \end{bmatrix}; \quad \ldots. \tag{5}$$

The matrix $\mathbf{M}$ determines the characteristic of interpolant hyper-surface.

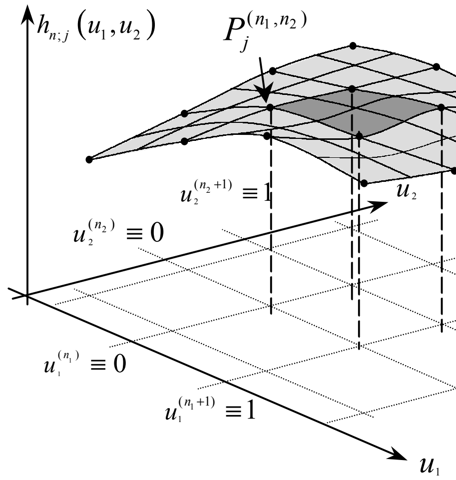

Figure 2 Example of control points 2D cubic spline interpolation

Referring to the network structure illustrated in Figure 1, we derive an algorithm to compute the activation of each neuron *j* during presentation of *n*-th example. Localization of net input vector into the grid is successively performed by considering control points in each directions which are equally spaced and symmetrically disposed respect to the axis origin. Let Δx_k and Q_k the fixed step between points and the number of points along the *k*-th dimension respectively, we have:

$$z_{k,j} = \frac{net_{k,j}}{\Delta x_k} + \frac{Q_k - 2}{2};$$

with constraints

$$z_{k,j} = \begin{cases} 1 & \text{if } z_{k,j} < 1 \\ z_{k,j} & \text{if } 1 \le z_{k,j} \le Q_k - 3 \\ Q_k - 3 & \text{if } z_{k,j} > Q_k - 3 \end{cases} \tag{6}$$

Constraints are necessary to keep the net input within the active volume that encloses the control points. Next step is to separate indices $z_{k,j}$ into integer and fractional part using the floor operator $\lfloor . \rfloor$:

$$n_{k;j} = \lfloor z_{k,j} \rfloor \quad \text{and} \quad u_{k,j} = z_{k,j} - n_{k;j} \tag{7}$$

The integer parts indicated as $n_{k;j}$ are used to address local control points of neuron *j*, while the fractional parts $u_{k,j}$ are passed as normalized inputs to multidimensional cubic spline function $h_{n;j}$. After training with N_e examples, activation function for *j*-th neuron can be expressed as:

$$\mathbf{\Phi}_j\left(\mathbf{w}_{1,j}\mathbf{x},..,\mathbf{w}_{M,j}\mathbf{x}\right) = \mathbf{\Phi}_j\left(net_{1,j},..,net_{M,j}\right) = \sum_{n=1}^{Ne} h_{n;j}\left(u_{1,j},u_{2,j},...,u_{M,j}\right) \tag{8}$$

3. EXPERIMENTAL RESULTS

In a digital communication system the transmitter sends a sequence $\{S[n]\}$ of symbol waveforms, extracted from a given alphabet at discrete time intervals. Going through the transmission channel, symbols are affected by both linear and nonlinear distortions. The linear part of the channel is commonly modeled by a finite impulse response filter [6]. The channel output $\{\hat{r}[n]\}$ is also corrupted by additive thermal noise, usually modeled as an additive Gaussian white process $\{q[n]\}$ with zero mean.

The received sequence $\{r[n] = \hat{r}[n] + q[n]\}$ must be processed to produce an estimate of the transmitted symbol $S[n]$. The equalizer structure is a *decision-feedback* architecture. The network input is constituted by feedforward delay chain of order *m*, and a feedback stage of order *l*. The filtering process introduces a decision delay *d*.

In our simulations, a multilayer neural network with two layers of connections and 2D neurons was employed. Inputs, outputs and network connection weights were defined as complex values. Activation functions and learning algorithm were reconsidered for the complex domain in accordance with definitions in [12]. The performances of the equalizer (SP2D_2) were evaluated for different channel models and compared with neural equalizers based on fixed sigmoidal (NN_20) and adaptive mono-dimensional spline (SP1D_6) activation function (the figures denote the number of hidden units). In every test the initial weight values were randomly fixed, while each adaptive activation function was formed by a grid of 30x30 control points, initially sampled from a signed two-dimensional sigmoid, with $\Delta x_1 = \Delta x_2$ =0.2. The activation function of the output neuron was linear. We report here results for two typical channels with transfer functions given by:

$$H_1(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \tag{9}$$

$$H_2(z) = \text{-}0.2052\text{-}0.5131z^{-1}\text{-}0.7183z^{-2}\text{+}0.3695z^{-3}\text{+}0.2052z^{-4} \tag{10}$$

A sequence of 2000 symbols was generated to train the equalizer for each SNR value. Once trained, the equalizer was tested on a sequence of other 100000 symbols. All tests were repeated for 10 different realization, where for each realization neural network parameters were reinitialized. Both experiments conducted with 2-PAM and 4-QAM modulations pointed out the better performances with respect to the others neural architectures. Similar results were obtained also for non-linear channels (Figure 3): a non-linearity was introduced at the channel output $\{\hat{r}[n]\}$. In particular the received sequence $x[n]$ was generated by the formula:

$$x[n] = r[n] + 0.2\left(r[n]\right)^2 \tag{11}$$

4. REFERENCES

[1] Vitushkin A.G. and Henkin G.M., "Linear Superposition of Functions", *Russian Math. Surveys,* vol. 22 pp. 77-125, 1967.

[2] Piazza F., Uncini A. and Zenobi M., "Neural Networks with Adaptive Polynomial Activation Function" *Proc. of the IJCNN Int. Joint Conf. on Neural Networks*, Beijing, China, vol.2 pp.343-349, Nov. 1992.

[3] S. Guarnieri, F. Piazza, A. Uncini, "Multilayer Neural Networks with Adaptive Spline-based Activation Functions", *Proc. of Word Congress on Neural Networks WCNN'95*, Washington D.C., USA, July 17-21, 1995.

[4] Vecci L., Campolucci P., Piazza F. and Uncini A., "Approximation Capabilities of Adaptive Spline Activation Function", *Proc. of International Conference on Neural Networks ICNN'97*, pp. 260-265, Houston TX, USA, Jun. 1997.

[5] Vecci L., Piazza F. and Uncini A., "Learning and Approximation Capabilities of Adaptive Spline Activation Function Neural Networks", *Neural Networks*, vol.11, No. 2, pp.259-270, Mar. 1998.

[6] Proakis J.G. "Digital Communications". 2° Edition. *McGraw-Hill International Ed,* New York, 1989.

[7] Mulgrew B., "Nonlinear Signal Processing for Adaptive Equalisation And Multi-User Detection", *Proc. of the EUSIPCO-98*, Vol.1, Rhodes,(Greece), Sept. 7-12, 1998, pp. 537-544.

[8] Gibson G.J., Chen S., Cowan C.F.N. and Grant P.M. "Adaptive Equalization of Finite Non-Linear Channels Using Multilayer Perceptrons," in *Signal Processing*, vol.20. pp. 107-109, 1990.

[9] Chen S., McLaughlin S. and Mulgrew B. "Complex Valued RBF Network, Part II: Application to Digital Communications Channel Equalization" in *Signal Processing*, vol.36, pp. 175–188, 1994.

[10] Parisi R., Di Claudio E., Orlandi G. and Rao B.D. "Fast Adaptive Digital Equalization by Recurrent Neural Networks" in *IEEE Trans. On Signal Processing,* vol. 45, N° 11, 1997.

[11] Haykin S., *Neural Networks-A Comprehensive Foundation*, IEEE Press, 1994.

[12] Benvenuto N. and Piazza F. "On the Complex BackPropagation Algorithm", *IEEE Trans. on Signal Proc.,* vol. 40 pp. 967-969, 1992.

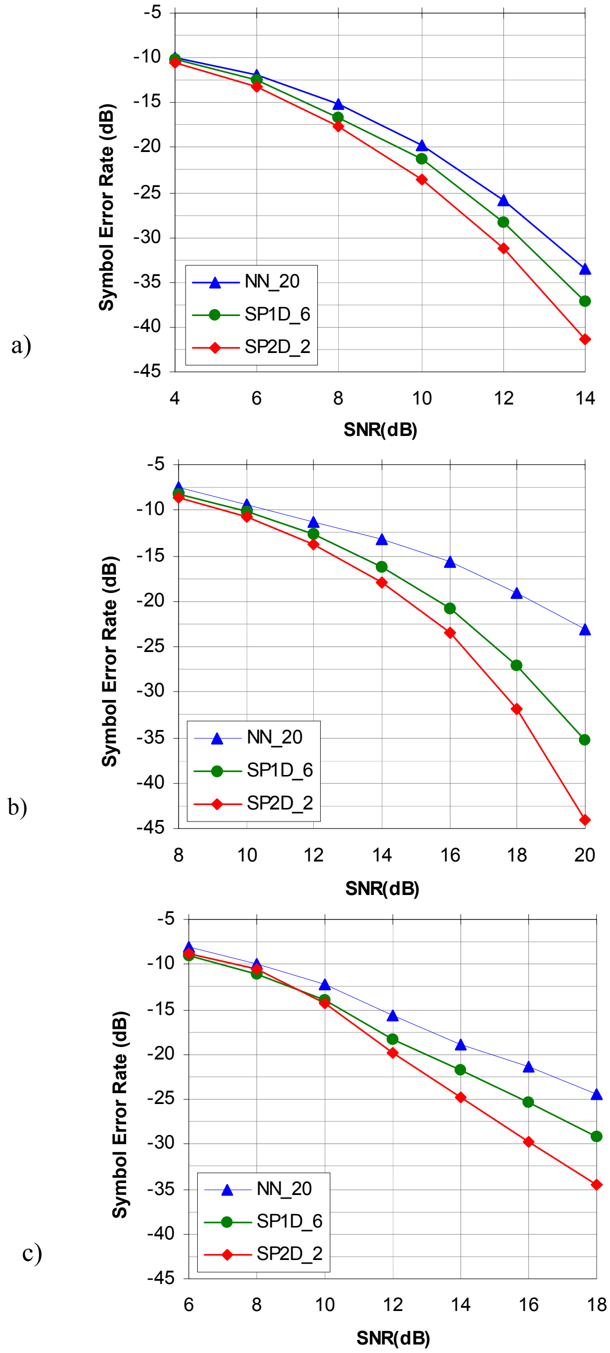

Figure 3 Average symbol error rate versus the SNR for the non-linear channels: a) $H_1(z)$,2-PAM; b) $H_1(z)$,4-QAM c) $H_2(z)$,4-QAM.