USING AN RBF NETWORK FOR BLIND EQUALIZATION: DESIGN AND PERFORMANCE EVALUATION

João Gomes *Victor Barroso*

Instituto Superior Técnico — Instituto de Sistemas e Robótica
Av. Rovisco Pais, 1096 Lisboa Codex, Portugal
{jpg,vab}@isr.ist.utl.pt

ABSTRACT

The design of adaptive equalizers is an important topic for practical implementation of efficient digital communications. In this paper, the application of a radial basis function neural network (RBF) for blind channel equalization is analysed. This architecture is well suited for equalization of finite impulse response (FIR) channels partly because the network model closely matches the data model. This allows a rather straightforward design of an optimal receiver, in a Bayesian sense. It also provides a simple framework for data classification, in which more complex nonlinear distortions can be accomodated with virtually no modifications. A clustering algorithm for dynamic creation and combination of local units is proposed, which eliminates the need for channel order estimation. An initialization procedure for the output linear layer is also presented. The network performance is illustrated with Monte Carlo simulations for a family of random channels.

1. INTRODUCTION

The problem of intersymbol interference cancellation using an adaptive equalizer has been studied for several years by the signal processing community. This effect occurs as a result of filtering performed by the physical channel or transmitter/receiver distortions, and can significantly impair the communications signal. Many of the techniques for equalizer design rely on the existence of a training sequence — a known preamble before the actual message begins — so that the channel can be identified. In one of the simplest and most widely used approaches, the sampled received signal is processed by a linear filter, adapted with the Least Mean Squares (LMS) criterion. More sophisticated techniques include decision-feedback equalization, fractionally-spaced sampling, maximum-likelihood sequence estimation and efficient updating algorithms such as recursive least squares.

The feasibility of applying neural networks for equalization of simple channels using multilayer perceptrons (MLP) was demonstrated in [1]. It showed that the best conventional receivers could be outperformed in terms of error probability and mean-square error by neural networks. It also highlighted some typical limitations of this type of neural network, such as long training times, convergence to undesirable performance extrema and strong dependence of estimation accuracy on the specific network topology.

The radial basis function neural network has attracted much attention since early works such as [2], which demonstrated that it could outperform multilayer perceptrons in some benchmarking problems. Moreover, its structure is deeply rooted in multivariable approximation theory, and allows a two-step learning algorithm that can converge to a solution several orders of magnitude faster than the MLP. In the specific application of channel equalization, there are additional physical motivations for using this network.

This work was partially funded by FEDER and PRAXIS XXI, under contract PRAXIS/3/3.1/TPR/23/94.

2. PROBLEM FORMULATION

A standard baseband equivalent model of a communications system is considered. A sequence of i.i.d. symbols $a(n)$ is transmitted through a finite memory distortive channel and corrupted by white Gaussian noise $\eta(n)$. The channel is assumed linear, with an impulse respose $h(n)$ spanning N_h symbols, and an output $y(n)$ of the form

$$y(n) = \sum_{k=0}^{N_h-1} h(k)a(n-k) + \eta(n)\,. \qquad (1)$$

The RBF network can also be used with more general channel models such as $y(n) = g\left(a(n), \ldots, a(n-N_h+1)\right)$, where g is an N_h-dimensional (mildly) nonlinear function. In this paper a binary sequence taking values from $\{-1, +1\}$ is used, although the RBF network is also applicable with M-ary real or complex signaling.

An equalizer estimates the current symbol $a(n)$ by processing an N-dimensional input vector $\mathbf{y}_f(n) = [y(n-N_1)\ldots y(n-N_2)]^T$, where $N = N_2 - N_1 + 1$ and $(\cdot)^T$ denotes vector transpose. In the absence of noise, the channel output can take only finitely many values. With binary signaling there are at most $N_s = 2^{N_h+N-1}$ possible combinations or *channel states* for the input vector $\mathbf{y}_f$. Conventional linear equalizers overlook this information, which should be exploited to the benefit of performance, as in the maximum likelihood receiver. In the presence of noise the data points form clusters around the channel states, and the pdf of $\mathbf{y}_f$ is obtained by a mixture of Gaussian density functions.

Most linear equalizers try to approximate the inverse of $h(n)$, so that the overall channel plus equalizer response approaches a discrete impulse. Alternatively, equalization may be viewed as a pattern classification problem in which the N-dimensional input space is partitioned into subsets corresponding to each desired symbol of the input alphabet. Using this approach, it may be shown that a minimum error probability symbol-by-symbol processor estimates $a(n)$ by maximizing the conditional probability $\Pr\{\mathbf{y}_f(n)|a(n)\}$ over the entire equiprobable source constellation [3]. For binary signaling and real channels, the associated Bayesian

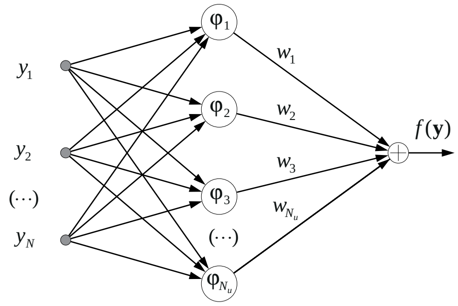


Figure 1. RBF network structure

risk function is

$$f(\mathbf{y}_f(n)) = \sum_{i\in N_s^+} \exp\left(-\|\mathbf{y}_f(n) - \mathbf{y}_{fi}^+\|^2/2\sigma^2\right) - \sum_{j\in N_s^-} \exp\left(-\|\mathbf{y}_f(n) - \mathbf{y}_{fj}^-\|^2/2\sigma^2\right) , \quad (2)$$

where $\mathbf{y}_{fi}^+$ and $\mathbf{y}_{fj}^-$ denote the channel states where $a(n)$ equals $+1$ or -1, respectively, and σ^2 is the noise variance.

3. RADIAL BASIS FUNCTION NETWORK

The structure of a radial basis function neural network is depicted in figure 1. It is formed by a single hidden layer of radially symmetric local units, which connect to an output linear layer. The network creates a mapping of the form

$$f(\mathbf{y}) = \sum_{k=1}^{N_u} w_k\, \phi\left(\|\mathbf{y} - \mu_k\|/\sigma_k\right) , \quad (3)$$

where N_u is the number of local units, $\phi : \mathbb{R} \to \mathbb{R}$ is a basis function, and w_k is the weight connecting unit k to the output. The nonlinear function ϕ takes significant values only near the origin, and determines the shape of all local units. It is almost invariably selected as $\phi(y) = \exp(-Ky^2)$, although other choices are possible. The center μ_k defines the point in input space where unit k is centered, and the radius σ_k adjusts its width.

Expression (3) has the same structure as (2), and for Gaussian units the two can be made equal by an appropriate choice of N_u, w_k, μ_k and σ_k. The real RBF network can implement the optimal decision function if a local unit is placed at the center of each input cluster, the weight connecting it to the output equals the desired symbol, and all radii satisfy $\sigma_k = \sqrt{2}\sigma$. This ideal solution requires an adequate placement of units at the channel states, which raises some problems. Firstly the number of units in an RBF network is usually considered fixed. Since N_s depends not only on the input dimension, but also on the channel order N_h, the latter must be known or estimated before the network can be used. Suitable channel order estimation methods are proposed in [4, 5]. If N_h is significantly underestimated, fewer units than channel states will be used, leading to severe performance degradation. Moreover, in time-varying channels the number and location of data clusters also changes in time. Assuming a fixed number of units may lead to undesirable solutions.

Knowledge of the noise variance is also needed to completely define the local unit response, but this is a rather simple problem. Additionally, this parameter has a reduced effect on the decision boundary $f(\mathbf{y}) = 0$, and it need not be determined very accurately.

For classification purposes only the decision boundary is relevant, and the mapping performed by the RBF will not necessarily produce a low output mean-square error (MSE). A more acceptable solution is to normalize the output of all local units by a factor [2]

$$\sum_{k=1}^{N_u} \phi\left(\|\mathbf{y} - \mu_k\|/\sigma_k\right) ,$$

which does not alter the decision boundary. When the centers are positioned at the channel states, the output of each unit becomes a symbol *a posteriori* probability, and a minimum MSE or MAP processor can be obtained by an appropriate selection of weights [3]. This normalization will be used throughout the simulations.

4. LEARNING ALGORITHM

A major drawback of nonlinear equalizers based on multilayer perceptrons or recurrent networks is the inability to determine the adjustable parameters in closed form even when the channel response and noise statistics are known *a priori*. This also means that convergence to local extrema in an adaptive implementation cannot be efficiently controlled. In this respect the RBF network is superior because the correctness of steady-state solutions can be imposed by a proper initialization of the free parameters based on the channel states.

The RBF learning algorithm is usually decoupled in two steps. First, clustering is performed on the input vectors to determine the unit centers and radii, and then the output weights are adjusted. The first step is typically accomplished using an unsupervised algorithm such as K-means, and the output weights are computed in closed form using the desired output symbols. Due to the special structure of the input data, modified versions of this procedure can be used in channel equalization. In [4] a simple supervised clustering algorithm is used, based on knowledge of the channel order and the transmitted symbols. The output weights are not adapted, but rather selected according to the desired ± 1 coefficients in the Bayesian decision function.

Since the transmitted symbols are not known in blind equalization, unsupervised learning must be used in both steps of the algorithm. In [6] an algorithm of this kind was proposed by the authors.

4.1. Updating of Centers and Radii

In [6] the local unit parameters were trained using a standard clustering algorithm that requires a fixed number of centers. While this algorithm is well suited for quantization of data uniformly distributed in a compact domain, it may lead to local minima which incorrectly identify the channel states when the input vectors form discrete clusters. It is also subject to the problems that other RBF equalization approaches face when the number of centers must be estimated beforehand. To overcome these limitations, a simple dynamic clustering algorithm is proposed in this paper, requiring only an estimate of the noise power. When a data point is "far" from all current centers, it is assumed that it belongs to an unidentified cluster, and a new unit is created at that location. When two centers are "close", they are regarded as belonging to the same physical cluster and subsequently combined. The definition of "far" and "close" depends on the spreading of clusters, which is a function of the noise power only. A data point is considered far from its channel state if the squared Euclidean distance is larger than $d_{\max}$, defined as

$$\Pr\{\|\mathbf{y}_f(n) - \mathbf{y}_{fi}\|^2 > d_{\max}\} < \epsilon\,, \tag{4}$$

where ϵ is typically smaller than 10^{-4}. The critical distance $d_{\max}$ is readily computable from (4) since $\|\mathbf{y}_f(n) - \mathbf{y}_{fi}\|^2/\sigma^2$ has a chi-square distribution with N degrees of freedom[1]. A critical distance $d_{\min}$ for combining centers is defined similarly by $\Pr\{\|\mathbf{y}_f(n) - \mathbf{y}_{fi}\|^2 \le d_{\min}\} < \delta$. The clustering algorithm may be summarized as follows:

$N_u = 1$
$n_1 = 1\,,\quad \mu_1(n_1) = \mathbf{y}_f(0)$

```
for n = 1:Ny
  d_i = ||y_f(n) - mu_i||^2,   1 <= i <= N_u
  k = arg min_i d_i
  if d_k > d_max
    N_u = N_u + 1
    n_{N_u} = 1 ,   mu_{N_u}(n_{N_u}) = y_f(n)
  else
    n_k = n_k + 1
    mu_k(n_k) = [lambda(1 - lambda^(n_k - 1)) mu_k(n_k - 1) + (1 - lambda) y_f(n)] / (1 - lambda^(n_k))
  end
  d_{i,j} = ||mu_i - mu_j||^2,   1 <= i, j <= N_u
  k, l = arg min_{i,j, i != j} d_{i,j}
  if d_{k,l} < d_min
    n_kl = n_k + n_l
    mu_kl = [n_k mu_k(n_k) + n_l mu_l(n_l)] / n_kl
    n_k = n_kl ,   mu_k(n_k) = mu_kl
    N_u = N_u - 1
    clear n_l, mu_l and recompute indices
  end
end
```

Each μ_k is updated with an exponentially weighted time average, which allows the centers to track channel variations and reduces the impact of undesirable center creations. The recursive expression for μ_k avoids the use of a gradient-type approximation that would require the selection of a suitable step size.

When a data block is processed, several spurious center creations and combinations may occur, but at least one center remains associated with each cluster (or group of partially overlapping clusters.) This ensures that the support set for the RBF mapping is properly defined, and the normalization of local unit outputs keeps the MSE at low levels. As in [4], it was found that the value of σ_k has little effect on the decision boundary, and a particularly simple choice is $\sigma_k = \sqrt{2}\sigma$ for real signals or $\sigma_k = \sigma$ in the complex case.

4.2. Initialization and Weight Update

According to (2), if the channel states were known the weights w_k would be fixed at ± 1. In a practical system, however, adapting the weights during data transmission is advisable, and a suitable blind equalization algorithm matches selected output and source moments by adaptively minimizing the cost function [6]

$$d(n) = \sum_{l=1}^{4} \alpha_l \left(\langle z^l\rangle_n - E\left\{a^l\right\}\right)^2 \tag{5}$$

$$\langle z^l\rangle_n = \frac{\lambda_0(1-\lambda_0^{n-1})\langle z^l\rangle_{n-1} + (1-\lambda_0)z^l(n)}{1-\lambda_0^n}\,. \tag{6}$$

In (6), $z(n) = f(\mathbf{y}_f(n))$ denotes the equalizer output, and $\langle z^l\rangle_n$ is a time average that estimates its moment of order l. The weighting constants $\alpha_1, \ldots, \alpha_4$ help define a generalized error signal.

This criterion can only compensate relatively small deviations of the weights from their optimal values, and a proper initialization procedure is crucial to ensure a valid steady-state RBF output. For binary signaling this is accomplished by partitioning the centers in two sets with opposite output desired symbols. All the weights w_k whose μ_k belongs to a given set are arbitrarily initialized with -1, and the remaining ones with $+1$. This may cause a sign reversal at the equalizer output, that can be easily corrected with differential encoding. To ensure that the selected partition could produce a valid set of channel states $\{\mathbf{y}_{fi}^+\}$, $\{\mathbf{y}_{fj}^-\}$, we first remark that for an invertible, linear, noiseless channel and a high enough input dimension N, there exists a vector $\mathbf{v}$ such that $\mathbf{v}^T\mathbf{y}_f(n) \approx a(n-d) = \pm 1$. Geometrically, this means that the data points $\mathbf{y}_f(n)$ are approximately located in two parallel planes orthogonal to the complex conjugate of $\mathbf{v}$, and the same must be true for the centers μ_k. The following cost function penalizes deviations from the planar model, and may be minimized iteratively by gradient descent

$$J = \sum_{k=1}^{N_u} \left(|\mathbf{v}^T\mu_k|^2 - 1\right)^2\,. \tag{7}$$

The requirement of linear separability of channel states (and centers) having different desired output symbols is rather restrictive, and may induce an unnecessarily high input dimension N. However, the lack of *a priori* knowledge about the shape of all possible decision boundaries forces us to make this assumption. It should also be pointed out that this separation does not significantly affect the convergence rate of the RBF equalizer because it occurs in parallel with the clustering process, and usually converges to a valid solution before enough centers have been reliably identified, so that the weight adaptation phase can be initiated.

5. RESULTS AND DISCUSSION

To better characterize the behaviour of this network under different conditions, a family of random multipath channels typical of an outdoor mobile communications environment in the GHz range was considered. Figure 2 depicts the normalized power delay profile, where fixed delays are assumed and the attenuations follow a Ricean fading model. Only the main path has a deterministic component, which accounts for 60% of its average power. The transmitted signal is formed by a binary sequence that modulates a train of raised cosine pulses with 100% rolloff. The received pulses may have very different shapes; in about 5% of all simulated channels very severe fading is observed, and virtually no signal energy is received. These extreme cases are clearly untractable, and were discarded.

The average received pulse has an effective duration of about five symbol intervals. Processing of longer responses

[1] For complex signals $\|\mathbf{y}_f(n) - \mathbf{y}_{fi}\|^2/(\sigma^2/2) \sim \chi^2_{2N}$.

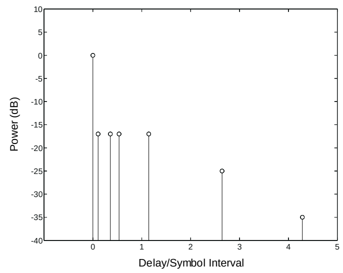
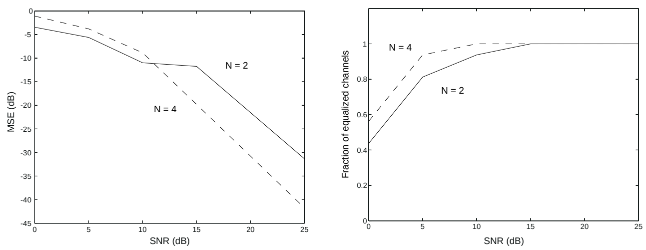


Figure 2. Power delay profile



(a) (b)

Figure 3. Random channel (a) Average MSE (b) Fraction of equalized channels

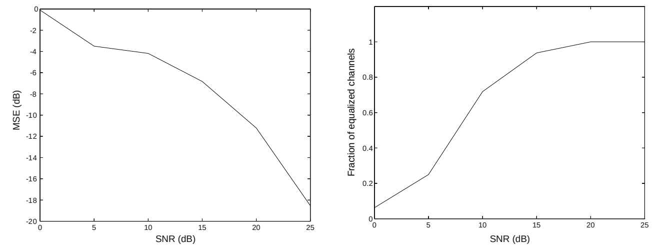
would require a higher input dimension N, which would in turn cause an exponential increase in the number of required local units. For this reason, practical RBF equalization is restricted to small signal alphabets and channels where the interference extends over four or five symbol intervals [3].

Figure 3(a) shows the average MSE as a function of mean signal to noise ratio[2] for $N = 2$ and 4. In each Monte Carlo run the channel is kept fixed, and a block of 5000 symbols is processed. Before decoding begins, 500 input vectors are processed to obtain estimates of the centers and the separating hyperplane. After this period the weights are initialized close to their steady-state values, and little transient behaviour is subsequently observed at the equalizer output. The clustering/separation algorithm is not stopped, so that channel variations can be tracked. If a center crosses the separating hyperplane during message decoding, it is considered that the new configuration is more reliable, and its associated weight is reinitialized. If clusters with different output symbols are indeed linearly separable and do not overlap, the RBF equalizer achieves a very low MSE of about -40 dB. As the noise power increases, the latter assumption will eventually be violated, and the error probability increases rapidly. Figure 3(b) shows the fraction of channels in the test set that were successfully equalized for different values of σ^2.

Figure 3 shows an improvement in overall performance when N is increased from 2 to 4, which may be attributed to a greater separation of clusters. However, the first learning stage becomes slower because more clusters have to be identified, and the hyperplane adaptation step has to be reduced by one order of magnitude. With $N = 4$ the centers are correctly separated in no more than 1500 iterations for most channels.

To test the effect of channel variations within a data block, the multipath attenuations were gradually changed

[2] The mean signal to noise ratio is defined as $E\{E_b\}/\sigma^2$, where E_b is the average received energy per symbol interval in a specific channel realization.



(a) (b)

Figure 4. Time-varying random channel, $N = 2$ (a) Average MSE (b) Fraction of equalized channels

every 100 symbols. To simulate a relatively slow drifting effect, zero mean lowpass noise was added to the gains in each channel considered previously. The disturbance in a given path is uncorrelated with all the remaining ones, and its power is 20 dB lower than the squared magnitude of the original gain. Using this model, the cluster signature of individual channels remains recognizable during the entire data block. Figure 4 shows the average MSE and the fraction of equalized channels under these conditions for $N = 2$. As expected, some performance degradation is observed when compared with figure 3. This effect is due not only to the more frequent overlapping of clusters, but mainly because the cluster drift occasionally causes one of the centers to cross the separating hyperplane, and its weight is then reset. It should be noted that these distortions are not directly related to the clustering process itself, whose tracking performance is quite good when a forgetting factor $\lambda = 0.5$ is used.

The simulations show that the RBF blind equalizer can be successfully applied in a variety of channels. The dynamic nature of the clustering process allows an adequate support for the mapping to be defined without prior knowledge of the number of clusters, even in time-varying channels. The major restriction is the requirement of linear separability of channel states, which is due to the non-convexity of the cost function used for weight adaptation.

REFERENCES

[1] G. Gibson, S. Siu, and C. Cowan, "Multilayer perceptron structures applied to adaptive equalisers for data communications", *in Proc. of ICASSP'89*, pp. 1183–1186, Glasgow, Scotland, May 1989.

[2] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units", *Neural Computation*, vol. 1, pp. 281–294, 1989.

[3] B. Mulgrew, "Applying radial basis functions", *IEEE Signal Proc. Magazine*, vol. 13, pp. 50–65, March 1996.

[4] S. Chen, B. Mulgrew, and P. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks", *IEEE Trans. on Neural Networks*, vol. 4, pp. 570–579, July 1993.

[5] J. Lee, C. Beach, and N. Tepedelenlioglu, "Channel equalization using radial basis function network", *in Proc. of ICASSP'96*, Atlanta, GA, May 1996.

[6] J. Gomes and V. Barroso, "Unsupervised learning using a neural network data receiver", *in Proc. of the Int. Conf. on Neural Networks (ICNN'95)*, Perth, Australia, November 1995.