On Evolutionary Programming for Channel Equalization
ADINA BURIAN, ARTO KANTSILA, MIKKO LEHTOKANGAS, JUKKA SAARINEN
Digital and Computer Systems Laboratory
Tampere University of Technology
P.O. BOX 553, FIN-33101, Hermiankatu 12C
FINLAND

Abstract: - The use of evolutionary programming for adaptive equalization of binary data bursts in a baseband digital communication system is studied. Evolutionary programming has been found to perform well for blind channel identification. In this paper it is shown that the usage of evolutionary programming techniques are beneficially also for adaptive channel equalization. We have also studied the effect of reducing the computational complexity of this technique on the bit error rates performances. Comparisons are made with multi-layer perceptrons equalizers.

Key-Words: - Channel equalizers, intersymbol interference, digital communications, evolutionary programming, offspring, population

1 Introduction

We start this paper by presenting in general terms the adaptive equalization concept, the very basic of a digital communication system and a few adaptive equalization methods. In the next section, we will present the used approach for adaptive equalization with evolutionary programming.

1.1 The Digital Communication System

Figure 1 illustrates the basic elements of a digital communication system [13]. The source encoder converts efficiently the output of an analog or digital source into a sequence of binary digits (data compression). This sequence is passed to the channel encoder which has the purpose to introduce in a controlled manner some redundancy in the binary information sequence. This redundancy can be used at the receiver to overcome the effect of noise and interference encountered in the transmission.

The output of the channel is passed to the digital modulator, which maps the coded sequence into signal waveforms. These are transmitted along with a carrier signal at a higher frequency. The obtained analog signal is passed through the communication channel - the physical medium used to send the signal from transmitter to the receiver.

At the receiver, the demodulator converts the continuous waveforms back to discrete domain. The obtained sequence is passed to the channel decoder which attempts to reconstruct the original sequence using the redundancy contained in the received data. Finally, the source decoder accepts the output sequence from the channel decoder, and from knowledge of the used source encoding method it tries to reconstruct the original signal. The
obtained signal at the output of the source decoder is an approximation of the original input.

One of the fundamental problems encountered in digital communication is the corruption of the transmitted signal by noise. Other undesired effects can occur during transmission: intersymbol interference (ISI), multipath propagation, ... ISI arises in digital communication systems when the channel impulse response lasts for more than one symbol interval. The communication channel which we will consider throughout this paper introduces ISI and additive white Gaussian noise to the transmitted signal. The ISI part of the channel is modeled as a finite impulse response (FIR) discrete-time filter.

To compensate for the ISI, channel equalization is needed. The purpose of equalization is to reduce the unwanted effects of signal distortion caused by the transmission channel, so that the transmitted symbols can be correctly interpreted. Equalization of digital communications channels is usually done by using a transmitted sequence also known to the receiver, during a preamble period. The channel response is time variant and unknown, and adaptive equalization methods are employed. In adaptive equalization the actual channel pulse response is estimated and an equalizer is automatically adjusted to equalize the channel [14].

In steady state, the adaptation of the equalizer is decision directed; this means that receiver decisions are used to generate the error signal. Decision directed equalizer adjustment is often not effective during initial acquisition, since the ISI can be so high that can initially cause a very high error rate.

Blind equalizers (self-learning) are used in order to provide the correct identification of transmitted symbols, when one does not have a training period, or when it is not practical to use such a strategy (i.e. digital TV broadcasting or multipath networks, where training has to be executed whenever one single receiver is inserted in the system). The transmission of training signals decreases communications throughput, although for time-invariant channels this is insignificant because only one training is necessary. For time-varying channels, the loss of throughput becomes an issue. Depending on the degree of channel time-variance, the repeated transmission of training sequences may leave the communication system with considerable overhead. In blind channel estimation, this overhead could be used for other purposes. The fundamental idea of blind channel estimation is to derive the channel impulse response from the received signal only, without access to the channel input signal by means of the training sequences.

Conventional approaches to solve the blind channel identification problem include high order statistics [20], which exploits the higher order statistical properties of the transmitted signal, and Bayesian estimation [15], when the channel is modeled by finite dimensional deterministic unknown parameters. Evolutionary Programming (EP) was also proposed for solving this problem [6]. EP was able to identify the channel parameters in almost all the situations tested in [6]. Channel estimates were not affected significantly by variations in the SNR, and the results were comparable with the ones given by genetic algorithms.

Optimum receivers in digital communication systems require the knowledge of the transmission channel. However, channel estimation is complicated by the fact that the transmission channel is frequency selective, mixed phase, and time-variant. Present state-of-the-art mobile communication systems transmit the so-called training sequences to assist the receiver in estimating the channel impulse response.

1.2 Traditional Channel Equalization Methods

The simplest equalizer mechanism is the Linear Transversal Equalizer (LTE) [9]. The past and current values of the received signal are linearly weighted by the complex-valued equalizer coefficients and tuned to produce the estimated output. There are various criteria for choosing the best equalizer coefficients, the most used being the MSE (Mean Square Error) criterion. Several coefficient adaptation algorithms have been developed; the best-known are LMS (Least Mean Square) and RLS (Recursive Least Square) algorithms.

Another traditional equalizer is Decision Feedback Equalizer (DFE). It usually consists of two FIR filters: a feedforward filter (FFF) that is basically a LTE, and a feedback filter (FBF). The decisions made on the equalized signal are feed back via the FBF. The FFF is used to remove ISI due to symbols transmitted in the future, while the FBF is used to cancel ISI due to symbols transmitted in the past. The FFF and FBF in DFE are typically optimized using either a zero-forcing criterion, or a minimum MSE criterion, the later being more prevalent. Both used criterions are equivalent in the limit of high signal-to-noise ratio (SNR). Optimization of DFE under a minimum symbol error probability criterion is usually not attempted.
Instead of symbol-by-symbol detection, the Maximum Likelihood Sequence Estimator (MLSE) treats the entire sequence at once and maximizes the mean time between error events. It is known that the Viterbi Algorithm (VA) can be used to implement MLSE of the input sequence in the presence of ISI and additive noise. The VA is a recursive structure that was originally invented to decode convolutional codes, but it was also analyzed for channel equalization purposes (MLSE/VA). The VA is used to determine the sequence that is closest in distance to the received sequence of noisy samples. The evaluation of the distance metric is done by using path metrics associated with states and branch metrics associated with transitions, in computing the path recursively. Unfortunately, the complexity of the VA grows exponentially with the duration of the channel impulse response. In applications which have high symbol rates and channel impulse responses which last for a large number of symbol intervals, it can be difficult to implement the VA and to meet overall system objectives of low power consumption and low cost. Consequently, suboptimum detection techniques - such as DFE - which offer good performances, are used.

1.3 Channel Equalization Using Classification Methods

Another approach is to consider the equalization of the received signal as a classification problem. In [5] the signal space was partitioned into appropriate decision regions and a class label was assigned to a particular region for all unclassified vectors belonging to that region. Top-down approach was used in tree induction, and splitting was done based on information gain criterion. Overfitting was avoided by utilizing a pruning algorithm. The advantages of this method are its simplicity and straightforwardness and thereby the reduction of computational complexity.

It was shown that the problem of finding finite length DFE filters that minimize the probability of symbol error at any SNR subject to a certain separation condition is a convex optimization problem [1]. The problem of determining DFE filters that minimize the probability of symbol error at high SNR it was shown to be equivalent to finding the hyperplane that maximally separates two given finite groups of points in a finite dimensional Euclidian space. This task is equivalent to finding the optimum separating hyperplane in support vector machines (SVMs) [16]. A SVM uses training data as an integral element of the function estimation model as opposed to simply training data to estimate parameters of an a priori model using maximum likelihood, which is the more traditional approach. Consequently, SVMs training is rather straightforward, requiring less ad hoc input from the designer. Once training of the SVM was completed, the equalization, or more appropriately the nonlinear detection, is efficient and comparable to Volterra filters and neural networks.

2 Channel Equalization using Evolutionary Programming

Evolutionary Programming (EP), originally conceived by Lawrence J. Fogel in 1960, is a stochastic optimization strategy that places emphasis on the behavioral linkage between parents and their offsprings [3][4].

The evolutionary process is simulated in the following manner: an initial population of solutions (N individuals) is typically chosen at random. These parents are measured in their individual ability to predict each next event in their experience, with respect to whatever payoff function has been prescribed (the fitness score - squared error, absolute error). Progeny are created through random mutation of these parents. Each parent creates a single offspring. The obtained offsprings are scored on their predictive ability in a similar manner to their parents, the ones that are most suitable are probabilistically selected to become the new parents. Pair-wise comparisons over the union of parents and offspring are conducted. For each comparison, if the individual’s fitness is no smaller than the opponent’s, it receives a ‘win’. An actual prediction is made when the predictive-fir score demonstrates that a sufficient level of credibility has been achieved or when the available computational limit has been exceeded. The disadvantages of using such algorithms for channel equalization are slow convergence, large error variance and high computational complexity.

The most obvious attempt to use EP for channel equalization can be considered to just use EP for the training of a neural network (which can be MLP) equalizer [2]. In [17] it was concluded that the use of an evolutionary algorithm provides a more effective mean of training an MLP to perform equalization of a non-minimum phase channel. The evolutionary algorithm would train the MLP to a more optimal solution more often than back propagation (BP). On average, the variance of the bit error rate performance of the MLPs trained with an evolutionary algorithm was much less than those
trained with BP. So, evolutionary algorithms offer an effective alternative training technique for MLPs to perform channel equalization.

A different approach is presented in [12]. Traditionally, equalization is based on linear FIR filters, but infinite impulse response (IIR) filter is a more general equalizer structure, with the disadvantages of a very slow convergence and always being stuck at a local minimum. A global optimum solution based on genetic algorithms (GA) for IIR lattice filter structure was obtained.

In this paper we propose a methodology in which an evolutionary programming-type search is used in combination with the gradient-descent method. The filter coefficients are evolved in a random manner when the filter is starting to have slow convergence rate. The used filter is a tunable IIR digital filter structure. The gradient-descent solution is used to initialize the EP method. The scheme of the used EP algorithm is summarized as follows:

1. Initialization:
An initial population of $N$ individuals is selected randomly from a feasible range in each dimension, around the solution given by the gradient descent method. Each individual is taken as a pair of real-valued vectors $(s_i, \eta_i) = S_i, \forall i = 1, \ldots, N$, where $S_i$ is a random vector, $s_i$ is the outcome of the random vector – the $n$ channel coefficients.
The distribution of initial trials $\eta_i$ is uniform.

2. Evaluation
Each $s_i, \forall i = 1, \ldots, N$ is assigned a fitness score $\phi(s_i)$ that is the mean absolute error (MAE) for the training sequence.

3. Creation of offspring
   (mutation)
Generate one offspring from each individual: each $s_i, \forall i = 1, \ldots, N$ is altered by adding a Cauchy random variable and assigned to $s_{i+N}$.
Cauchy mutation is used because it performs better than Gaussian mutation: it has a higher probability of making longer jumps [19]. The mutation is done according to:

$$\eta_i(j) = \eta_i(j) \exp\left[ \frac{1}{N(0,1) + \tau N(j,1)} \right]$$

$$s_i(j) = s_i(j) + \eta_i(j) C_j,$$

where $N(0,1)$ is a normally distributed random number with zero mean and unit variance, $N_j(0,1)$ is the same as $N(0,1)$ but is regenerated for every $j$, and $C_j$ is a Cauchy distributed random variable with unit scale parameter. The factors $\tau$ and $\tau'$ are (commonly) set to $\left( \frac{\sqrt{2\sqrt{n}}}{n} \right)$ and $\left( \sqrt{2n} \right)^{-1}$, respectively. Therefore, individuals including parents and offspring exist in a common competing pool.

4. Evaluation
Evaluate the offspring: to each $s_{i+N}, \forall i = 1, \ldots, N$ it is assigned a fitness score.

5. Competition and selection
Each individual in the competing pool must stochastically strive against other members of the pool, based on the function $\phi(s_i)$. Every individual in the population (channel coefficients in this case) is compared with $r$ randomly selected opponents. The individuals with the best function values are selected to form a survivor set according to the decision rule. The better half of the population, with the largest number of wins, is selected to become the new parents for the next generation.

6. Stopping rule
The process of generating new trials and selecting those with best function values is continued until the function values are not obviously improved (the obtained MAE is small enough) or a given count of maximum number of generations $G$ is reached.
3 Experimental Results

In obtaining the results, we used the Global System for Mobile Communications (GSM) [11]. The information signal consists of bursts of binary symbols taking values of either 1 or –1. Each burst begins with a training sequence, which is 26 bits in length and known to the receiver. Therefore it can be used to adapt the equalizer. The latter part of the burst contains 116 bits of data payload, which is not known to the receiver. The total length of the burst is 142 bits, and the burst is oversampled at the transmitter by a factor of 3. The information signal is transmitted as two-leveled baseband signal. For each transmitted burst, we have first used the training sequence of the burst, and then the equalization was performed on the data sequence of the burst with the obtained trained solution.

The communication channel used in simulations has both ISI and additive white Gaussian noise. The ISI part, resulting from multipath propagation, is modeled as a FIR discrete-time filter. We can formalize the relationship between the channel outputs $y$ and the transmitted binary signal $a$ using equation (3), where $h_i$ are the channel coefficients and $\delta$ represents additive noise.

$$y_n = \sum_{i=0}^{N} h_i a_{n-i} + \delta_n$$ (3)

The obtained results using evolutionary programming were compare with the corresponding results given by a cascade-correlation trained multilayer perceptron neural network equalizer [7]. The evolutionary programming and neural network learning are performed for each burst separately. We used a fixed channel impulse response:

$$h = [h_0 \ h_1 \ldots \ h_4] = [0.5 -0.3 \ 0.6 -0.7 -0.8]^T,$$ (4)

and signal to noise ratio (SNR) of 5, 10, and 15 dB. A white Gaussian noise with zero mean and unit variance was added to the channel output.

In all simulations the maximum number of generations $G$ was set to 10000. The results presented are obtained by averaging over 10 runs. The obtained bit error rates for the two considered structures in all the considered situations are presented in Table 1. It appears from this table that the use of EP improves the obtained BER when compared with MLP equalizer. Computational complexity of the neural network depends on the number of inputs (in our case 2), training epochs (50) and hidden units (max. 6). Computational complexity of evolutionary programming depends on the number of evolving individuals. This gives a higher complexity to this technique. When a small number of evolving individuals $N$ was used, a bigger number of generations $g$ was needed, but a good solution was found (see Table 2). In both tables the bit error rate is computed as the decimal logarithm of the error probability.

<table>
<thead>
<tr>
<th>SNR</th>
<th>BER NN</th>
<th>BER EP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-1.236</td>
<td>-1.237</td>
</tr>
<tr>
<td>10</td>
<td>-1.638</td>
<td>-1.741</td>
</tr>
<tr>
<td>15</td>
<td>-1.860</td>
<td>-1.991</td>
</tr>
</tbody>
</table>

Table 1. Comparative results between NN and EP with 100 evolving individuals.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$g$</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>7799</td>
<td>-1.443</td>
</tr>
<tr>
<td>50</td>
<td>6130</td>
<td>-1.652</td>
</tr>
<tr>
<td>100</td>
<td>5540</td>
<td>-1.741</td>
</tr>
<tr>
<td>150</td>
<td>5509</td>
<td>-1.726</td>
</tr>
<tr>
<td>200</td>
<td>5460</td>
<td>-1.742</td>
</tr>
</tbody>
</table>

Table 2. Comparative results for EP with different number of evolving individuals for SNR=10 dB

Future work ideas include also the evolving of the used neural network. Eventually, to use an embedded training: both cascade correlation and evolutionary programming. The purpose of using such a system is to reduce the searching space of neural network training, and also to assist the neural network training in the search of the global minimum.

4 Conclusions

We have studied the use of evolutionary programming for equalization purposes in a time-varying communications channel, where the channel introduces intersymbol interference and additive Gaussian noise to the transmitted signal. The usage of evolutionary programming gives an increase in performances (smaller error rates) comparatively with multi-layer perceptrons networks, at the cost of an increased computational complexity. The computational complexity is a very important factor in mobile environments. Because of that we studied also the possibility of decreasing the number of the evolving individuals, which reduces the computational complexity of evolutionary programming. For a very small number of evolving individuals, similar values of the error rates with MLP networks were obtained, for a bigger number of generations requested.
References:


