

Adaptive Deconvolution and Cross Equalization

By: Dr. M. Turhan (Tury) Taner - mt.taner@rocksolidimages.com

August - 1998

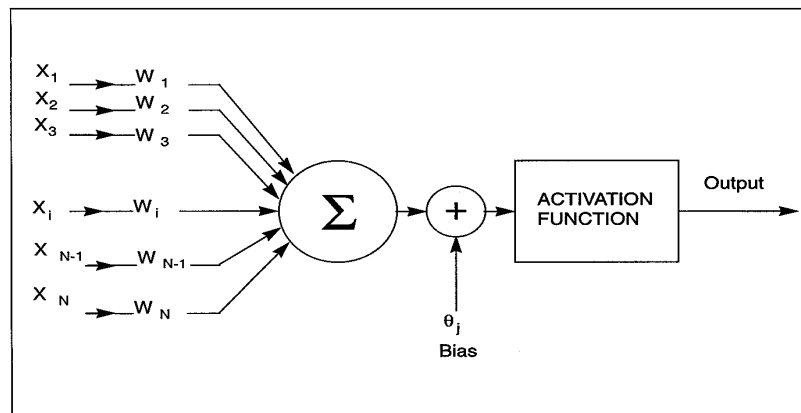
Introduction:

Adaptive filtering have been introduced by Widrow, which later led to the development of Neural Networks. I am taking a lot of liberties by borrowing many concepts introduced by Widrow for the purpose of reintroducing the concepts of adaptation into deconvolution. I recommend reading of the book by Widrow and Stearns for further understanding and proofs of the concepts discussed below. I have also studied the method presented by Griffiths et al (1977), which is somewhat similar to Widrow's approach and adaptation method used in the ProMax Adaptive Deconvolution modules.

Introduction:

Most of the methods described below are based on the updating the operators in the direction of steepest descent towards the optimum solution. I have found the "Delta Rule" used in the Neural Network training computations gives the simplest explanation of the operator updating, based on the least mean error squares computation. This method is thoroughly covered in the book by Widrow and Stearns.

Generalized Delta Rule:



Perceptron or "NEURON"
Fig : 1

One of the earliest works on Neural networks goes back to Widrow's introduction of Perceptron. This consisted of a single Neuron, which could be trained to respond in an adaptive manner. Widrow (1962) showed that the Perceptron could be trained to perform linear filtering or discrimination.

Let $\mathbf{X} = \{x(1), x(2), \dots, x(N)\}$ represent the input training pattern in the form of column vectors. We wish to solve for a column vector \mathbf{w} of M elements such that;

$$\mathbf{X} \mathbf{w} = \mathbf{b} \quad (1)$$

where the elements of \mathbf{b} are the output values specified by the training set. In binary classification $\mathbf{b}=\mathbf{1}$ specify the corresponding \mathbf{X} set belong to the class \mathbf{A} and $\mathbf{b}=\mathbf{0}$ specify the corresponding set \mathbf{X} belongs to the class \mathbf{B} . Since \mathbf{X} matrix has N number of columns and M number of rows and $M < N$, then we can solve this set of equations by the classical *least mean error squares* method. We can obtain the normal equation square matrix by pre-multiplying both sides by the transpose of the \mathbf{X} matrix;

$$\mathbf{X}^T \cdot \mathbf{X} \cdot \mathbf{w} = \mathbf{X}^T \mathbf{b} \quad (2)$$

and solve for w ;

$$w = (X^T . X)^{-1} . X^T b \quad (3)$$

This expression can be simplified to ;

$$w = \hat{X} . b \quad (4)$$

where \hat{X} is the pseudo-inverse of the original rectangular $(X^T . X)^{-1} . X^T$ matrix,

$$\hat{X} = (X^T . X)^{-1} . X^T . \quad (5)$$

This inverse theoretically can be computed directly. However we may get some non-sensical values not representing the real situation. This is computed by an iterative procedure called the linear perceptron algorithm which leads us to the Delta rule.

We wish to compute a single set of w weights to yield correct set of outputs b for all input patterns $x(p)$. We start with an arbitrary set of values of $w^{(1)}(i)$ then update it by the following rule;

$$w^{(k+1)}(j) = w^k(j) + r[b(j) - w^k(j)x(j)].x(j) \quad (6)$$

This updating continues until all of the patterns are classified correctly, at which time the negative of gradient $[b(j) - w^k(j)x(j)]$ becomes zero or very small. In practice this cannot be reached in most of the cases, hence the iteration would be stopped when sum of the squares of errors reaches below some prescribed threshold value.

We can write the expression 6 in the form;

$$\Delta(W) = h.dX \quad (7)$$

where d is the difference between the desired output and the computed actual output produced by the perceptron. This is the delta-rule expression, which states that the change in the weight vector should be proportional to the delta (the error) and to the input pattern. We can also show expression 6 in a more familiar form as;

$$w^{(k+1)}(j) = w^k(j) + h.dX \quad (8)$$

Widrow shows that the product of dX is the movement in the steepest descent direction.

General Optimization and LMS Method of Widrow:

This is the method described by Widrow and Stearns and Griffith et al. The method is based on the fact that the formulation of least mean error square solution also represent the negative of the gradient at the initial solution coordinates. To demonstrate this, we will review the development of predictive deconvolution operator computation. We wish to design a set of operators, when convolved with the data set will predict its future values at some prediction distance. The least mean error squares method help design these operators that minimizes the sum of squares of differences between the actual and the predicted trace.

Let the trace be given by $f(t)$ and the predictor operators given by $a(n)$;

$$\mathbf{e}^2 = \sum_{t=I_1}^{I_2} \{f(t+L) - \tilde{f}(t+L)\}^2, \quad (9)$$

where predicted trace is the result of convolution of data with operators;

$$\tilde{f}(t+L) = \sum_{n=0}^N a(n)f(t-n). \quad (10)$$

By substituting equation (2) back into equation (1), we will get;

$$\mathbf{e}^2 = \sum_{t=I_1}^{I_2} \left\{ f(t+L) - \sum_{n=0}^N [a(n)f(t-n)] \right\}^2 \quad (11)$$

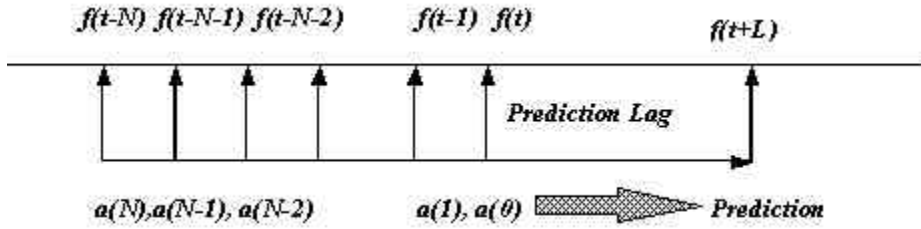


Figure 2. Relationship between operators (at bottom) seismic trace (at top) and predicted position. Note that in convolution operators are in reverse order with respect to seismic trace.

In conventional computation, we determine the operators by setting partial derivatives of the equation (11) with respect to the unknown operators equal to zero.

$$\nabla \mathbf{e}^2 / \nabla a(m) = -2 \cdot \left\{ \dot{\mathbf{a}} \left[f(i+L) \cdot f(i-m) \right] - \dot{\mathbf{a}} f(i-m) \dot{\mathbf{a}} \left[a(n)f(i-n) \right] \right\} \quad (12)$$

Equation (12) is the expression of the gradient of the least mean error squares surface at the initial estimate of $a(n)$. By setting these expressions to zero we determine the minimum coordinates of the error surface, which gives us the desired set of operators, that minimizes the error in a least mean error squares manner.

$$\nabla \mathbf{e}^2 / \nabla a(m) = \mathbf{0} \quad (13)$$

Therefore, if we have an initial estimate of the operator, its correction will be towards the minimum following the negative of the gradient direction;

$$w^{(k+1)}(t) = w^{(k)} + h \cdot \left\{ \dot{\mathbf{a}} \left[f(i+L) \cdot f(i-m) \right] - \dot{\mathbf{a}} f(i-m) \dot{\mathbf{a}} \left[w^{(k)}(n)f(i-n) \right] \right\} \quad (14)$$

Here, we go back to the original formulation of the equations, as shown on equations (9) and (10), and write the equation (14) in more compact form;

$$w^{(k+1)}(t) = w^{(k)}(t) + h \cdot \left\{ \dot{\mathbf{a}} f(i-m) \left[\dot{\mathbf{a}} f(i+t) - \tilde{f}(i+t) \right] \right\} \quad (15)$$

This is the general minimization solution. Note that this update is based on minimization over a time window. Widrow suggested the use of instantaneous values instead of windowed averages. The error at time t is;

$$e^2(t) = \left\{ f(t+L) - \sum_{n=0}^N [a(n)f(t-n)] \right\}^2 \quad (16)$$

Similarly the gradient will be ;

$$\nabla e^2(t) / \nabla a(m) = -2 \cdot \{ f(t+L) \cdot f(t-m) - \sum_{n=0}^N [a(n)f(t-n)] \} \quad (17)$$

Therefore, corrections in the steepest descent direction will be in the form of;

$$w^{(k+1)}(t, m) = w^{(k)}(t, m) + h \cdot \{ f(t+L) - \sum_{n=0}^N [a(n)f(t-n)] \} \quad (18)$$

This (Equation 18) formulation is termed by Widrow as the LMS algorithm, which is implemented in the ProMax module. Widrow's studies indicate that to achieve convergence, h , the learning rate should be kept $0 < h < 2 / \lambda_{\max}$, where λ_{\max} is the largest eigenvalue of the autocorrelation matrix of the normal equations.

It is interesting that, one could reach the same conclusion as equation (18) by an heuristic approach. We could say that we would like to adjust operators gradually at each output sequence so the difference between the desired and actual output is minimized. We try to keep the adjustment amounts slowly varying with time. However, it is difficult heuristically prove that this approach will converge.

ProMax Implementation:

This is the simplest application of the adaptation of predictive deconvolution. It is the implementation of Widrow's LMS method, which is an economical way of adapting the actual output to the desired output in a sample by sample continuous manner. This method is included with the ProMax modules. The method is capable to update the operators by comparing the predicted output with the actual desirable output data. This comparison could be made sample by sample, as implemented in the ProMax module, or by measuring the RMS error value over some moving preset time window. The help file gives the following explanation:

" A trace sample is predicted from a subset of past trace values, and the error is the actual trace value minus this predicted value. (The length of the subset is probably equal to the length of the operator computed one of several ways available in the module.) If the error is zero, the output sample is zero, and the unchanged prediction filter is left to predict the next sample. If the error is not zero, each filter coefficient is increased by an amount equal to the error times the corresponding trace value times the rate of adaptation. The filter is now ready to predict the next trace value."

This procedure, in essence, similar to Widrow's approach, operators are corrected with respect to error amount at each sample. By keeping the learning rate as a small number, the amount of modification is kept under control.

Both L1 and L2 norm operator design algorithms and Burg algorithm are offered as deconvolution operator design procedures. We can express this approach as;

$$w^{(k+1)}(j) = w^k(j) + h[b(j) - \sum_{n=0}^N [w^k(j)x(j-n)] \cdot x(j)] \quad (19)$$

Adaptive Subtraction or Cross Equalization:

Predictive deconvolution estimates future values of the seismic trace from its past values. In the case of cross equalization we predict values of one trace from another trace. Therefore the trace $f(t)$ will be replaced by the predictor trace $g(t)$ on the convolution shown in equation (11). We assume that the trace to be predicted $f(t)$ is the sum of signal $s(t)$ and some form of noise $h(t)$ which could be predicted from the second trace channel $g(t)$.

$$f(t) = s(t) + h(t) \quad (20)$$

Let the actual output be;

$$\tilde{h}(t) = \sum_{n=-N}^N a(n)g(t-n) . \quad (20)$$

And the error function be given as;

$$\mathbf{e}^2(t) = \{s(t) + h(t) - \sum_{n=-N}^N a(n)g(t-n)\}^2 . \quad (21)$$

Partial derivatives of the error function will give the gradient as;

$$\partial \mathbf{e}^2(t) / \partial a(m) = -2. \{g(t-m)[s(t) + h(t) - \sum_{n=-N}^N [a(n)g(t-n)]]\} \quad (22)$$

Assuming that correlation between $g(t)$ and $s(t)$ is negligible, then the gradient formula will simplify to;

$$\partial \mathbf{e}^2(t) / \partial a(m) = -2. \{g(t-m)[h(t) - \tilde{h}(t)]\} \quad (23)$$

Therefore the updating procedure will be;

$$w^{(k+1)}(t, m) = w^{(k)}(t, m) + \mathbf{h}.[h(i+t) - \tilde{h}(i+t)]g(t-m) \} \quad (24)$$

This is similar to the updating procedure of predictive deconvolution. Except, here I have used two sided filter in order to accommodate wavelet shape differences between $h(t)$ and $g(t)$.

Computational Procedure:

Since the updating algorithm follows the steepest descent route, operators will eventually approach the least mean error squares solution for stationary time series. Therefore, we can start with any simple filter weights as we wish. In Neural Network training all of the weights are chosen as random variables. In predictive deconvolution or the cross equalization case, we can start with a reasonable set of operators weights. We can assume that initially the time series are stationary and start with the conventional Wiener filters computed over sufficiently long windows. We know that these filters are optimum for the time window, but they may not be optimum for non-stationary data sets. Since we are starting with overall statistics of the data set, these filter weights will be reasonably close to the local optimum solution, thus they can be used as the starting values of the filter set. ProMax module uses several different filter designs as the starting values, as pointed above.

Conclusion:

I have presented two basic methods for adaptive deconvolution. One is based on minimization over some time window and the second is based on instantaneous values of the gradient function. Both of the method are viable in predictive deconvolution and cross equalization (adaptive subtraction) applications. I have found the book by Widrow and Stearns very informative and gives very systematic explanation of all of the theorems involved in the adaptive procedures. This provided me good background to understand the paper by Griffiths et al.

I have also discussed the "Delta" rule utilized in Neural Network computation. Lastly, I have discussed the adaptive cross equalization (adaptive subtraction), which is developed in the similar lines as the predictive deconvolution.

References:

Griffiths, L. J., Smolka, F. R., and Trembly, L. D., 1977, Adaptive deconvolution: a new technique for processing time-varying seismic data; *Geophysics*, 42, No. 4, 742-759.

ProMax Help File, Adaptive Deconvolution; Advance Geophysical Co.

Widrow, B. and Stearns S. D., 1985, Adaptive signal processing; Prentice-Hall Book Co.

Widrow, B., 1962. Generalization and information storage in networks of Adaline "Neurons". In M.C. Yovits, G. T. Jacobi, and G. D. Goldstein (Eds.), *Self Organizing Systems*, pp 435-461, Spartan Books, Washington, D.C.