Bernard Widrow

Thinking About Thinking: The Discovery of the LMS Algorithm

In this issue, our guest is Dr. Bernard Widrow. Born during the winter holiday season of 1929 in a small town in Connecticut, Dr. Widrow gladly remembers the advice received in his youth to have the courage to apply to the Massachusetts Institute of Technology (MIT), even though he didn't know a soul. Bernard Widrow applied to MIT, was admitted, and then completed his S.B. (1951), S.M. (1953), and Sc.D. (1956) degrees, all in electrical engineering. After spending a few more years at MIT as a faculty member, he joined Stanford University in 1959 and later became a professor of electrical engineering there.

Over the past half century, Dr. Widrow's work has focused on numerous aspects of adaptive digital signal processing: noise canceling, antennas, inverse control, and nonlinear filtering. He coauthored the books *Adaptive Signal Processing* (1985), *Adaptive Control* (1996), and *Quantization Noise* (to appear). Bernard Widrow has been awarded prestigious distinctions, including the IEEE Centennial Medal (1984), the IEEE Alexander Graham Bell Medal (1986), the IEEE Neural Networks Pioneer Medal (1991), and the IEEE Millennium Medal (2000). He was also inducted into the National Academy of Engineering (1995) and the Silicon Valley Engineering Council Hall of Fame (1999).

Nicknamed "Doc" by his students, Bernie Widrow values a "can do" attitude in his collaborators and appreciates their faith in him. He confesses that getting a major new idea approximately every five years stimulates him greatly, whereas the times when a new idea doesn't work keep him grounded. Between research, decade-long collaborations (such as that with John McCool on adaptive filtering and applications), and teaching, he finds balance by enjoying opera, symphony, and ballet; collecting art; going to museums; and watching movies. For the curious journalist, "Doc" admits that his artistic interests may be partly inherited, as he is related to the famous painter Marc Chagall, who was his grandmother's cousin. Needless to say, sharing impressions with Bernard Widrow on the daunting Chagall exhibition held at the San Francisco Museum of Modern Art (2003) is a delight. "Doc" also likes traveling, visiting his children and grandchildren, and going out for walks. We invite you to join him as he is "thinking about thinking" and recalling the events related to the discovery of the LMS algorithm.

—Adriana Dumitras and George Moschytz
"DSP History" column editors
adrianad@ieee.org,
moschytz@isi.ee.ethz.ch

It was the summer of 1956. I was at the Massachusetts Institute of Technology (MIT) and had just finished my doctoral thesis on the theory of quantization noise, in the field of digital signal processing. During the past academic year, I had been working very hard on my thesis and teaching courses in the fields of radar, digital signal processing, and digital controls. The summer was a pleasant time, much more relaxed. I was looking forward to the fall semester, when I would join the MIT faculty as an assistant professor of electrical engineering. A colleague in our laboratory, Ken Shoulders, told me about an ongoing seminar that summer at Dartmouth College on the subject of artificial intelligence (AI). He planned to go to Dartmouth to learn about AI and about the progress that had been made in the field. I agreed to go with him. The Dartmouth seminar was really the beginning of the field of AI. The founders and pioneers were there. The discussions were highly stimulating and inspirational. We joined the seminar for one week and then tore ourselves away and returned to MIT.

Some people at the Dartmouth AI meeting were seriously considering building an artificial brain. I was so taken by this that I never got over it. I spent the next six months thinking about thinking. I forgot all about digital signal processing and the theory of quantization noise.

I began to see a connection between problem solving and game playing, and I began to contemplate building a problem-solving machine that could perform simple reasoning. I concluded, however, that it would take about 25 years to do this, given the state of electronics at that time. Ken Shoulders was working on some basic ideas for integrated circuits, but they were years away. Being interested in teaching and academic research, I realized that a 25-year time horizon for practical realization was too far out, and with the "publish or perish" system, I couldn't work on this subject and succeed as an academic. I was lucky to have realized this at an early stage. I dropped out of AI, but I never lost my interest in it. Almost 50 years have gone by since then, and we are not even close to building an artificial brain. Maybe we will be able to do it during the next 50 years.

After working on AI for six months, I was very anxious to get back to something with a more near-term payoff. I returned to the field of digital signal processing. I was familiar with Wiener filter theory in both its continuous and discrete forms. To design a Wiener filter, you need to know the autocorrelation

1053-5888/05/$20.00©2005IEEE

function of the input signal and the cross-correlation function between the input and desired response signals. This is fine when you are doing homework exercises, but what can you do in practice when no one gives you the input statistics? All you have are input signals.

I puzzled over this and then developed a simple idea: Let the input signal flow into a finite impulse response (FIR) digital filter. By sensing an error signal and estimating its mean squared error, the coefficients of the filter (the filter weights) could be adjusted to minimize the mean squared error. Thus, the filter would learn the Wiener solution (i.e., the optimal impulse response). The result would be an adaptive Wiener filter. A simple learning process could therefore be used to make a self-optimizing filter. This would be a lot easier than making an artificial brain that learns. My idea was to use filter performance to control the impulse response. I called this *performance feedback.*

I first applied an adaptive FIR filter to the problem of Wiener prediction. The objective was to predict a random input signal, Δ time samples into the future. The weights of the adaptive filter were adjusted to minimize the mean squared of the prediction error. The overall result was a best least squares linear predictor that learned from its input data.

The structure of the FIR adaptive filter used in the Wiener predictor consisted of a tapped delay line with variable weights connected to the taps. The output was a weighted sum of present and past input samples. The impulse response of this filter was equal to the sequence of weight values. The error was the difference between the desired response and the actual output response and was used by the adaptive algorithm to adjust the weights. The adaptive algorithm controlled the impulse response of the filter.

The basic adaptive element used in the FIR filter was a linear combiner with variable weights. I was able to analyze the linear combiner and to show that the mean squared error was a quadratic function of the weights. If there were only two weights, w_1 and w_2, the mean squared error could be pictured as a paraboloid. With more weights, this would be a hyperparaboloid. The optimal operating point was at the bottom of the paraboloidal bowl, and this was the Wiener solution.

Initially, one would not know where the bottom of the bowl was, so the starting point would be an initial guess. To find the bottom of the bowl, I used the method of steepest descent (i.e., following the gradient with a series of steps). It was possible to model the steepest descent as a feedback process for controlling the weights. The gradient was an error vector in a multivariable feedback system. Because the mean squared error was a quadratic function of the weights, its derivative (the gradient) was a linear function of the weights. Therefore, the feedback was linear and could be analyzed. I found that the relaxation toward the Wiener solution was of exponential nature, like transients in linear systems. All this was worked out and verified by simulation in 1957–1958.

To effect learning, each component of the gradient was estimated (one component at a time) by incrementing the corresponding weight and measuring the mean squared error, decrementing the weight and measuring the mean squared error, and then finding the difference in the mean squared errors and dividing by the total weight change. By simulation, it was possible to verify theoretically derived learning rates and to verify the mean squared error performance of the adaptive predicting filter and the adaptive linear combiner.

In 1958, Richard L. Mattson, a new master's student, approached me about conducting research on a *neural element* that turned out to be the linear combiner followed by a two-level output quantizer. The neuron output was binary, and Mattson made the inputs binary. Since all the inputs were binary and the output was binary, the neural element was a logic device whose logic function depended on the weight values. We had a novel logic device that could be varied by an adaptive process. Only certain logic functions were realizable with this single neuron, however, and they were called the linearly separable logic functions. Mattson did an M.S. thesis on this subject and was the one who started me thinking about artificial neural elements.

I continued to work on adaptive filters and adaptive Wiener predictors. In the autumn of 1959, I left MIT to join the electrical engineering faculty at Stanford. This was a great change for me, not only because Palo Alto was a lot smaller and prettier than Cambridge and the weather a lot nicer, but also because Stanford was in a growth mode, stealing people left and right from MIT, Harvard, Bell Labs, and other East Coast institutions. Stanford was highly receptive to new ideas that could lead to breakthrough technologies. I came with my head full of ideas about adaptive and learning systems, and this was well received. Prof. Hugh Skilling was chair of the Electrical Engineering Department, and Prof. John Linvill was his closest advisor, particularly regarding the acquisition of new faculty and the development of new research areas. John's twin brother Bill was my thesis advisor at MIT, and that was the connection that brought me to Stanford.

One day in the autumn of 1959, I received a call from John Linvill, who told me about one of his student advisees, Marcian E. ("Ted") Hoff, Jr. Ted was a very bright fellow who was looking for a subject for Ph.D. research. John had made some suggestions but had the feeling that Ted was looking for something else. He didn't know what this might be. He asked me to talk to Ted and see if I could propose something that would interest him.

I met Ted for the first time on a Friday afternoon in the fall of 1959. I was at my office blackboard explaining adaptive filters and the trainable neural element. I explained how the components of the gradient could be obtained by rocking each weight back and forth in the manner of elementary calculus and how steepest descent could be used with a series of steps to direct the weight vector toward the Wiener solution. The next weight vector equals the present weight vector plus a change proportional to the negative gradient, as follows:

$$W_{k+1} = W_k + \mu(-\nabla_k). \qquad (1)$$

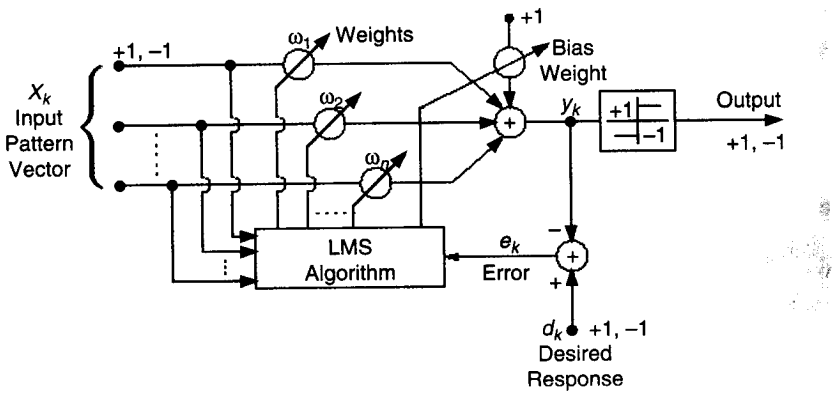
On the blackboard, I had written expressions for the error and the square

of the error for a linear combiner. In the course of discussion, a new idea "popped up" about differentiating the expression for the error to find the gradient. The true gradient is a vector of partial derivatives of the mean squared error with respect to the weights. The new idea used the square of a single value of error in place of the mean squared error. This was differentiated and gave an approximate gradient or a gradient estimate. Using this gradient estimate with steepest descent, we obtained a new adaptive algorithm as follows:

$$\begin{cases} W_{k+1} = W_k + 2\mu\varepsilon_k X_k \\ \varepsilon_k = d_k - X_k^T W_k \end{cases} \qquad (2)$$

where μ is known as the *learning rate*.

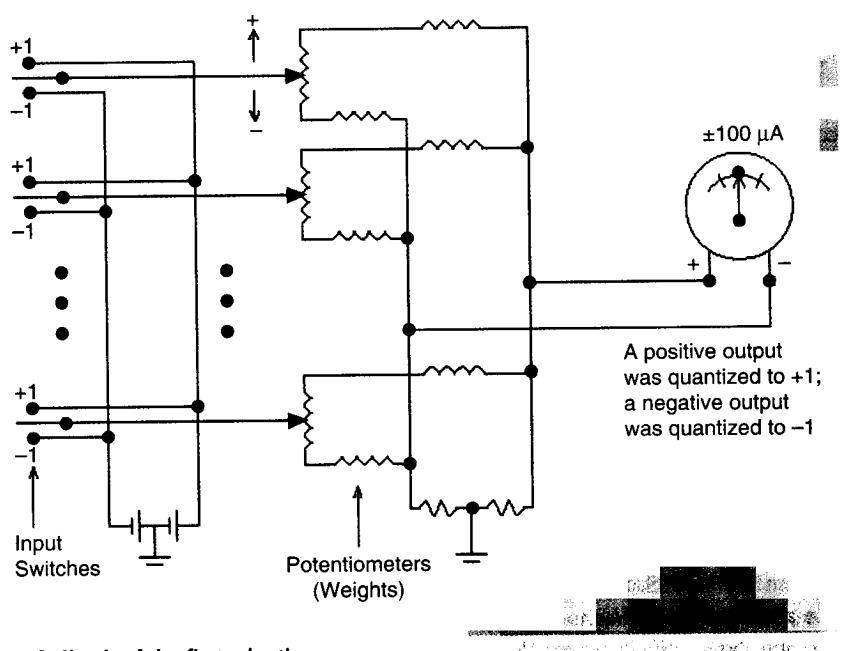
We didn't have a name for this algorithm. A year or so later, another one of my Ph.D. students, James S. Koford, gave it the name LMS algorithm for "least mean square," and the name stuck. The new algorithm used a very crude form of gradient based on a single error sample. This gradient estimate was obtained without squaring the error, taking averages, or differentiating. All components of the estimated gradient were obtained at once. The gradient estimate was proportional to the product of the instantaneous error and the vector of the inputs. This was an instantaneous gradient that was easily calculated. Seeing this algorithm on the blackboard, we knew that we had a significant discovery. But would this algorithm with its crude gradient actually work? Anxiety was mixed with elation. We could hardly wait to try it.



[FIG1] Block diagram of the ADALINE. When the ADALINE is trained using the LMS algorithm, weighted sums of the input patterns X_k are computed at each iteration k (with a bias included for each weight) to obtain the output values y_k. These output values are then compared with the desired output values d_k, yielding the instantaneous errors e_k given by (2). Then, the estimate of the gradient of the error function is obtained as follows: the gradient estimate is proportional to the product of the instantaneous error e_k and the vector of inputs. Using this gradient estimate, the new weight values for iteration $k + 1$ are computed using (1) or the weight update part of (2).

Prof. Gene Franklin had an analog computer that occupied a rather large room directly across the hall from my office. Ted was able to program its plugboard to implement the algorithm. Within a half hour after the discovery of LMS, it was up and running and we had an adaptive neuron being trained to do simple pattern classification tasks. We were very excited. We decided to build a single neuron in a small box for easier demonstration and experimentation. We needed an aluminum chassis, potentiometers, switches, batteries, a microammeter, etc. It was late Friday afternoon, and the Electronics Research Lab stockroom was already closed for the weekend.

That evening, we drew up a circuit and made a list of parts. The next day, we went to Zack Electronics in downtown Palo Alto and bought the parts. By late Sunday afternoon, the single neuron was fabricated and working. We were training it with all sorts of input patterns. We gave it the name ADALINE for "adaptive linear neuron." Its block diagram is shown in Figure 1, and its circuit is shown in Figure 2. The input signals were obtained from a set of switches mounted in a 4×4 array. Each input pixel was +1 or −1. The weights were analog and could take positive or negative values as required. A typical training experiment would cause the meter to



[FIG2] Circuit of the first adaptive neuron.

(continued on page 106)

(continued from page 102)

read positive values for an X-pattern, negative values for a T-pattern, positive values for a C-pattern, and negative values for a J-pattern. Within two or three iterations through those training patterns, the correct responses were learned. We began to have confidence in the LMS algorithm.

As time went by, we began to apply the LMS algorithm to adaptive signal processing. Work was commenced on adaptive antennas (adaptive arrays) and adaptive noise canceling. These were very successful applications. Use of the LMS algorithm really took off after the invention of the adaptive equalizer by R.W. Lucky and the adaptive echo canceller by M.M. Londhi and A.J. Presti at Bell Telephone Laboratories in the 1960s and 1970s. These were among the enabling technologies that made the Internet possible. Today, 45 years after its discovery, the LMS algorithm is still the world's simplest and most widely used learning algorithm.

REFERENCES

[1] B. Widrow and M.E. Hoff, Jr., "Adaptive switching circuits," in *Proc. IRE WESCON Conf. Rec.*, part 4, 1960, pp. 96–104.

[2] B. Widrow and M. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sept. 1990.

[3] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[4] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[5] S. Haykin and B. Widrow, *Least-Mean Square Adaptive Filters*. New York: Wiley, 2003.

[SP]