GTE TR88-509.4

NADALINE: A Normalized Adaptive Linear Element that Learns Efficiently

Richard S. Sutton
GTE Laboratories Incorporated

Abstract

This paper introduces a variant of the ADALINE in which the input signals are normalized to have zero mean and unit variance, and in which the bias or “threshold weight” is learned slightly differently. These changes result in a linear learning element that learns much more efficiently and rapidly, and that is much less dependent on the choice of the step-size parameter. Using simulation experiments, learning time improvements of from 30% to hundreds of times are shown. The memory and computational complexity of the new element remains $O(N)$, where N is the number of input signals, and the added computations are entirely local. Theoretical analysis indicates that the new element learns optimally fast in a certain sense and to the extent that the input signals are statistically independent.

NADALINE: A Normalized Adaptive Linear Element that Learns Efficiently

Richard S. Sutton
GTE Laboratories Incorporated

Summary

Widrow and Hoff's ADALINE is one of the most effective, most well understood, and most widely used connectionist learning components (Widrow and Hoff, 1960). Here we propose some small changes and additions to the ADALINE that result in large increases in its learning rate. The most important change is the addition of a preprocessing step in which each input signal is normalized to have zero mean and unit variance. Figure 1a shows a standard ADALINE, and Figure 1b shows an ADALINE with normalized input signals, i.e., a normalized ADALINE or NADALINE. The special input line that always carries the value +1 is not normalized; the associated bias or "threshold" weight is learned slightly differently in the NADALINE, as described below. The learning of all other weights, as well as the overall input-ouput transformation, is done exactly as it is in the standard ADALINE, but of course using the normalized rather than the original input signals. That is:

$$y(t) = \sum_{i=1}^{n} w_i(t) x_i'(t) + b(t)$$

and

$$w_i(t+1) = w_i(t) + \rho(d-y)x_i'(t),$$

where ρ is a positive step-size parameter and all other symbols are as given in Figure 1b.

Normalization is done by a local, linear process as follows. Let $\bar{x}_i(t)$ and $\sigma_i(t)$ be estimates at time t of the mean and standard deviation of the original input signal x_i. In the experiments reported here, these estimates were the sample mean and sample standard deviation for the input signal values seen up through time t, computed according to the standard incremental formulas.[*] The normalized signals $x_i'(t)$ were then computed as

$$x_i'(t) = \frac{x_i(t) - \bar{x}_i(t)}{\sigma_i(t)}.$$

[*] See, e.g., Iman & Connover, 1983. For the first few time steps it is possible for these sample statistics to be undefined; these cases were handled such that there was no consequent change in the weights.

It is clear that in the NADALINE the best value for the bias weight b is the mean of the desired or target response signal d.** Accordingly, $b(t)$ was set to the sample mean of the d values seen up to (but not including) t, computed as above for $\bar{x}_i(t)$. For non-stationary problems it would be better not to use sample statistics in this way, but to instead use running averages.

Figure 2 shows a simple comparison of the learning speeds ADALINE and NADALINE for a variety of values for the step-size parameter. The task was to learn (mimic) randomly selected training sets of 17 examples, each example consisting of an input vector of 20 binary (0/1) components and a corresponding desired response of either +1 or -1. Plotted is the average error per example after 25 presentations of the training set. The best errors for ADALINE are about four times those for NADALINE. Figure 3 shows a similar comparison for the case in which the binary input signals are chosen from $\{-1, +1\}$ rather than from $\{0, 1\}$. This is the case in which NADALINE's advantage is the least, as the input signals are already nearly normalized a priori. Nevertheless, a speedup of about 30% is obtained.

Notice that NADALINE's performance varies in approximately the same way as a function of the step-size parameter in both experiments, whereas ADALINE's dependence on ρ is different in the two experiements. Unless the input signals are very highly correlated, NADALINE reliably performs well at an ρ value of approximately $\frac{1}{n+1}$. This means that NADALINE can usually be used without any search for parameter values.

So far we have considered input signal values in $\{0, 1\}$ and $\{-1, +1\}$. Figure 4 is a summary comparison of learning times for ADALINE and NADALINE for a range of input signal values from $\{6, 7\}$ down to $\{-5, -4\}$. The x-axis is labeled with the first member of the pair, called the *base*; the second member is greater by 1. Thus, ADALINE performs best when the base is -0.5, so that the two values are symmetrically distributed around 0. The learning time is defined as the number of presentations of the training set required before all actual responses y are of the same sign as the desired responses d.

** This follows because we want, as a minimum, that $E\{d\} = E\{y\}$; but $E\{y\} = E\left\{\sum w_i \frac{x_i(t)-\bar{x}_i}{\sigma_i} + b\right\} = \sum w_i \frac{E\{x_i(t)\}-\bar{x}_i}{\sigma_i} + b) = b$, thus $E\{d\} = b$.

The ADALINE data was obtained by trying a variety of values for ρ, and taking the best learning time. The NADALINE data was obtained by using the single nominal learning rate value $\rho = \frac{1}{n+1}$. The simulations also used the "momentum" technique (Rumelhart, Hinton & Williams, 1985, see figure caption), which slightly speeds the learning of both elements.

These results show that ADALINE learning times are severely lengthened, in some cases by over a hundred times, if the mean value of an input signal departs very far from zero, whereas the NADALINE learning times are unaffected by such changes, and are always faster, even at the best base levels for ADALINE, and even when ADALINE's step size parameter is chosen to give it maximum advantage. Although we have considered only single units here, the advantages of normalization may have particular significance within networks. Although we may able to arrange for the original inputs to a network to be in $\{+1, -1\}$, or otherwise nearly normalized, the input signals from interior "hidden" units will not be, even if a symmetric squashing function is used (Stornetta & Huberman, 1987).

Although the theory of normalization remains to be fully worked out, and is anyway beyond the scope of this short paper, part of the reason for its effectiveness is immediately clear. It is well known that the convergence rate of the ADALINE is governed by the eigenvalues of the input cross-correlation matrix $R = [E\{x_i x_j\}]$. The larger the "spread", or ratio of the largest to the smallest eigenvalue, the slower convergence must be (Widrow and Stearns, 1985). If the input signals are statistically uncorrelated, then the normalization process converts the input cross-correlation matrix to the identity matrix. Thus, all of the eigenvalues are 1, the spread is minimized, and so therefore is convergence time. Normalization's effect on the spread in the case of statistically dependent signals, or on the misadjustment in either case, remain open questions for further research.

REFERENCES

Iman, R. L., & Connover, W. J. (1983). *A Modern Approach to Statistics.* New York: Wiley.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (Institute for Cognitive Science Technical Report 8506). La Jolla, Ca: University of California, San Diego. Also in D. E. Rumehart, & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations.* Cambridge, MA: MIT Press.

Stornetta, W. S., & Huberman, B. A. (1987). An improved three-layer, back propagation algorithm. *Proceedings of the First IEEE International Conference on Neural Networks, Vol. II*, pp. 637–643, San Diago, CA

Widrow B., & Hoff, M. E. (1960). Adaptive switching circuits. *1960 WESCON Convention Record Part IV* (pp. 96–104).

Widrow, B., & Stearns, S. D. (1985). *Adaptive signal processing.* Englewood Cliffs, NJ: Prentice-Hall.

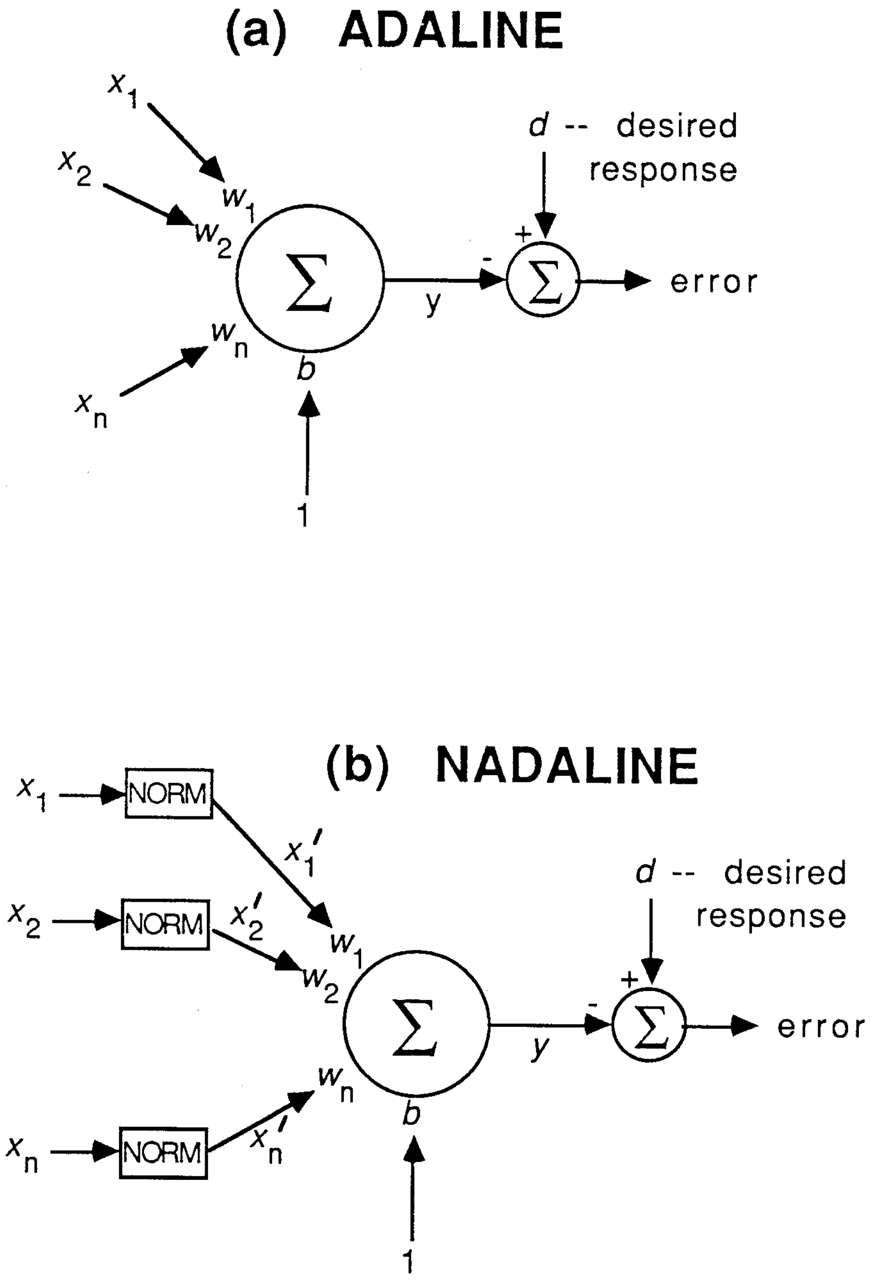

Figure 1. The primary difference between the classic ADALINE and the NADALINE is a local normalization of the input signals.

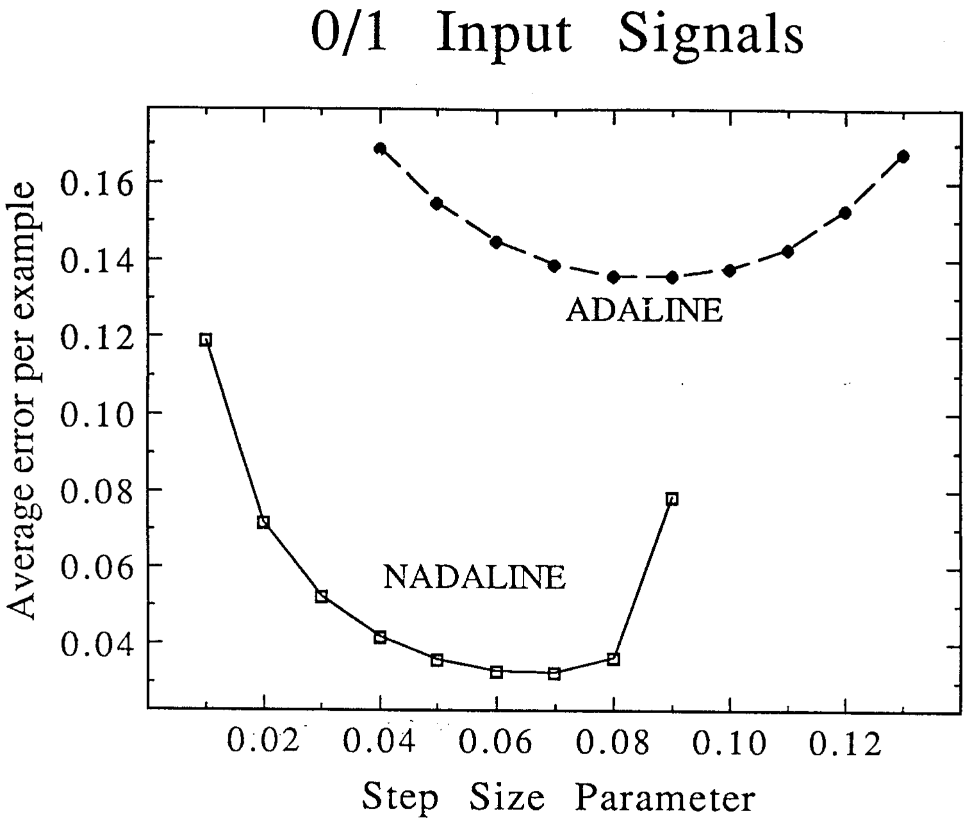

Figure 2. Comparison of ADALINE and NADALINE on a random binary mapping problem. The input signal values are all either 0 or 1. Plotted is the average error per example after 25 presentations of the training set as a function of ρ, the step-size parameter. Each data points represents the mean performance on 100 randomly selected training sets. The standard error of these means is approximately 0.01.

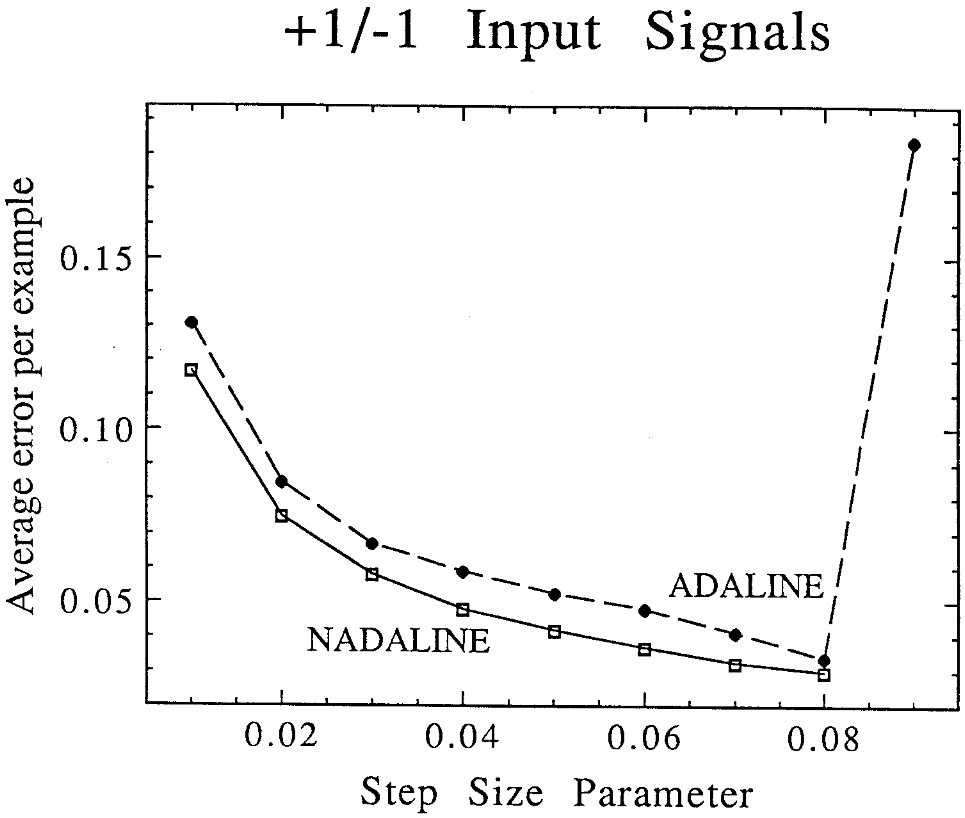

Figure 3. Comparison of ADALINE and NADALINE on a random binary mapping problem. The input signal values are all either +1 or -1. Plotted is the average error per example after 25 presentations of the training set as a function of ρ, the step-size parameter. Each data points represents the mean performance on 100 randomly selected training sets. The standard error of these means is approximately 0.01.

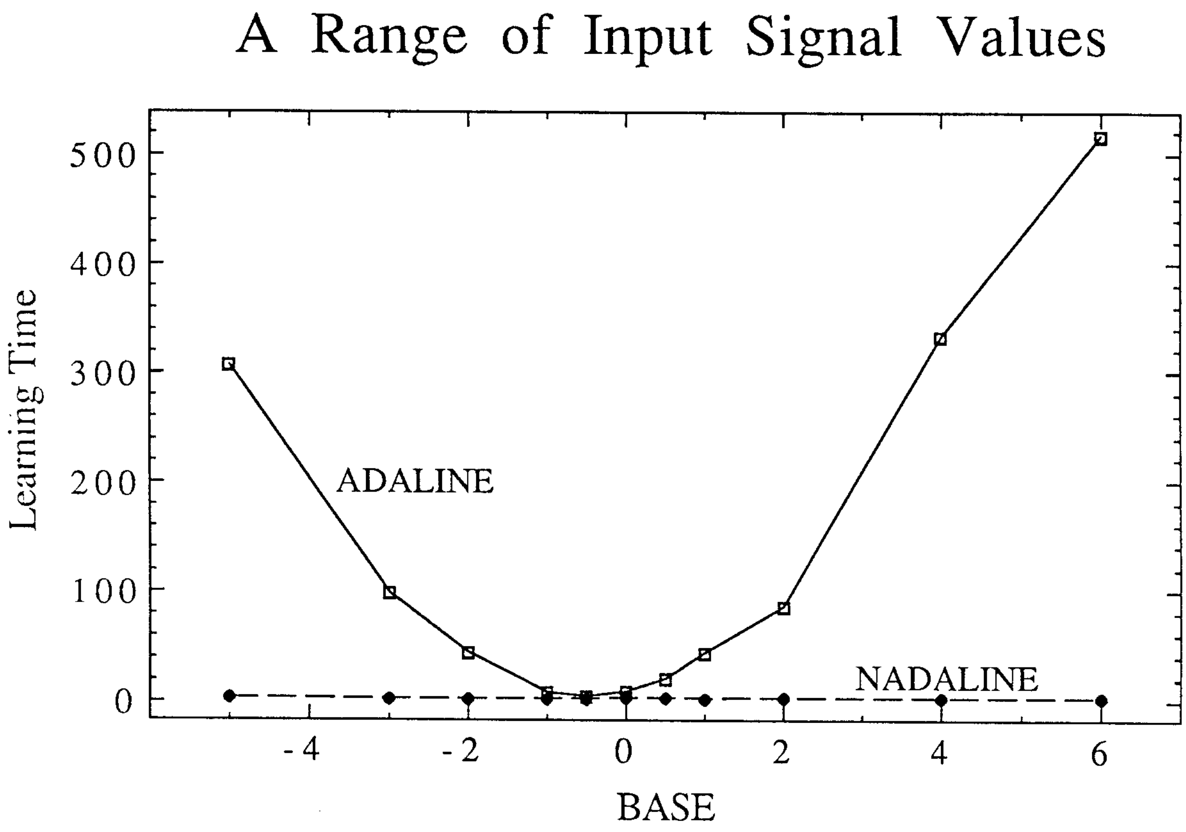

Figure 4. Summary comparison of ADALINE and NADALINE on a suite of tasks as in Figure 2 and 3. The input signal values are all either BASE or BASE+1. The learning time is the number of presentations of the training set before all responses are of the correct sign (targets were +1 and -1). The means are accurate to within 10% at the 0.95 level of significance. The ADALINE data points were obtaining by using many different ρ values and then taking the best performance. In obtaining these data, the momentum technique was also used for both algorithms.