Fuzzy ADALINEs for gray image recognition

Biswajit Paul¹, Amit Konar, Ajit K. Mandal* 

Department of Electronics and Telecommunication Engineering, Jadavpur University, Calcutta - 700 032, India

Received 8 September 1997; accepted 4 November 1998

Abstract

The paper aims at extending the scope of application of Widrow-Hoff’s ADALINE model from binary to gray-level (fuzzy) pattern recognition. The condition of stability for the extended ADALINE model has been derived and the algorithm for training the multilayered feedforward neural net consisting of ADALINE neurons have been presented. The time required for training the neural net is insignificantly small. Moreover, the scheme for the recognition of objects from their gray level images, using fuzzy ADALINE model, is translation, rotation and size invariant. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy ADALINE; Gray image; Translation; Rotation and size-invariant object recognition

1. Introduction

In early 1960s, Prof. Widrow of Electrical Engineering Department, Stanford University, opened up a new frontier of research on Neurocomputing, which had a far-reaching impact on intelligent signal processing for years to come. He introduced the concept of ADALINEs, the electrical analogue of the so-called biological neurons, which were self-adaptive and responsive to changes in input–output pattern by adjusting their weights autonomously. A non-linear signum type inhibiting element rendered the output of the ADALINES to binary \{ +1, −1\} level. The artificial neural net proposed by him is of multilayered feedforward geometry, consisting of...
a number of cascaded layers of ADALINES. The training algorithm for adaptation of
weights consists of (i) identification of an ADALINE at a given layer based on
minimum disturbance principle [4,10–14] and (ii) adjustment of weights by using
Delta rule [13]. The most important contribution of Widrow’s work, however, was
the selection of the weight matrices of ADALINES in retinal planes, in order to keep
the output of the ADALINE-planes insensitive to translation, rotation and magnification
of input binary patterns. The invariant scrambled binary patterns thus obtained
could be descrambled by a two-layer trained neural net.

The delta learning rule, also called the least-mean-square (LMS) algorithm [1],
proposed by Widrow-Hoff was successfully employed in solving many complex
problems of signal processing and pattern recognition. For example, Robert W.
Lucky of AT&T Bell Labs invented a new scheme of adaptive equalization of
telecommunication channels by employing a ‘decision oriented learning’ technique
[5] following the delta rule. At present the LMS algorithm is extensively used by
telecommunication industries for adaptive equalization of channels [13]. The ‘inverse
modeling problem’ [1] realised with ADALINE found applications in adaptive
control and ‘deconvolution’ in geophysical signal processing [13]. Even after the
invention of the backpropagation training algorithm [9], Widrow-Hoff’s model is
still used in many real-world applications, where hard non-linearity in the neuronal
model is essential. Recently, Hui and zak [2] applied the LMS algorithm to McCulloc–
Pitts-type neurons [6] with soft (differentiable) non-linearity and analyzed
its convergence. They have shown the robustness of the algorithm in presence of
imperfections in the non-linear activation function [6].

The present work is an extension of Widrow-Hoff’s ADALINE to enhance its
scope of application from binary to gray images [8]. In order to handle gray images,
the weights as well as the non-linear element of the ADALINEs are required to be
adaptive. Further, the functional form of the non-linearity and its position in the
neuronal structure also needs modifications. One possible type of the non-linearity is
the two level clipper [7], that could be positioned inside the loop of ADALINE, which
is in contrast to that of Widrow’s model, where the non-linear element was placed
outside the loop. Besides, to make the response of the system insensitive to translation,
rotation and scale change, the MAJ operator [3] in the retinal planes in Widrow’s
model should be replaced by AVG (Average) operator, that computes the arithmetic
average of the input signals.

The paper presents the condition for stability of the neuronal parameters that
guarantees the convergence of the Learning algorithm. Other important issues
covered in the paper are (i) the concurrent training of neurons in the entire network
and (ii) the design of the scrambler network that yields a unique output pattern, when
presented with input patterns having rotation, translation and size variance.

The paper has been divided into six sections. Section 2 of the paper deals with the
proposed model, while the condition for stability of system weights are discussed in
Section 3. Section 4 delineates the training algorithm of the modified multilayered
neural net consisting of ADALINE neurons. The aspects of translation, rotation and
size-invariant pattern recognition are discussed in Section 5. Finally, the conclusions
are summerized in Section 6.
2. The proposed model of ADALINE

In order to widen the scope of applications of the ADALINE neurons to include the multilevel (gray) image recognition, the following modifications are envisaged.

1. The signum-type non-linearity is to be replaced by a two-level clipper.
2. The position of the non-linear function block is shifted inside the loop, unlike the case of Widrow’s model, where it was on the forward path outside the loop.
3. A provision for stretching the linear portion of the non-linearity is to be incorporated in the model.

With these modification, the proposed model of ADALINE would look like the one in Fig. 1.

In the proposed model, with the input and output constrained in the range \([0,1]\), the unity slope in the nonlinearity of the neuron will fail to yield the desired output whenever the target signal is higher than the average of the inputs.

In order to circumvent this situation, an adaptive algorithm for adjustment of the slope of the linear portion in the non-linear clipper function is necessary. The detailed scheme for slope adjustment is presented in Fig. 2.

The clipper-type nonlinearity in the proposed ADALINE model can preferably be described by the following expression:

\[
Out_k = \left[ \left( (Net_k \land m_2') \lor m_1 \right) + \left( (Net_k \land m'_2) \lor m_1 \right) \right] \delta m, \quad (1)
\]

\[
\Delta w = \alpha \varepsilon_i x_i \sum w_i / x_i
\]

Delta Learning Rule

Fig. 1. Proposed model of ADALINE that supports fuzzy input-output pattern when \(d_k < \sum_{i=1}^{n} x_i/n\).
where “∧” and “∨” denote fuzzy AND (minimum) and OR (maximum) operators, respectively. The rest of the notations in expression (1) are evident from Fig. 3.

Unity slope of the linear region AC in Fig. 3 is increased by δm so that AC assumes new position AB.

**Theorem 1.** The functional representation of expression (1) describes the non-linearity PABQ.

**Proof.** The non-linear function PACQ, in Fig. 3 can be described as follows:

\[
\text{Out}_k = \begin{cases} 
  m_1, & \text{Net}_k \leq m_1, \\
  m_2, & \text{Net}_k \geq m_2, \\
  \text{Net}_k, & m_1 < \text{Net}_k < m_2.
\end{cases}
\]

Combining the above expressions, we find

\[
\text{Out}_k = (\text{Net}_k \wedge m_2) \vee m_1.
\]
When the slope of the central (linear) region is changed from AC to AB, the equation of line AB is given by

\[ \text{Out}_k = (1 + \delta m)\text{Net}_k + c \quad \text{where} \ c \ \text{is a constant.} \quad (3) \]

However, to keep Eq. (3) valid for \( m_1 < \text{Net}_k < m_2 \), we rewrite it as follows:

\[ \text{Out}_k = (1 + \delta m)[(\text{Net}_k \land m_2') \lor m_1] + c \quad \text{when} \ m_1 < \text{Net}_k < m_2'. \quad (4) \]

The discontinuity of the non-linearity at points A and B can be made (pseudo) continuous by replacing the symbol “<” in Eq. (4) by “≤”.

Further, the equations for regions PA and BQ are given by Eqs. (5) and (6), respectively,

\[ \text{Out}_k = m_1 \quad \text{when} \ \text{Net}_k \leq m_1, \quad (5) \]
\[ \text{Out}_k = m_2 \quad \text{when} \ \text{Net}_k \geq m_2'. \quad (6) \]

Now to compute the constant “c”, we consider the point A in Fig. 3 by using expressions (4) and (5), which yields

\[ m_1 = (1 + \delta m)[(m_1 \land m_2') \lor m_1] + c \]
\[ \Rightarrow m_1 = (1 + \delta m)m_1 + c \]
\[ \Rightarrow c = -\delta m m_1. \quad (7) \]
Substituting the value of “c” from Eq. (7) into Eq. (4), we find

\[ Out_k = (1 + \delta m)[(Net_k \land m'_2) \lor m_1] - \delta m m_1 \]

\[ = \left[ [(Net_k \land m'_2) \lor m_1] + \{(Net_k \land m'_2) \lor m_1) - m_1\} \delta m \right] \]

(Eq. (1) rewritten) \hspace{1cm} (8)

which satisfies expressions (4)–(6) uniquely. \qed

**Corollary.** When \( \delta m \to 0 \), \( m'_2 \to m_2 \), signifying that the linear portion is changed from \( AB \) to \( AC \).

The above corollary is, however, for academic interest only.

3. **Stability analysis for the convergence of weights of the proposed ADALINE model**

Consider the fuzzy ADALINE neuron of Fig. 2 where \( d_k \) and \( Out_k \) denote the target and the output signal of the ADALINE at iteration \( k \). Thus the error signal \( e_k \), at iteration \( k \), is given by

\[ e_k = d_k - Out_k = d_k - \left[ [(Net_k \land m'_2) \lor m_1] + \{(Net_k \land m'_2) \lor m_1) - m_1\} \delta m \right] \]

\[ = d_k - \left[ Net_k + (Net_k - m_1) \delta m \right], \quad \text{if} \quad m_1 < Net_k < m'_2, \hspace{1cm} (9) \]

where

\[ Net_k = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}. \]

It may be verified easily that when \( Net_k < m_1 \) or \( Net_k > m'_2 \), the training algorithm is unconditionally stable. We, thus attempt to find the condition for stability of neuronal parameters for \( m_1 < Net_k < m'_2 \).

Assuming that the input signals of the ADALINE at instant \( k \) be a vector \( X_k \) whose \( i \)th component \( x_i \) denotes the scalar signal at the \( i \)th input. Analogously, let \( W_k \) be the weight vector at instant \( k \), whose \( i \)th component \( w_i \) is the weight for the signal at the \( i \)th input.

Thus,

\[ \sum_{i=1}^n w_i x_i = X_k^T \cdot W_k \hspace{1cm} (10) \]

Substituting Eq. (10) in Eq. (9) we find,

\[ e_k = d_k - \frac{X_k^T \cdot W_k}{\sum_{i=1}^n w_i} - \left( \frac{X_k^T \cdot W_k}{\sum_{i=1}^n w_i} - m_1 \right) \delta m, \hspace{1cm} (11) \]
Hence,
\[ \Delta e_k = \Delta d_k - \frac{1}{\sum_{i=1}^{n} w_i} (X_k^T \cdot \Delta W_j (1 + \delta m)) + \Delta (m_1 \delta m) , \]
\[ = - \frac{1}{\sum_{i=1}^{n} w_i} X_k^T \cdot \Delta W_k (1 + \delta m) . \] (12)

Now, we shall use the following form of Widrow-Hoff’s delta rule:
\[ \Delta W_k = \alpha e_k \sum_{i=1}^{n} w_i \frac{X_k}{|X_k|^2} . \] (13)

Substituting expression (13) into expression (12), we find
\[ \Delta e_k = - \alpha e_k (1 + \delta m) . \] (14)

The solution of the above difference equation is given by
\[ e_k = e_0 [1 - \alpha (1 + \delta m)]^k . \] (15)

For the stability of the system, it is sufficient that the magnitude of error at iteration \( k \) should be less than that at iteration \( k - 1 \), i.e.,
\[ |1 - \alpha (1 + \delta m)| < 1 , \]
which yields,
\[ 0 < \alpha < \frac{2}{(1 + \delta m)} . \] (16)

For instability, \( e_k \to \infty \) as \( k \to \infty \) which yields
\[ |1 - \alpha (1 + \delta m)| > 1 \]
\[ \Rightarrow \alpha > \frac{2}{(1 + \delta m)} , \] since \( \alpha < 0 \) is not feasible. (17)

Lastly, for oscillatory behaviour, one can easily find that
\[ \alpha = \frac{2}{(1 + \delta m)} . \] (18)

4. Training of the proposed neural net

The proposed neural net consists of a number of layers with a number of ADA-
LINE neurons in each layer (see Fig. 5). Before describing the training process of the
neural net, we first present the training of a single ADALINE neuron.
4.1. Training algorithm of ADALINEs

The training algorithm of an ADALINE neuron having input vector $X$ and target scalar $d_k$ is presented below.

Procedure Adjust_Adaline;
Begin
Repeat
If $d_k \leq \frac{\sum_{i=1}^{n} x_i}{n}$
then do
begin
$\Delta W_k := \alpha \delta_k \sum_{i=1}^{n} w_i \frac{X_k}{|X_k|^2}$;
$\delta m := 0; W_{k+1} := W_k + \Delta W_k$;
end
else $\delta m := \beta \delta_k$;
if $d_k > m_2$
then $m_2 := d_k$;
if $d_k < m_1$
then $m_1 := d_k$;
m_2 := (m_2 + m_1 \cdot \delta m)/(1 + \delta m);
$\delta_k := d_k - [(1 + \delta m) \{Net_k \land m_2^k \lor m_1^k \} - (\delta m)m_1^k]$;
Until $\delta_k \leq$ pre-assigned_limit
end.

Example 1. The algorithm for training a single ADALINE neuron was simulated on a computer. The adaptation of one of the weights for different value of $a$, satisfying $d_i < \sum_{i=1}^{n} x_i/n$, has been presented in Fig. 4. It is clear that as $d_i < \sum_{i=1}^{n} x_i/n$, no adaptation of slope is required. Thus the range of stability of the system is $0 < a < 2$, since $\delta m = 0$. It is evident from the figure that as $a \to 2.00000$, the weight demonstrates temporal oscillatory behaviour. It may also be noted that the weights demonstrate underdamped, overdamped and critically damped behaviour for $2 > a > 1.2$, $0 < a < 1.2$ and $a = 1.2$, respectively.

4.2. Training of the neural net

For a given input and output pattern vector $X$ and $Z$, respectively, the components are mapped at nodes $x_i$ and $z_j$ for all $i, j$ (see Fig. 5a and b). The algorithm for training the neural net requires the adaptation of the weights and/or slopes of linear region of the ADALINEs in the network. The training algorithm may be initiated with arbitrary value of signals at the nodes of all intermediate layers. The algorithm
for weight adaptation of all neurons may be executed concurrently as described below.

Procedure Train–Neuralnet;
Begin
Randomize output of all ADALINEs excluding those at the last layer in the interval [0,1];
While error of each neuron > preassigned limit2
    Cobegin
        Adjust weights and nonlinearity of each neuron by procedure Adjust–Adaline;
        Coend;
    endwhile;
end.

4.3. Training with multiple input–output patterns

The algorithm for training the neural net with $m$-sets ($m > 1$) of input–output patterns is presented below. The algorithm yields a set of steady-state weights and non-linearity for each ADALINE of the network.
Fig. 5. (a) Symbolic representation of Fuzzy ADALINE \( AD_{ij} \) with inputs \([x_1, x_2, \ldots, x_n]^T\) and output \( y \).
(b) A typical feedforward neural net of Fuzzy ADALINE with hidden layers.

Procedure \textit{Train–with–multiple–I/O–patterns};
Randomize output of all ADALINES, excluding those in the last layer in [0,1];
For \( r := 1 \) to max–pass do // \( r = \text{pass–number} // \)
Begin
Repeat
\( PI_r := 0; // PI_r \equiv \text{performance index in } r\text{–th pass} // \)
For \( k := 1 \) to \( m \) do
Begin
For the k-th pair of I/O patterns do
Begin
(1) Adjust weights and nonlinearity of all ADALINEs once only;
(2) Estimate error vector \( E_k \) that includes components of error (\( \epsilon \)) of all
ADALINEs in the network;
(3) \( s_k := \sum_{j} e_j^{1/3} \), where \( e_j \) is the j-th component of \( E_k \);
\( P_{I_r} := P_{I_r} + S_k \);
Until \( P_{I_r} - P_{I_{r+1}} < \text{pre-assigned_positive_limit} \);
End_for;
End
End_for.
End.

It may be noted from the above procedure that steady state value of weights and
nonlinearity of the ADALINEs depend on the initial random output value of ADA-
LINEs.

One should thus run the above procedure for different set of randomized output
patterns of ADALINE and save the steady-state value of weights and nonlinearity for
that set of random pattern for which the performance index (\( P_{I_r} \)) is the smallest. An
alternative way to tackle the above problem might be to employ a Genetic Algorithm
to select the randomized output of ADALINEs for minimizing \( P_{I_r} \).

5. Translation, rotation and size invariant gray pattern recognition

In the first step, we design an invariance network which transforms a shifted,
rotated or scale-changed image into a unique scrambled image pattern. Such
scrambled pattern may be descrambled with the help of a trained neural net of the
proposed topology to reproduce the original patterns in “standard” position, orienta-
tion, scale, etc. The overall scheme of the system is presented in Fig. 6.

The invariance network consists of a number of planes of ADALINE (see Figs. 7
and 9) \([13]\), each of which receives signals from the retinal plane (Figs. 8 and 9). The
weights assigned to the input terminals of an ADALINE are taken together to form
a matrix, called weight matrix. Each retinal plane is assumed to correspond to
a number of such weight matrices.

5.1. Design of translation invariance network

Let us assume that a 2D view of an object moves horizontally in front of the retinal
plane but remains within the field of view of the plane. It is further assumed that each
weight matrix in the plane has \( p \times p \) pixels, whereas the entire plane consists of \( n \times n \)
pixels, where \( n \gg p \).
For translation invariance, we organize, following Widrow, the weight matrices of all ADALINEs, first by selecting the weight matrix for ADALINE AD\textsubscript{11} and then constructing the other weight matrices by the following procedure. For AD\textsubscript{ij}, the weight matrix should be horizontally rolled right and vertically down by \(i\) and \(j\) position, respectively, compared to AD\textsubscript{11}. 

Fig. 6. The overall scheme of gray pattern recognition.
Now if the 2D view of the object shifts right horizontally by \( m \) pixels where \( p < m < 2p \), the ADALINE closest to the right terminal of the 2D view will yield more signal strength and the ADALINE closest to the left terminal of the 2D view will have a decrease in output signal strength. As a consequence, the average of the output of the ADALINEs remains constant. We, therefore, connect an AVG module to take the average of the output of the ADALINEs. The details of each ADALINE plane is given in Fig. 7.

5.2. Design of rotational invariance network

For 90°, 180° and 270° rotational invariance we group, following Widrow [13], four planes of ADALINEs and thus resolution of the system is reduced to 1/4. The outputs of four planes of ADALINEs within one group are passed through an AVG module. The weight matrices of AD\(_{11}\) of the 2nd, 3rd and the 4th planes are constructed by rotating the weight matrix of AD\(_{11}\) under the first plane by 90°, 180° and 270° with respect to a fixed pivot. The other weight matrices under each plane of ADALINE are
designed satisfying translational invariance. The complete scheme for the system is given in Fig. 10.

5.3. Design of size invariance network

For size invariance recognition of gray image, we need not make any modification to Widrow Hoff’s scheme.

6. Conclusions

Widrow-Hoff’s ADALINE-based scheme for recognition of binary images has been extended in this paper to make it appropriate for gray images. For gray input–output patterns, both weights and nonlinearity of the ADALINE require adaptation, which is contrary to Widrow’s scheme, where only the weights require adaptation. The slope of the inhibiting two-sided clipper-type nonlinearity requires enhancement, when \( \text{Net}_i > d_i \). This, however, causes a further reduction in the range of \( \alpha \) for attaining stability. Under extreme cases, when \( \delta m \) approaches 1, \( \alpha(= 2/(1 + \delta m)) \) becomes unity, i.e. the permissible range of \( \alpha \) for stability of the system is \( 0 < \alpha \leq 1 \).

The adaptation of weights and nonlinearity of each neuron in the proposed neural net can be carried out in parallel, thereby reducing the training time significantly.

The proposed scheme for training with multiple input–output patterns has been simulated on IBM PC/AT and the results thus obtained support the theoretical foundations developed in this paper.
Fig. 9. (a) The geographical orientation of the ADALINE planes in the invariance network, where each ADALINE of each plane can receive signals from the butterfly as in Fig. 9b. (b) Each ADALINE of the plane receiving signals from the butterfly.
Fig. 10. Grouping of 4-ADALINE planes together to obtain 0°, 90°, 180°, and 270° rotational invariance.

Acknowledgements

The authors thank the referees for their valuable comments in connection with the improvement of the paper.

References


Biswaajit Paul received his B.E. degree in Electronics and Telecommunication Engineering from Bengal Engineering College, Calcutta in 1983 and currently is a Ph.D. (Engg.) candidate, working for his dissertation on artificial intelligence. After serving Oil and Natural Gas Commission, India as an Assistant Executive Engineer for around one year, Mr. Paul joined Indian Telecommunication services in 1985. There he was responsible for the maintenance of the microwave, satellite and optical fibre transmission systems until 1989, when he was promoted to Divisional Engineer (DE). As a DE, he was responsible for system analysis and design, programming and the maintenance of computer networks. In 1996, he was promoted to Deputy General Manager in Calcutta Telephones and was involved with the installation of “new technology” digital telephone exchanges. Currently, he is an Area Manager, where he is in charge of maintenance of telecommunication network in the southern part of the city.

Amit Konar received a B.E. degree in electronics and telecommunication engineering from Bengal Engineering College, Calcutta in 1983. He also earned an M.E. degree in control engineering, an M. Phil. Degree in environmental science and a Ph.D. (Engg.) degree in artificial intelligence from Jadavpur University, Calcutta in 1985, 1988 and 1994, respectively. Dr. Konar joined the department of electronics and telecommunication engineering, Jadavpur University, as a lecturer in 1985 and currently is a Reader in the same department. He supervised a number of Ph.D. and M.E. theses in different areas of artificial intelligence including logic programming, neural networks, cognitive systems, stochastic and fuzzy models of uncertainty, fuzzy algebra, image understanding and navigational planning of mobile robots. He has published more than 60 papers in international journals and conferences and is an invited author of a chapter in an edited book to appear shortly from Academic Press. Dr. Konar is a recipient of the 1997 young scientist award, offered by the All India Council for technical Education (AICTE) for his significant contribution in artificial intelligence. He is the founder of “Robotics and Vision Laboratory” in his department and actively participated as an investigator in a number of Government funded research projects.

Ajit K. Mandal holds a M.Tech Degree in Radio Physics and Electronics and earned a Ph.D. from the University of Calcutta, India. He is currently a Professor in the Department of Electronics and Telecommunication Engineering, Jadavpur University, Calcutta and served it as its Head from August 1992 to July 1994. He was the Principal Investigator of the project “Automatic Diagnostic System for Heart Diseases” sponsored by the Dept. of Electronics, Govt. of India, and the Scientist in Charge of the Eastern Regional Centre of “Appropriate Automation Promotion Programme” funded by Dept. of Electronics, Govt. of India in collaboration with UNDP (1988–1990). His research interests are fuzzy logic, neural networks, and genetic algorithm and machine learning with applications to pattern recognition and digital image processing. Dr. Mandal is a senior member of IEEE and the Vice-Chairperson of Computer Chapter of IEEE Calcutta Section. He is a fellow of the Institution of Engineers, India.