NOISE-TOLERANT SPECTRAL SIGNATURE CLASSIFICATION IN NON-RESOLVED OBJECT DETECTION USING ADAPTIVE LATTICE NEURAL NETWORKS

Mark S. Schmalz mssz@cise.ufl.edu
Gerhard X. Ritter ritter@cise.ufl.edu
Eric T. Hayden hayden@cise.ufl.edu

Department of Computer and Information Science and Engineering
University of Florida, Gainesville FL 32611-6120

Gary Key
gkey@fti-net.com

Frontier Technology, Inc.
Altamonte Springs, FL 32714-2019

ABSTRACT

Accurate spectral signature classification is key to reliable nonimaging detection and recognition of spaceborne objects. In classical hyperspectral recognition applications, especially where linear mixing models are employed, signature classification accuracy depends on accurate spectral endmember determination. In previous work, it has been shown that class separation and classifier refinement results in Bayesian rule-based classifiers and in classical neural nets (CNNs) based on the linear inner product tend to be suboptimal. For example, the number of signatures accurately classified often depends linearly on the number of inputs. This can lead to potentially severe classification errors in the presence of noise or densely interleaved signatures. Such problems are exacerbated by the presence of input nonergodicity.

Computed pattern recognition, like its human counterpart, can benefit from processes such as learning or forgetting, which in spectral signature classification can support adaptive tracking of input nonergodicities. For purposes of simplicity, we model learning as the acquisition or insertion of a new pattern into a classifier's knowledge base. For example, in neural nets (NNs), this insertion process could correspond to the superposition of a new pattern onto the NN weight matrix. Similarly, we model forgetting as the deletion of a pattern currently stored in the classifier knowledge base, for example, as a pattern deletion operation on the NN weight matrix, which is a difficult goal with classical neural nets (CNNs). In practice, CNNs have significant disadvantages of poor classification accuracy, limited information storage capacity, poor convergence, and long training times, which have been remedied by the development of neural networks based on lattice algebra. The first two authors have elsewhere shown that such *lattice neural networks* (LNNs) can be configured as auto- or hetero-associative memories and are amenable to pattern insertion or deletion operations on the LNN weight matrix.

In this paper, we detail the implementation of pattern insertion and deletion in lattice associative memories (LAMs), in support of signature classification. It is shown that, for an n-input LAM having an nxn-element weight matrix, pattern insertion and deletion from the weight matrix can be computed exactly in $\mathbf{O}(n)$ addition operations, with a small proportionality constant. Adaptive classifiers based on LNN technology can thus achieve accurate signature classification in the presence of time-varying noise, closely spaced or interleaved signatures, and imaging system optical distortions. As proof of principle, we exemplify classification of multiple closely spaced, noise corrupted signatures from a NASA database of space material signatures.

Keywords: Automated signature detection, Pattern recognition

1. INTRODUCTION

Non-resolved detection and classification of space objects using features such as spectral signatures, or mixtures of such signatures, requires accurate, comprehensive signature classification technology [1-3]. Although passive remote sensing research has advanced significantly over the past decade, yielding imaging devices with increasing spectral coverage and resolution, the development of classifier technology has progressed more slowly. In the linear mixing models from which hyperspectral image understanding derives, *spectral endmembers* represent fundamental materials that are mixed via various abundance fractions to yield signatures that characterize remotely sensed objects. Regardless of the type of demixing and classification of abundance fraction vectors that are employed in hyperspectral image understanding, if one cannot accurately classify spectral endmembers under various noise constraints, then classification of signatures based on mixed endmembers is not feasible in practice.

Additionally, in the development of adaptive pattern recognition systems, nonergodic input is often encountered in realistic applications. Nonergodicity can affect the accuracy of pattern classification, because input statistics tend to drift away from the statistical assumptions upon which a classifier is based. In previous research, we have found that it is useful to be able to add or remove patterns from a pattern database, to improve classifier performance in response to input nonergodicity [4]. These processes of addition or removal are respectively similar to notions of human remembering or forgetting. In pattern classification technology, such *classifier updates* can help maintain correspondence between input statistics and underlying classifier assumptions, and can keep the classifier's knowledge base (pattern memory) at fixed size, to achieve and maintain storage efficiency.

In the case of neural network based classifiers, one desires that pattern updates could be performed incrementally, in real or near-real time. Unfortunately, classical neural networks (CNNs) have many disadvantages, including not being able to add representations of input patterns to their weight matrices without recomputation of the entire NN weight matrix. Due to the computational cost of multiplications inherent in the CNN linear inner-product formulation, the concept of incremental or continuous pattern update remains primarily of pedagogic interest for CNNs with large weight matrices.

Fortunately, lattice associative memories (LAMs) have much lower computational cost in their training and classification steps, very high classification accuracy, and theoretical maximum information (pattern) storage capacity [5]. As such, LAMs exhibit many advantages over CNNs, including but not limited to very high classification accuracy in noisy environments, theoretical maximum information storage capacity, fast computation, fast convergence and training. Due to their foundations in lattice algebra, we have found that LAMs are good candidate platforms for the development of pattern insertion and deletion algorithms, in support of adaptive classifier technology.

In this paper, we first overview lattice associative memory theory (Section 2), then show how patterns are added or removed in a type of LAMs called *lattice autoassociative memories* (LAAMs, Section 3). We extend this theory to *lattice heteroassociative memories* (LHAMs Section 4), and discuss how this extension overcomes problems associated with incremental updates due to existence of fixed points in LAAMs. Pattern insertion and deletion operations for LAAMs and LHAMs are exemplified in Section 5. Preliminary conclusions and suggestions for future work are given in Section 6.

2. OVERVIEW OF LATTICE ASSOCIATIVE MEMORIES

Let a source domain $\mathbf{X}$ = {1,2, …, n}, such that patterns in $\mathbf{X}$ are given by $\mathbf{x}^1 = 1, \ldots, \mathbf{x}^n = n$. Let the weight matrices M and W be formed for a lattice auto-associative memory, as follows:

$$m_{ij} = \bigvee_{k=1}^{n} (\mathbf{x}_i^k - \mathbf{x}_j^k) \quad \text{and} \quad w_{ij} = \bigwedge_{k=1}^{n} (\mathbf{x}_i^k - \mathbf{x}_j^k) \,. \tag{1}$$

Note that $W = -M'$, where unary minus denotes negation (sign flip) of the elements of M transposed.

The correct operation of M can be verified using the additive minimum operation in image algebra [6]:

$$M \boxtimes \mathbf{x}^k = \mathbf{x}^k \,, \tag{2}$$

and similarly using W and additive maximum. M and W can also be denoted as M_{XX} and W_{XX}, since as autoassociative memories, they map $\mathbf{X}$ to $\mathbf{X}$. The case of M_{XY} and W_{XY}, where $\mathbf{Y}$ is not the same as $\mathbf{X}$, is called *heteroassociative memories*, and is discussed in Section 4.

Given this basic overview of LAMs, we next progress to the problem of dynamically inserting a pattern into an LNN weight matrix, without recomputation of the entire matrix.

3. PATTERN OPERATIONS IN LATTICE AUTO-ASSOCIATIVE MEMORIES

Suppose we have a lattice autoassociative memory (LAAM) W_{XX}, where $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$. Consider the insertion of a new pattern $\mathbf{x}^{k+1}$ such that W_{XX} will recognize $\mathbf{x}^{k+1}$ presented as input to W_{XX}, for example, using a formulation similar to Equation (2).

3.1. Pattern Insertion in LAAMs

To compute the augmented memory $\underline{W}_{XX}$, with the new pattern set $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k, \mathbf{x}^{k+1}\}$, we simply calculate

$$\underline{w}_{ij} = \begin{cases} w_{ij} & \text{if } w_{ij} \le \mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1} \\ \mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1} & \text{if } w_{ij} > \mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1} \end{cases}, \ \forall i, j \in \{1, \ldots, n\} \tag{3}$$

Note that the size of W (i.e., n x n elements) does not change.

Proof. Using the fact that for $a, b, c \in \mathbf{R}$, $a \wedge b \wedge c \le a \wedge b$, we always have that

$$\underline{w}_{ij} = \bigwedge_{\varsigma=1}^{k+1} (\mathbf{x}_i^\varsigma - \mathbf{x}_j^\varsigma) \le \bigwedge_{\varsigma=1}^{k} (\mathbf{x}_i^\varsigma - \mathbf{x}_j^\varsigma) = w_{ij} \cdot$$

Therefore, $\underline{w}_{ij} \le w_{ij} \quad \forall i, j \in \{1, \ldots, n\}$.

Now suppose that $w_{ij} \le (\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1})$ for some i, j. Then,

$$w_{ij} = w_{ij} \wedge (\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1}) = \left[\bigwedge_{\varsigma=1}^{k} (\mathbf{x}_i^\varsigma - \mathbf{x}_j^\varsigma) \right] \wedge (\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1}) .$$

$$= \bigwedge_{\varsigma=1}^{k+1} (\mathbf{x}_i^\varsigma - \mathbf{x}_j^\varsigma) = \underline{w}_{ij}$$

Thus, in this case, $w_{ij} = \underline{w}_{ij}$.

Next suppose that $\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1} < w_{ij}$ for some i, j. Then,

$$\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1} = w_{ij} \wedge (\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1}) = \left[\bigwedge_{\varsigma=1}^{k} (\mathbf{x}_i^\varsigma - \mathbf{x}_j^\varsigma) \right] \wedge (\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1})$$

$$= \bigwedge_{\varsigma=1}^{k+1} (\mathbf{x}_i^\varsigma - \mathbf{x}_j^\varsigma) = \underline{w}_{ij}$$

Therefore, $\underline{w}_{ij} = \mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1}$, in this case. ■

Remark 1. If $(\mathbf{x}_i^{k+1} - \mathbf{x}_j^k) \le w_{ij}$, then the computational cost can be reduced by setting $\underline{w}_{ij} = w_{ij}$, else the lower part of Equation (3) is implemented. Again, the computational cost involves only one subtraction operation.

3.2. Pattern Deletion in LAAMs

Suppose we have an autoassociative memory W_{XX}, where $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$, as before. Consider deletion of a pattern $\mathbf{x}^\lambda$ from W_{XX}, where $1 \le \lambda \le k$. We first let $\underline{\mathbf{X}} = \mathbf{X} \setminus \{\mathbf{x}^\lambda\}$, where (\) denotes set subtraction (i.e., deletion of $\mathbf{x}^\lambda$ from X). Instead of directly computing the new memory $W_{\underline{XX}}$, which incurs $\mathbf{O}(k^2)$ operations, we let w_{ij} denote the $(i, j)^{\text{th}}$ entry of W_{XX} and $\underline{w}_{ij}$ denote the $(i, j)^{\text{th}}$ entry of $W_{\underline{XX}}$, as before, then perform the following:

$$\underline{w}_{ij} = \begin{cases} w_{ij} & \text{if} \quad w_{ij} < \mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda} \\ \bigwedge_{\zeta=1,\, \zeta \neq \lambda}^{k} (\mathbf{x}_i^{\zeta} - \mathbf{x}_j^{\zeta}) & \text{if} \quad w_{ij} = \mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda} \end{cases} \tag{4}$$

Remark 2. Note that $\mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda} < w_{ij}$ is impossible, since

$$w_{ij} = \bigwedge_{\zeta=1}^{k} (\mathbf{x}_i^{\zeta} - \mathbf{x}_j^{\zeta}) \leq \mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda}, \ \forall \lambda \in \{1,\ldots,k\} \text{ and } \forall i, j\,.$$

Defining the entries $\underline{w}_{ij}$ according to Equation (4) results in the same memory as computing $W_{\underline{XX}}$ from scratch (i.e., by complete recomputation of $W_{\underline{XX}}$), but without the computational cost.

Proof. If $w_{ij} < \mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda}$ for some $i, j \in \{1,\ldots,k\}$, then

$$w_{ij} = \bigwedge_{\varsigma=1}^{k} (\mathbf{x}_i^{\varsigma} - \mathbf{x}_j^{\varsigma}) = (\mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda}) \wedge \left[\bigwedge_{\varsigma=1,\, \varsigma \neq \lambda}^{k} (\mathbf{x}_i^{\varsigma} - \mathbf{x}_j^{\varsigma}) \right]$$

$$= (\mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda}) \wedge \underline{w}_{ij} = \underline{w}_{ij}$$

This equation follows from the fact that if $a, b, c \in \mathbf{R}$ with $a < b$, then $a = b \wedge c \Rightarrow c = a$. So in the preceding equation, let $a = w_{ij}$, $b = \mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda}$, and $c = \underline{w}_{ij}$.

Next suppose $\underline{w}_{ij} = \mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda}$. In this case, we have

$$\mathbf{x}_i^{\lambda} - \mathbf{x}_j^{\lambda} = w_{ij} = \bigwedge_{\varsigma=1}^{k} (\mathbf{x}_i^{\varsigma} - \mathbf{x}_j^{\varsigma}) \leq \bigwedge_{\varsigma=1,\, \varsigma \neq \lambda}^{k} (\mathbf{x}_i^{\varsigma} - \mathbf{x}_j^{\varsigma}) = \underline{w}_{ij}$$

or $w_{ij} \leq \underline{w}_{ij}$. Note that if $w_{ij} \leq \underline{w}_{ij}$ for at least one pair $\{i, j\} \subset \{1,\ldots,n\}$, then the pattern will have been successfully removed. However, if $\underline{w}_{ij} = w_{ij} \ \forall i, j \in \{1,\ldots,n\}$, then the pattern is *lattice dependent* on the set $\mathbf{X} \setminus \{\mathbf{x}^{\lambda}\}$ and will not have been erased. This follows from the fixed point theorems for W_{XX} and M_{XX} stated in [7]. ■

This problem with fixed points can be remedied using lattice heteroassociative memories, as follows.

4. PATTERN OPERATIONS IN LATTICE HETERO-ASSOCIATIVE MEMORIES

Suppose we have a lattice heteroassociative memory (LHAM) W_{XY}, where $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$, as before, and $\mathbf{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^k\}$, such that $\mathbf{Y} = \{$*one, two, …, letter-n*$\}$, that is, patterns in $\mathbf{Y}$ are given by $\mathbf{y}^1 = one$, …, $\mathbf{y}^n = letter - n$, and so forth.

4.1. Pattern Insertion in LHAMs

Further suppose that we want to add a new association $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})$ such that W_{XY} will output $\mathbf{y}^{k+1}$ when $\mathbf{x}^{k+1}$ is presented as input to W_{XY}. To compute the augmented memory $\underline{W}_{XY}$ with the new pattern sets $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k, \mathbf{x}^{k+1}\}$ and $\mathbf{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^k, \mathbf{y}^{k+1}\}$, we simply calculate

$$\text{If } (\mathbf{y}_i^{k+1} - \mathbf{x}_j^{k+1}) > w_{ij}\text{, then } \underline{w}_{ij} = w_{ij}\,. \quad (5a)$$

$$\text{else } \underline{w}_{ij} = w_{ij} \wedge (\mathbf{y}_i^{k+1} - \mathbf{x}_j^{k+1})\,,\ \forall i, j \in \{1,\ldots,n\} \quad (5b)$$

As before, note that the size of W (i.e., n x n elements) does not change.

4.2. Pattern Deletion from LHAMs

Let $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$, as before. To remove a pattern $\mathbf{x}^\lambda$ from the memory W_{XY}, where the pattern index $1 \leq \lambda \leq k$, we perform the following:

$$\text{If } (\mathbf{y}_i^{\lambda} - \mathbf{x}_j^{\lambda}) > w_{ij}\text{, then } \underline{w}_{ij} = w_{ij}\,. \quad (6a)$$

$$\text{else if } (\mathbf{y}_i^{\lambda} - \mathbf{x}_j^{\lambda}) = w_{ij}\text{, then } \underline{w}_{ij} = \bigwedge_{\zeta \neq \lambda \text{ and } \zeta \neq 1}^{k} (\mathbf{y}_i^{\zeta} - \mathbf{x}_j^{\lambda}) \quad (6b)$$

The proofs of Equations (5) and (6) follow from the discussion of Section 4 and the observation that, in MHAMs, fixed points are not a problem because fixed points do not exist in heteroassociative memories, as the mapping (e.g., W_{XY}) is between different sets **X** and **Y**. Unfortunately, the MHAMs W_{XY} and M_{XY} do not possess the perfect recall features of MAAMs. We are currently investigating solutions to this challenging situation, using dendritic lattice associative memories (DLAMs, [5]).

5. EXAMPLES OF PATTERN INSERTION / DELETION IN LAAMS AND LHAMS

Given the 16 x 16 pixel input patterns shown in Figure 1, a LAAM was constructed using the procedure described in Section 2, which yielded the M and W weight matrices shown in Figure 2. This memory stored all patterns (1 through 5), and recalled all patterns correctly, such that $M \boxed{\wedge} \mathbf{x}^k = \mathbf{x}^k$ and $W \boxed{\vee} \mathbf{x}^k = \mathbf{x}^k$.

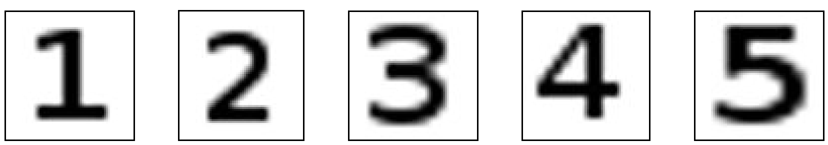


Figure 1. Input images (16 x 16 pixels) for MAAM test example.

We then deleted pattern "3" from the MAAM, via the procedure described in Section 4. This yielded the revised M and W weight matrices shown in Figure 3, which show subtle differences when compared with Figure 2. Application of the input $\mathbf{x}^3$ = pattern "3" (Figure 4a) to M and W (Figure 3) respectively yielded the results shown in Figure 4b-c. To obtain the correct result of zero output for input $\mathbf{x}^3$ not represented in M and W, we multiply image 4b by image 4c, i.e., $(W \boxed{\vee} \mathbf{x}^3)(M \boxed{\wedge} \mathbf{x}^3) = 0$. This can be exhaustively shown to work correctly for any pattern not stored in the MAAM. Finally, re-insertion of $\mathbf{x}^3$ into the revised MAAM (Figure 3) yielded weight matrices shown in Figure 2, and the correct result $W \boxed{\vee} \mathbf{x}^3 \simeq M \boxed{\wedge} \mathbf{x}^3$ shown in Figure 4a.

We also applied LHAMs to classification of numerical patterns (set **X**) and pictures of words (in set **Y**) that corresponded to the numbers in **X**. In each case, we were able to employ our LAM testbed shown in Figure 5, varying the noise (standard deviation in dialog box at bottom-right-hand side of Figure 5) until imperfect classification occurred, typically, at noise standard deviation $\sigma \geq 0.06$.

In an additional experiment, the spectral signatures illustrated in Figure 6 were classified with LHAMs using the testbed shown in Figure 5. In preliminary results, perfect classification was achieved for each spectrum with respect to all other spectra for noise standard deviation $0 \leq \sigma \leq 0.04$ at 100 percent noise cross-section, with respect to a range of [0,1], and for mean values $0 \leq \mu \leq 0.5$. These tests will be extended to various types and levels of noise in future research.

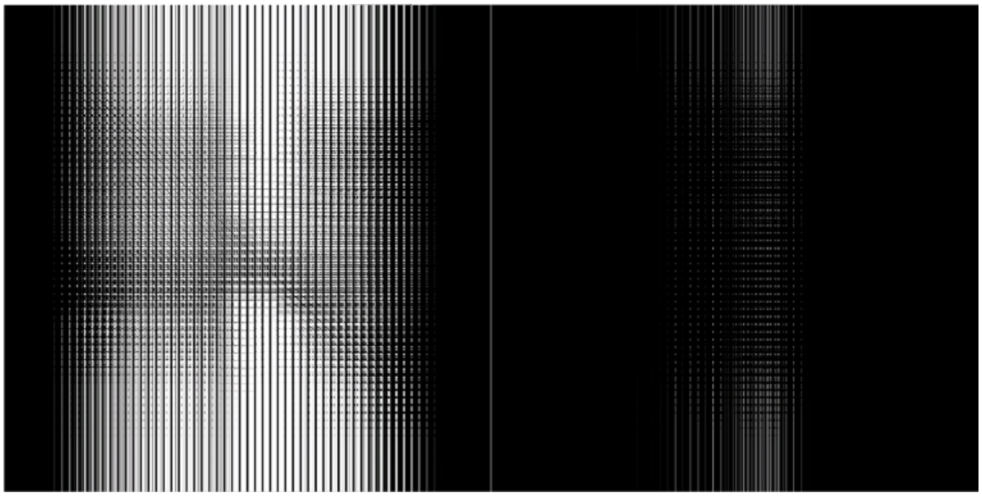


Figure 2. *W* and *M* matrices of MAAM constructed with patterns from Figure 1.

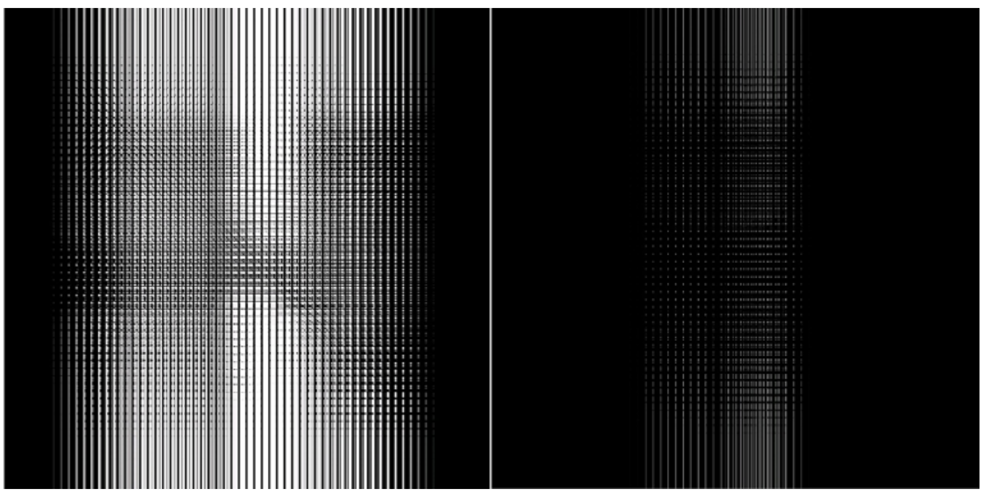
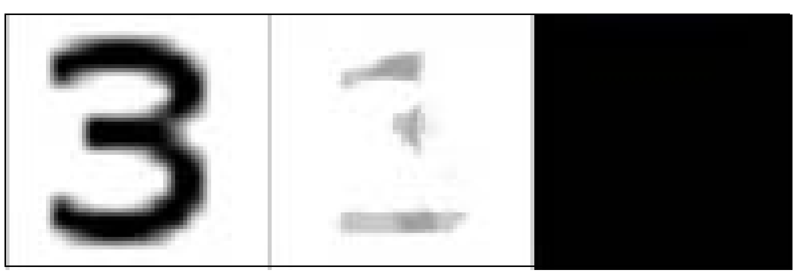


Figure 3. *W* and *M* matrices of MAAM from Figure 2, after pattern “3” is deleted.



(a) (b) (c)

Figure 4. Results of MAAM pattern deletion: (a) MAAM input and output with original weight matrices (Figure 2); after $\mathbf{x}^3$ = pattern “3” deleted we have (b) result of $W \boxed{\vee} \mathbf{x}^3$, and (c) result of $M \boxed{\wedge} \mathbf{x}^3$. It is readily seen that $(W \boxed{\vee} \mathbf{x}^3)(M \boxed{\wedge} \mathbf{x}^3) = 0$.

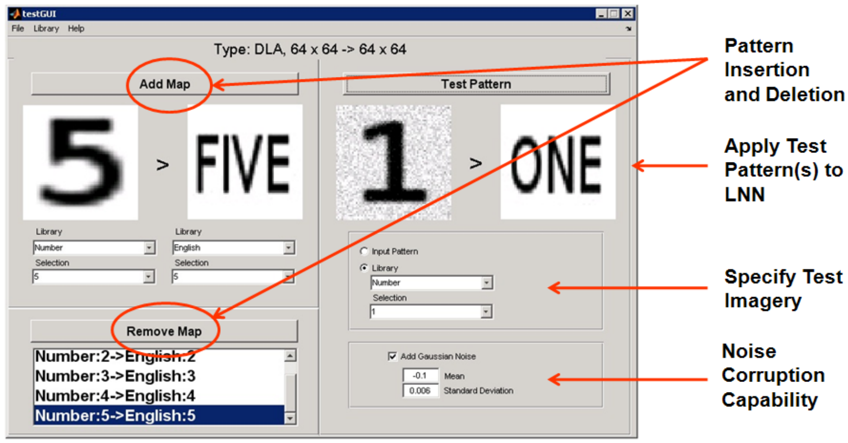


Figure 5. Example computational testbed for LAM development.

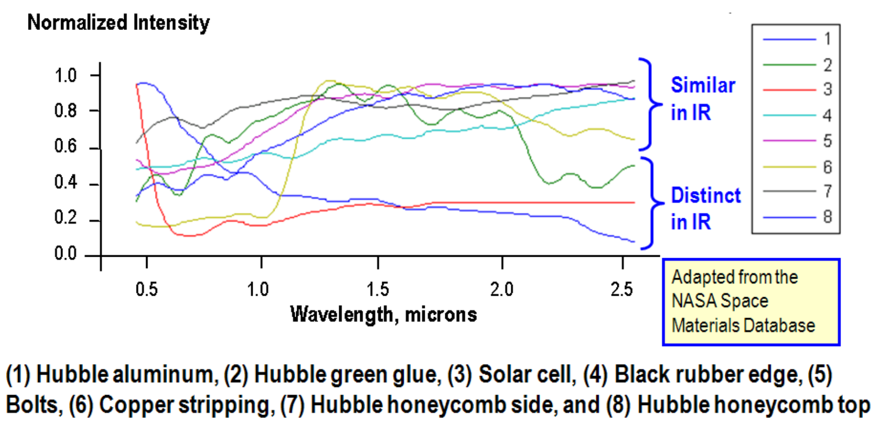


Figure 6. Normalized test spectra, adapted from NASA database of space materials.

6. CONCLUSIONS AND FUTURE WORK

We have presented a simple technique for inserting and deleting patterns from lattice auto- and hetero-associative memories (LAMs), similar in concept to human practices of remembering and forgetting. This approach is useful for various purposes including (a) maintaining storage efficiency of a LAM, (b) tracking input statistical changes by changing the statistics of the memory to match input statistics, as well as (c) expanding or contracting the pattern database stored in the memory, as application scope changes. We successfully demonstrated this approach for lattice auto-associative memories, using a simple recall of images of Arabic numbers and corresponding names of numbers. In a separate experiment, we demonstrated perfect classification with LAMs of space materials spectra under moderate Gaussian noise.

Unlike classical associative memories based on a linear inner product, which require recomputation of their entire weight matrices whenever a pattern is inserted or deleted, lattice associative memories can insert or delete patterns from their weight matrices with a much smaller amount of computation. This is due to the exploitation of locality of pattern storage in the memory, and to the lattice memory's significantly lower computational cost.

Future work involves extended testing with different noise levels and distributions, as well as augmentation of the presented techniques using dendritic lattice associative memories. We are also investigating techniques for improving the recall accuracy of lattice heteroassociative memories, which will form the basis for the majority of our forthcoming research in adaptive spectral classification.

7. REFERENCES

[1] Winter, M.E. “Fast autonomous spectral end-member determination in hyperspectral data,” in *Proceedings of the 13th International Conference On Applied Geologic Remote Sensing*, Vancouver, B.C., Canada, pp. 337-44 (1999).

[2] N. Keshava, “A survey of spectral unmixing algorithms,” *Lincoln Laboratory Journal* **14**:55-78 (2003).

[3] Ritter, G.X., G. Urcid, and M.S. Schmalz. “Autonomous single-pass endmember approximation using lattice auto-associative memories”, *Neurocomputing* 72: 2101-2110 (2009).

[4] Schmalz, M.S. and G.X. Ritter. “Spectral discrimination in noisy sensors using morphological neural computation“, in *Proceedings of the AMOS 2006 Conference, Maui Hawaii* (2007).

[5] Ritter, G.X. and M. Schmalz. “Learning in lattice neural networks that employ dendritic computing”, *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 7-13 (2006).

[6] Ritter, G.X. and J.N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*, Second Edition, CRC Press (2001).

[7] Ritter, G.X. and P. Gader. “Fixed points of lattice transforms and lattice associative memories”, *Advances in Imaging and Electron Physics* **144**: 165-242 (2006).