*International Journal of Computer and Information Technology (ISSN: 2279 – 0764)*
*Volume 02– Issue 05, September 2013*

# Modified Self Organizing Feature Maps for Classification of Mathematical Curves

Lionel Puengue F
College of Engineering/Chemical
Wayne State University
Detroit, MI 48202 USA

Daniel Liu*
Stephen M. Ross School of Business
University of Michigan
Ann Arbor, MI 48109 USA
Email: *liudani@umich.edu

Mohamed Zohdy
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI 48309

*Abstract*—**In this paper, we applied versions of a two-dimensional Self Organizing Feature Maps (SOFM) to the categorization of mathematical objects in the form of families of curves. We have considered two different categories of curves: functions and relations. The features have been extracted from the joint independent variable-frequency space obtained by transformations of the curves to spectrograms. New contributions have been attempted, such as the extraction of features from the joint independent variable-frequency space as well as modifications to the learning algorithm, namely the saturation of the learning rate. Although this study is significant, extensions to other space objects such as surfaces and spheres will be considered and later on several applications of the SOFM will evolve namely in the financial sector, the chemical field and physical applications.**

***Keywords--*** ***Self Organizing feature Map, Mathematical curves, learning rate, correlation, neighborhood function, winning nodes history, saturation, initialization. Introduction.***

## I. INTRODUCTION

The Self Organizing Feature Map (SOFM) often referred to as Kohonen's Map is a form of competitive, unsupervised, self-organizing computer learning. The SOFM provides a method of visualizing multidimensional data in lower dimensional space; usually in 1 or 2 dimensions. The SOFM's learning algorithm has been used for almost 30 years now and is still used today. There have numerous different applications of SOFM namely on recycling data, GPS data [1], digit recognition data, Animal Communication and sound discrimination including noises data sets [2], etc. Applications and extensions of SOFMs will continue to be a popular topic.

SOFM performance depends primarily on critical learning parameters namely the learning rate, neighborhood function, and weight initialization. The SOFM also uses competition to find best matching nodes to decide the area of the map to be updated. From there, a neighborhood function is utilized to decide which nodes in the selected regions are updated. The learning rate determines the intensity of this process. The nodes autonomously organize themselves thus learning from the input data and storing that discovered knowledge in the map.

In this paper, we propose contributions and by modifying the SOFM algorithm in its applications to the classifications of mathematical space curves, specifically functions and relations. Contributions include the extraction of features from mathematical curves converted to the joint independent variable-frequency space via fast Fourier transformation as well as the saturation of the learning rate.The next section will give a brief overview of the basic SOFM learning algorithm. The third section explains our contributions to the SOFM. The fourth section describes our novel application of the SOFM to math curves. Results and analysis are reported in the fifth section. Finally, the conclusion and future extensions are included in the last section.

## II. SUMMARY OF THE SELF ORGANIZING FEATURE MAP

### *Learning Algorithm*

The Learning Algorithm is the process by which the learning map autonomously organizes itself to effectively represent the inputted data. The goal of learning in the self-organizing map [4] is to cause different parts of the network to respond similarly to certain input patterns. The weights of the nodes are initialized either to small random values or in our case will be adjusted to see the impact on the learning map either using not only random but also K-mean. The training utilizes competitive learning. When a training example is fed to learning map, Its Euclidean distance to all weight vectors is computed [8]. The node with weight vector most similar to the input is called the best-matching node (BMN). The weights of the BMN and its surrounding neighborhood of nodes are adjusted according to the input vector

*Step 1: Initialization: Choose random values for the initial weights W(0)*

*Step 2: Find the winner : Find the best matching node j(k)*

$$j(k) = argmin\|x(k) - w_j\| \quad \textbf{(1)}$$
$$j = 1, \dots, N^2$$

Where $x(k) = [x_1(k), \dots, x_n(k)]$ represents the $k^{th}$ input pattern, and $N^2$ is the total number of inputs, and $\|.\|$ indicates the use of the Euclidean norm.

*Step 3:* Updating weights: Adjust the weights of the winning node and its neighborhood using the following equation:

$$W(k+1) = W(k) + \alpha(k) * \beta(k) * (x(k) - W(k) \quad \textbf{(2)}$$

*W (k)* : Node Weight
$\alpha(k)$ : Learning rate function
$\beta(k)$ : Neighborhood function

The learning rate determines the magnitude of each update based on the number of iterations. The greater the learning rate, the more aggressively the program learns. There exist many possible expressions for the learning rate[1], including constant values, reciprocal functions, logarithmic functions, exponential functions, and double exponential functions. Traditional SOFMs typically use a constant learning rate and in order to optimize our learning algorithm, we will saturate the learning rate and observe its impact.

$$\propto (t) = \alpha_0 \;\; where \; 0 < \alpha_0 \; < 1 \qquad (3)$$
$$\alpha_0 \;\; : initial\ learning\ rate$$

The neighborhood function determines the amount each node is updated based on the distance for the updating node and the BMN.

$$\beta\,(t) = \exp(-d_{BMU}^2\, / 2(\sigma^2(k)) (4)$$

Where $\sigma(t) = \alpha_0 \; \exp\left(-\frac{k}{\gamma}\right)$, (5)

And $\gamma$ is the size of the neighborhood

As $\sigma(t)$ decreases monotonically with the number of iterations, the size of the neighborhood follows.

This process is repeated for each input vector depending for each iteration (t). The learning map associates output nodes with groups or patterns in the input data set. The measurement of similarity or distance is fundamental in the cluster analysis process as most clustering begin with the calculation of distances [14]

## III. PROBLEM SOLUTIONS: CONTRIBUTION TO THE SOFM

### *1. Learning rate saturation*

We study the convergence behavior of the learning map as it is applied to the registration of mathematical space curves. For the purposes of our research, we used a double exponential expression, because it tends to yield the clearest learning map. Using this learning rate expression, the learning rate is initially set to a value 0.3, and decreases through each epoch. The learning equation through which the training occurs is defined by:

$$\propto (t) = \alpha_0 \; \exp\left(-\frac{t}{\lambda}\right) \quad where \;\; 0 < \alpha_0 \; < 1 \qquad (6)$$

$$\alpha_0 \;\; : initial\ learning\ rate$$
$$\alpha(t) is\ the\ final\ learning\ rateat\ iteration$$
$$t\ which\ decreases\ monotonically\ with\ time$$
$$\lambda\ is\ the\ moment\ of\ inertia$$

Unsaturated learning rates approach 0, thus limiting convergence. To improve the learning algorithm, we saturate the learning rate at a percentage of the initial learning rate, allowing the SOFM to continue learning through the duration of iterations. [21]

For these simulations, the learning rate was initialized to α = 0.3; the reason being that if α is too large, the algorithm will learn aggressively and will never find the minimum distance so no pattern will be seen on the converged map. On the other hand, if α is too small the algorithm will learn very slowly due to the fact that each step is only changing its location by a small amount so more epochs will be necessary to achieve a converged map. [21]

### *2. Normalization*

Due to the large variance of values in the spectrogram, the data must be normalized and interpolated. The ranges of the nodes on the initialized map and the input matrix should be similar, so that the nodes on the map can effectively represent the input data. Normalization of the inputs increases the organization of the map as well as the speed of convergence. The effect of normalization is greater with fewer dimensions. Nevertheless, normalization still noticeably improves SOFMs with more dimensions. [20]

### *3. Independent Variable-Frequency Joint Space and Spectrograms:*

Converting mathematical space curves into spectrograms in the x-f joint space provides a more detailed representation of mathematical curves, which allows for more accurate node-matching and differentiation between curves. This is useful considering how similar some families of mathematical curves are in the x-domain. Since data points are defined by functions and relations, noise is not an issue.

The independent variable-frequency joint space is a power density spectrum. We input the absolute value of the power density spectrum as a spectrogram. We used the entire data matrix because of the unique 'fingerprint' representation of the data that seems to provide a clearer more pronounced representation of the curves than the curves themselves (in the x-domain). For the purposes of our research, we consider the spectrogram obtained from the x-f joint space as an image and we extracted data points as a matrix of elements, which turned out to be a 1032 x 8 matrix, due to the format of the spectrogram function in "MATLAB".

## IV. APPLICATION TO MATHEMATICAL CURVES

In order to test our modifications and contributions to the SOFM, we decided to first apply the learning algorithm to classify a variety of mathematical curves. We selected families of functions and relations.

### *A) Functions*

Functions are a set of mathematical operations performed on one or more inputs (variables) that results in an output. We classified different types of functions including parabola(figure 1) and camel hump (figure 2) as spectrograms in the x-f joint space of the designated function and consider it as a picture so we can extract every data point and present it to the learning map as a matrix. (Fig.1)

### *B) Relations*

Relations are commonly defined as a special type of functions. A relation from $X$ to $Y$ is a set of an ordered pair defines a function as a type of relation [3]. We studied relations throughout this project by first transferring them from the polar coordinate into a Theta-R domain where Theta represent the Horizontal axis and R the vertical axis; From there , we extracted features in the x-domain, the frequency domain and the X-F joint space presenting obtained matrices to the learning map for each domain.

## V. SIMULATION RESULTS AND DISCUSSION

### *i) Simulation*

For each individual family of function and relation, we generated 12 unique curves by inputting 12 random values for each constant and scalar. The domain for each curve is $-5<x<5$. The 'y' values were extracted for each interval of 0.1. The power spectral density of the curves was then generated via fast Fourier transformation.

A 10 x 10 network is initialized using random weights. Two versions of the learning program were applied to the spectrogram data in order to analyze the effect of using a saturated learning rate.. Both methods use the same weight updating algorithm (3) and neighborhood function (4)(5). The learning equation (6) was adjusted for each method. Method 1 uses the basic Learning Algorithm with an unsaturated learning rate. Method 2 incorporates a saturated learning rate For each trial we used the following parameters:

*Iterations: $t = 2000$ iterations*
*Initial learning rate: $\alpha_0 = 0.3$*
*Size of neighborhood: $\gamma = 1000/\log(sigN)$*
*Wheresigma0 = N/3 and N is 10*
*Constant used for learning rate: $\lambda = 1000$*

### *ii) Results:*

*Function Results:*
Figure [3] & [4] shows the resultant feature maps constructed for the function curves data set using method 1 and method 2. The maps shown are after 1, 250,1000, and 2000 epochs.

*Relation Results:*
Figure [7]& [8] shows the resultant feature maps constructed for the relation curves data set using method 1 and method 2. The maps shown are after 1, 250,1000, and 2000 epochs.

### *iii) Discussion*

It is difficult to see the SOFM's learning because the output nodes have multiple dimensions. The picture representations do not fully display all of the learning and representations of the data, but to ease visualization of the learning, we manually outline the borders between the nodes of the different families of curves.

*Function Data Results:*

The SOFM has difficulty distinguishing between linear and parabolic curves, because of how similar their spectrograms look. But the fact that the SOFM correctly segregates the camel humps from the linear and parabolic curves shows how the map is learning. Figure [1] shows how both method 1 and 2 produce converged maps, correctly ordering the nodes to represent to the density distribution of the map.

As early as the 10th epoch, we can see the early stages of convergence, as a topologically ordered map begins to appear. By the 250th epoch, we can see that the maps have converged, but more iterationis required to further adjust the map, in order to correctly represent the distribution of the input. By the 1,000th epoch the map is topologically ordered, but further tuning is still needed to fully represent the correct densities. At this point, the SOFM's learning becomes negligible under method 1. However, under method 2, the map continues to fine-tune itself, better representing the data. However if the SOFM is allowed to continue to learn under method 2, we would see that the map eventually over learns and becomes unstable.

*Relations Data Results:*

The Self Organizing Feature Map produces better results when used with relations rather than functions. However, the convergence method follows the same path as functions. Relations appear to have unique and very distinctive features from each others. Three types of relations have been study in this paper namely the "Lemiscate of Bernouli", the " tricuspoid" and the"hurricane" like function. Twelve data points have been chosen for each relation (More or less data points can be chosen), and after been normalized and randomized, the matrix of (1032 * 36) is fed to the learning algorithm and the results are illustrated in figures (7) . The same data matrix is then fed to a learning program but at a saturated learning rate for this second trial. The results obtained are shown in figure (8). The close comparison of the saturated learning rate versus unsaturated learning rate maps at each outputted epoch shows that the clustering area of relations are interchanged. An in depth analysis also shows that the convergence is attained faster when the learning rate is saturated.

## VI. CONCLUSION

In this research, we have been able to modify the "SOFM" algorithm and effectively applied it to the learning of a variety of important mathematical curves. We can make the following conclusions:

1. The SOFM can distinguish both functions and relations from features in the frequency domain and x-f joint space. The SOFM does a better job of learning relations.
2. Normalizing data produces better results.
3. The SOFM shows a better learning map when the learning rate is saturated at suitable value and the number of iterations is appropriate.

Our research helps improve the efficiency and accuracy of SOFM. However, there are still limitations that must be examined in the future; such as increasing the number of iterations of the learning maps with adjusted parameters, increasing the number of nodes, and possible saturation of the neighborhood function as well.

Future work will be done to further modify and improve the SOFM's application to mathematical space curves. One current idea is to add a best matching node history function that would improve the effectiveness of the neighborhood function. Other ideas include more effective initialization methods, such as k-means.

Extensions to other space objects such as surfaces, spheres and families of elliptic curves will be considered. Further, different applications in the financial sector, chemistry/chemical engineering field as well as physical applications of the SOFM will evolve.

ACKNOWLEDGEMENT

First and foremost, we would like to thank the "UnCoRe" program at Oakland University. We would like to thank Professor Mohamed Zohdy for his guidance and mentorship; without him, none of this would have been possible. Finally, we would like to thank our colleague Jonathan Poczatek for his assistance with Matlab.

This research was funded by the NSF under the grant number CNS – 1062960 and conducted at Oakland University in the UnCoRe program – REU. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the National Science Foundation.

<u>Functions</u>

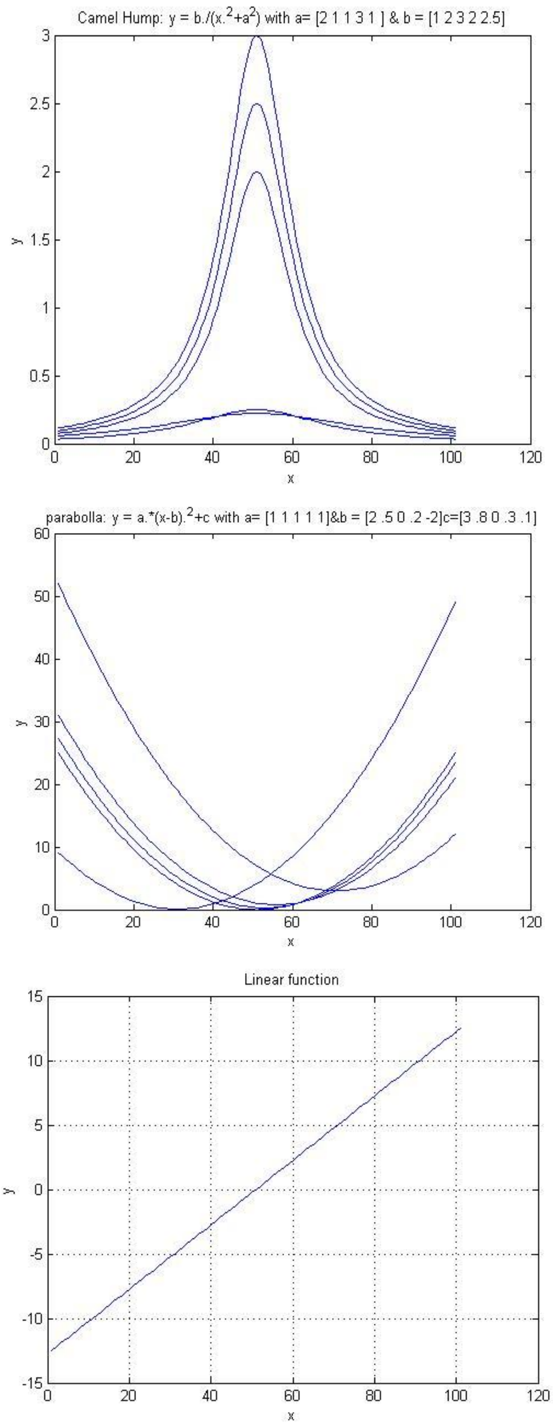



**Figure 1: List of functions**

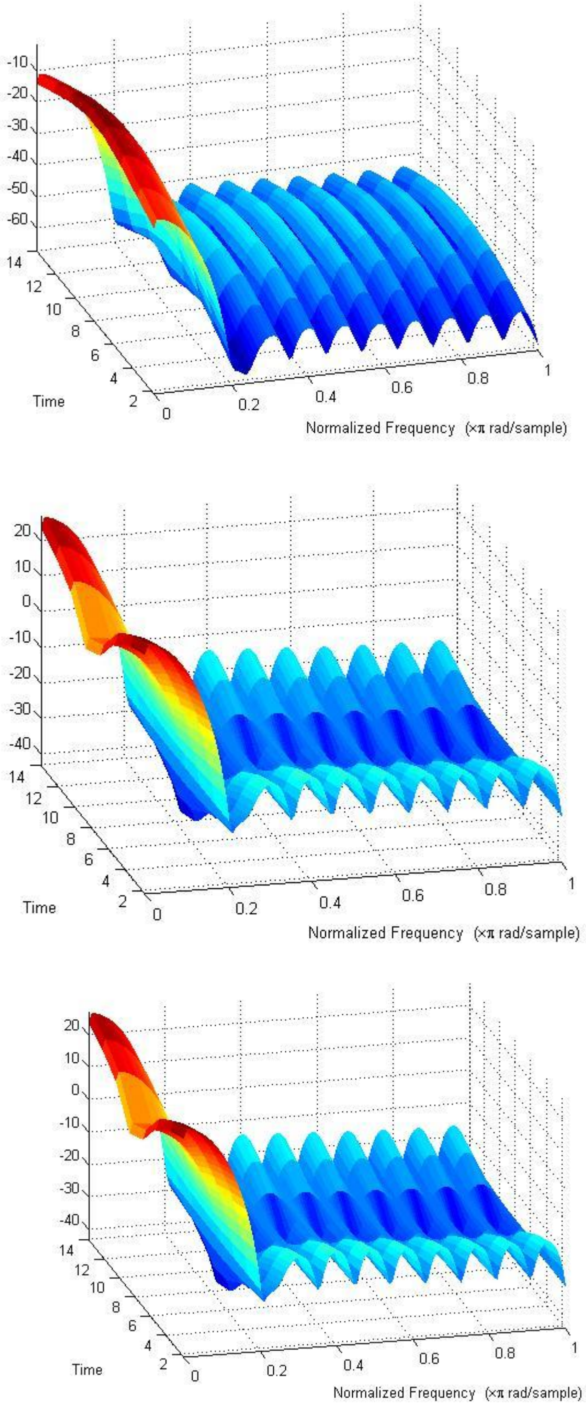



**Figure 2: spectrogram pertaining to each function**

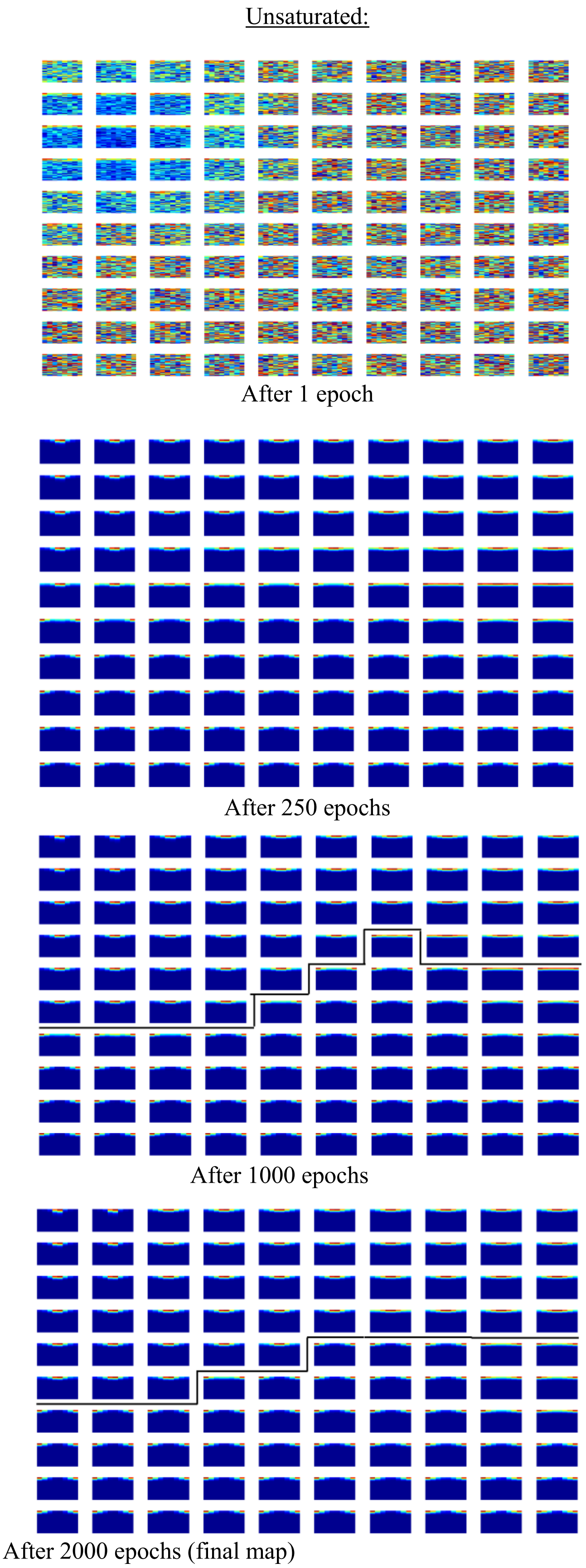



**Figure 3: Unsaturated maps for functions**

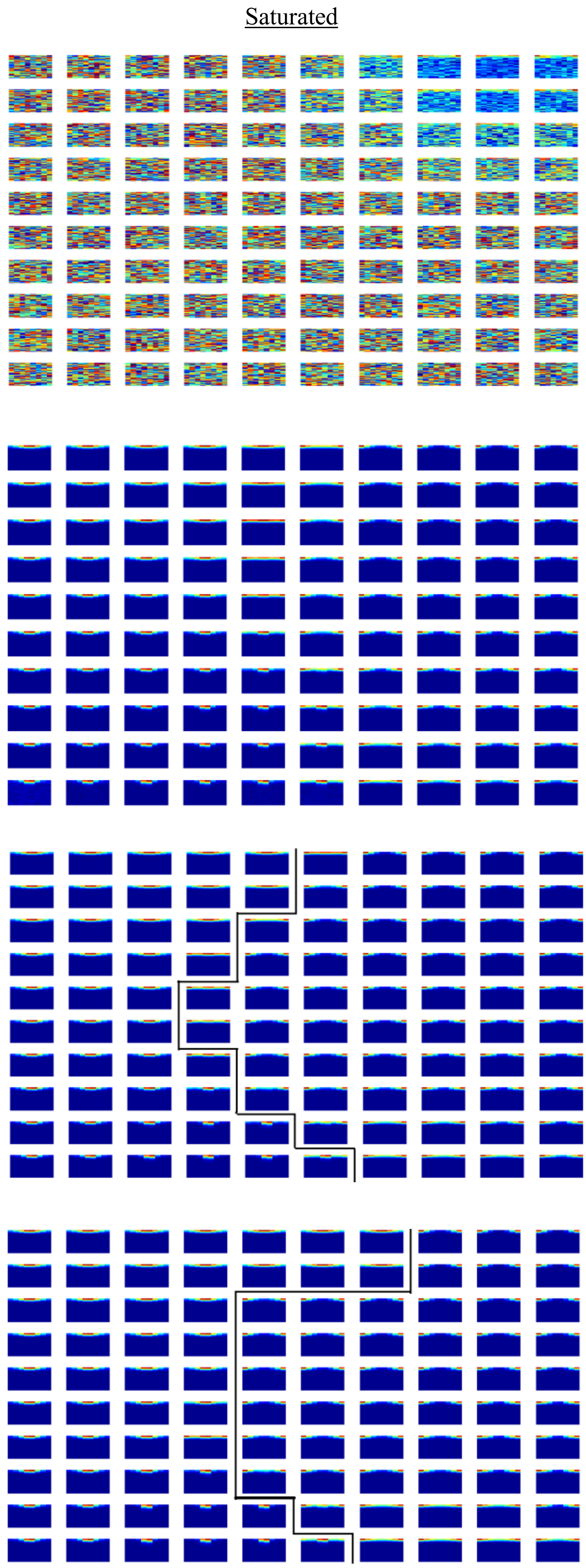



**Figure 4: Saturated Maps for functions**

<u>Relations</u>

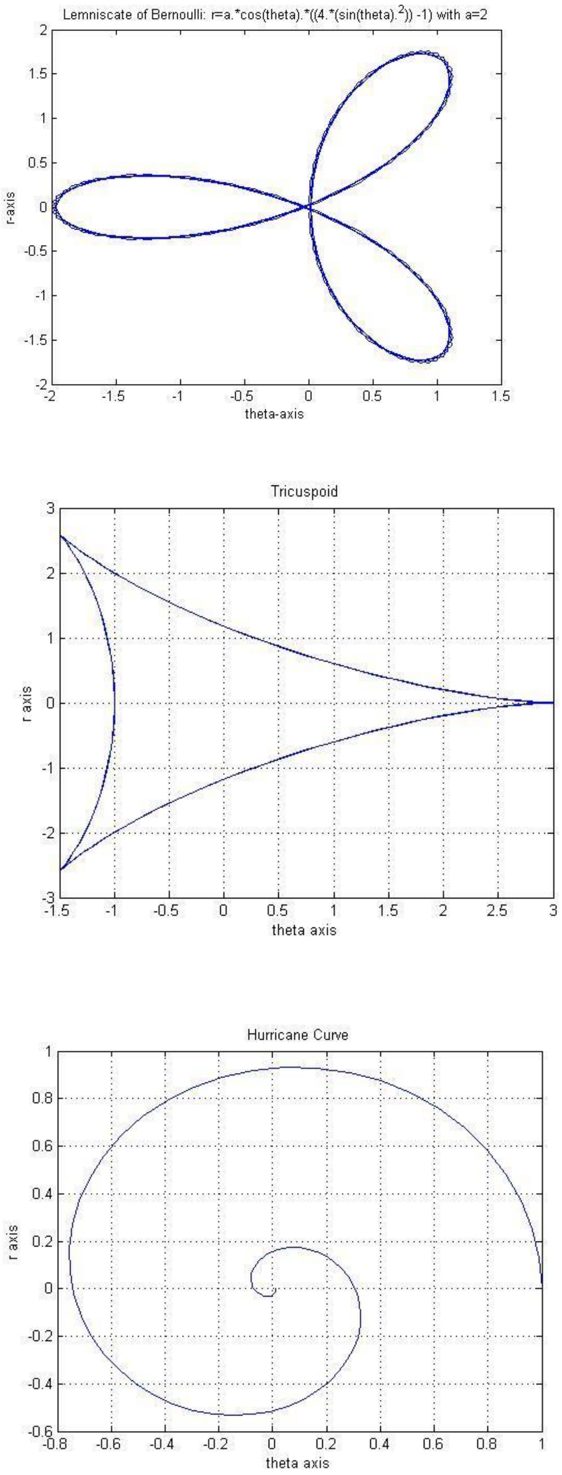



**Figure 6: List of relations**

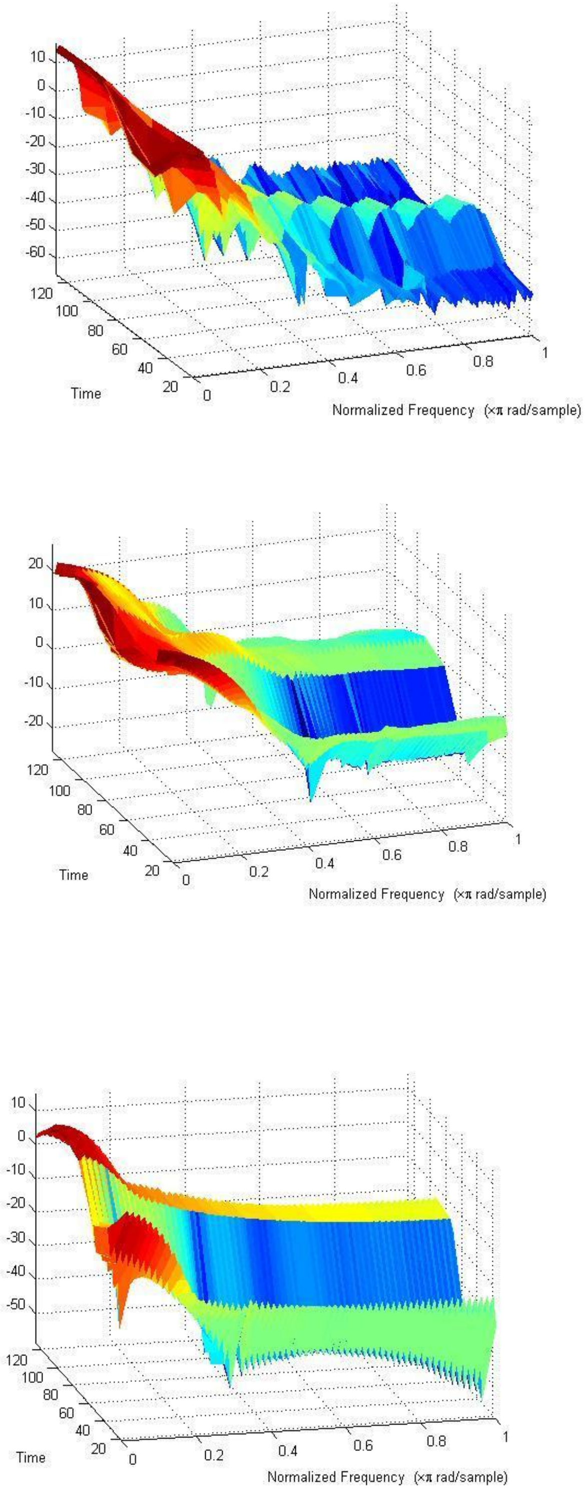
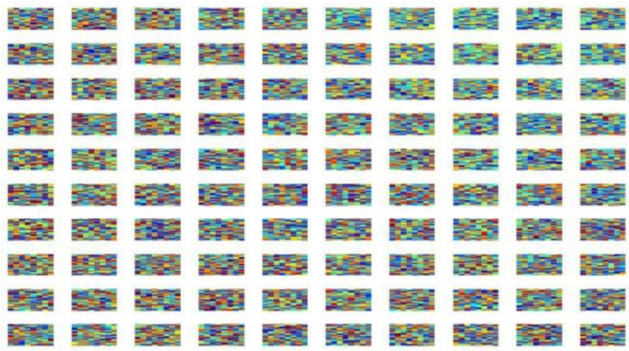
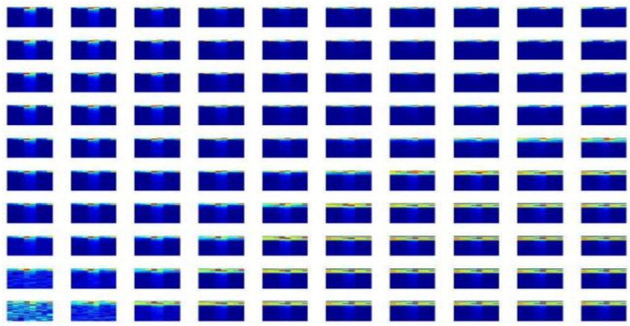


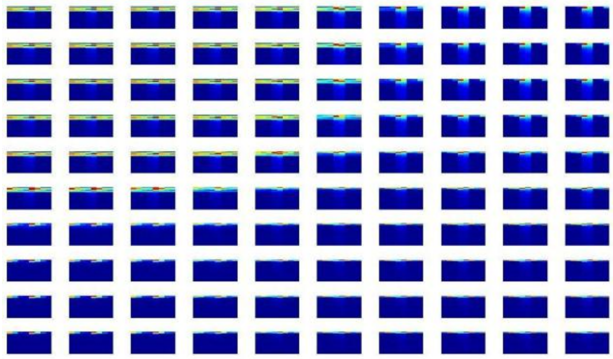

**Figure 5: Spectrograms pertaining to each relation**

Results:

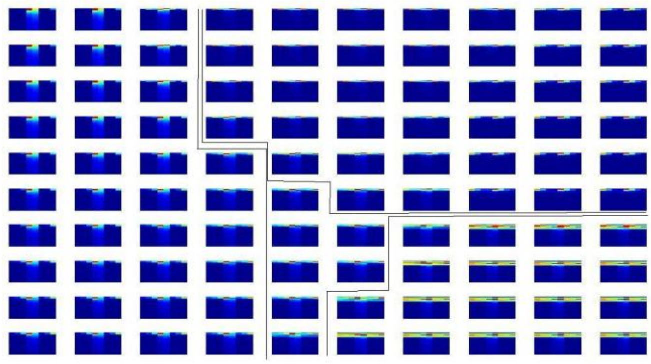


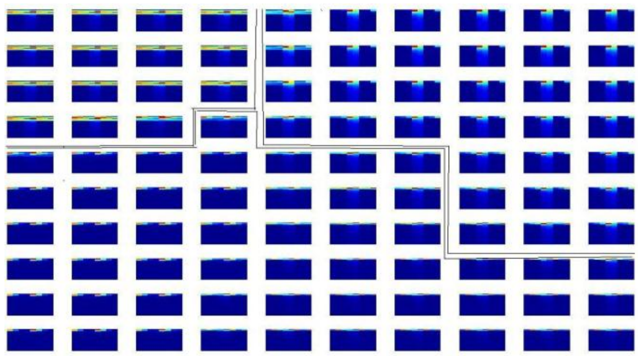
After 1 epoch



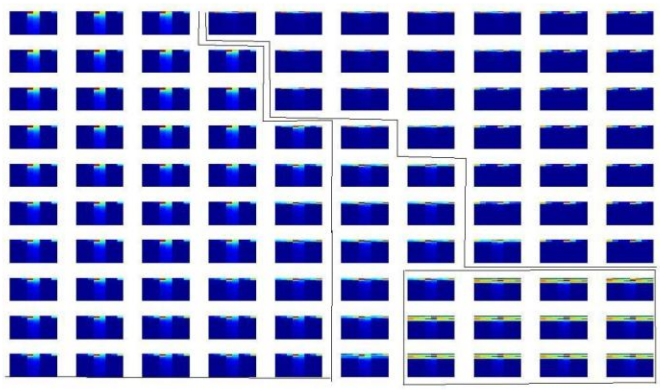


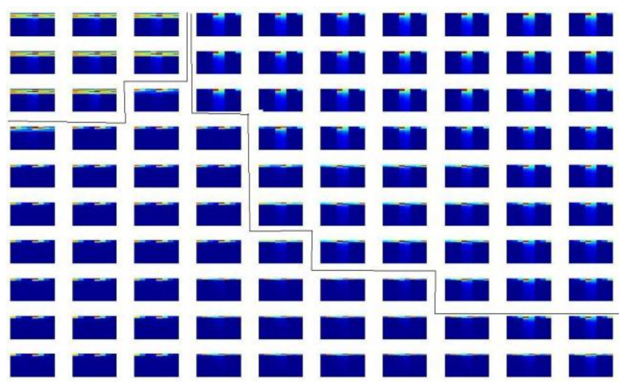
After 250 epochs





After 1000 epochs





After 2000 epochs

**Figure 7: Relations maps unsaturated**

**Figure 8: Relations maps saturated**

REFERENCES

[1] Ahmad R. Nsour, Pr Mohamed A Zohdy "Self Organized Learning Applied to Global Positioning System(GPS)Data".. Lisbon, Portugal September 22-24, 2006

[2] Matthew Bradley, Kay Jantharasorn, Keith Jones"Application of Self Organized Neural Net to Animal Communications".. Oakland University REU-UnCoRe 2010

[3] David H. Von Seggern., "CRC Standard Curves and Surfaces" Dec 15, 1992 CRC Press.

[4] H.S. Abdel-Aty-Zohdy, M. & A. Zohdy "Self Organizing Feature Maps". The Wiley Encyclopedia for Electrical Engineering,. Dec 27, 1997. pp.767-772

[5] Kohonen, T.; , "Things you haven't heard about the self-organizing map,", *IEEE International Conference onNeural Networks*, pp.1147-1156 vol.3, 1993

[6] P.V.S. Balakrishnan, M.C. Cooper, V.S. Jacob, P.A. Lewis "A study of classification capabilities of neural networks using unsupervised learning: a comparison with *K*-means clustering " Psychometrika, 59 (4) (1994), pp. 509–525

[7] Kohonen T. A simple paradigm for the self-organized formation of structured feature maps. In: Amari S, Berlin M. editors. Competition and cooperation in neural nets. Lecture Notes in Biomathematics. Berlin: Springer, 1982.

[8] L. Han, "Initial weight selection methods for self-organizing training", *Proc. IEEE Int. Conf. Intelligent Processing Systems*, pp.404 -406 1997

[9] H. J. Ritter, T. Martinetz, and K. J. Schulten, *Neural Computation and Self-Organizing Maps: An Introduction*, 1992 :Addison-Wesley

[10] M. Y. Kiang, U. R. Kulkarni, M. Goul, A. Philippakis, R. T. Chi, and E. Turban, "Improving the effectiveness of self-organizing map networks using a circular Kohonen layer", *Proc. 30th. Hawaii Int. Conf. System Sciences*, pp.521 -529 1997

[11] T. Kohonen, "The self-organizing map", *Proc. IEEE*, vol. 78, no. 9, pp.1464 -1480 1990

[12] H. Shah-Hosseini and R. Safabakhsh, "TASOM: a new time adaptive self-organizing map", *IEEE Trans. Syst., Man, Cybern. B*, vol. 33, no. 2, pp.271 -282 2003

[13] J. A. Starzyk, Z. Zhu, and T.-H. Liu, "Self-organizing learning array", *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp.355 -363 2005

[14] N. Keerati Pranon and F. Maire, "Bearing similarity measures for self-organizing feature maps", *Proc. IDEAL*, pp.286 -293 2005

[15] R. Iglesias and S. Barro, "SOAN: self organizing with adaptive neighborhood neural network", *Proc. IWANN*, pp.591 -600 1999

[16] Laaksonen, J. and Honkela, T. (eds.) (2011). "Advances in Self-Organizing Maps, WSOM 2011", Springer, Berlin.

[17] Koblitz, Neal (1993). *Introduction to Elliptic Curves and Modular Forms*. Graduate Texts in Mathematics. **97** (2nd ed.). Springer-Verlag.

[18] Mu-Chun Su, Ta-Kang Liu and Hsiao-Te Chang. "Improving the Self-Organizing Feature Map Algorithm Using an Efficient Initialization Scheme" .Tamkang Journal of Science and Engineering, Vol. 5, No. 1, pp. 35-48 (2002)

[19] Abdel-Badeeh M. Salem, Mostafa M. Syiam, and Ayad F. Ayad. "Improving Self-Organizing Feature Map (SOFM) Training Algorithm Using K-Means Initialization".

[20] Blayo, Francois. "Kohonen Self-Organizing Maps: Is the Normalization Necessary?.

[21] Béla G. Lipták (2003). *Instrument Engineers' Handbook: Process control and optimization* (4 ed.). CRC Press. p. 100