

Brief Review of Self-Organizing Maps

Dubravko Miljković

Hrvatska elektroprivreda, Zagreb, Croatia
dubravko.miljkovic@hep.hr

Abstract - As a particular type of artificial neural networks, self-organizing maps (SOMs) are trained using an unsupervised, competitive learning to produce a low-dimensional, discretized representation of the input space of the training samples, called a feature map. Such a map retains principle features of the input data. Self-organizing maps are known for its clustering, visualization and classification capabilities. In this brief review paper basic tenets, including motivation, architecture, math description and applications are reviewed.

I. INTRODUCTION

Among numerous neural network architectures, particularly interesting architecture was introduced by Finish Professor Teuvo Kohonen in the 1980s, [1,2]. Self-organizing map (SOM), sometimes also called a Kohonen map use unsupervised, competitive learning to produce low dimensional, discretized representation of presented high dimensional data, while simultaneously preserving similarity relations between the presented data items. Such low dimensional representation is called a feature map, hence map in the name. This brief review paper attempts to introduce a reader to SOMs, covering in short basic tenets, underlying biological motivation, its architecture, math description and various applications, [3-10].

II. NEURAL NETWORKS

Human and animal brains are highly complex, nonlinear and parallel systems, consisting of billions of neurons integrated into numerous neural networks, [3]. A neural networks within a brain are massively parallel distributed processing system suitable for storing knowledge in forms of past experiences and making it available for future use. They are particularly suitable for the class of problems where it is difficult to propose an analytical solution convenient for algorithmic implementation.

A. Biological Motivation

After millions of years of evolution, brain in animals and humans has evolved into the massive parallel stack of computing power capable of dealing with the tremendous varieties of situations that it can run into. The biological neural networks are natural intelligent information processors. Artificial neural networks (ANN) constitute computing paradigm motivated by the neural structure of biological systems, [6]. ANNs employ a computational approach based on a large collection of artificial neurons that are much simplified representation of biological neurons. Synapses that ensure communication among

biological neurons are replaced with neuron input weights. Adjustment of connection weights is performed by some of numerous learning algorithms. ANNs have very simple principles, but their behavior can be very complex. They have a capability to learn, generalize, associate data and are fault tolerant. The history of the ANNs begins in the 1940s, but the first significant step came in 1957 with the introduction of Rosenblatt's perceptron. The evolution of the most popular ANN paradigms is shown in Fig. 1, [10].

B. Basic Architectures

An artificial neural network is an interconnected assembly of simple processing elements, called artificial neurons (also called units or nodes), whose functionality mimics that of a biological neuron, [4]. Individual neurons can be combined into layers, and there are single and multi-layer networks, with or without feedback. The most common types of ANNs are shown in Fig. 2, [11]. Among training algorithms the most popular is backpropagation and its variants. ANNs can be used for solving a wide variety of problems. Before the use they have to be trained. During the training, network adjusts its weights. In supervised training, input/output pairs are presented to the network by an external teacher and network tries to learn desired input output mapping. Some neural architectures (like SOM) can learn without supervision (unsupervised) from the training data without specified input/output pairs.

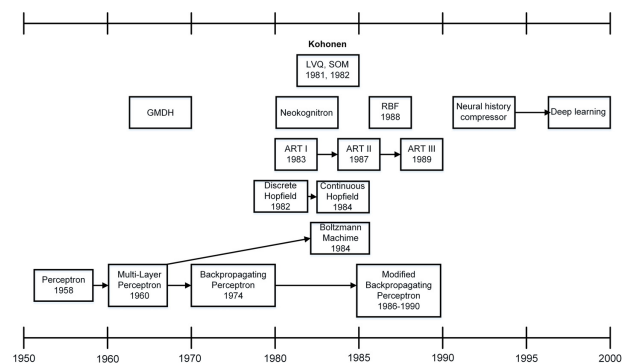


Figure 1. Evolution of artificial neural network paradigms, based on [10]

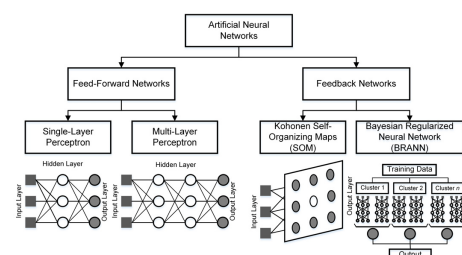


Figure 2. Most common artificial neural networks, according to [11]

III. SELF-ORGANIZING MAPS

Self-organized map (SOM), as a particular neural network paradigm has found its inspiration in self-organizing and biological systems.

A. Self-Organized Systems

Self-organizing systems are types of systems that can change their internal structure and function in response to external circumstances and stimuli, [12-15]. Elements of such a system can influence or organize other elements within the same system, resulting in a greater stability of structure or function of the whole against external fluctuations, [12]. The main aspects of self-organizing systems are increase of complexity, emergence of new phenomena (the whole is more than the sum of its parts) and internal regulation by positive and negative feedback loops. In 1952 Turing published a paper regarding the mathematical theory of pattern formation in biology, and found that global order in a system can arise from local interactions, [13]. This often produces a system with new, emergent properties that differ qualitatively from those of components without interactions, [16]. Self-organizing systems exist in nature, including non-living as well as living world, they exist in man-made systems, but also in the world of abstract ideas, [12].

B. Self-Organizing Map

Neural networks of neurons with lateral communication of neurons topologically organized as self-organizing maps are common in neurobiology. Various neural functions are mapped onto identifiable regions of the brain, Fig. 3, [17]. In such topographic maps neighborhood relation is preserved. Brain mostly does not have desired input-output pairs available and has to learn in unsupervised mode.

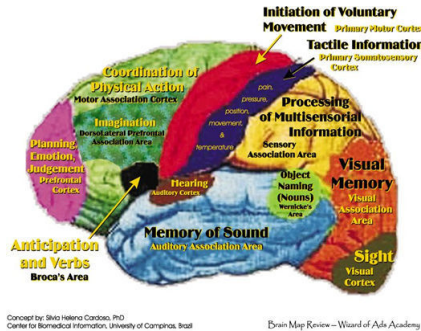


Figure 3. Maps in brain, [17]

A SOM is a single layer neural network with units set along an n -dimensional grid. Most applications use two-dimensional and rectangular grid, although many applications also use hexagonal grids, and some one, three, or more dimensional spaces. SOMs produce low-dimensional projection images of high-dimensional data distributions, in which the similarity relations between the data items are preserved, [18].

C. Principles of Self-Organization in SOMs

Following three processes are common to self-organization in SOMs, [7,19,20]:

1. Competitive Process

For each input pattern vector presented to the map, all neurons calculate values of a discriminant function. The neuron that is most similar to the input pattern vector is the winner (best matching unit, BMU).

2. Cooperative Process

The winner neuron (BMU) finds the spatial location of a topological neighborhood of excited neurons. Neurons from this neighborhood may then cooperate.

3. Synaptic Adaptation

Provides that excited neurons can modify their values of the discriminant function related to the presented input pattern vector by the process of weight adjustments.

D. Common Topologies

SOM topologies can be in one, two (most common) or even three dimensions, [2-10]. Two most used two dimensional grids in SOMs are rectangular and hexagonal grid. Three dimensional topologies can be in form of a cylinder or toroid shapes. 1-D (linear) and 2-D grids are illustrated in Fig. 4, with corresponding SOMs in Fig. 5 and Fig. 6, according to [19].

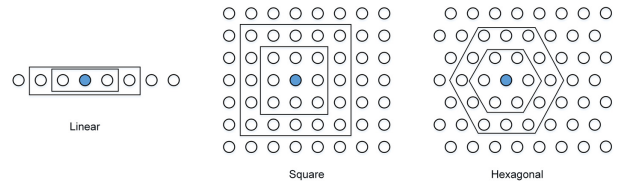


Figure 4. Most common grids and neuron neighborhoods

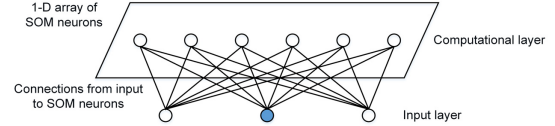


Figure 5. 1-D SOM network, according to [19].

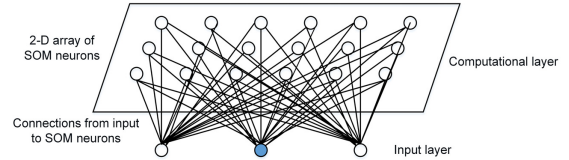


Figure 6. 2-D SOM network, according to [19].

IV. LEARNING ALGORITHM

In 1982 Professor Kohonen presented his SOM algorithm, [1]. Further advancement in a field came with the Second edition of his book "Self-Organization and Associative Memory" in 1988, [2].

A. Measures of Distance and Similarity

To determined similarity between the input vector and neurons measures of distance are used. Some popular distances among input pattern and SOM units are, [21]:

- Euclidian
- Correlation
- Direction cosine
- Block distance

In a real application most often squared Euclidean distance is used, (1):

$$dj = \sum_i (x_i - w_{ij})^2 \quad (1)$$

B. Neighborhood Functions

Neurons within a grid interact among themselves using a neighbor function. Neighborhood functions most often assume the form of the Mexican hat, (2), Fig. 7, that has biological motivation behind (rejects some neurons in the vicinity to the winning neuron) although other functions (Gaussian, cone and cylinder) are also possible, [22]. Ordering algorithm is robust to the choice of function type if the neighbor radius and learning rate decrease to zero. The popular choice is the exponential decay.

$$h_g(w_{ij}, w_{mn}, r) = e^{-\frac{1}{2} \left(\frac{\sqrt{(i-n)^2 + (j-m)^2}}{r} \right)^2} \quad (2)$$

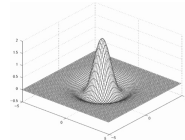


Figure 7. Mexican hat function

C. Initialization of Self-Organizing Maps

Before training SOM, units (i.e. its weights) should be initialized. Common approaches are, [2,23]:

1. Use of random values, completely independent of the training data set
2. Use of random samples from the input training data
3. Initialization that tries to reflect the distribution of the data (Principal Components)

D. Training

Self-organizing maps use the most popular algorithm of the unsupervised learning category, [2]. The criterion D , that is minimized, is the sum of distances between all input vectors \mathbf{x}_n and their respective winning neuron weights \mathbf{w}_i calculated at the end of each epoch, (3), [21]:

$$D = \sum_{i=1}^k \sum_{n \in c_i} (\mathbf{x}_n - \mathbf{w}_i)^2 \quad (3)$$

Training of self-organizing maps, [2,18], can be accomplished in two ways: as sequential or batch training.

1. Sequential training
 - single vector at a time is presented to the map
 - adjustment of neuron weights is made after a presentation of each vector
 - suitable for on-line learning
2. Batch training
 - whole dataset is before any adjustment to the neuron weights is made
 - suitable for off-line learning

Here are steps for the sequential training, [3,7,19,22]:

1. Initialization
 - Initialize the neuron weights (iteration step $n=0$)
2. Sampling
 - Randomly sample a vector $\mathbf{x}(n)$ from the dataset
3. Similarity Matching
 - Find the best matching unit (BMU), c , with weights $\mathbf{w}_{bmu} = \mathbf{w}_c$, (4):

$$c = \arg \min_i (\|\mathbf{x}(n) - \mathbf{w}_i(n)\|) \quad (4)$$

4. Updating

- Update each unit i with the following rule:

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \alpha(n) h(\mathbf{w}_{bmu}(n), \mathbf{w}_i(n), r(n)) (\mathbf{x}(n) - \mathbf{w}_i(n)) \quad (5)$$

5. Continuation

- Increment n . Repeat steps 2-4 until a stopping criterion is met (e.g. the fixed number of iterations or the map has reached a stable state).

For the convergence and stability to be guaranteed, the learning rate $\alpha(n)$ and neighborhood radius $r(n)$ are decreasing with each iteration towards the zero, [22].

SOM Sample Hits, Fig. 8, show the number of input vectors that each unit in SOM classifies, [24].

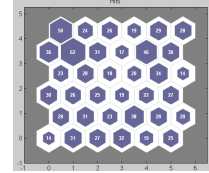


Figure 8. SOM Sample Hits, [24]

During the training process two phases may be distinguished, [7,18]:

1. Self-organizing (ordering) phase:
Topological ordering in the map takes place (roughly first 1000 iterations). The learning rate $\alpha(n)$ and neighborhood radius $r(n)$ are decreasing.
2. Convergence (fine tuning) phase:
This is fine tuning that provides an accurate statistical representation of the input space. It typically lasts at least (500 x number of neuron) iterations. The smaller learning rate $\alpha(n)$ and neighborhood radius $r(n)$ may be kept fixed (e.g. last values from the previous phase).

After the training of the SOM is completed, neurons may be labeled if labeled pattern vectors are available.

E. Classification

Find the best matching unit (BMU), c , (5):

$$c = \arg \min_i (\|\mathbf{x} - \mathbf{w}_i\|) \quad (5)$$

Test pattern \mathbf{x} belongs to the class represented by the best matching unit c .

V. PROPERTIES OF SOM

After the convergence of SOM algorithm, resulting feature map displays important statistical characteristics of the input space. They are also able to discover relevant patterns or features present in the input data.

A. Important Properties of SOMs

SOMs have four important properties, [3,7]:

1. Approximation of the Input Space
The resulting mapping provides a good approximation to the input space. SOM also performs dimensionality reduction by mapping multidimensional data on SOM grid.
2. Topological Ordering
Spatial locations of the neurons in the SOM lattice are topologically related to the features of the input space.
3. Density Matching
The density of the output neurons in the map approximates the statistical distribution of the input space. Regions of the input space that contain more training vectors are represented with more output neurons.

4. Feature Selection

Map extracts the principal features of the input space. It is capable of selecting the best features for approximation of the underlying statistical distribution of the input space.

B. Representing the Input Space with SOMs of Various Topologies

1. 1-D

2D input data points are uniformly distributed in a triangle, 1D SOM ordering process shown in Fig. 9.

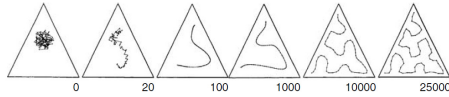


Figure 9. 2D to 1D mapping by a SOM (ordering process), [2]

2. 2-D

2D input data points are uniformly distributed in a square, 2D SOM ordering process shown in Fig. 10.

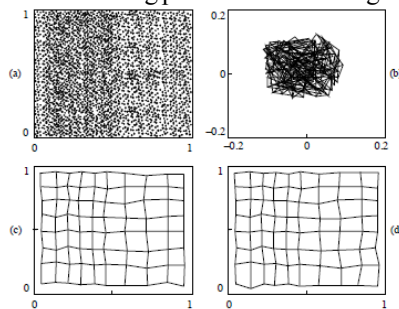


Figure 10. 2D to 2D mapping by a SOM (ordering process), [3]

3. Torus SOMs

In conventional SOM, the size of neighborhood set is not always constant because the map has its edges. This problem can be mitigated by use of torus SOM that has no edges, [25]. However torus SOM, Fig. 11, is not easy to visualize as there are now missing edges.

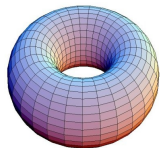


Figure 11. Torus SOM

4. Hierarchical SOMs

After previous topologies, hierarchical SOMs should also be mentioned. Hierarchical neural networks are composed of multiple neural networks concentered in a form of an acyclic graph. The output of one SOM can be used as the input for another SOM, Fig. 12, [10]. Such input can also be formed of several output vectors coming from Best Matching Units (BMU) of many SOMs.

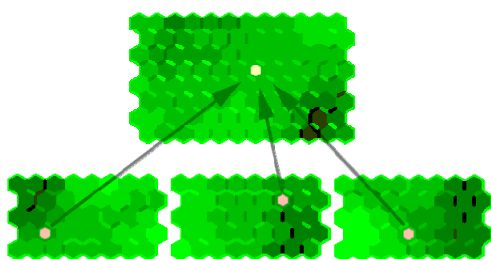


Figure 12. Hierarchical SOM. The topmost SOM takes input from the best matching units of the three lower level SOMs, [10]

VI. APPLICATIONS

Despite its simplicity, SOMs can be used for a various classes of applications, [2,26,27]. This in a broad sense includes visualizations, generation of feature maps, pattern recognition and classification. Kohonen in [2] came with the following categories of applications: machine vision and image analysis, optical character recognition and script reading, speech analysis and recognition, acoustic and musical studies, signal processing and radar measurements, telecommunications, industrial and other real world measurements, process control, robotics, chemistry, physics, design of electronic circuits, medical applications without image processing, data processing linguistic and AI problems, mathematical problems and neurophysiological research. With such an exhaustive list provided here, as space permits, it is possible only to mention some of them that are interesting and popular.

A. Speech Recognition

The neural phonetic typewriter for Finnish and Japanese speech was developed by Kohonen in 1988, [28]. The signal from the microphone proceeds to acoustic preprocessing, shown in more detail in Fig. 13, forming 15-component pattern vector (values in 15 frequency beans taken every 10 ms), containing a short time spectral description of speech. These vectors are presented to a SOM with the hexagonal lattice of the size 8 x 12.



Figure 13. Acoustic Preprocessing

After training resulting phonotopic map is shown in Fig. 14, [7]. During speech recognition new pattern vectors are assigned category belonging to a closest prototype in the map.

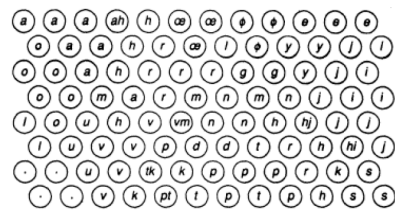


Figure 14. Phonotopic map, [7]

B. Text Clustering

Text clustering is the technology of processing a large number of texts and gives their partition. Preparation of text for SOM analysis is shown in Fig. 15, [29], and

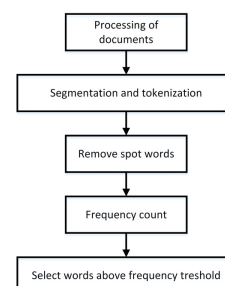
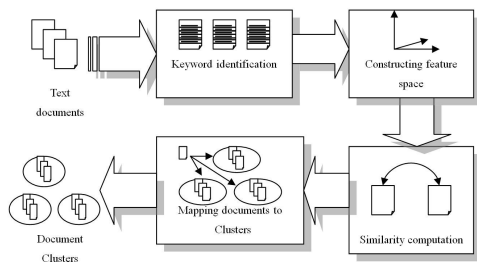


Figure 15. Preparation of text for SOM analysis, according to [29]

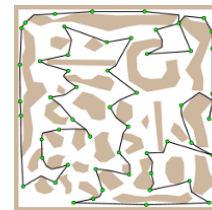


complete framework in Fig. 16., [29]. Massive document collections can be organized using a SOM. It can be optimized to map large document collections while preserving much of the classification accuracy. Clustering of scientific articles is illustrated in Fig. 17, [30].

C. Application in Chemistry

D. Medical Imaging and Analysis

E. Maritime Applications

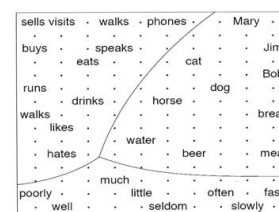


F. Robotics

Some applications of SOMs are control of robot arm, learning the motion map and solving traveling salesman problem (multi-goal path planning problem), Fig. 20, [32].

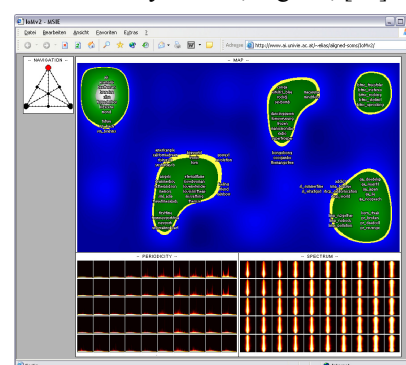
G. Classification of Satellite Images

One example is the categorization of words by their local context in three word sentences of the type subject-predicate-object or subject-predicate-predicative that were constructed artificially. The words become clustered by SOM according to their linguistic roles in an orderly fashion, Fig. 22, [18].



I. Exploring Music Collections

Similarity of music recordings may be determined by analyzing the lyrics, instrumentation, melody, rhythm, artists, or emotions they invoke, Fig. 23, [34].



J. Business Applications

Customer segmentation of the international tourist market is illustrated in Fig. 24, [35]. Another example is classifying world poverty (welfare map), [36]. Ordering of items with the respect to 39 features describing various quality-of-life factors, such as state of health, nutrition, and educational services is shown in Fig. 25. Countries with similar quality of life factors clustered together on a map.

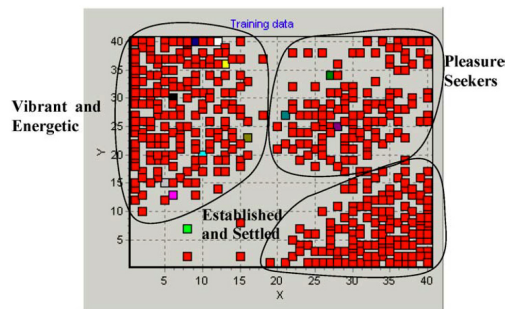


Figure 24. Customer segmentation of the international tourist market, [35]

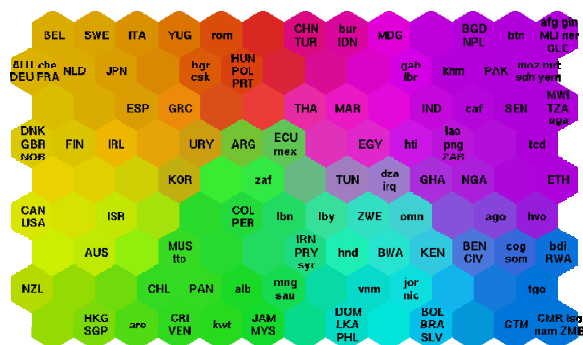


Figure 25. Poverty map based on 39 indicators from World Bank statistics (1992), [36]

VII. CONCLUSION

Self-organizing maps (SOMs) are neural network architecture inspired by the biological structure of human and animal brains. They become one of most popular neural network architecture. SOMs learn without external teacher, i.e. employ unsupervised learning. Topologically SOMs most often use a two-dimensional grid, although one-dimensional, higher-dimensional and irregular grids are also possible. SOM maps higher dimensional input onto the lower dimensional grid while preserving topological ordering present in the input space. During competitive learning SOM use lateral interactions among the neurons to form a semantic map where similar patterns are mapped closer together than dissimilar ones. They can be used for broad type of applications like visualizations, generation of feature maps, pattern recognition and classification. Humans can't visualize high-dimensional data, hence SOMs by mapping such data to two-dimensional grid are widely used for data visualization. SOMs are also suitable for generation of feature maps. Because they can detect clusters of similar patterns without supervision SOMs are a powerful tool for identification and classification of spatio-temporal patterns. SOMs can be used as an analytical tool, but also in myriad of real world applications including science, medicine, satellite imaging and industry.

REFERENCES

- [1] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biol. Cybern.* 43, pp. 59-69, 1982
- [2] T. Kohonen, *Self-Organizing Maps*, 2nd ed., Springer 1997
- [3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall PTR Upper Saddle River, NJ, USA, 1998
- [4] K. Gurney, *An introduction to neural network*, UCL Press Limited, London, UK, 1997
- [5] D. Kriesel, *A Brief Introduction to Neural Networks*, <http://www.dkriesel.com>
- [6] R. Rojas: *Neural Networks, A Systematic Introduction*, Springer-Verlag, Berlin, 1996
- [7] J. A. Bullinaria, *Introduction to Neural Networks - Course Material and Useful Links*, <http://www.cs.bham.ac.uk/~jxb/NN/>
- [8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1997
- [9] R. Eckmiller, C. Malsburg, *Neural Computers*, NATO ASI Series, Computer and Systems Sciences, 1988
- [10] P. Hodju and J. Halme, *Neural Networks Information Homepage*, <http://phodju.mbnet.fi/nenet/SiteMap/SiteMap.html>
- [11] Káthia Maria Honório and A. B. F. da Silva, "Applications of artificial neural networks in chemical problems", in *Artificial Neural Networks - Architectures and Applications*, InTech, 2013
- [12] W. Banzhaf, "Self-organizing systems", in *Encyclopedia of Complexity and Systems Science*, 2009, Springer, Heidelberg.
- [13] A. M. Turing, "The chemical basis of morphogenesis", *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, Vol. 237, No. 641, pp. 37-72, Aug. 14, 1952
- [14] W. R. Ashby, "Principles of the self-organizing system", *E:CO Special Double Issue Vol. 6, No. 1-2*, pp. 102-126, 2004
- [15] C. Fuchs, "Self-organizing system", in *Encyclopedia of Governance*, Vol. 2, SAGE Publications, 2006, pp. 863-864
- [16] J. Howard, 2010+ | *Research Perspectives of the Max Planck Society*, pp. 28-29
- [17] The Wizard of Ads Brain Map - Wernicke and Broca, <https://www.wizardofads.com.au/brain-map-brocas-area/>
- [18] T. Kohonen, *MATLAB Implementations and Applications of the Self-Organizing Map*, Unigrafia, Helsinki, Finland, 2014
- [19] Bill Wilson, *Self-organisation Notes*, 2010, www.cse.unsw.edu.au/~billw/cs9444/selforganising-10-4up.pdf
- [20] J. Boedecker, *Self-Organizing Map (SOM)*, ppt, Machine Learning, Summer 2015, Machine Learning Lab, Univ. of Freiburg
- [21] L. Grajciarova, J. Mares, P. Dvorak and A. Prochazka, *Biomedical image analysis using self-organizing maps*, *Matlab Conference 2012*
- [22] V. J. A. S. Lobo, "Application of Self-Organizing Maps to the Maritime Environment", *Proceedings of IF&GIS 2009*, 20 May 2009, St. Petersburg, Russia
- [23] A. A. Akinduko and E. M. Mirkes, "Initialization of self-organizing maps: principal components versus random initialization. A case study", *Information Sciences*, Vol. 364, Is. C, pp. 213-221, Oct. 2016
- [24] MathWorks, *Self-Organizing Maps*, <https://www.mathworks.com/help/nnet/ug/cluster-with-self-organizing-map-neural-network.html>
- [25] M. Ito, T. Miyoshi, and H. Masuyama, "The characteristics of the torus self organizing map", *The 6th Int. Conf. on Soft Computing (IIZUKA'2000)*, Iizuka, Fukuoka, Japan, Oct. 1-4, 2000, pp. 239-44
- [26] M. Johnsson ed., *Applications of Self-Organizing Maps*, InTech, November 21, 2012
- [27] J. I. Mwasiagi (ed.), *Self Organizing Maps - Applications and Novel Algorithm Design*, InTech, 2011
- [28] T. Kohonen, "The 'neural' phonetic typewriter", *IEEE Computer* 21(3), pp. 11-22, 1988
- [29] Yuan-Chao Liu, Ming Liu and Xiao-Long Wang, "Application of self-organizing maps in text clustering: a review", in *"Self Organizing Maps - Applications and Novel Algorithm Design"*, InTech, 2012
- [30] K. W. Boyack et al., *Supplementary information on data and methods for "Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches"*, *PLoS ONE* 6(3): e18029, 2011
- [31] A. Aslantas, D. Emre and M. Çakiroğlu, "Comparison of segmentation algorithms for detection of hotspots in bone scintigraphy images and effects on CAD systems", *Biomedical Research*, 28 (2), pp. 676-683, 2017
- [32] J. Faigl, "Multi-goal path planning for cooperative sensing", PhD Thesis, Czech Technical University of Prague, February 2010
- [33] D. Lairy, *Machine Learning for Scientific Applications*, slides, <https://www.slideshare.net/davidlary/machine-learning-for-scientific-applications>
- [34] E. Pampalk, S. Dixon and G. Widmer, "Exploring music collections by browsing different views", *Computer Musical Journal*, Vol. 28, No. 2, pp. 49-62, Summer 2004
- [35] J. Z. Bloom, "Market segmentation - a neural network application", *Annals of Tourism Research*, Vol. 32, No. 1, pp. 93-111, 2005
- [36] World Poverty Map, SOM research page, Univ. of Helsinki, <http://www.cis.hut.fi/research/som-research/worldmap.html>