

## Neural Network Change Detection Model for Satellite Images Using Textural and Spectral Characteristics

<sup>1</sup>A.K. Helmy and <sup>2</sup>Gh.S. El-Taweel

<sup>1</sup>National Authority for Remote Sensing and Space Sciences, Egypt

<sup>2</sup>Department of Computer Science, Faculty of Computers and Informatics,  
Suez Canal University, Egypt

---

**Abstract: Problem statement:** Change detection is the process of identifying difference of the state of an object or phenomena by observing it at different time. Essentially, it involves the ability to quantify temporal effects using multi-temporal data sets. Information about change is necessary for evaluating land cover and the management of natural resources. **Approach:** A neural network model based on both spectral and textural analysis is developed. Change detection system in this study is presented using modified version of back-propagation-training algorithm with dynamic learning rate and momentum. Through proposed model, the two images at different dates are fed into the input layer of neural network, in addition with Variance, Skewness and Eculedian for each image that represent different texture measure. This leads to better discrimination process. **Results:** The results showed that the trained network with texture measures achieve 23% higher accuracy than that without textural parameters. **Conclusion:** Adding textural parameters of satellite images through training phase increases the efficiently of change detection process also, it provides adequate information about the type of changes. It also found, when using dynamic momentum and learning rate, time and effort needed to select their appropriate value is reduced.

**Key words:** Change detection, neural network, image texture

---

### INTRODUCTION

Change detection in imagery is quite useful generally (Radke *et al.*, 2005), but it has particular value in the remote sensing context. The aim of change detection is to find pixels in pairs of co-registered images that correspond to real changes on the ground. Differences those are due to variations in the environment (illumination and atmospheric distortion) or the sensor (focus and calibration) are generally of less interest. These less interesting differences are often pervasive, with the effect visible over the whole image. The more interesting changes, on the other hand, are often anomalous and involve only a few pixels in the image. Schaum and Stocker (1997) and Clifton (2003) have argued that the interesting changes are the anomalous changes and (Theiler and Perkins, 2006; Hwang *et al.*, 2008) proposed a framework that built on the machine learning formalism for anomaly detection, but recast the problem in terms of binary classification: pervasive differences versus anomalous changes.

Based on the function of the current techniques, change detection is classified into two broad categories:

- Change Mask Development (CMD)
- Categorical Change Extraction (CCE)

In CMD changed and non-changed areas are separated by preset threshold according to the spectral characteristic of images. There are mainly two CMD techniques: image difference and image ratio (Singh, 1989).

In image difference the two spatially registered images of time  $t_1$ ,  $t_2$  are subtracted pixel by pixel to produce further image, which represents the change between two images. In image ratio the two images from different dates are divided by each other. If the output is around zero in image difference (or one in image ratio) then the pixel represents non-changed area, otherwise the pixel represents change. However, there are major problems associated with these techniques as The types of changes are unknown directly from these techniques and need to be identified by further pattern recognition system (Singh, 1989), beside there is no previous information about the value of the threshold (Fung and LeDrew, 1998).

---

**Corresponding Author:** A.K. Helmy, National Authority of Remote Sensing and Space Sciences, Egypt

In CCE explicit categorical changes are detected directly based on spectral gray level of the data. There are mainly three CCE techniques.

**Change vector analysis (Lambin and Strahler, 2004):** In change vector analysis a vector derived from each image pixel, with coordinates corresponding to pixel's brightness values in the spectral components (RGB). If the pixel undergoes a change from 1-t2 a vector describing a change can be defined by subtracting the two vectors. The decision that the change has occurred is made if the magnitude of the change vector exceeds pre-specified threshold value and the direction of the change vector contains the information about the type of changes. Again there is a problem that there is no previous information about the value of the threshold.

**Post classification comparison:** Classification is the process of sorting pixels into a finite number of individual classes, or categories of data, based on some similarity measure (Denq *et al.*, 2008). If a pixel satisfied a certain set of criteria, the pixel is assigned to the class that corresponds to those criteria. This process is also referred to as image segmentation. In post classification comparison technique, the two images with different date are classified separately and the comparison is made between two classified images.

**Direct multi-date classification:** In direct multi-date classification technique (Warner *et al.*, 2009), the two images with different dates are combined together to produce one single image, the classification is made on the output combined image. However, there are problems associated with these techniques namely; the accuracy of these techniques is critically dependent upon the accuracy of the classification process.

**Artificial neural network:** A neural network is a computational structure inspired by the study of biological neural processing. There are many different types of neural networks; from relatively simple to very complex, just as there are many theories on how the biological neural works. Based on the learning algorithm used, a neural network is classified into two main categories:

**Unsupervised learning:** In this mode the competitive learning (winner-take-all) strategy (Parisi *et al.*, 2004) is used; the network consists of two layers: an input layer and an output layer. The inputs are fed into each neuron in the input layer and each neuron determines its output according to a weighted sum formula:

$$\text{Output} = \sum w_{ij} x_i$$

The neuron with the largest output is the winner. This neuron has a final output of 1 and all other neurons in the layer have an output of zero, leading that Different input patterns end up fires different winner neuron. So it's usually used for clustering the input data.

**Supervised learning:** In this mode (Windrow and Lehr, 1990; Hush and Horne, 2003) the actual output of a neural network is compared to the desired output (previously defined). The network then adjusts its weights, which are usually randomly set to begin with, so that the next iteration will produce a closer match between the desired and actual output. The learning process tries to minimize the current errors of all processing elements.

In change detection of digital images, the input image is not classified into different clusters but changed and non-changed area need to be separated and the types of changes may need to be identified. In this study supervised training algorithm is used using modified Backpropagation training algorithm.

**Back-propagation neural networks:** Artificial neural networks are large networks of individual processing elements that are interconnected. Figure 1 represents three-layer feed forward neural network, they are designed to perform a specific pattern recognition task by specifying their architecture (Windrow and Lehr, 1990) (the input layer, the output layer, the hidden layers, the number of processing elements in each layer and the weight of each connection). The formulation of the input and output layer is determined from the definition of the problem and the way you want to solve this problem, the hidden layers learn to provide a representation for the input-output pattern. Running the network consists of two phases forward and backward passes.

**Forward pass:** The output at each neuron is calculated as well as the error at the output units.

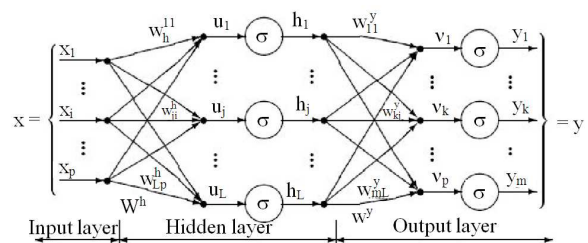


Fig. 1: A general three-layer backpropagation network

**Backward pass:** The output unit error is used to update the weights on the output units. Then the error at the hidden node is calculated (by back-propagating the error at the output units through the hidden layers) and the weights on the hidden nodes will be changed using these values.

For each data pair to be learned a forward pass and backward pass is performed. This will be repeated over and over again until the error is low enough or we give up. Backpropagation is derived by minimize the error at the output units over all input patterns (delta rule). The weights that minimize the error at the output units over the entire input patterns are to be calculated as follows. Let:

$$O_k = \frac{1}{1 + e^{-net_k}}$$

where,  $net_k = \sum_j w_{jk} O_j$  :

$$O_j = \frac{1}{1 + e^{-net_j}}$$

$O_k$  = Be the output of neuron k in output layer  
 $O_j$  = Be the output of neuron in hidden layer j  
 $w_{jk}$  = Be the weight associated with the connection between neuron j and k

E gives the error at the output units at output layer over all the input patterns:

$$E = \frac{1}{2} \sum_p \left( \sum_k (t_{pk} - o_{pk})^2 \right)$$

Where:

$t_{pk}$  = The target value of the output unit k for pattern p  
 $o_{pk}$  = The actual output value of the output layer unit k for pattern p

The weight update in the output layer equals:

$$\eta \delta_k o_k$$

Where:

$\delta_k = o_k (1 - o_k) (t - o_k)$   
 $\eta$  = Learning rate, small positive constant

The weight update equal in the hidden layer equals to:

$$\eta \delta_j o_j$$

Where:

$\delta_j = o_j (1 - o_j) \sum_k \delta_k w_{jk}$   
 $\delta_k = (1 - o_k) o_k (t - o_j)$

The speed of learning process is governed by the learning rate value. Some researchers include a term, momentum, which includes a proportion of the last weight change. The new formula of weight update with momentum:

$$\Delta w_{ij}(n) = \eta \delta_j o_j + \mu \Delta w_{ij}(n-1)$$

## MATERIALS AND METHODS

While the use of spatial information, for instance via Markov random field models (Bruzzone and Prieto, 2000), can improve change detection performance, the approach here will be to concentrate on the spectral information in the pixels. Many authors reported that in many cases spectral information is not enough to discriminate between different types of changes (Van Oort, 2007). Through this work we use derived textural information side by side with the spectral information to improve the accuracy of change detection namely:

- **Mean Euclidean distance:** Iron and Petersen (1981):

$$\text{Mean Euclidean Distance} = \frac{\sum \left[ \sum_{\lambda} (x_{c\lambda} - x_{ij\lambda})^2 \right]^{\frac{1}{2}}}{n-1}$$

Where:

$x_{ij\lambda}$  = DN value for spectral band  $\lambda$  and pixel (i,j) of a multispectral image  
 $x_{c\lambda}$  = DN value for spectral band  $\lambda$  of a window's center pixel  
 $n$  = Number of pixels in a window

- **Variance:**

$$\text{Variance} = \frac{\sum (x_{ij} - M)^2}{n-1}$$

Where:

$x_{ij}$  = DN value of pixel (i,j)  
 $n$  = Number of pixels in a window  
 $M$  = Mean of the moving window, where:

$$M = \frac{\sum x_{ij}}{n}$$

- **Skewness:**

$$\frac{\sum |(x_{ij} - M)^3|}{(n-1)(v)^{3/2}}$$

Where:

- $x_{ij}$  = DN value of pixel (i,j)
- n = Number of pixels in a window
- M = Mean of the moving window
- V = Variance

The algorithm is designed to accept any number of digital images (eights images in our case, the two original images plus variance, Euclidean and Skewness for each image Fig. 2). The output of neural network represents changed image where changed and non-changed areas are assigned to different codes, more over different types of changes and non-changes are included and assigned another codes.

**Network input (problem definition):** The two satellite images with different dates plus variance, Euclidean and Skewness for both images are fed into the network and it is expected that network will be able to differentiate between changed and no changed areas. The input data to the network consist of eight 256 gray-level images representing Ismailia governorate. The structure of data input to the network is to read one pixel from one image and the corresponding one from another image at a time. The pixels of the whole images are processed sequentially on a pixel-by-pixel basis. The all images are spatially registered and scaled to the range of 0-1. The whole process consists of three phases. Firstly learning phase, in which samples of changed and non-changed area is fed into neural network. Secondly, execution phase, in which the two satellite images with different dates and auxiliary textural images are entered to the neural network for classification process. Finally, change extractor, in which the values of output neurons for each input vector are assigned to different codes according to its rank. These phases are explained in Fig. 3.

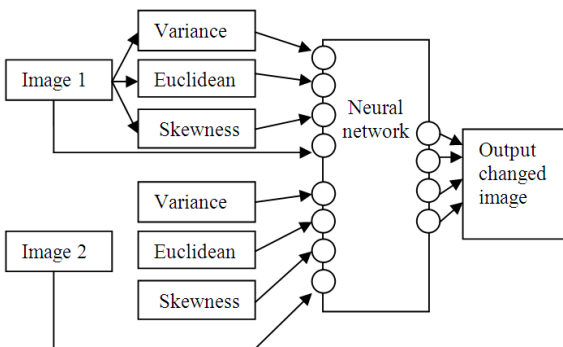


Fig. 2: Block diagram of change detection system

**Learning phase:** In this stage of network, the network weights are randomly generated between (-1, 1) and the training process will try to adjust the weights so that the actual output come out closer to the expected output (Hush and Horne, 1993). The procedures of the learning phase are summarized as follows:

1. One of the patterns to be learned will be put on the input units
2. Values of the output of hidden layers units and output layer units are calculated
3. The errors on the output layer units are calculated
4. The modified learning rule algorithm (Werntges, 1993) will adjust the weights leading to the output units

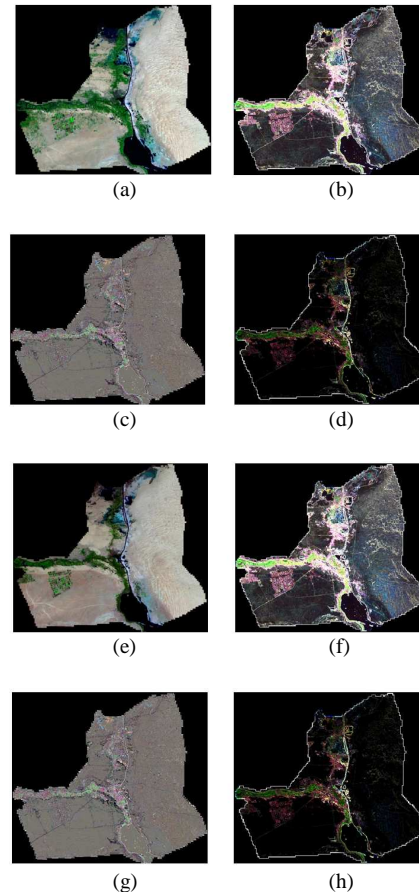


Fig. 3: The input images to the neural network (a) Satellite image for Ismailia governorate acquired 2007 (b) Mean Euclidean Distance of (c) the Skewness image (d) variance image. (e) Satellite image for Ismailia governorate acquired 2000 (f) Mean Euclidean distance of (g) the Skewness image (h) variance image

5. The errors on the hidden units are calculated
6. The weights leading into the hidden layers will be adjusted
7. Steps from 1-6 are repeated for all the input patterns
8. Steps from 1-7 are repeated until the actual output is closer to desired output

**Dynamic learning rate optimization:** Since the backpropagation error curve (Werntges, 1993) usually consists of a large amount of flat regions as well as extremely steep regions. So there is a great difficulty about how to choose an appropriate value of learning rate and momentum. Several authors have tried to characterize the error curves of backpropagation networks (Hush and Horne, 2003; Werntges, 1993). It has been found when the error curve is far from the form of quadratic bowl, as is the case of linear adaptive filter (Chen *et al.*, 1991). They usually consist of a large amount of flat regions as well as long and narrow extremely steep regions. Such as, the back-propagation algorithm with a fixed learning rate would be low efficient (Hush and Horne, 2003). This because the learning rate need to be selected with high value to increase the convergence time, on the other hand the learning rate has to be kept small to prevent over-adjustment of the weights that may occur in the error curve regions with extreme steepness (Werntges, 1993). This study considers straightforward modification of learning rule by dynamically adjust the learning rate as follows:

- If the squared errors (over the entire training set) increase by more than some set percentage  $\xi$  after the weights update. Then the weights update is discarded and the learning rate is multiplied by some factor  $\rho$ , where  $0 < \rho < 1$  and the momentum coefficient is set to zero
- If the squared errors decrease after the weights update, then the weights update is accepted, the learning rate is multiplied by some factor  $\eta$ , where  $\eta > 1$  and the momentum coefficient is returned to its original value if it was previously set to zero
- If the squared errors increase by less than  $\xi$  then the weights update is accepted then, the learning rate and momentum coefficient are unchanged
- $\xi = 4\%$ ,  $\rho = 0.7$ ,  $\eta = 1.05$  were used in this study. Figure 4a shows the squared error, learning rate value and momentum value versus number of cycles
- As shown in Fig. 4, learning rate tends to increase when the output error decreases. When the output error increases by more than 4% the learning rate is reduced and the momentum is eliminated. When the output error increases by less than 4% the learning rate and momentum are kept constant

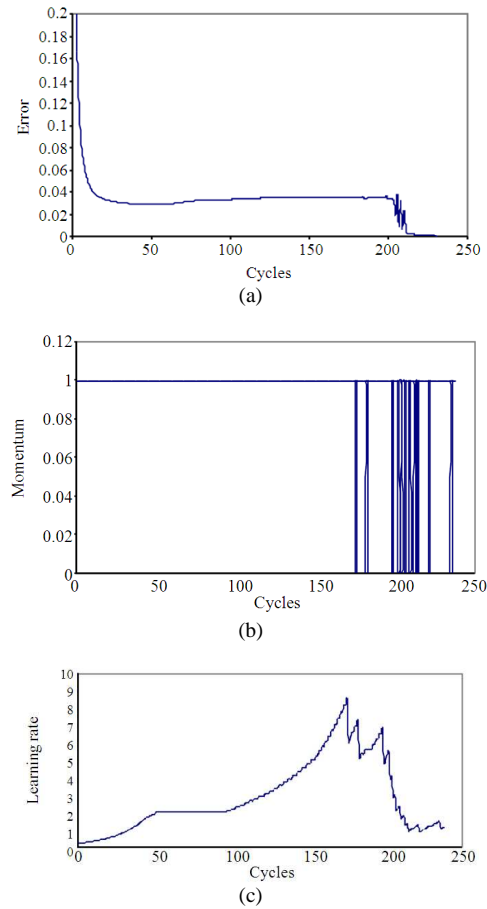


Fig. 4: (a) Output error pattern produced in dynamic learning rate optimization; (b) Variation of learning rate versus number of iterations; (c) Variation of momentum versus number of iterations

**Execution phase:** The eight input images are fed to the network one pixel from one image and the corresponding one from the other image at a time and a set of pre-calculated weights matrix, in training phase, is used to calculate the output values in forward direction (Riedmiller and Braun, 2003).

**Change extractor:** In this part the values of output neurons for each input vector are compared to each other and the order of neuron with the highest value with respect to other neurons represent the output class for the given input vector.












## RESULTS

The previously change detection system is implemented using two TM satellite images representing ismaila governorate acquired at 2000 and 2007, EGYPT to quantify the areas of land cover/land use changes.

Table 1: Sample of training file used in satellite images

	First image		Second image			Expected output
0.396078	0.474510	0.482353	0.019608	0.000000	0.239216	10000000000
0.396078	0.450980	0.494118	0.019608	0.000000	0.239216	01000000000
0.019608	0.000000	0.239216	0.313725	0.196078	0.239216	00100000000
1.00000	1.000000	0.019608	1.000000	1.000000	0.184310	00010000000
0.619608	0.600000	0.580392	1.000000	0.819608	0.376471	00001000000
0.160784	0.611765	0.137255	0.160784	0.635294	0.172549	00000100000

Table 2: Definition of output classes in changed image

	Cover changes	Color	Area (km <sup>2</sup> )
Class 1	Water to water (no-change)		33.00
Class 2	Vegetation to vegetation (no-change)		24.91
Class 3	Sapkha to saphead (no-change)		2.000
Class 4	Soil to soil (no-change)		19.60
Class 5	Coastal line erosion (changes)		7.870
Class 6	Sand to sapkhas (Changes)		2.700
Class 7	Sediment (changes)		0.620
Class 8	New reclaimed areas (changes)		8.100
Class 9	Soil changes (changes)		1.956
Class 10	Soil changes (changes)		7.540
Class 11	Soil changes (changes)		2.670

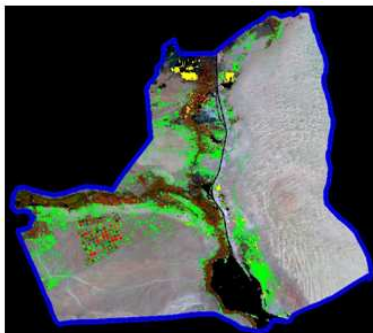


Fig 5: The output changed image

Figure 3 shows the input images to the change detection system. During the learning phase 0.22% of the whole images are learned to the system using binary encoding. Table 1 shows a sample of the training vectors used in training phase.

The network architecture is set to twenty four neurons in input layer, two hidden layers, fifteen neurons in each hidden layer and eleven neurons in the output layer. Dynamic learning rate with momentum term equal to 0.2 is used. Figure 5 shows the output-changed image contains eleven different categories of changed and non-changed areas as described in Table 2.

### DISCUSSION

The neural network model using textural images as auxiliary information for digital change detection shows a great potential as a change detection technique compared with that without auxiliary information. Especially if the two images contain different classes

with same spectral characteristics, since using textural information overcome the spurious effect on change and non-change classes.

The validation was carried out by using ground truth information provided by the NARSS<sup>1</sup>, it has been shown the proposed system achieve accuracy of 87%. On the other hand, using neural network system without using of textural information achieve accuracy of 72%.

### CONCLUSION

As learning parameters and momentum values, have no rules for assigning specific values for these parameters since they depend mainly on the nature of the problem and the shape of the error curve. The only way to find out their values is by experiments. In this study it was found that, when using dynamic momentum and learning rate optimization will reduce the time and effort needed to select an appropriate value of learning rate and momentum.

### REFERENCES

Bruzzone, L. and D.F. Prieto, 2000. Automatic analysis of the difference image for unsupervised change detection. *IEEE Trans. Geosci. Remote Sens.*, 38: 1171-1182. DOI: 10.1109/36.843009

Chen, S., C.F.N. Cowan and P.M. Grant, 1991. Orthogonal least squares learn in for radial bias function basis function networks. *IEEE Trans. Neural Networks*, 2: 302-309.

Clifton, C., 2003. Change detection in overhead imagery using neural networks. *Applied Intell.*, 18: 215-234. DOI: 10.1023/A:1021942526896

- Denq, J.S., K. Wanq, Y.H. Denq and G.J. Qi, 2008. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. *Int. J. Remote Sens.*, 29: 4823-4838. DOI: 10.1080/01431160801950162
- Fung, T. and E. LeDrew, 1998. The determination of optimal threshold levels for change detection using varies accuracy indices. *Photogram. Eng. Remote Sens.*, 54: 1449-1454.
- Hush, D.R. and B.G. Horne, 1993. Progress in supervised neural networks. *IEEE Sign. Process. Mag.*, 10: 8-39. DOI: 10.1109/79.180705
- Hwang, Y., J.S. Kim and I.S. Kweon, 2008. Change detection using a statistical model in an optimally selected color space. *Comput. Vis. Image Understand.*, 112: 231-242. DOI: 10.1016/j.cviu.2008.05.001
- Lambin, E.F. and A.H. Strahler, 2004. Change vector analysis in multitemporal space: A tool to detect and categorize land-cover change processes using high multitemporal resolution satellite data. *Remote Sens. Environ.*, 48: 231-224. DOI: 10.1016/0034-4257(94)90144-9
- Parisi, R., E.D. di Claudio, G. Orlandi and B.D. Rao, 2004. A generalized learning paradigm exploiting the structure of feedforward neural network. *IEEE Trans. Neural Networks*, 7: 1450-1460. DOI: 10.1109/72.548172
- Radke, R.J., S. Andra, O. Al-Kofahi and B. Roysam, 2005. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Process.*, 14: 294-307. DOI: 10.1109/TIP.2004.838698
- Riedmiller, M. and H. Braun, 2003. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proceeding of the IEEE International Conference on Neural Networks, San Francisco.*
- Schaum, A. and A. Stocker, 1997. Subclutter target detection using sequences of thermal infrared multispectral imagery. *Proc. SPIE*, 3071: 12-22. <http://cat.inist.fr/?aModele=afficheN&cpsidt=2039738>
- Singh, A., 1989. Digital change detection techniques using remotely sensed data. *IEEE Int. J. Remote Sens.*, 10: 989-1003. DOI: 10.1080/01431168908903939
- Theiler, J. and S. Perkins, 2006. Proposed framework for anomalous change detection. *Proceeding of the 23rd ICML Workshop on Machine Learning Algorithms for Surveillance and Event Detection, (MLASED'06), CITESEERX, USA.*, pp: 7-14.
- Van Oort, P.A.J., 2007. Interpreting the change detection error matrix. *Remote Sens. Environ.*, 108: 1-8. DOI: DOI: 10.1016/j.rse.2006.10.012
- Warner, T.A., A. Almutairi and J.Y. Lee, 2009. Remote Sensing and Land Cover Change. In: *The SAGE Handbook of Remote Sensing*, Warner, T.A., M.D. Nellis and G.M. Foody, (Eds.). Sage Publications Ltd., London, UK., ISBN: 978-1412936163, pp: 459-472.
- Werntges, H.W., 1993. Partitions of unity improve neural network function approximation. *Proceeding of the IEEE International Conference on Neural Networks, Mar. 28-Apr. 1, IEEE Xplore Press, San Francisco, CA., USA.*, pp: 914-918. DOI: 10.1109/ICNN.1993.298679
- Windrow, B. and M.A. Lehr, 1990. 30 years of adaptive neural network: Preceptron, madaline and Backpropagation. *Proc. IEEE*, 78: 1415-1441. DOI: 10.1109/5.58323