EXTREME LEARNING MACHINE FOR CLASSIFICATION OF HIGH RESOLUTION REMOTE SENSING IMAGES AND ITS COMPARISON WITH TRADITIONAL ARTIFICIAL NEURAL NETWORKS (ANN)

Shailesh Shrestha[1], *Zbigniew Bochenek*[1], *and Claire Smith*[2]

1. Institute of Geodesy and Cartography, Warsaw, Poland; {shailesh.shrestha / zbigniew.bochenek}(at)igik.edu.pl
2. University of Leicester, Leicester, UK; cls53(at)leicester.ac.uk

ABSTRACT

Artificial Neural Networks (ANN) is an important technique for land cover classification of high resolution images. However, there are many inherent limitations of ANN based on Multi-Layer Perception (MLP) with back propagation such as the necessity of fine-tuning the number of input parameters and slow convergence time. Therefore, an attempt has been made to introduce and explore the potential of an Extreme Learning Machine (ELM) which deviates from iterative weight adjustment of neurons during the learning process and is extremely fast at classifying high resolution images. A detailed comparison of the ELM is made with back propagation ANN with better learning algorithms (Scaled Conjugate Gradient (SCG) and Levenberg-Marquardt (LM)) in terms of accuracy, the time required for fine tuning different parameters, and generalisation capability. A high resolution QuickBird satellite image collected over an area of Warsaw, Poland was used for the analysis. Experimental results showed that the ELM produced classification accuracy comparable to that achieved with newer state-of-the-art ANN. The benefits of employing the ELM over conventional ANN are the need of determining only one user parameter, namely the number of neurons in the network, and the significantly lower computational costs. The simplicity of needing to determine only one parameter and the extremely high speed of the ELM could be extremely helpful for different applications when fast but accurate classification is desired.

INTRODUCTION

There are many algorithms for classification, but parametric classifiers produce sub-optimal results especially with high resolution satellite images. Even with non-parametric classifiers, there is a wide array of options such as Support Vector Machine (SVM), Random Forest (RF), and Artificial Neural Network (ANN). The most widely used ANN variant for classification of RS images is the Multi-Layer Perception (MLP) with back propagation learning based on the Gradient Descent (GD) learning rule (1,2,3,4,5). However, there are many inherent limitations of MLP with back propagation such as the need to fine-tune the number of input parameters and the slow convergence time. In order to alleviate these limitations of MLP a better learning strategy that converges faster and requires less input parameters such as the Scaled Conjugate Gradient (SCG) (6) or the Levenberg-Marquardt (LM) (7) algorithms has been devised. Nevertheless, these better learning strategies still require the heuristics of a trial-and-error approach to determine the topology of an ANN (number of hidden layers and number of hidden neurons in each layer) that is crucial for the efficient performance of ANN. In addition, these learning algorithms are still not available in Commercial-off-the-Shelf (COTS) remote sensing software; therefore, they have been used sparingly for image classification. Slow convergence and the necessity of numerous trials and errors in order to determine critical parameters like the number of hidden layers and the number of neurons in each layer have been key obstacles to wide and efficient utilisation of ANN in the remote sensing community (1,2).

Recently, an innovative modified version of ANN, known as the Extreme Learning Machine (ELM) has been proposed (8). The working principle of the ELM is radically different from the conventional working principle of ANN. First, it does not require any adjustment to the iterative tuning of the input weights of processing neurons and secondly it only uses Single Layer Feedforward Net-

DOI: 10.12760/01-2014-2-02

work (SLFN), so eliminating the complexity of determining the number of hidden layers that is an intrinsic characteristic of the back propagation based ANN. The only parameter, to be tuned in the ELM algorithm, is the number of neurons in the network. For a detailed description of the algorithm, interested readers are encouraged to refer to (8). Being a relatively new approach that was originally developed for the field of signal processing and computer vision, the potential benefits of ELM have not yet been fully harnessed for the classification of RS images. In fact, only one attempt has been made so far to utilize ELM for classification of medium resolution Enhanced Thematic Mapper (ETM+) of spatial resolution 30 m over an agricultural area (4). ELM was compared with standard back propagation ANN and it was concluded that ELM works equally well in terms of classification accuracy in comparison to back propagation neural networks (4). However, the traditional Gradient Descent (GD) learning algorithm was used in the employed ANN architecture.

Consequently, a detailed analysis of application of ELM for the classification of high resolution images provided by sensors such as IKONOS, QuickBird or WorldView is still lacking. Furthermore, the current analysis is based on an image collected over highly structured heterogeneous urban areas consisting of numerous land covers classes with similar kinds of spectra (e.g. roofs, roads or parking lots made of asphalt or cement; bright man-made structures, bare soils; different types of tiles, roads) where differentiation between different classes is more difficult.

Therefore, it is attempted to introduce and explore the potential of ELM for classification of high resolution images and comparison with back propagation ANN with the better learning algorithms SCG and LM in terms of accuracy, time required for fine-tuning different parameters and generalization capability.

METHODICAL APPROACH

The graphical depiction of the methodology employed is presented in Figure 1. The methodology could be divided into three distinct parts: (i) data preparation, (ii) parameterization of ANN, and (iii) classification. The details are presented in the following sections. It is to be noted that all the calculations were performed with an in-house developed script using MATLAB. For ANN, a Neural Network toolbox was used, whereas a GUI developed in-house (9) script written in MATLAB was used for ELM. The comparison of computational cost is based on a dual core Pentium processor based desktop with 8 GB of RAM.

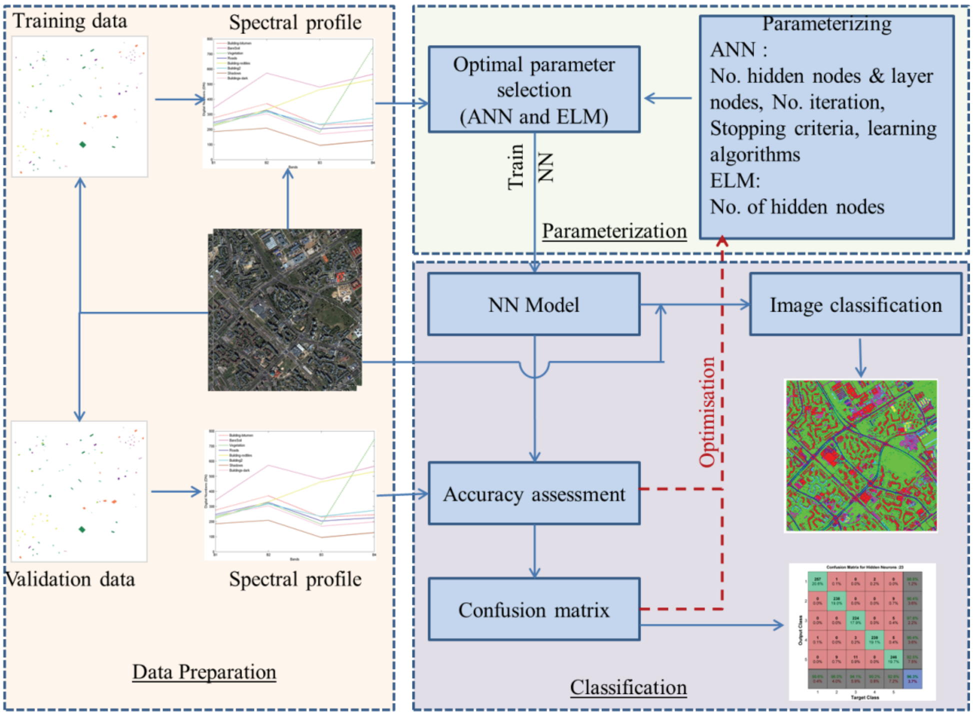


Figure1: Methodology adopted for the classification of the high resolution image.

Dataset and data preparation

A high resolution image with four multi-spectral bands, collected by commercial satellite QuickBird over an area of Warsaw, Poland, on 9th May 2002 was used for the analysis. The data was pan-sharpened by adopting a Gram-Schmidt spectral sharpening algorithm available in the ENVI software package. The image has a spatial dimension of 2644 × 2674 pixels. After a rigorous visual inspection of the image, eight different classes (BareSoil, Building-bitumen, Building-dark, Building-redtiles, Road, Building2, Shadows, and Vegetation) that are prevalent and well characterize the study area were identified. Subsequently, the training set was collected by visual inspection and manual digitization. If the validation set is derived by portioning the training set, it is not independent because of high spatial autocorrelation between neighbouring pixels in remote sensing images (1) and, in that case, trained ANN may fail to detect the problem of overfitting the training set. Therefore, another set of homogeneous regions was collected as a validation set. When collecting the validation set, it was ensured that the validation set was at least three times bigger than the training set. Both training data and validating data were rescaled into [-1,1]. Image data was also rescaled into [-1,1] before being fed into the trained network for the classification purpose. The output encoding method of assigning 1 to true class and 0 to others was employed. The Log-sigmoid transfer function was used as activation function. After the ANN have been trained, the competitive approach (winner-take-all) has been considered to decide on the final classification response.

Parameterization of ANN

Although it has been found that ANN provides a good approach for classifying remote sensing images, the sheer difficulty of fine-tuning many parameters and the lack of an incorporation of efficient implementation of ANN classifiers have limited its wide applicability in remote sensing (1). For a given problem, a large number of architectures are possible. Therefore, considerable time and effort are required to find an optimal architecture selection that would exhibit good classification performance. For example, training of 59 different ANN models with four different internal parameter settings that affect the classification accuracy was required for the optimal selection of parameters for classifying Landsat imagery in Georgia, U.S.A (5). One of the crucial and most challenging tasks in designing an ANN architecture is to determine the number of hidden layers and the number of the neurons in the hidden layer(s) in the network (1-5,10). Many optimal ANN architecture searching strategies are available in the literature such as heuristic suggestions summarized in (2), network pruning and genetic algorithm search (19). There is still a debate among researchers about the number of layers required for optimal performance. It has been stated that a single hidden layer should be sufficient for classification tasks (1). Use of a single hidden layer is more common in the literature for both medium resolution data (4,11) and high resolution data (5). However, based on the analysis of high resolution QuickBird data, it has been argued that the use of a supplementary second layer would be helpful in extracting additional information from what has already been discriminated by the first layer and would thus enhance the discriminatory power of the ANN (3).

In our analysis, both one hidden layer and two hidden layers were considered and the best topology was selected based on the overall accuracy produced on independent validation samples sets. The number of hidden neurons in each layer was searched by an iterative trial-and-error approach. The lower and upper limits for the search were fixed by taking into account heuristic suggestions found in the literature (2). Specifically, the number of neurons varied from 4 to 24, while the numbers of neurons in the second hidden layer varied from 4 to 16 with a step interval of 2 neurons. Two criteria were used for the stopping iterative training of neurons a) the number of iterations for training and b) the estimated mean-sum-square-error (MSSE). The adopted MSSE was 0.05 in this case. The number of iterations was varied beginning with 100 iterations with a step interval of 100 iterations until the MSSE criterion was met with a single layer topology. There are more sophisticated stopping criteria known as Early Stopping (ES) that divides a training dataset into three sets – the training set, and independent validation and testing sets (12). If the training error continues to decrease but the validation set error starts to increase, the training procedure is stopped regardless of the specified number of iterations and MSSE with that method. While we acknowledge that

ES provides a better generalization capability of ANN based on iterative tuning, ES stopping criteria were not used in this case because of tradeoffs between training time and more accurate generalization.

There are several learning algorithms designed to minimize the objective function and hence to train the ANN to produce minimal differences between the actual outputs and the desired outputs by adjustment of neurons weights. Three different learning algorithms (GD, SCG and LM) were considered for comparisons with the ELM. All representative ANN algorithms considered train networks with iterative adjustment of neurons in hidden layers; however, each algorithm updates weights of neurons differently.

Classification

After parameterization of ANN, the accuracy assessment was performed with both the training dataset and the validation dataset. A complete confusion matrix with user's accuracy and producer's accuracy for each class, and overall accuracy was computed. A final classification of the image is performed with only the best optimal parameters.

RESULTS

Single layer ANN topology meeting MSSE requirement

As expected, the GD method, which uses a constant learning rate during the whole iteration process, required considerably more iterations than the other two algorithms. The GD method, the SCG method and the LM method required 2000 iterations, 400 iterations and 100 iterations respectively. Figure 2 illustrates the disparity in the overall accuracy for different learning algorithms for single layer topology with a varying number of hidden neurons. In all cases, training overall accuracy is higher than validation overall accuracy. As the complexity of the network grows with an increasing number of neurons in the hidden layer, the overall accuracy also increases. GD exhibited a lower overall accuracy than that of both SCG and LM algorithms. After a topology of 4-12-8, the increase of the number of hidden neurons did not change the overall accuracy and accuracy curves attained a plateau with both SCG and LM algorithms. In the aforementioned topology, 4 is the number of input nodes governed by the number of bands in the image used, 12 is the number of hidden neurons and 8 is the number of classes in the classification. In addition, it is clearly seen, that there is no significant difference between validation overall accuracies for SCG and LM algorithms. The maximum overall validation accuracy of 82.76 % was obtained with the SCG algorithm with the topology of 4-18-8, while the maximum overall accuracy of 81.34 % was achieved with the LM algorithm with the topology of 4-18-8 as well.

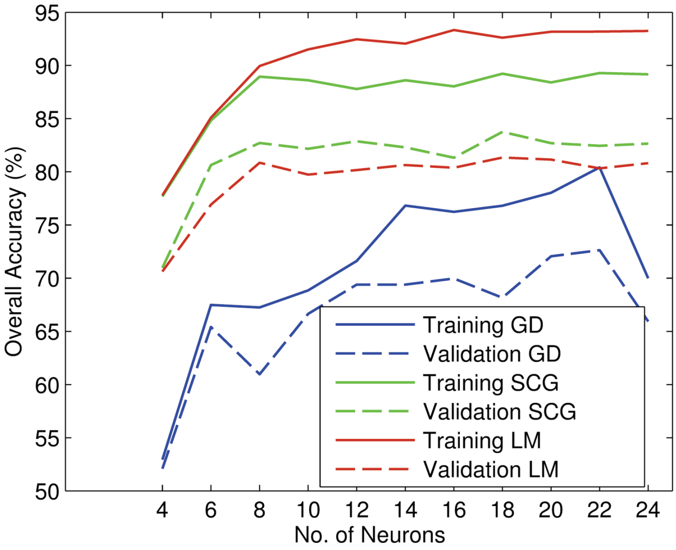


Figure 2: Comparison of overall accuracy for three different learning algorithms for a single layer topology with varying number of hidden neurons.

Comparison of computational cost for different algorithms for a single layer topology with varying number of hidden neurons is shown in Figure 3. The training time is a function of network complexities governed by input nodes, output nodes, the number of hidden neurons and the number of training samples used to train the network. When the number of hidden neurons was less than 12, the GD algorithm took a longer time to train the network. With lower numbers of hidden neurons, the LM algorithm was pretty fast. As the network complexities grow, the LM algorithm performs poorly starting from the topology with 4-14-8. Another interesting fact that can be distinguished in Figure 3 is that the SCG algorithm computation cost is less dependent on the number of hidden neurons as compared to the other two algorithms. SCG training time was lower in all cases except for the one with 4 hidden neurons. So considering both overall accuracy and computational cost, the SCG algorithm was more suitable in this case and topology 4-18-8 with SCG algorithm judged as best when only a single layer of hidden neurons is considered.

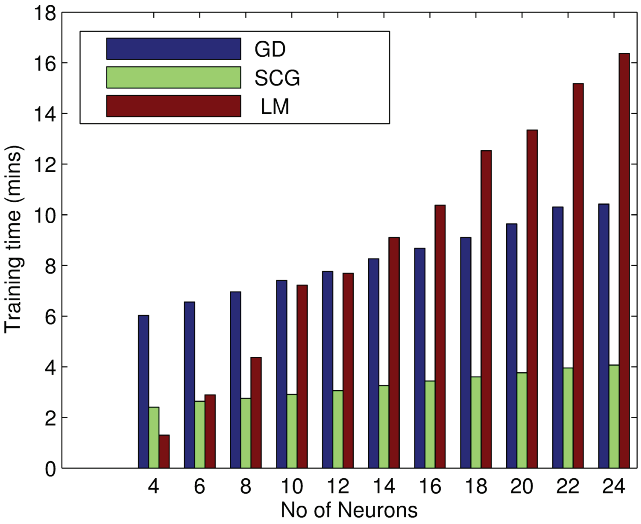
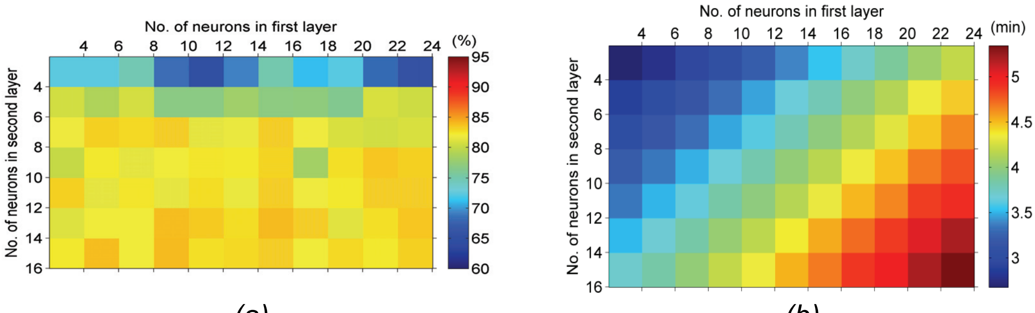


Figure 3: Comparison of computational cost for three different learning algorithms for single layer topology with varying number of hidden neurons.

Addition of second layer in the ANN topology

For the experiment with an additional second layer of hidden neurons, results of ANN architecture with only SCG and LM are reported. The performance of the SCG algorithm with two layers of hidden neurons is illustrated in Figure 4. The best performance was achieved with 4-18-8 topology with overall validation accuracy of 82.76% and computational cost of 3.60 minutes only with a single layer of hidden neurons. Many more double-layer topologies resulted in higher overall validation accuracy than the best single-layer topology. Altogether, 28 double-layer topologies produced a better result. The best result of overall validation accuracy of 84.10% was obtained with a topology of 4-10-16-8. The computational cost for topology 4-10-16-8 was only 4.19 minutes which demonstrates that increase in computational cost is negligible (increase of only 0.59 minutes as compared to the best single-layer topology). The gain in overall validation accuracy was 1.34%, which is a significant increase considering the fact that the total number of validation samples used was 68,690.



(a) *(b)*

Figure 4: Performance of SCG with two hidden layers. (a) Overall accuracy (b) Computational cost.

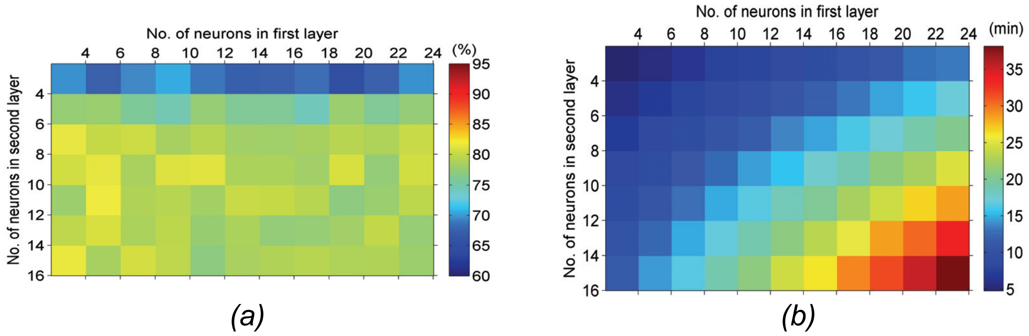


Figure5: Performance of LM with two hidden layers. (a) Overall accuracy (b) Computational cost.

The performance of the LM algorithm with two layers of hidden neurons is illustrated in Figure 5. The best overall validation accuracy of 81.88% was obtained with the topology 4-6-12-8. As compared to the best single topology with LM, the increase in overall validation accuracy was only 0.54%, however, the double-layer topology took a significant time (10.71 min) to train the network. As depicted in Figure 4b) and Figure 5b), the training time for the SCG algorithm is fairly less compared to that of the LM algorithm. For the double layer with a topology (4-24-16-8), the training time for the LM algorithm is 38.15 minutes, whereas it is only 5.34 minutes for the SCG algorithm. This provides evidence that as the complexities of network grow, the LM algorithm takes a significant training time, whereas the training time is less dependent on the network complexities for the SCG algorithm. The results indicated that the additional second layer of hidden neurons provided a better discriminatory power than that of the single-layer topology. Based on the overall validation accuracy, the double-layer topology of 4-10-16-8 with the SCG algorithm was deemed to be optimal for this case. Hence, it was used for comparison with ELM for classifying the entire image. It is to be noted that some variations in reported overall accuracy in Figure 4a) and Figure 5a) can be due to an effect of random initialization of weight and bias of neurons during the training phase. The effect of randomness can be minimized by repeating the training process many times and averaging the results. However, it was not performed due to a trade-off between very accurate generalization capability of the trained ANN and the computation time.

Extreme Learning Machine (ELM)

Figure 6 provides the performance of the ELM algorithm in terms of overall validation accuracy and computational cost.

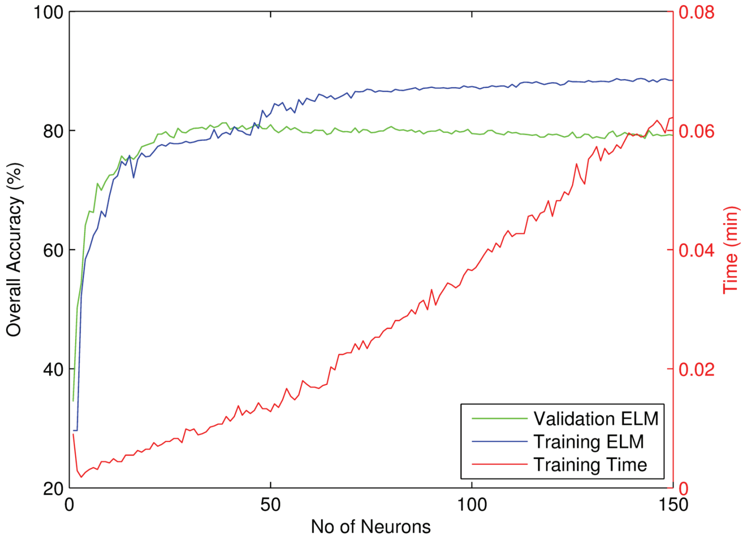


Figure 6: Performance of ELM in terms of overall accuracy and computational cost.

In the ELM algorithm, the number of neurons is the only parameter to be tuned. The number of neurons varied from 1 to 150. The result indicated that ELM also follows a similar trend: Training overall accuracy is lower than that of validation overall accuracy. The overall accuracy increases with the number of neurons up to a certain number and after that it achieves a stagnant plateau, where overall accuracy no longer significantly depends on the number of neurons. The optimal topology for the ELM algorithm was found to be 4-39-8, which produced an overall validation accuracy of 81.27%. Increasing the number of neurons beyond that point resulted in higher training overall accuracy, but lowered the testing validation accuracy, which exemplifies the overtraining of the network.

In terms of computational cost, the time required for training networks by the ELM algorithm was found to be linearly proportional to the numbers of neurons used. The ELM is very efficient and fast in terms of training time; even with more than 150 neurons it took considerably less than a minute. The average training time for a single iteration is 0.028 minute. The cumulative overall training time for numbers of neurons varying from 1 to 150 neurons is around 5 minutes only. The optimal topology 4-39-8 required only 0.012 minute for the training.

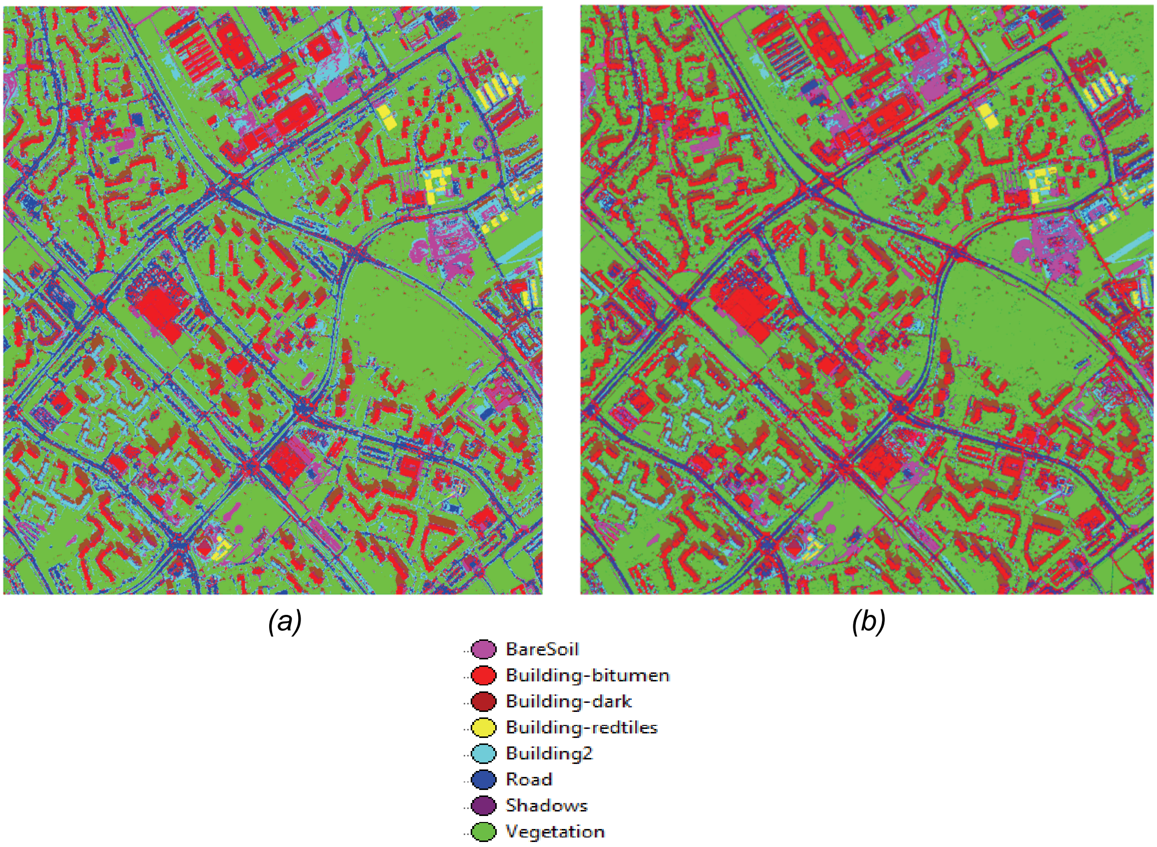
Comparison of optimal ANN and ELM networks

Table 1 presents a detailed comparison between the optimal topology of ANN based on the SCG learning algorithm and the newly introduced ELM algorithm. For all classes, accuracy was expressed as the percentage of correctly classified test pixels. The numbers of pixels used for each class for both training and validation are also reported. The SCG algorithm produced classification results with Overall Accuracy (OA) of 84.1% with a Kappa (K) coefficient of 0.796, while the newer technique ELM produced classification results with OA of 81.27 % and a K coefficient of 0.765. In terms of both OA and K, SCG performed slightly better than ELM. Other values such as Average Producers Accuracy (APA) and Average Users Accuracy (AUA) are more or less the same. In both classifications, the highest classification accuracy was obtained for the class *Vegetation*, whereas the lowest classification accuracy was obtained for the class *Building-dark*. The class *Vegetation* shows accuracies of 99.97% and 99.44% in SCG and ELM, respectively, which indicates that there is no problem of discriminating the class *Vegetation* from the rest of the classes, as the spectral signature of the class *Vegetation* is quite different from the rest of the classes. Similarly, both algorithms SCG and ELM have difficulty in correctly identifying the class *Building-dark* from *Shadows and Roads* as their spectral signatures are more or less similar. The classification results obtained from the optimal trained SCG based on iterative tuning of neurons and the single layer ELM without iterative adjustment are shown in Figure 7a) and Figure 7b), respectively.

Table 1: Detailed comparision with SCG and ELM.

	Train Set	Val. Set	Pixel-wise methods		Computational cost	
			SCG	ELM	SCG	ELM
OA			**84.1**	**81.27**	*Iteration: 88* *Total time: 343 min*	*Iteration:150* *Total Time: 5 min*
APA			80.52	79.49		
AUA			81.89	79.25		
K			**0.796**	**0.765**		
Building-bitumen	7,825	23,711	88.21	89.27	*Training time for optimal topology* 4.19 min	*Training time for optimal topology* 0.012 min
BareSoil	2,374	6,212	86.83	90.74		
Vegetation	6,159	12,219	99.97	99.44		
Roads	3,906	8,149	67.80	52.61		
Building-redtiles	724	2,874	93.63	87.27	*Minimum Parameters* 3 *(No. of iteration, no. of neurons, no. of layers)*	*Minimum Parameters* 1 *(No. of neurons)*
Building2	2,310	5,787	87.07	84.07		
Shadows	1,243	3,169	97.60	98.93		
Building-dark	1,869	6,569	34.04	31.66		

The computational cost of a classification represents a significant proportion of the cost involved in land cover classification. Table 1 indicates that the ELM algorithm is far more effective and provides significant advantages over the SCG algorithm. The training time for the optimal topology of the ELM algorithm is considerably less than 1 minute (~0.012 minute), while the training of the SCG algorithm required 4.19 minutes. The ELM required less training time for the optimal architecture by the factor of 349 which is a significant advantage. The advantage is even more apparent when the time required for the whole training process to select optimal architecture is considered. The time required for the whole process of finding optimal parameter and architecture for the SCG algorithm is 343 minutes (5.71 hours). As compared to this, the ELM required 5 minutes only for the whole selection process. Another significant advantage of the ELM algorithm over the SCG algorithm is that the ELM algorithm has only one parameter to be tuned, i.e. the number of neurons in a single hidden layer given, while SCG has a minimum of three parameters to be tuned (number of iterations, number of neurons and number of layers). Given the fact that both classification algorithms produced comparable results and that ELM requires minimal time for parameterizing optimal parameters, using ELM is advantageous as compared to SCG. Furthermore, the ELM algorithm is much simpler than the learning algorithms for neural networks and it can be easily programmed in scripting languages like MATLAB, C++, python or any other programming language. For a detailed description of a MATLAB-based GUI for the ELM algorithm, refer to (9).


(a)

(b)

Figure 7: The thematic classification maps of a QuickBird image of Warsaw, Poland; (a) classification with SCG (b) classification with ELM.

CONCLUSION

With the increasing availability of high resolution spatial data, there is a considerable impetus towards machine learning-related classifiers for the production of land use/ land cover thematic maps. Among many other algorithms, the use of ANN in remote sensing has been increasing, but many people are quite reluctant to switch to applying ANN in their research because of the difficulty in the adjustment of many of the parameters that are required for optimal performance of the classifier as well as because of the longer time required for parameter adjustment. As an alternative option, the ELM approach with SLFF with very quick learning was presented and compared with state-of-the-art back propagation based learning algorithms, particularly the SCG algorithm and the LM algorithm, for classifying high resolution QuickBird imagery over an urban area in Warsaw, Poland. For the dissemination of ELM, a MATLAB-based GUI has been developed. The results suggested that the ELM could be a viable alternative for classical ANN with iterative learning approach, since ELM produced comparable classification accuracy with ANN based on newer learning algorithms. The significant benefit of employing the ELM algorithm over conventional ANN is the need to parameterize only one user parameter, namely the number of neurons in the network, as well as the significantly lower computational cost.

ACKNOWLEDGEMENTS

The study has been carried out within the scope of the GIONET project funded by the European Commission, Marie Curie Programme, Initial Training Networks, Grant Agreement number PITN-GA-2010-264509. The authors would like to thank the two anonymous reviewers for their valuable constructive comments.

REFERENCES

1 Mas J F & J J Flores, 2008. The application of artificial neural networks to the analysis of remotely sensed data. International Journal of Remote Sensing, 29(3): 617-663

2 Kavzoglu T & P Mather, 2003. The use of backpropagating artificial neural networks in land cover classification. International Journal of Remote Sensing, 24(23): 4907-4938

3 Frate F D, F Pacifici, G Schiavon & C Solimini, 2007. Use of Neural Networks for Automatic Classification from High resolution Images. IEEE Transactions on Geoscience and Remote Sensing, 45(4): 800-809

4 Pal M, 2009. Extreme learning machine based land cover classification. International Journal of Remote Sensing, 30(4): 3835-3841

5 Jiang J, J Zhang, G Yang, D Zhang & L Zhang, 2010. Application of back propagation neural network in the classification of high resolution remote sensing image. In: 18th International Conference on Geoinformatics (Peking University, Beijing) 1-6

6 Möller M F, 1990. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, 6: 525-533

7 Levenberg K, 1944. A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics, 2(2): 164-168

8 Huang G B, Q Y Zhu & C K Siew, 2006. Extreme learning machine: Theory and applications. Neurocomputing, 70: 489-501

9 Shrestha S, Z Bochenek & C Smith, 2012. Artificial Neural Network (ANN) beyond cots remote sensing packages: Implementation of Extreme Learning Machine (ELM) in MATLAB. In: Geoscience and Remote Sensing Symposium (IGARSS). IEEE International, 6301-6304

10 Stathakis D, 2009. How many hidden layers and nodes? International Journal of Remote Sensing, 30(8): 2133-2147

11 Dixon B & N Candade, 2008. Multispectral landuse classification using neural networks and support vector machines: one or the other, or both? International Journal of Remote Sensing, 29(4): 1185-1206

12 Prechelt L, 1998. Early Stopping - But When? In: Neural Networks: Tricks of the Trade, edited by G Orr & K-R Müller (Springer: Berlin, Heidelberg) 55-69