Earth Observation Laboratory
Tor Vergata University
Rome - ITALY

Colorado Center for Astrodynamics Research
University of Colorado at Boulder
Colorado - USA

2007 REMOTE SENSING DATA FUSION CONTEST: NEURAL NETWORKS FOR DATA FUSION

By

Fabio Pacifici

William J. Emery **Fabio Del Frate**

e-mail: f.pacifici@disp.uniroma2.it

Submitted to
IEEE Data Fusion Technical Committee

When publishing results based on this report, please, refer to:

F. Pacifici, F. Del Frate, W. J. Emery, P. Gamba, J. Chanussot, "Urban mapping using coarse SAR and optical data: outcome of the 2007 GRS-S data fusion contest", *IEEE Geoscience and Remote Sensing Letters* vol. 5, no. 3, Jul. 2008

Table of Contents

Chapter 1

Introduction

In 2007, the Data Fusion Contest [1] was related to urban mapping using radar and optical data. A set of satellite radar and optical images (ERS amplitude and Landsat multi-spectral data) was available with the task of obtaining a classified map as accurate as possible with respect to ground truth data, depicting land cover and land use classes for the urban area under test.

The data set contains different SAR amplitude images, referring to the urban test area of Pavia, Northern Italy (45.11N, 9.09E) acquired by sensor ERS 1 and ERS 2 during the period between 1992 and 1995 and two Landsat images, one acquired on 1994 and the other on 2000 as reported in Table 1.1.

Sensor	**Acquisition date**	**Image ID**
ERS	August 13, 1992	SAR Date 1
ERS	October 22, 1992	SAR Date 2
ERS	June 24, 1993	SAR Date 3
ERS	November 11, 1993	SAR Date 4
ERS	October 3, 1994	SAR Date 5
ERS	November 9, 1994	SAR Date 6
ERS	July 22, 1995	SAR Date 7
ERS	July 23, 1995	SAR Date 8
ERS	August 27, 1995	SAR Date 9
Landsat	April 7, 1994	Landsat Date 1
Landsat	October 8, 2000	Landsat Date 2

Table 1.1: Data Fusion Contest data set.

The site has been chosen because it well represents the diversity of urban land covers,

uses, and features. Pavia is a small town with a very densely built center, some residential areas, industrial suburbs, and the Ticino River runs across it [2].

The classification processing is summarized by the following three steps:

1. *Preprocessing*: Principal Component Analysis (PCA) of SAR imagery concatenated with the remaining Optical data
2. *Neural Network design*:
    - Multilayer Perceptron (MLP)
    - error minimization by Scaled Conjugate Gradient(SCG)
    - pruning by Magnitude Based algorithm
3. *Post-processing*: minimum mapping unit (*sieve* and *clump*)

Chapter 2

Preprocessing

Different classification schemes could be developed in order to achieve (when possible) the desired classification accuracy. The first step to be considered regards the choice of the inputs which will successively feed the Neural Network. The simplest scheme, shown in Figure 2.1(a), feeds the Neural Network with 21 inputs (9 SAR and 6+6 Optical) which generally requires a long training time. Therefore, the reduction of the input dimensionality is generally desirable: Principal Component Analysis (PCA) was applied to decrease the number of inputs used to train the NN.

PCA maps image data into a new, uncorrelated coordinated system in which the data have greatest variance along its first axis, the next largest variance along a second mutually orthogonal axis, and so on. The later principal components would be expected, in general, to show little variance [3]. In Figure 2.2 are shown the PCA Eigenvalues for the cases relative to dates 1:6 and 7:9 of the SAR imagery. As expected, only the first principal component shows a large variance. Therefore, the remaining components do not contribute to separability and could be ignored, reducing the dimensionality of the classification space.

The input space reduction can applied to both SAR and Optical imagery. Considering the different characteristics of the data, PCA should be applied to similar features, resulting in 8 inputs: 2 from SAR data (using the first component of dates 1:6 and 7:9) and 6 from

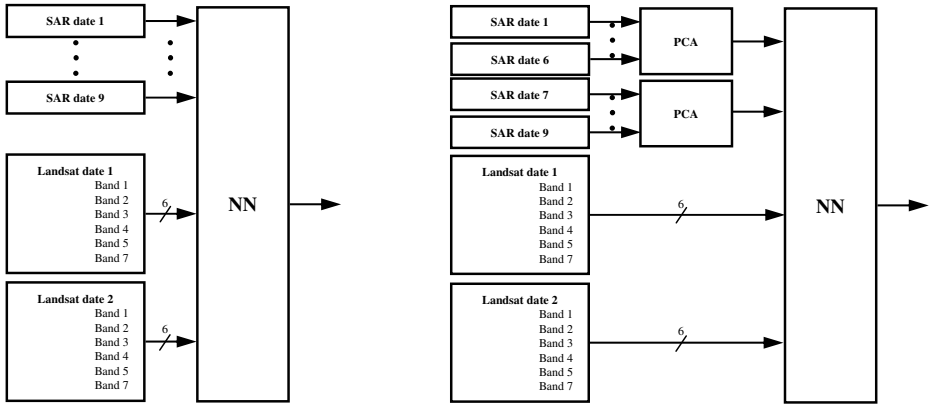

Figure 2.1: Classification schemes: (a) all 21 inputs feed the neural network. The features reduction is obtained by applying (b) PCA only to SAR imagery and considering the first component of dates 1:6 and 7:9.

Optical (using, for each couple of bands, only the first PCA component).

These 8 inputs did not appear to contain enough information, since a poor classification accuracy resulted. Therefore, the PCA method was applied only to SAR imagery, resulting in 14 inputs: 2 from SAR data (again, considering the first component of dates 1:6 and 7:9) and 6+6 Optical, as shown in Figure 2.1(b).

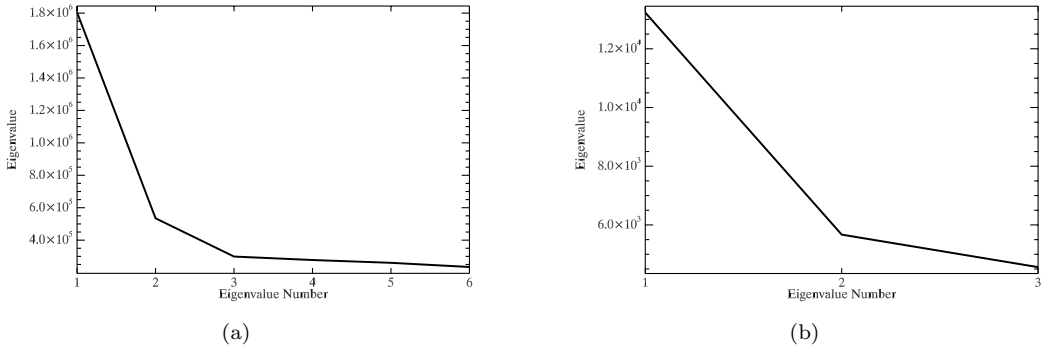

Figure 2.2: PCA Eigenvalues for dates: (a) 1:6 and (b) 7:9 of the SAR imagery.

Chapter 3

Neural Network Classification

An artificial Neural Network may be viewed as a mathematical model composed of non-linear computational elements, named neurons, operating in parallel and connected by links characterized by different weights.

3.1 Neuron Structure

The basic building block of a NN is the neuron. As described by many models over the years [4]-[9], a single neuron is an information processing unit generally characterized by several inputs and one output. As shown in Figure 3.1, each neuron consists of three main parts:

- the weight vector
- the net function
- the activation function

The net function used is a weighted linear combination of inputs such as:

$$o = \sum_{i=1}^{N} x_i w_i + \theta \tag{3.1.1}$$

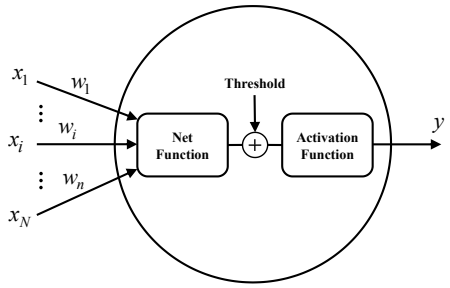

Figure 3.1: Neuron model: the input vector is combined with the weight vector through the net function. Then, the activation function is applied, producing the neuron output.

where o denotes the net function output, $x_1, x_2, ..., x_N$ and $w_1, w_2, ..., w_N$ are the components of the neuron input and synaptic weight vectors, respectively, while θ is called the bias or threshold.

The neuron output is achieved using the activation function related to the net function output through a linear or non-linear transformation. The following Sigmoid activation function was used:

$$y(o) = \frac{1}{1 + e^{-o}} \tag{3.1.2}$$

In general, the activation functions are characterized by saturation at a minimum and a maximum value: this is also the case of the Sigmoid. Therefore, to avoid saturation within the network, it has been necessary to scale all the values of the input vector in the range $[-1; +1]$. At the same time, the component of the output vector corresponding to the true class has been set to 1 while the others are set to 0.

3.2 Multilayer Perceptron

Although a single neuron can resolve simple information processing functions, the advantage of the neural computation approach comes from connecting neurons in networks, where several units are interconnected to form a distributed architecture. Due to its high generalization ability, the Multilayer Perceptron (MLP) is the most widely used neural network for

solving decision-making problems for many different applications [10] [11]. It approximates an unknown input-output relationship providing a nonlinear mapping between its inputs and outputs.

The architecture consists of different layers of neurons, while the interconnections are provided only between neurons of successive layers of the network. The first layer merely distributes the inputs to the internal stages of the network: there is no processing at this level. The last layer is the output which provides the data processed. The layers between the input and the output are called hidden layers. The number of neurons which compose the input and output layers are directly related to the dimension of the input space and to the dimension of desired output space, respectively. The five classes of interest considered for the contest are *City Center*, *Residential Areas*, *Sparse Buildings*, *Water*, *Vegetation*. The number of training and validation samples are reported in Table 3.1. The training pixels used were part of the larger set provided, while the validation samples were selected by careful visual inspection of the scene.

Class	**Color**	**Tr. Set**	**Val. Set**
City Center	Yellow	3783	4000
Residential Areas	Red	7572	8000
Sparse Buildings	Blue	5994	8000
Water	Cyan	893	1000
Vegetation	Green	591	10000

Table 3.1: Classes of interest with relative color map, number of training and validation samples.

Once the input and the output of the network are established, the critical step is to find the optimal number of units to be considered in the hidden layers. In fact, if the number of these neurons is too small, the input-output associative capabilities of the network are too weak. At the same time, this number should not be too large, otherwise these capabilities might show a lack of generality being too narrowly tailored to the training set, and the computational complexity of the algorithm would be increased in vain [12]. Therefore, a compromise has to be found to select the most suitable number of hidden neurons for the

optimal neural network topology.

Two different approaches can be used to find the best architecture:

- the *growing method* in which the starting network is small and the neurons are subsequently added until the optimization criteria is reached;
- the *pruning method* in which the starting network is relatively large and the neurons are subsequently removed until the optimization criteria is satisfied.

The approach used here is the latter. Therefore, after a reasonable evaluation in term of classification accuracy, the chosen topology was 14-200-200-5. This estimation involved the analysis of the output variance of different topologies characterized by an increasing number of hidden neurons [13].

3.3 Learning Algorithm

During the training phase, the network learns how to approximate an unknown input-output relation by adjusting the weight connections. This is done by receiving both the raw data as inputs and the target as outputs which are used to minimize an error function. There are several learning algorithms designed to minimize this error function related to the differences between the inputs $x_1, ..., x_N$ and the target outputs $t_1, ..., t_N$. The learning algorithm used here is the Scaling Conjugate Gradient method (SCG) [14] which is a supervised learning algorithm for feedforward neural networks.

The Conjugate Gradient Methods (CGM) belong to the second order techniques that minimize the error function. Second order means that these methods make use of the second derivatives of the goal function, while first-order techniques, like standard backpropagation, only use the first derivatives. A second order technique generally finds a better way to a (local) minimum than a first order technique, but at a higher computational cost. Backpropagation always proceeds down the gradient of the error function, while Conjugate

Gradient methods proceed in a direction which is conjugate to the directions of the previous steps. This approximation gives more complete information about the search direction and step size [15].

Basically in most applications problems, the error function decreases monotonically towards zero for a increasing number of learning iterations. Scaling Conjugate Gradient adds to the Conjugate Gradient methods a scaling approach to the step size in order to speed up the algorithm. The number of epochs is not relevant when comparing SCG to other algorithms like standard backpropagation. Indeed one iteration in SCG needs the computation of two gradients, and one call to the error function, while one iteration in standard backpropagation needs the computation of one gradient and one call to the error function. Möller [14] defines a *complexity unit* (cu) to be equivalent to the complexity of one forward passing of all patterns in the training set. Then computing the error costs 1 cu while computing the gradient can be estimated to cost 3 cu. According to Möller's metric, one iteration of SCG is as complex as around 7/4 iterations of standard backpropagation.

The parameters that specify the SCG algorithm are four:

1. σ: it should satisfy $0 < \sigma < 10^{-4}$. Empirically, Møller [14] has shown that bigger values of σ can lead to a slower convergence.

2. λ: it should satisfy $0 < \sigma < 10^{-6}$. The values of λ directly scale the step size in the way, that the bigger λ, the smaller the step size.

3. d_{max}: it is the maximum difference $d_i = t_i - y_i$ between a target value t_j and an output y_j of an unit which is tolerated. Typical values of d_{max} are 0.0, 0.1 or 0.2.

4. ε: it is the value of the termination criterion used to avoid floating point exceptions.

Table 3.2 shows the values of the SCG parameters used to train the Neural Network.

Parameter	Value
σ	10^{-4}
λ	10^{-6}
d_{max}	0
ε	10^{-8}

Table 3.2: Values of the SCG parameters used to train the Neural Network.

3.4 Pruning

A central problem in pattern recognition consists of minimizing the system complexity. In Neural Networks this issue often consists of minimizing the number of connection weights [16]. Pruning algorithms try to make neural networks smaller by pruning unnecessary links or units. The advantages are:

- the cost of a NN can be reduced (runtime, memory and cost for hardware implementation)
- the generalization of the NN can be improved
- unnecessary input units can be pruned in order to give evidence of the relevance of input values

To decide which links or units are less important and thus susceptible to removal, it is necessary to assess the relative importance of weight: this process is called *saliency*. The most common pruning algorithms perform training and pruning of a NN alternately, according to the following algorithm:

1. choose a reasonable NN architecture
2. train the NN with any learning function into a minimum of the error
3. compute the saliency (relevance for the performance of the network) of each element (link or unit respectively)
4. prune the element with the smallest saliency

5. retrain the NN into a (new) minimum

6. if the net error does not exceed a definite value, repeat the procedure from step (3)

There are different approaches to determine the saliency of an element. The Magnitude Based Pruning (MBP) is the simplest weight pruning algorithm [17] which is based on deleting links with small saliency (since this deletion will have the least effect on the training error). After a reasonable initial training, the link having the smallest magnitude value is removed (thus the saliency of a link is just the absolute size of its weight). The network is then retrained and the process is iteratively repeated until the training error reaches a specified limit. Although this method is very simple, it rarely yields worse results than the more sophisticated algorithms such as Optimal Brain Damage (OBD), Optimal Brain Surgeon (OBS) or Skeletonization [15][18].

Based on the validation set, the pruning processing removed neurons following two distinct goals:

- to reach the minimum of the classification error;
- to reach the minimum number of connections taking into account a maximum decrease of the classification accuracy of about 10% with respect of the full connected network. Note this task is only academic and it does not regard the contest.

The functions and the parameters used to prune the full connected neural network are listed in Table 3.3.

Pruning Function	Magnitude Based
Maximum error increase in %	0.0
Accepted Error	0.0
Learning Function	Scaling Conjugate Gradient
Learning cycles for first training	10
Learning cycles for re-training	10
Minimum Error to stop	1

Table 3.3: Parameters used to prune the Neural Network with the goal of reaching the minimum of the classification error.

Concerning the first task, the full connected neural network, consisting in 43800 connections (Overall Accuracy 86.4%), has been pruned reaching the minimum of the classification error with 43657 connections (Overall Accuracy 89.7%) as shown in Figure 3.2. Therefore, less than 0.5% of the initial connections were removed for an improvement in terms of classification accuracy of 3.3%.

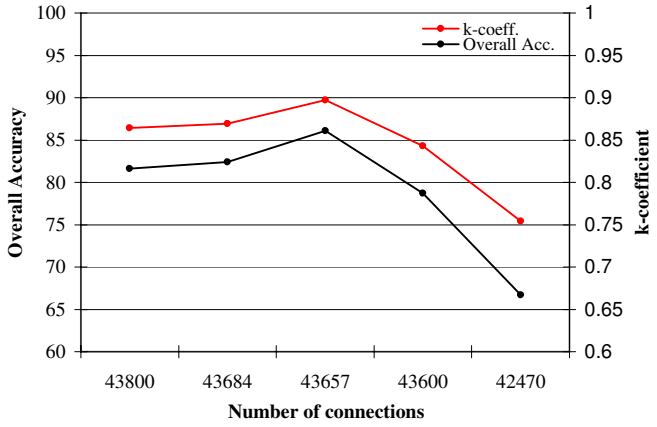

Figure 3.2: Accuracies of the different topologies obtained after the pruning process. The minimum of the classification error was reached by removing less than 0.5% of the initial number of connections, resulting in 43657 links.

Successively, the pruning continued to further reduce the number of connections expecting a decrease of the classification accuracy. As shown in Table 3.4, which lists the accuracy details of the different network topologies, 57 connections were sufficient to decrease the classification accuracy from 89.7% to 84.3%. The decrease of the classification accuracy of about 10% with respect of the full connected network was reached with 42470 connections (3.0% of the initial connections).

Net ID	Number of Connections	Overall Accuracy (%)	*k*-coeff
Full Connected	43800	86.4	0.816
net1	43684	86.9	0.824
net2	43657	89.7	0.861
net3	43600	84.3	0.787
net4	42470	75.4	0.667

Table 3.4: Accuracy details of the of the different topologies.

These accuracies were based on the validation set obtained by visual inspection of the scene and were only used to have a rough information on the selection of the most suitable network topology.

Chapter 4

Postprocessing

Classified images often suffer from a lack of spatial coherency which results in speckle or holes in homogeneous areas. Therefore, this noise phenomenon appears as many isolated pixels or small groups of pixels whose classifications are different from those of their neighbors [19].

Post-classification processing techniques can be applied to further increase the classification accuracy. This is often achieved by analyzing the neighborhood for each pixel and removing the isolated pixels (*sieve* process), and then merging the small groups of pixels together to make more continuous and coherent units (*clump* process) [20].

Post-classification data processing procedures, such as *sieve* and *clump*, were used here to reduce the effect of isolated pixels in land cover maps. As a result, the minimum mapping unit was selected removing all regions smaller than the designated size which was 142 pixels. Therefore, even though the smaller clusters may have been correctly classified, they were considered not reliable.

Bibliography

[1] Data Fusion Contest, *http://tlclab.unipv.it/dftc/*, last visited July, 2007.

[2] F. DellAcqua, P. Gamba, G. Lisini, "Improvements to Urban Area Characterization Using Multitemporal and Multiangle SAR Images", *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1996-2004, Sept. 2003.

[3] J. A. Richards, X. Jia, "Remote sensing digital image analysis", Spinger, Berlin, 2006.

[4] C. Bishop, "Neural Networks for pattern recognition", Oxford University Press, New York, 1995.

[5] Y.H. Hu, J.N. Hwang, "Handbook of Neural Network Signal Processing", CRC press LLC, 2002.

[6] N.K. Kasabov, "Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering", The MIT press, Cambridge, Massachusetts, 1996.

[7] V. Kecman, "Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models", The MIT press, Cambridge, Massachusetts, 2001.

[8] R. M. Hristev, "The Artificial Neural Networks", available at http://www.igi.tugraz.at/lehre/NNA/WS03/downloads/ANN.pdf

[9] M. A. Arbib, "The handbook of brain theory and neural networks", The MIT press, Cambridge, Massachusetts, 2003.

[10] S. K. Pal, S. Mitra, "Multilayer perceptron, fuzzy sets, and classification", *IEEE Trans. on Neural Networks*, vol. 3, no. 5, pp. 683-697, September 1992.

[11] F. Del Frate, A. Petrocchi, J. Lichtenegger, G. Calabresi, "Neural networks for oil spill detection using ERS-SAR data", *IEEE Trans. Geosci. Remote Sens.*, Vol. 38, No. 5, Sept. 2000.

[12] F. Del Frate, F. Pacifici, G. Schiavon and C. Solimini, "Use of Neural Networks for Automatic Classification From High-Resolution Images", *IEEE Trans. Geosci. Remote Sens.*, Vol. 45, No. 4, Apr. 2007.

[13] F. Pacifici, F. Del Frate, D. Solimini, "Urban land cover in Rome, Italy, monitored by single-parameter multitemporal SAR imagery", to be submitted on *IEEE Trans. Geosci. Remote Sens.*.

[14] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, vol. 6, issue 4, pp. 525-533, 1993.

[15] A. Zell *et al.*, "SNNS, Stuttgart Neural Network Simulator, user manual v. 4.2", University of Stuttgart, Stuttgart, Germany

[16] B. Hassibi, D.G. Stork and G. Wolf, "Optimal brain surgeon and general network pruning", in *Proceedings of the 1993 IEEE International Conference on Neural Networks*, pp.293-300, San Francisco, CA, Apr. 1993.

[17] T. Kavzoglu, P. M. Mather, "Pruning artificial neural networks: an example using land cover classification of multi-sensor images", in *Int. J. Remote Sensing*, vol. 20, no. 14, pp. 2787-2803, 1999.

[18] T. Ragg, H. Braun, H. Landsberg, "A Comparative Study of Neural Network Optimization Techniques", in *Proceedings of the International Conference on Adaptative and Natural Computing Algorithms - ICNNGA 97*, Norwich, UK, 1997.

[19] H. Huang, J. J. Legarsky, S. Gudimetla, C. H. Davis, " Post-classification smoothing of digital classification map of St. Louis, Missouri", in *Proceedings of the International Geoscience and Remote Sensing Symposium - IGARSS 2004*, vol. 5, pp. 3039-3041, Sept. 2004.

[20] R. A. Schowengerdt, "Techniues for image processing and classification in remote sensing", *Academic Press*, New York, 1983.