CLASSIFICATION OF MULTISPECTRAL REMOTE-SENSING IMAGES BY NEURAL NETWORKS

F. Roli$^{(1)}$, S.B. Serpico$^{(2)}$, L. Bruzzone$^{(2)}$, and G. Vernazza$^{(1)}$
$^{(1)}$ Dept. of Electrical and Electronic Eng., University of Cagliari
Piazza d’Armi, I-09123, Cagliari, Italy
tel: +39 70 6755897; fax: +39 70 6755900
e-mail: vernazza@elettro1.unica.it
$^{(2)}$ Dept. of Biophysical and Electronic Eng., University of Genoa
Via All’ Opera Pia, 11A, 16145, Genova, Italy
tel: +39 10 3532752; fax: +39 10 3532134
e-mail: vulcano@dibe.unige.it

ABSTRACT

This paper addresses the classification of multispectral remote-sensing images by the neural-network approach. In particular, an experimental comparison on the performances provided by different neural models for classifying multisensor remote-sensing data is reported. Four neural classifiers are considered in the comparison: the Multilayer Perceptron, Probabilistic Neural Networks, Radial Basis Function networks and a kind of Structured Neural Networks.

1 INTRODUCTION

Supervised classification of multispectral remote-sensing images has been usually performed by statistical parametric methods [1]. These methods exploit a model to represent the statistical distribution of data (e.g., the gaussian model). However, the parametric classifiers are not suited for classification of multisensor images as the data acquired by different sensors often require different statistical models. Then, non-parametric classifiers are more appropriate as they do not rely on a priori models. In the context of the non-parametric classifiers for multisensor remote-sensing images, neural networks are among the most promising one [2-17]. Neural networks exhibit interesting properties, such as learning capability, fast classification time, and intrinsic parallelism [5]. On the other hand, depending on the model adopted, their use also gives rise to some problems, such as the lack of precise rules for defining the network architecture and a long training time.

In this paper we report the results of an experimental analysis on the use of different neural models for the classification of multisensor remote-sensing data. In particular we consider four neural classifiers: the Multilayer Perceptron (MLP), Probabilistic Neural Networks (PNNs), Radial Basis Function (RBF) networks, and a kind of Structured Neural Networks (SNNs). A brief description of both data set and architectures of each neural model used in the experiments is reported. Results are compared and discussed in order to point out the peculiarity of the different neural classifiers.

2 DATA SET DESCRIPTION

The considered data set referred to an agricultural area near the village of Feltwell (UK) [15]. We selected a section (250 x 350 pixels) of the scene acquired with two imaging sensors installed on an airplane: a Daedalus 1268 Airborne Thematic Mapper (ATM) scanner and a PLC-band, fully polarimetric, NASA/JPL SAR sensor. Images were registered by using the radar image as reference. The ground truth was used to prepare a reference map to assess the classification accuracy. We considered the five numerically most representative agricultural classes (55657 pixels). Agricultural fields were randomly subdivided into two disjoint sets; 5124 training pixels were taken from the fields of one set, 5820 test pixels from the fields of the other set. Fifteen channels were selected to form an "attribute vector" for each pixel. We selected the six ATM channels corresponding to TM channels in the visible and in the infrared spectrum, and the nine SAR channels in the HH, HV, and VV polarizations. The noise affecting the intensity values was reduced by applying a simple running mean filtering to both the ATM (5x5 window) and the SAR (9x9 window) channels.

3 EXPERIMENTAL RESULTS

3.1 Multilayer Perceptron

For the MLPs [18,19] used in our experiments, we considered in the input layer a number of neurons equal to the number of attributes that characterize each pixel to be classified (i.e., 15). The output layer was composed by as many neurons as the number of data classes (i.e., 5). The number of hidden layers and the number of neurons per layer (i.e., the number of weights and biases) were selected by using a simplified version of the rule proposed by Baum and Haussler [20]. Taking into account this indication, we experimented five different fully connected architectures, with one or two hidden layers and various numbers of units per layer.

Each neuron had a sigmoidal activation function defined as:

$$S(net_i) = \left(1 + e^{-net_i}\right)^{-1} \qquad (1)$$

The classification of each pixel was carried out by assigning each pixel to the class corresponding to the output unit with the highest output. A Winner-Takes-All (WTA) decision block was used to this end, with no threshold margin between the maximum output and the other outputs.

Training was carried out by the error backpropagation learning procedure [21] by using a value of "learning rate" η=0.01. In Table 1 the results provided by the considered MLPs on both the training and the test sets are shown.

Table 1: Percentages of correctly classified pixels by using MLPs. Different architectures with one or two hidden layers and various numbers of units per layer were used. The "learning rates" η=0.01 was used.

Architecture	Training set	Test set
15-15-8-5	98.7%	79.9%
15-7-5-5	96.9%	82.3%
15-15-5	97.3%	87.9%
15-30-5	97.2%	88.2%
15-8-5	96.5%	89.6%
Mean value	97.3%	85.6%

3.2 Probabilistic Neural Networks

The PNNs [22] we used in our experiments were composed of an input layer, one hidden layer and an output layer. As in our MLPs, the number of neurons in the input layer was equal to the number of attributes (i.e., 15). The hidden layer had as many neurons as the number of training patterns (i.e., 5124); each neuron had a Gaussian activation function ("kernel") centred on the attribute vector of each training pixel. The output layer had as many neurons as the number of data classes considered (i.e., 5); the activation function of each neuron computed the sum of inputs. Analogously to MLPs, classification was carried out by using the WTA decision rule to identify the most probable class.

Training in PNNs consisted in the selection of the Gaussian kernel width by trials with different values of the “smoothing parameters” σ. As σ we adopted:

$$\sigma = c\left(\frac{v}{M^{1/p}}\right)^{1/2} \qquad (2)$$

where v is the average quadratic distance, in the attribute space, between vectors of training pixels and mean vectors of the related classes; M is the number of training pixels; and p is the number of attributes. The optimal value of coefficient c was obtained experimentally (i.e., c=0.1) as a result of the training phase. The classification accuracies on the test set obtained by using the PNNs are summarized, class by class, in Table 2.

3.3 Radial Basis Function networks

The RBF networks [19,23] we considered in our experimental analysis had an input layer, one hidden layer, and an output layer. As for previously described neural classifiers, the input layer had as many neurons as the number of attributes (i.e., 15) and the output layer had as many neurons as the number of classes (i.e., 5). The neurons of the input layer just propagated the attribute values; the neurons of the output layer had a linear activation function. With regard to the number of neurons of the hidden layer, it must be greater than or equal to the number of natural clusters of input data. In our experiments, we performed trials with different numbers of neurons. Each neuron of the hidden layer had a gaussian activation function characterized by both the vector of the centre and the distance scaling parameter which determines over what distance in the input space the neuron have a significant influence.

The training of RBF networks was performed in two step: in the first step, we used an unsupervised clustering algorithm to determine the parameters of the hidden layer; in the second step, the supervised least-square solution to the linear output weights were computed [23]. Unsupervised clustering was performed by the k-means clustering algorithm [23]. Each cluster centre corresponded to a neuron of the hidden layer. The distance scaling parameters of units were computed by the P-nearest neighbour heuristic [23].

The best overall accuracy on the test set (i.e., 88.8%) was obtained with a number of neurons in the hidden layer equal to 200 (the classification accuracies, class by class, provided by this architecture are shown in table 2).

3.4 Structured Neural Networks

The SNNs we utilized in our comparative investigation are MLPs with structured architectures tailored to multisensor classification problems [15]. Their main feature is the possibility of allowing a quantitative and detailed interpretation of the network operation.

SNNs have one-net-one-class architectures, that is, a separate network is devoted to each class. Each Class-Related Network (CRN) has a tree-like architecture with an input layer, two hidden layers, and an output layer. The input layer has as many neurons as the number of sensor channels. Each neuron of the first hidden layer applies a constraint on the intensity in a sensor channel; a neuron is included in the second hidden layer for each sensor, which provides the output of a sensor-related subnet. The output layer is composed by one neuron which provides an estimate of the probability that an input pixel belongs to the class associated to the considered class-related network. A WTA decision block makes the final decision about classification. SNNs we utilized in our experiments were trained by the same learning procedure adopted for MLPs. Table 2 reports the classification results provided by the SNNs on the test set.

3.5 Comparison

The classification accuracies provided by the different neural classifiers considered in our experiments were compared. The best classification accuracy was obtained by the MLP with the architecture 15-8-5: it performed slightly better than the RBF networks and the PNNs (i.e., there was a difference of about 1% on the test set). The RBF networks provided accuracies similar to those of the PNNs and slightly better than those of the SNNs (there was a difference of about 2% on the test set). On the other hand, the SNNs yielded accuracies on the test set that were slightly better than the mean accuracy computed for the set of fully connected MLPs considered. In addition, the SNNs had the advantage that they allow the interpretation of the network behaviour.

The PNNs had the advantage to have a very fast learning time as the learning was limited to search the optimal width of the kernel function (search in one dimensional space). However, the size of the network and the computational time grow proportionally to the number of training pattern (i.e., the number of neurons of the hidden layer). The RBF networks overcame this problem of the PNNs by choosing a fixed number of neurons in the hidden layer and performing clustering to determine the parameters of each neuron. The RBF networks were also faster in the training phase as the training of the hidden layer and of the output layer was decoupled. However, according to the results of other researchers, we verified a critical dependence of the accuracies provided by the RBF networks on the number of neurons selected in the hidden layer. Then, as in the MLPs, trials with different architecture were necessary in order to obtain a configuration that provides good accuracies. On the contrary, the SNNs and the PNNs do not required trials with different architectures.

Table 2: Percentages of correctly classified test pixels by using PNNs, RBF networks, and PNNs .

Class	PNNs	RBF networks	SNNs
Sugar beets	97.84%	98.63%	99.51%
Stubble	82.42%	87.38%	85.92%
Bare soil	79.64%	77.48%	79.28%
Potatoes	81.79%	80.99%	74.21%
Carrots	89.35%	83.66%	75.18%
Overall accuracy	88.62%	88.8%	86.49%

Acknowledgments

This research was partially conducted within the framework of the cooperative research project “Integrated assessment of environmental degradation connected with forest fires in European areas”, funded by the ECC (Environment Programme II, Contract N. EV5V CT94 0481), and partially within the framework of the Italian project “Sviluppo di metodi integrati di classificazione agroecologica tramite dati di telerilevamento per la gestione delle risorse naturali”, funded by Italian Space Agency. Both support are gratefully acknowledged.

References

[1] P.H.Swain and S.M.Davis, *Remote Sensing: The Quantitative Approach.* New York: McGraw-Hill, 1978.

[2] S.E.Decatur, “Applications of neural networks to terrain classification”, *Proc. Int. Joint Conf. on Neural Networks 89*, Washington D.C., vol. 1, pp. 283-288,1989.

[3] J.A.Benediktsson, P.H.Swain, O.K.Ersoy, “Neural Network Approaches versus Statistical Methods in Classification of Multisource Remote-Sensing Data,” *IEEE Trans. on Geosc. and Remote Sens.*, vol.28, n.4, pp.540-552, July 1990.

[4] J.A.Benediktsson, O.K. Ersoy, and P.H. Swain, "A Consensual Neural Network", *Proc. of the 1991 Int. Geoscience and Remote Sensing Symp., IGARSS 91*, pp. 2219-2222, Espoo, Finland, June 3-6, 1991.

[5] H.Bischof, W.Schneider, and A.J.Pinz, "Multispectral classification of Landsat-images using neural networks", *IEEE Trans. on Geosc. and Remote Sens.*, vol. 30, n.3, pp. 482-490, May 1992.

[6] M.R.Azimi-Sadjadi, S.Ghaloum, and R.Zoughi, “Terrain Classification in SAR Images Using Principal Components Analysis and Neural Networks”, *IEEE Trans. on Geosc. and Remote Sens.*, vol. 31, n. 2, pp. 511-515, March 1993.

[7] M.S.Dawson and A.K.Fung, "Neural networks and their applications to parameter retrieval and classification", *IEEE Geoscience and Remote Sensing Society Newsletter*, pp. 6-14, September 1993.

[8] M.S.Dawson, A.K.Fung, and M.T.Manry, "Sea ice classification using fast learning neural networks", *Proc. of IGARSS '92*, pp. 1070-1071, Houston, Texas, May 1992.

[9] L.Mascarilla, E.H.Zahzah, J.Desachy, "Neural networks classifiers based on geocoded data and multispectral images for satellite image interpretation", *Proc. 5th Int. Conf., CAIP '93*, pp. 830-837, Budapest, Hungary, Sept. 1993.

[10] O.K.Ersoy, and D.Hong, "Parallel, Self-Organizing, Hierarchical Neural Networks, *IEEE Trans. on Neural Networks*, vol. 1, n. 2, pp.167-178, June 1990.

[11] P.D.Heermann and N.Khazenie, "Classification of multispectral remote sensing data using a back-propagation neural network", *IEEE Trans. on Geosc. and Remote Sens.*, vol. 30, n. 1, pp. 81-88, January 1992.

[12] J.N.Hwang, S.R.Lay, and R.Kiang, “Robust Construction Neural Networks for Classification of Remotely Sensed Data”, *Proc. World Congress on Neural Networks, WCNN’ 94*, , Vol. 4, pp. 580-584, Portland, Oregon, USA, July 11-15, 1993.

[13] J.Lee, R.C.Weger, S.K.Sengupta, R.M.Welch, "A neural network approach to cloud classification", *IEEE Trans. on Geoscience and Remote Sensing*, vol. 28, n. 5, pp.846-855, September 1991.

[14] Y.Salu, and J.Tilton, “Classification of multispectral image data by the Binary Diamond neural network and by nonparametric, pixel-by-pixel methods”,

IEEE Trans. on Geosc. and Remote Sens., vol. 31, n. 3, pp. 606-617, May 1993.

[15] S.B.Serpico and F.Roli, "Classification of multisensor remote-sensing images by structured neural networks", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, n. 3, pp.562-578, May 1995.

[16] J.D.Paola and R.A.Schowengerdt, "A Detailed Comparison of Backpropagation Neural Network and Maximum-Likelihood Classifiers for Urban land Use Classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, n. 4, pp. 981-996, July 1995.

[17] J.A.Benediktsson, J.R.Sveinsson, and K. Arnason, "Classification and Feature extraction of AVIRIS Data", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, n. 5, pp. 1194-1205, September 1995.

[18] D.E.Rumelhart, J.L.McClelland, and the PDP Research Group, *Parallel Distributed Processing*, vol. 1-2, Cambridge, MA: MIT Press, 1986.

[19] D.R.Hush and B.G.Horne, "Progress in supervised neural networks", *IEEE Signal Processing Magazine*, vol. 10, n. 1, pp. 8-39, 1993.

[20] E.B.Baum and D.Haussler, "What Size Network Gives Valid Generalization", *Neural Computation*, vol. 1, pp. 151-160, 1989.

[21] J.Hertz, A.Krogh, R.G.Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley Pub. Co., The Advance Book Program, 1991.

[22] D.F.Specht, "Probabilistic Neural Networks", *Neural Networks*, Vol.3, pp. 109-118, 1990.

[23] J.A.Leonard and M.A.Kramer, "Radial Basis Function Networks for Classifying Process Faults", *IEEE Control Systems Magazine*, pp. 31-38, April 1991.