Fuzzy classification of pixels using neural networks

Theo E. Schouten

Katholieke Universiteit Nijmegen, Computing Science Institute,
Toernooiveld 1, 6525 ED Nijmegen The Netherlands

ABSTRACT

Imaging spectrometers acquire images in many narrow spectral bands. Because of the limited spatial resolution the measured spectrum of a pixel is often a composition of a number of basic spectra. The purpose of fuzzy classification is to determine the presence and abundance of the basic spectra in a measured spectrum. Previous work demonstrated that a neural network could perform fuzzy classification. In this paper we study a more realistic situation of 10 basic spectra using 12 band airborne data and of 16 basic spectra using 6 band LANDSAT data. Available for this study were images and sets of pixels, which have been classified by inspection on the ground. For the LANDSAT case the set was not very pure and not very large. A method to purify and to expand data sets using image processing methods was therefore developed. Mixed pixels training and testing sets were generated from each original and generated set using a linear mixture model, where a mixed pixel could have a contribution from up to three classes. For each of the training sets a 1 hidden layer backpropagation neural network was trained to do the fuzzy classification. Testing the networks showed that they performed up to 20% better than the developed AnaML method, which is a combination of two classical methods.

Keywords: Remote Sensing, Spectral Micture Analysis, Neural Networks, Fuzzy Classification

1. INTRODUCTION

Measured spectra are often a composition of a number of basic spectra. In remote sensing, imaging spectrometers[1,2] acquire images simultaneously in many narrow spectral bands from visible to infrared. Due to the limited spatial resolution, a pixel often contains more than one ground cover e.g. grass, mais and a water filled ditch in between. The purpose of fuzzy classification or mixed pixel analysis is to determine the fractions of the ground cover categories (called endmembers) present in each pixel.

Usually a statistical linear mixture model[3] is used to model the composition, the signal received for a pixel in band i is assumed to be:

$$s_i = \sum_{j=1}^{c} f_j M_{ij} + e_i, \quad i = 1, 2, \ldots, n \tag{1}$$

Here n is the total number of bands, c is the total number of end-members in the pixel, M_{ij} is the reflectance of the jth end-member in the ith band, f_j is the fraction of the pixel covered by end-member j and e_i models the error in the ith band. On the j's there are the sum-to-1 and the positivity constraints:

$$\sum_{j=1}^{c} f_j = 1 \quad \text{and} \quad 0 \leq f_j \quad j = 1, \ldots, c \tag{2}$$

and they complicate the proces of finding a solution for equation (1).

The simple least squares solution $\hat{f} = (M^T M)^{-1} M^T s$, which minimises the cost function $\|e\|^2$, in general does not satisfy the constraints. We defined a cost function[4] which includes also deviations from the constraints:

$$\|e\|^2 + \sum_{j=1}^{c} (if\ f_j < 0\ then\ f_j^2\ else\ 0) + (\sum_{j=1}^{c} f_j - 1)^2 \tag{3}$$

Further author information:
E-mail: ths@cs.kun.nl
Home Page: http://www.cs.kun.nl/~ths

and minimize it with a fast iterative method. The method is fast in the sense that the time complexity of each iteration is proportional to c and not to n.

In other classical methods[5] the error is modelled, assuming it to have a multivariate normal distribution with zero mean and covariance matrix N_f determined by the covariance matrices of the endmembers: $N_f = \sum_{j=1}^{c} f_j N_j$. Following the maximum likelihood approach a solution $\hat{f}$ is searched for which minimizes: $e^T N_f^{-1} e + ln|N_f|$ subject to the constraints. The searching may involve iterative minimising methods or exhaustive searches through the f space.

When all N_j are equal to say N analytical solutions can be derived. Ignoring the constraints on f gives: $\hat{f}_0 = UM^T N^{-1} s \;\; with \;\; U^{-1} = M^T N^{-1} M$. Imposing the sum-to-one constraint gives: $\hat{f}_1 = \hat{f}_0 + \alpha(1 - I^T \hat{f}_0)UI \;\; with \; \alpha = (I^T UI)^{-1}$, I is the identity vector. The positivity constraint is then usually taken into account by setting negative $\hat{f}_{1j}$'s to zero and then renormalising their sum to 1 to obtain the final $\hat{f}$ solution. Fortunately N can be approximated[3] by $N = \frac{1}{c} \sum_{i=1}^{c} N_i$ without much accuracy loss when the distance between the end-members is large compared to the variations in the end-members.

To model the local variations in the end-member spectra we developed physical linear mixture models[6,7]:

$$s_i = \sum_{j=1}^{c} f_j (M_{ij} + E_{ij}), \quad i = 1, 2, \ldots, n \tag{4}$$

where E can be partly derived from neighbouring pure pixels using image processing methods. It was shown[7] that using these methods the area estimation of agricultural fields could be improved by more than 30%.

In previous studies[8,9] we showed that a backpropagation neural network[10] is able to perform fuzzy classification of mixed pixels with at least as good accuracy as maximum likelihood based methods obtain. Other studies[11,12] indicate the same.

In this paper we study a more realistic situation of 10 classes using 12 band airborne data and of 16 clases using 6 band LANDSAT data, where a mixed pixel could have a contribution from up to three classes.

In Sect. 2 the used neural neural network and their implementaions are described. The used images and data sets are described in Sect. 3. As one of the data sets was not very pure, a purification method was developed. In Sect. 4 this method and its results are described. The way in which the mixed pixels were generated is described in Sect. 5. To have a method to compare the neural network with, the AnaML method was developed which is a combination of the above given analytical and maximum likelihood methods. It is described in Sect. 6. The results of the neural networsk in comparison to the AnaML results are given and evaluated in Sect. 7, also expressions for evaluating the time needed for both methods is given there. The final conclusions are given in the last section.

2. NEURAL NETWORK

A multi-layer backpropagation neural network was used. The simulations were performed with our parallel network simulator program which is implemented in C using PVM for communication, allowing it to be ported onto a wide range of parallel and distributed systems. Currently a 16 node PowerXplorer is used as execution platform. Based on our previous experiences with neural networks for fuzzy classification and other appplications, the following strategies and parameters were used.

The number of input neurons is equal to the number of bands (n) of the used spectrometer and each input neuron i receives s_i, linearly scaled to the range [-1.0,. . .,1.0], as its input value. The number of output neurons is equal to the number of endmembers (c), during training the target value of each output neuron i is set to the then known f_i, lineary scaled to the range [-1.0,. . .,1.0]. The transfer function for a neuron i was set to $tanh(\beta_i a_i)$ for the neurons of the hidden layers and to the linear function $\beta_i a_i$ for the neurons of the output layer. Here a_i is the activation of each each neuron calculated from its input values and the weight and threshold values. The value β_i is the steepness of each neuron which is also learned[13] during training. For this study the learning of the steepness decreased the number of training epochs needed for reaching a certain accuracy by a factor 2 to 4 while increasing the time needed per epoch by only 3 to 8%. In Fig. 1 this is shown for a 12-64-10 * network with about 120000 training mixed pixels.

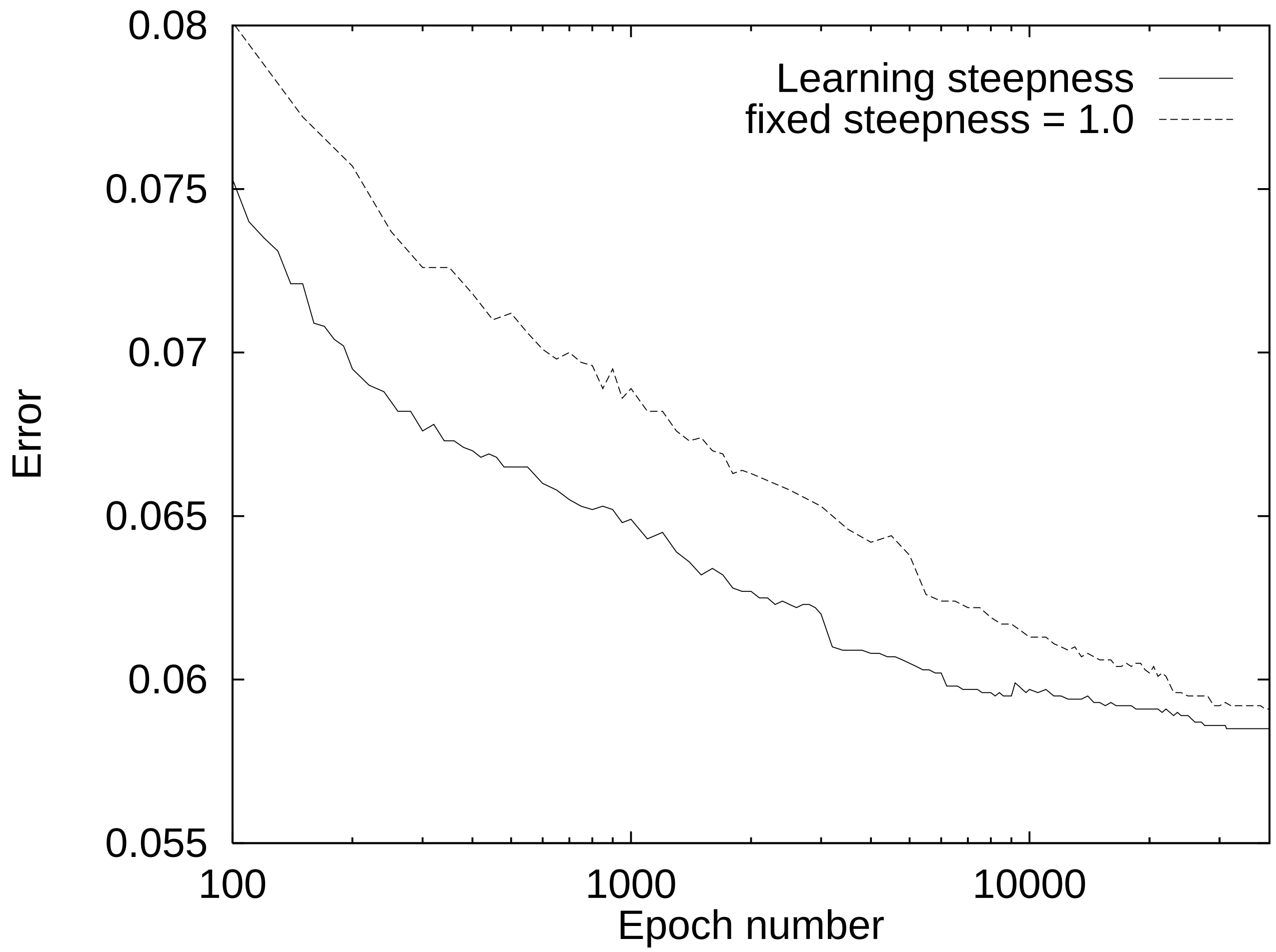


Figure 1. Influence of learning steepness on the neural network error as function of epoch

The observed decreases in epochs to reach the errors 0.075, 0.07, 0.065 resp. 0.06 are the factors 2.0, 3.2, 3.3 resp. 3.5 while the increase in training time per epoch is 3.6% (from 13.6s to 14.1s).

The learning rate was varied by hand during training, starting with a high rate (typically 0.2) and decreasing it following the sequence (0.1, 0.05, 0.02, 0.01, etc.) whenever the error starts oscillating or the decrease in error is too small. This learning rate is divided by the number of training patterns used for updating. The momentum rate was kept at a constant value of 0.8.

Training was done in a semi-epoch mode: weights, thresholds and steepnesses were updated several times (typically 100) during each epoch. This semi-epoch mode approximates on-line learning which in our experience usually produces better results than full epoch learning when there are a large number of noisy training samples. On a sequential machine on-line learning is usually faster than epoch learning: the weights with best results on the test data are quicker obtained. On parallel machines different paralellization strategies[14] can be used for on-line and epoch learning, making the latter faster as less communication operations are needed. For epoch and semi-epoch training dataset decomposition[15] is used where each processor has its own copy of the full neural network. The training patterns are equally divided over a number of slave processors and the master processor gathers and broadcast results and calculates the update of weights, thresholds and steepnesses. This produces a rather linear speedup of training time proportional to the number of slave processors used: for a 6-64-16 network with 120000 training patterns epoch learning takes 8.7s using 7 slaves and 4.3 s using 15 slaves, for 1% semi-epoch learning the times are 15.1s and 7.5s. For on-line learning network decomposition[15] could be used where the neurons of a network are

[*] The size of a network is notated as i-h1-[h2]-o, with i the number of input neurons, o the number of output neurons and hn the number of neurons in the hidden layer(s).

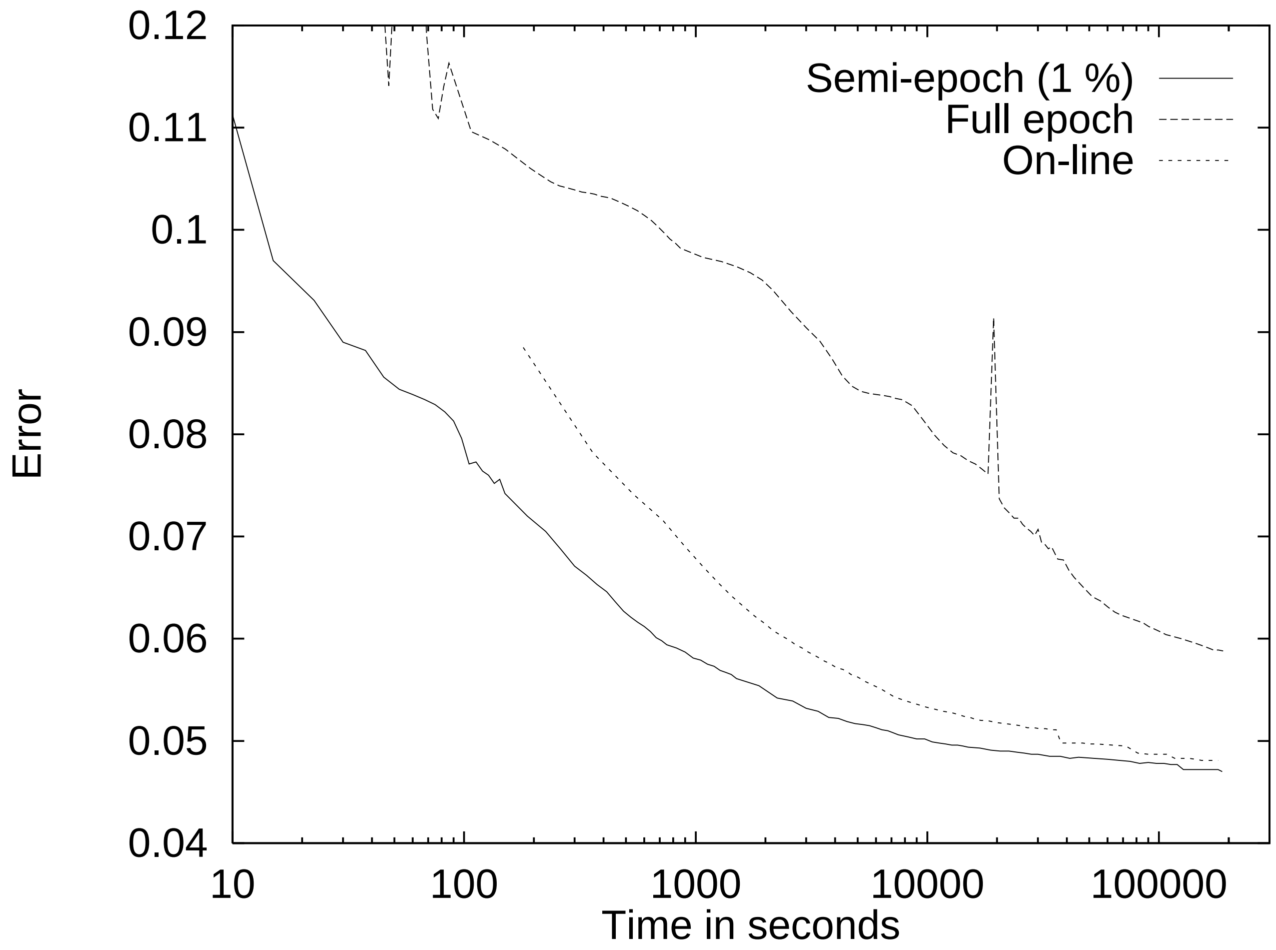


Figure 2. Neural network error as function of time

divided over the slave processors. But for the sizes of the networks used in this study this gives a too large ratio of communication time to calculation time on the used execution platform. For the above mentioned case on-line learning with 1 slave processor takes 533s per epoch, while using 3 slaves increases the time to 691s. For comparison the time for a sequential version of on-line learning using 1 node of our execution platform is 180s.

In Fig. 2 the training error is shown as function of time for epoch and 1% semi-epoch learning using 15 slave processors and for on-line learning in sequential mode. A 6-64-16 network was used with about 120000 mixed pixels as training pattern. As expected the obtained errors for on-line and 1% semi-epoch learning are much lower than for full epoch learning. The 1% semi-epoch learning gives a smaller error than on-line learning, probably there is too much variation in the gradients when the update is done after each pattern. The error on the test set appeared to be within 2% of the eroor on the training set, no overlearning was observed.

Training the 1 hidden layer networks went rather smoothly, not much fiddling around with parameter settings was needed and the obtained accuracy did not very much depend on the number of hidden neurons. Networks with 2 hidden layers were tried in the hope to find a configuration which gives the same accuracy as the best 1 hidden layer network but which uses less neurons and weights thus decreasing the time needed for the fuzzy classification of mixed pixels. Training these networks appeared to be very difficult, no good results were obtained.

The above observations on speed, accuracy and ease of training are off course dependent on many factors: the type of problem and the structure of its error surface, the size of the network, the number of training patterns and the ratio of communication to calculation speed of the used execution platform.

The used execution platform is a 16 node PowerXplorer parallel machine, each node consists of a PowerPC-601 80MHz processor with 32MB memory and a T805 30MHz processor providing 4 20Mbit/s bidirectional links with 4 neighbouring nodes. The calculation capacities of a node are about 1.9 SPECint95 and 2.3 SPECfp95, for comparison a Pentium-II 400MHz is rated at about 15.8 SPECint95 and 12.4 SPECfp95 while a SUN Ultra5 333MHz machine reaches 14.1 SPECint95 and 18.3 SPECfp95. The 16 nodes are arranged in a 4 by 4 grid which can be divided in upto 4 partitions for 4 simultanious users.

The code of the simulator on the PowerXplorer was optimized for speed regarding calculation time (unfolding of loops, linear approximation of the *tanh* function, etc.) and communication time (broadcast and gathering operations depending on the topology, joining of messages to reduce setup time, etc.). As more than 80% of the effort of developing a neural network simulator goes into auxillary code (user interface, reading and writing of parameters, networks and patterns), this optimization is a relative small effort which pays off because training times are long, in the order of days.

3. IMAGES AND DATA SETS

Images from two spectrometers were available, each having a set of pixels which were classified by ground survey. From a 6 band LANDSAT image of Portugal a set of 12505 pixels classified into 16 classes[16,17] was available. The number of pixels per class varied from 172 for "sand" to 3331 for "grass". Per class about 2/3 of the pixels were randomly selected to be used as basis for training the neural networks (the L-TRAIN set) and the remaining pixels were used for testing the performance of trained nets (the L-TEST set).

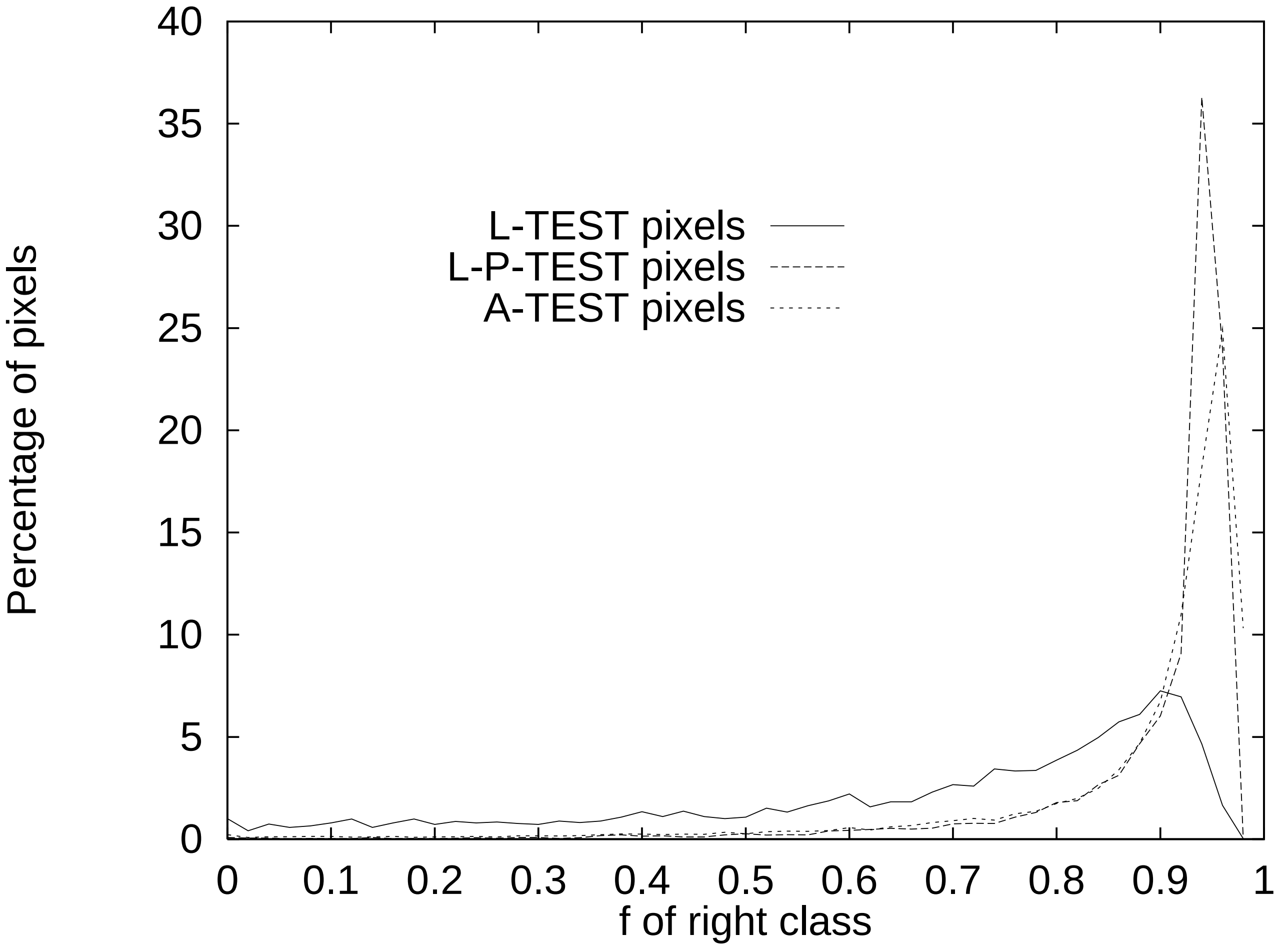


Figure 3. Distribution of f's of the right class

To test the purity of the ground survey data, the pure L-TRAIN pixels were used to train a 6-64-16 network. After training, the L-TEST pixels were passed through the network and the distribution of the obtained f's of the expected classes is shown in Fig. 3. Instead of the expected sharp peak at little below 1, a broad peak is observed, indicating that the ground survey data contain many mixed pixels. In Table 1 the percentage of pixels having their maximum f in one of the neural network output classes is given for each input class. Most of the classes show sizable contribution of other classes, especially "weeds", "garigue", "grass" and "deciduous forest".

class		nn output class with maximum f															
name	number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
soil	1	71	1	2	5	3	6	2	3	0	0	0	3	1	1	3	0
fresh water	2	2	77	0	2	1	2	3	4	4	0	2	1	1	0	1	0
vineyard	3	1	0	75	3	5	3	3	1	0	0	0	2	2	2	1	0
weeds	4	3	0	2	54	5	14	8	2	1	0	0	3	4	1	2	0
garigue	5	2	0	3	5	60	3	12	7	1	0	0	2	2	1	1	0
grass	6	2	0	1	7	3	78	2	1	0	0	0	3	1	1	1	0
dec. forest	7	2	1	3	10	20	4	46	6	1	0	0	2	3	1	1	0
con. forest	8	3	2	1	7	18	4	9	48	1	0	0	2	2	0	2	0
rice	9	2	2	1	1	2	2	2	3	80	0	1	1	1	0	1	0
sea water	10	0	2	0	0	1	2	0	0	0	91	2	0	1	0	0	0
estuary	11	1	1	0	1	1	1	1	0	0	1	92	0	1	0	0	0
wheat	12	5	0	5	8	2	15	4	2	1	0	0	49	3	2	2	0
barley	13	2	0	1	16	9	6	6	1	0	0	0	3	50	1	2	0
maize	14	2	0	10	4	3	5	4	1	1	1	1	4	3	59	1	0
tiled	15	4	1	4	8	3	4	3	3	0	0	0	3	3	1	63	0
sand	16	1	0	1	2	1	3	2	2	0	0	0	1	1	1	2	83

Table 1. Neural network classification of L-TEST, the average of the diagonal terms is 67.9%

class		nn output class with maximum f															
name	number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
soil	1	86	0	0	1	1	2	2	1	0	0	0	2	1	0	4	0
fresh water	2	0	94	0	0	0	0	0	1	0	1	3	0	0	0	0	0
vineyard	3	1	0	89	2	0	3	1	0	0	0	0	1	2	1	1	0
weeds	4	1	0	0	76	1	8	2	0	0	0	0	1	1	0	1	0
garigue	5	1	0	0	2	85	1	5	2	0	0	0	2	2	1	1	0
grass	6	1	0	0	2	0	90	1	0	0	0	0	2	2	0	1	0
dec. forest	7	1	0	0	2	5	3	84	1	0	0	0	1	1	0	2	0
con. forest	8	0	0	0	0	1	0	0	98	0	0	0	0	0	0	0	0
rice	9	0	0	0	0	0	0	0	0	96	0	1	1	0	0	0	0
sea water	10	0	0	0	0	0	0	0	0	0	98	0	0	0	0	0	0
estuary	11	0	1	0	0	0	0	0	0	0	0	97	0	0	0	0	0
wheat	12	3	0	1	3	3	15	4	1	0	0	0	63	3	0	3	0
barley	13	1	0	0	9	1	7	2	0	0	0	0	2	75	1	1	1
maize	14	1	0	1	1	1	1	2	0	0	0	0	2	1	89	1	0
tiled	15	4	0	0	2	2	3	3	1	0	0	0	2	1	0	81	0
sand	16	0	0	0	1	0	1	1	0	0	0	0	0	2	0	0	93

Table 2. Neural network classification of L-P-TEST, the average of the diagonal terms is 87.1%

The second image was the moderate dimension image acompanying the MultiSpec[18] computer program, which was taken in 1966 by an airborne spectrometer. From this 12 band image a large set of pixels classified into 10 classes was available. Per class about 2/3 of the pixels were randomly selected to be used for training the neural networks (the A-TRAIN set) and the remaining pixels were used for testing the performance of trained nets (the A-TEST set). To test the purity of the ground survey data, the A-TEST pixels were passed through a 12-64-10 neural network trained with the A-TRAIN pixels. The resultion f distribution in Fig. 3 shows a sharp peak at 0.96 and the average of the diagonal terms of the classification table was 85.8%. This showes that the ground truth set for the airborne image is rather pure.

4. PURIFICATION METHOD

One could purify the ground survey data by requiring a minimum value of the f for the right class, but then one would select a set of pixels suitable for neural network processing. There is no guarantee that one would obtain a representative sample of pure basic spectra and it would also reduce the number of ground truth pixels.

Another method was therefore developed to purify and possible enlarge the data set with known classification. Each pixel in the image was assigned to the class of the nearest example in the ground truth set, provided it was close enough. As distance metric to select the nearest example the city-block distance was used: $d_{abs} = \sum_{i=1}^{n} |s_i^j - s_i^g|$, with s^j the intensity of a pixel j in the image and s^g the intensity of a pixel g in the ground truth set. To define "close enough" two ways can be selected: (1) requiring $d_{abs} \leq \epsilon_{MS}$ called MS or (2) by placing restrictions on each band: $|s_i^j - s_i^g| \leq \epsilon_{MB} \forall i = 1, \ldots, n$ called MB. Then those image pixels were selected which have the same class as their neighbourhood, where the neighbourhood size can be varied from 4 pixels for classes with a low number of pixels to 24 for classes with a large number of pixels.

After some experimentation the MB method with $\epsilon_{MB} = 3$ was choosen and the selected pixels were divided in a training set (L-P-TRAIN) and a testing set (L-P-TEST). The distribution of the obtained f's, after training a 6-64-16 network with L-P-TRAIN and evaluating it with L-P-TEST, is given in Fig. 3 and shows a sharp peak at 0.94. In Table 2 the percentage of pixels having their maximum f in one of the neural network output classes is given for each input class. Compared to Table 1 the average of the diagonal terms is increased from 67.9% to 87.1%.

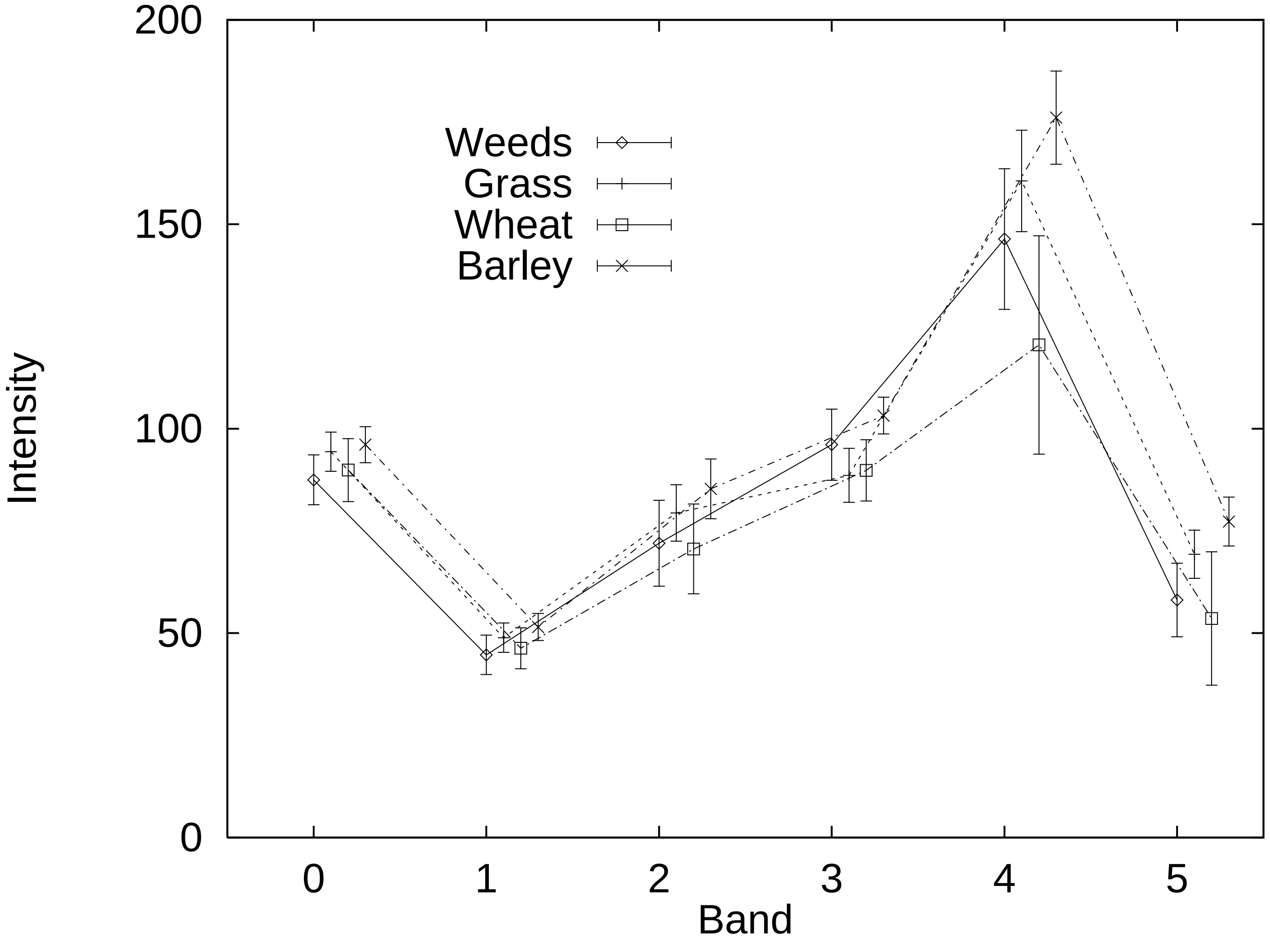


Figure 4. Average and RMS of intensity per band for various classes

The classes "weeds" and "wheat" are contaminated with "grass" and "barley" with "grass" and "weeds". These contaminations are probably due their large overlap in intensities as shown in Fig. 4. The contaminations by the classes "garigue" and "deciduous forest" were largely removed, they were probably due to bad ground survey or bad registration between ground survey and image data.

The number of pixels was decreased by up to 25% for 4 classes ("sand", "wheat", "weeds" and "barley") and increased for the other classes, sometimes with a large factor. For two classes ("sea water" and "estuary") the number of pixels was reduced by random selection of pixels in order not to have a too large variation in the number of pixels per class used for training the neural networks.

For the airborne data the MB method with $\epsilon_{MB} = 3$ improved the average of diagonal terms of the classification table a little from 85.8% to 86.5%. However, it decreased the number of pixels in the "water" class from 43 to 37, so it was decided not to used the improved data.

5. MIXED PIXEL GENERATION

Mixed pixels for all combinations of 3 classes were generated because for land use and agricultural applications usually not more than 3 classes are present in a pixel.

For each combination the full f space allowed by the constraints given in equation (2) was covered with a 3-D grid with spacing 0.02. By allowing f's to be 0, also pure classes and combinations of 2 classes were considered. At each point of the grid a mixed spectra was calculated as : $s_i = \sum_{j=1}^{3} f_j M_{ij}$ with M randomly selected from the pure pixels under consideration. This models a kind of physical linear mixture model as defined in equation (4).

From A-TRAIN and A-TEST training and test mixed pixel sets were so generated with each set containing 163270 mixed pixels. The sets generated from L-P-TRAIN and L-P-TEST each contained 717520 mixed pixels. To speed up training for the LANDSAT data the neural networks for them were first trained with a mixed pixel set generated from L-P-TRAIN with a grid spacing of 0.05, containing 118720 pixels and then further trained with the full set.

6. COMPARISON METHOD

The quality of a fuzzy classification method was defined as a root mean square per-pixel error:

$$e_{pp} = \sqrt{\frac{1}{2}\sum_{i=1}^{c}(\hat{f}_i - f_i)^2}, \tag{5}$$

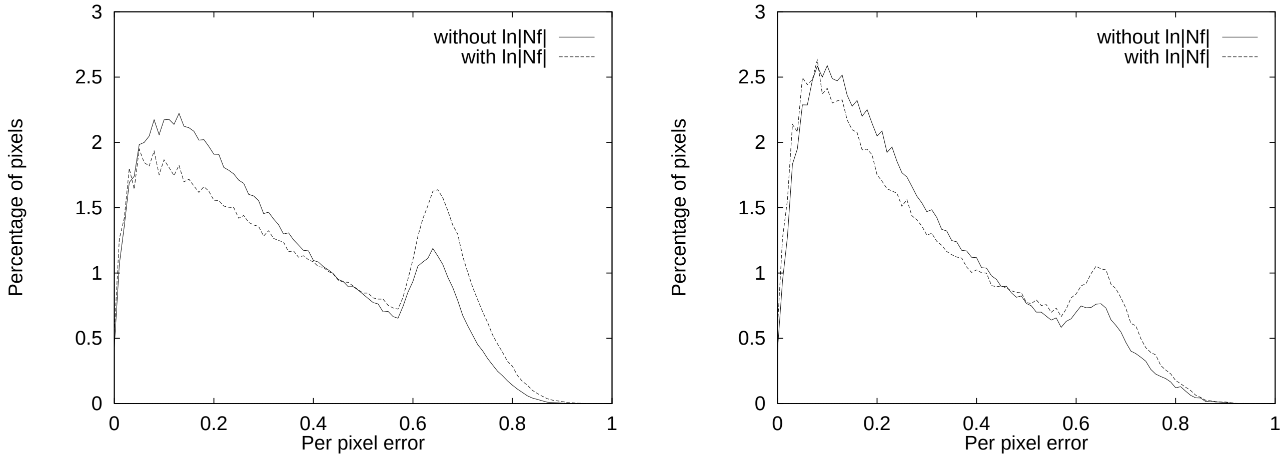
where c is the total number of classes (16 or 10), $\hat{f}_i$ is the f determined by the fuzzy classification method and f_i is the true f. The factor $\frac{1}{2}$ ensures that e_{pp} is in the range 0 to 1.

To evaluate the results of the neural networks another fuzzy classification method was needed. As for the LANDSAT data the number of classes is much larger than the number of bands, the maximum likelihood and the analytical methods described in Sect. 1 could not be used directly. For the airborne data the number of classes (10) is comparable to the number of bands, making it unlikely that those methods would produce good results.

As the number of classes in the mixed pixels was limited to 3, a combination of the two methods, called AnaML, could be developed. For each combination of 3 and 2 classes and for the single classes, a solution $\hat{f}$ was calculated using the analytical method. The matrix N was approximated using the average of the classes in consideration and the positivity constraint was handled by setting negative $\hat{f}_j$'s to 0 and then renormalizing so that the sum-to-1 constraint was satisfied again. Then from the set of $\hat{f}$'s the one was selected which had the minimal maximum likelihood value. As shown in Fig. 5 better results are obtained when the $ln|N_f|$ is left out: the peak at the end, caused by selecting the wrong combination of classes, is smaller and the RMS average e_{pp} decreases by about 10%. Thus it was decided not to use the logaritme of the determinant term in the AnaML method.

7. NEURAL NETWORK RESULTS

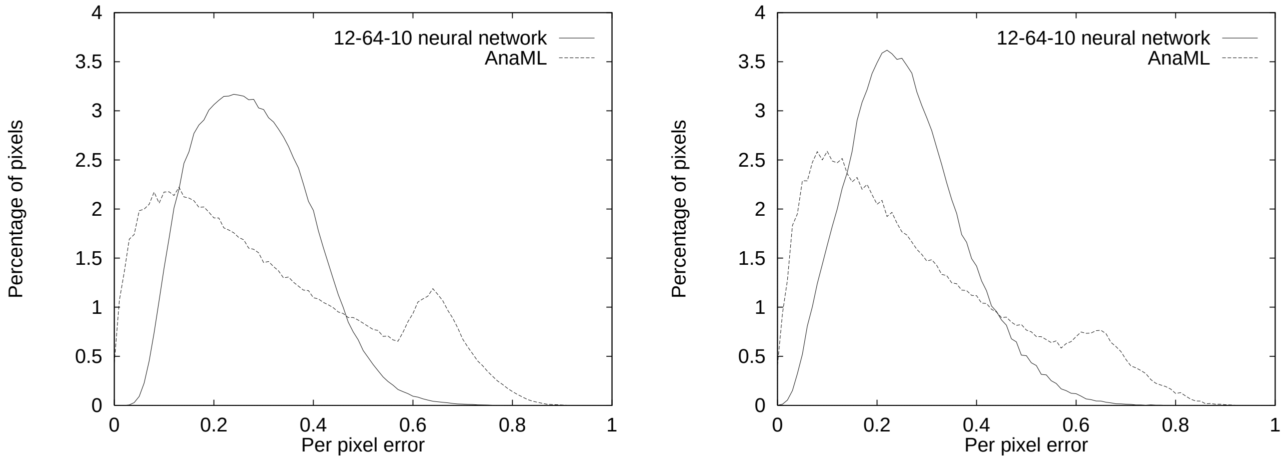
In Fig. 6 the e_{pp} distributions are shown for the neural network (64 hidden neurons) and AnaML methods on the mixed pixel sets generated from the L-P-TEST and A-TEST sets. The neural network results do not have the peak at the end, probably because the output of the neural network is not forced to contain only 3 classes. Trying to restrict the output of the neural networks to 3 classes by taking the 3 highest f values and rescaling them so that the sum-to-1 constraint is satisfied, also worsened the e_{pp} distribution. Thus for the neural network method less



(a) L-P-TEST

(b) A-TEST

Figure 5. e_{pp} distributions with and without $ln|N_f|$ term



(a) L-P-TEST

(b) A-TEST

Figure 6. e_{pp} distributions for neural network (64 hidden neurons) and AnaML methods

pixels have a large error that for the AnaML method: for L-P-TEST (A-TEST) the percentage of pixels with an error larger than 0.6 is 1.0% (0.5%) for the neural network and 17.2% (10.2%) for AnaML. The RMS average e_{pp} is about 20% smaller for the neural network: 0.305 compared to 0.378 for L-P-TEST and 0.289 compared to 0.349 for A-TEST.

Although training a neural network takes a considerable amount of time, once the weights are determined fuzzy classification by a neural network is much faster than by the AnaML method, as shown in Table 3. The neural network is more than 200 times faster in the examined cases. Counting the number of multiply-add operations in the computer code gives the following expressions for the number of those operations:

$$N_{ma}^{N}N = h(n+c) \tag{6}$$

$$N_{ma}^{A}naML = \{\frac{1}{6}c(c-1)(c-2)+\frac{1}{2}c(c-1)\}\{\frac{1}{2}n^3+\frac{11}{2}n^2+4n+cn+2c\} \tag{7}$$

The values calculated for the execution time per multiply-add operation and shown in Table 3 indicate that the above expressions can be used to predict the execution time to within 15%.

	Neural Network			AnaML		
name	time	operations	time/oper	time	operations	time/oper
L-P-TEST 6-64-16	0.172 ms	1408	0.122 μs	44.6 ms	311440	0.143 μs
A-TEST 12-64-10	0.196 ms	1408	0.139 μs	43.0 ms	304260	0.141 μs

Table 3. Timing results for the fuzzy classification methods on 1 pixel on 1 node of the PowerXplorer

The fuzzy classification of a full LANDSAT image (about 6000 by 6000 pixels) would take 30 hours for AnaML and a few minutes for the neural network on a 16 node PowerXplorer. The training of the neural network however will take up to 1 week.

Examining the case of $n = 250$ and $c = 16$ the AnaML method will take 180s per pixel, to process a 6000 by 6000 image would take about 14 years using 16 nodes. For the neural network the times depend on the value of h that is needed. Our previous results[8] on 220 band AVIRIS data seems to indicate that 64 hidden neurons might be sufficient, due to the fact that the bands are correlated. In that case processing the full image would take less than 2 hours and the training time would be in the order of 3 months.

8. CONCLUSION

A 1 hidden layer backpropagation neural network can succesfully perform the fuzzy classification of mixed pixels. This was demonstrated on pixels which were a linear combination of up to three ground covers, selected from a larger number of ground covers: 16 for 6 band LANDSAT data and 10 for 12 band airborne data.

The neural network simulations were performed with our parallel program which is implemented in C using PVM for communication, allowing it to be easily ported from our current execution platform (a 16 node PowerXplorer) to other parallel or distributed systems. It was shown that the implemented learning of the steepnesses of the transfer functions decreased the training time needed to reach a certain accuracy by a factor 2 to 4. For the updating of network weights the semi-epoch mode was developed were the update is done several times per epoch, typically after 1% of the patterns have been processed. This mode gave lower errors than using on-line mode and it is faster because it can be effectively parallelized.

The ground truth set for the LANDSAT image appeared to be not very pure, large mixing between the classes was observed. A method to purify the set was developed by classifying all pixels in the image using a nearest neighbour method and then selecting pixels which have the same class as their neighbour pixels. The mixing between classes due to bad ground survey or bad registration was largely removed, the remaining mixture is mainly between 4 classes with show a large overlap in their intensity distributions. The number of pixels was increased for most classes. The parameters of the nearest neighbour method and the size of the neighbourhood in the image can be varied to trade-off purity and number of obtained pixels.

To compare the results of the neural network with, the AnaML fuzzy classification method was developed. It uses the analytical method on all combinations of 3 classes and then selects the best combination using a maximum likelihood method. The logaritme of the determinant term was left out as it increased the number of wrongly selected combinations. The comparison between the two fuzzy classification methods was based on the RMS per-pixel error e_{pp}.

The neural networks gave a 20% smaller RMS average e_{pp} than the AnaML method and gave also less pixels with a high e_{pp}.

Fuzzy classification of pixels using a neural network is much faster than using the AnaML method, but the needed training of the neural network takes a large amount of time. Expressions for calculating the time needed for neural network and AnaML classification were determined. The speed advantage of the neural network increases with increasing number of bands or classes, at large enough values for them and depending the number of pixels to proces, the neural network method, including the training time, will be faster.

REFERENCES

1. G. Vane *et al.*, "The airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment* **44**, pp. 127–143, 1993.
2. Y. Q. Xue, "The technical development of airborne scan imaging systems," *Journal of Infrared and Millimeter Waves (China)* **11**, pp. 168–191, 1992.
3. H. Horwitz *et al.*, "Estimating the proportion of objects within a single resolution element of a multispectral scanner," in *Proc. of the 7th International Symposium on the Remote Sensing of the Environment*, pp. 1307–1320, (Ann Harbor, Michigan), 1991.
4. T. E. Schouten, M. S. klein Gebbinck, Z. K. Liu, and S. W. Chen, "Fast algorithm for spectral mixture analysis of imaging spectrometer data," in *Image and Signal Processing for Remote Sensing III*, J. Desachy, ed., *Proc. SPIE* **2955**, pp. 198–207, (Taormina), 1996.
5. J. J. Settle and N. A. Drake, "Linear mixing and the estimation of ground cover proportions," *International Journal of Remote Sensing* **14**, pp. 1159–1177, 1993.
6. M. S. klein Gebbinck and T. E. Schouten, "Decomposition of mixed pixels," in *Image and Signal Processing for Remote Sensing II*, J. Desachy, ed., *Proc. SPIE* **2579**, pp. 104–115, (Paris), 1995.
7. M. S. klein Gebbinck, *Decomposition of mixed pixels in remote sensing images to improve the area estimation of agricultural fields*, Universal Press, Veenendaal, The Netherlands, 1998. PhD Thesis, Katholieke Universiteit Nijmegen, ISBN 90-9012027-0.
8. T. E. Schouten and M. S. klein Gebbinck, "A neural network approach to spectral mixture analysis," in *Neurocomputation in Remote Sensing Data Analysis: Proceedings of the Concerted Action COMPARES*, I. Kanellopoulos, G. G. Wilkinson, F. Roli, and J. Austin, eds., pp. 79–85, (York), 1997.
9. T. E. Schouten and M. S. klein Gebbinck, "Fuzzy classification of spectra," in *Neural networks: Best practice in Europe*, pp. 198–201, (Amsterdam), 1997.
10. D. E. Rumelhart, D. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed processing*, D. E. Rumelhart and J. L. McClelland, eds., pp. 318–362, MIT Press, 1986.
11. G. M. Foody, "Fully fuzzy supervised classification of land cover from remotely sensed imagery with an artificial neural network," *Neural Computing & Applications* **5**, pp. 238–247, 1997.
12. G. M. Foody, "Land cover mapping from remotely sensed data with a neural network: accommodating fuzziness," in *Neurocomputation in Remote Sensing Data Analysis: Proceedings of the Concerted Action COMPARES*, I. Kanellopoulos, G. G. Wilkinson, F. Roli, and J. Austin, eds., pp. 28–37, (York), 1997.
13. A. Sperduti and A. Starita, "Speed up learning and network optimization with extended back propagation," *Neural Networks* **6**, pp. 365–383, 1993.
14. L. G. Vuurpijl, T. E. Schouten, and J. Vytopil, "Performance prediction of large mimd systems for parallel neural network simulations," *Future Generation Computer Systems* **11**, pp. 221–232, 1995.
15. L. G. Vuurpijl, *Platforms for artificial neural networks - Neurosimulators and Performance prediction of MIMD-parallel systems*, Louis Vuurpijl, 1998. PhD Thesis, Katholieke Universiteit Nijmegen, ISBN 90-9011347-9.
16. R. P. H. M. Schoenmakers, G. G. Wilkinson, and T. E. Schouten, "Results of a hybrid segmentation method," in *Image and Signal Processing for Remote Sensing I*, J. Desachy, ed., *Proc. SPIE* **2315**, pp. 113–126, (Rome), 1994.
17. R. P. Schoenmakers, *Integrated methodology for segmentation of large optical satellite images in land applications of remote sensing*, Office for Official Publications of the European Communities, EUR 16292 EN, Luxembourg, 1995. PhD Thesis, Katholieke Universiteit Nijmegen.
18. D. Landgrebe, *Multispectral Data Analysis: A Moderate Dimension Example.* Purdue University, June 1996. http:/dynamo.ecn.purdue.edu/ biehl/MultiSpec/.