BACKPERCOLATION TRAINING OF NEURAL NETWORKS FOR AGRICULTURAL LAND USE CLASSIFICATION WITH LANDSAT–TM DATA

A. K. Kaifel, F. Löher

Center for Solar Energy and Hydrogen Research
Hessbruehlstr. 61, 7000 Stuttgart 80
Germany

ABSTRACT:

The recently published backpercolation algorithm for the training of neural networks will be compared with the backpropagation and quickpropagation algorithm by means of "artificial" classification problems (e.g. XOR, M-N-M decoder) and serveral others. Within all classification schemes the backpercolation algorithm is much more efficent and even successful where the other training schemes are without success.

Several different neural networks are trained by backpercolation in order to classify agricultural land use within multitemporal LANDSAT-TM scenes and to show the efficency of various network configurations (e.g. number of neurons in the hidden layers, number of hidden layers) and the impact of training parameters.

In the classification fourteen different classes are discriminated. A relaxation process after the classification is able to improve the accuracy of the result up to 15 % in various classes. For all classes the achieved classification result is better than 80 % up to 90 % or single classes on a pixel bases.

KEY WORDS: Neural Network, Backpercolation, Classification, LANDSAT-TM, Remote Sensing Applications, Agricultural Land Use, Multispectral, Multitemporal.

1. INTRODUCTION

In order to improve classification results of land use classification within SPOT images obtained by conventional multivariate classification methods [Kaifel, 1990] we made a new approach with neural networks. Furthermore, statistical classification methods don't work good enough applied to multitemporal LANDSAT-TM images to detect up to 15 different agricultural land use classes. The reason for these poor results might be that the multitemporal satellite sets has a non-gaussian distribution.

Neural network approaches outperform statistical classification schemes if datasets are distribution-free [Benediktsson, 1990]. At first we tested and compared commonly employed training techniques for feedforward multilayer perceptron neural networks like backpropagation [Rummelhart, 1986a, 1986b] and quickpropagation [Fahlmann, 1988] and furthermore the more recent and less popular backpercolation algorithm [Jurik, 1990] by means of "artificial" classification problems.

The results are discussed in section 3 and the most efficient one is applied to multitemporal LANDSAT-TM data sets followed by a relaxation process for further improvements of the classification results as described in section 4.

2. LEARNING ALGORITHMS

2.1 Description

The first general training algorithm for multilayer preceptron networks was the backpropagation algorithm of Rummelhart. The adaptation of weights and thresholds in a network will be calculated from the error function at the network output back through the network by means of steepest descent techniques. Backpropagation disadvantages are very slow learning, frequent inferior solutions when the problem to be solved is moderately complex and oscillation between increasing and decreasing weight values.

In order to omitt oscillations due to inferior solutions Fahlman modified the backpropagation algorithm to speed up learning (quickpropagation). He proposes that weight adaptation and output error of a network can be described by a parabolic function and that the change of weights are correlated with the slope of the function.

In comparison to backpropagation and quickpropagation the backpercolation algorithm published by M. Jurik [Jurik, 1990] assign each cell of a network its own output error and weights descend their cell's own output error surface.

As our result in the next section will confirm backpercolation is very stable and permit large training rates for fast convergence.

At this place I don't want to go more in mathematical details. More interested persons should have a look to the original references of the different learning schemes mentioned in section 1.

2.2. Comparison

Nomenclature: For all tables in this subsection the following symbols are used:

$\overline{E_p}$: mean number of epochs for successful learning,
max : maximum number of epochs,
min : minimum number of epochs,
N_F : number of training procedures without success,
N_{sim} : total number of training procedures,
λ : learning rate,
σ : standard deviation of the number of epochs for successful learning,
θ : amplitude of random generator,
? : the questionmark indicates that this value is not mentioned in the references used.

For benchmark tests of learning algorithms mostly artificial classification problems like exclusiv-OR (XOR) and N-M-N decoders are used. Furthermore some random patterns and parity bit classification problems are compared.

With each problem Nsim number of simulations are performed. Due to random inititialization of the weights and thresholds each simulation has different results.

XOR: Quickpropagation learning (Fahlmann, 1988) got with two neurons in the hidden layer his best results. Table 1 compares it to the results we achieved with backpercolation.

Decoder: N-M-N-decoder are special for learning algorithms because the input and output layers have the same size and only one neuron of each pattern has the value one. All others are zero. Table 2 show the results of learning different N-M-N decoders from Fahlmann [Fahlman, 1988] and Table 3 shows our results we achieved with backperclation learning of the same decoder (neural network) configurations.

Parity Bit: For this benchmark test the neural net has only one output neuron. This should be set to one if the number of input values set to one (high) is odd.

Discussion: If we compare backpropagation with backpercolation learning (Table 4) backpercolation is by a factor of about 50 in the mean value faster than backpropagation. Jurik also mentioned too, that backpropagation cannot handle learning rates higher than 0.5 and that the much higher stability of backpercolation learning allows learning rates up to 2.0 [Jurik, 1990].

The difference of quickprogation with backpercolation (Table 1, 2, 3) is not so dramatic but backpercolation outperforms quickpropagation in speed of learning (nb. of epochs) up to two times in the mean. The standard deviation of used epochs for successful learning of the different patterns is less for backpercolation. This means that the

Lern mode	λ	θ	max	min	σ	N_F	N_{sim}	$\overline{E_p}$
QP	4.0	1.0	66	10	16.3	14	100	24.2
Perc	2.0	2.0	59	10	11.8	0	100	23.1
Perc	4.0	2.0	34	6	6.1	15	100	15.4

Table 1. Comparsion of quickpropagation learning (QP) [Fahlman, 1988] with backpercolation learning (Perc) by means of XOR-problem. The networks has all one hidden layer with 2 neurons.

decoder	λ	θ	max	min	σ	N_F	N_{sim}	$\overline{E_p}$
8-2-8	1.0	4,0	155	26	37,7	?	25	102,8
8-3-8	2.0	2,0	42	12	5,6	?	100	21,99
8-4-8	3.0	2,0	23	10	2,8	?	100	14,88
8-5-8	3.0	2,0	19	9	2,0	?	100	12,29
8-6-8	3.0	2,0	15	7	1,6	?	100	10,13
8-8-8	3.0	2,0	12	6	1,3	?	100	8,79
8-16-8	3.0	2,0	10	4	1,0	?	100	6,28
16-4-16	1.2	2,0	48	20	5,4	?	100	28,93
10-5-10	0.35	2,0	21	9	2,1	?	100	14,01

Table 2. Results of quickprogation learning with different N-M-N decoders [Fahlmann, 1988].

decoder	λ	θ	max	min	σ	N_F	N_{sim}	$\overline{E_p}$
8-2-8	1,2	1,0	158	63	28,7	0	50	98,5
8-3-8	2,0	1,0	20	10	2,9	0	50	13,8
8-4-8	2,0	1,0	12	7	1,7	0	50	9,3
8-5-8	3,0	1,0	9	5	1,3	0	50	7,4
8-6-8	3,0	1,0	10	4	1,7	0	50	6,9
8-8-8	3,0	1,0	8	4	1,3	0	50	6,1
8-16-8	4,0	1,0	5	4	0,5	0	50	4,5
16-4-16	2,0	1,0	29	15	4,27	0	50	20,9
10-5-10	2,0	1,0	9	5	1,76	0	50	6,9

Table 3. Results of backpercolation learning with the same decoder configuration as shown in Table 2.

Lern mode	λ	θ	max	min	σ	N_F	N_{sim}	$\overline{E_p}$
BP	0.5	1.0	2538	1134	738.6	14	20	1968.7
Perc	1.0	2.0	1046	116	280.0	2	22	33.5

Table 4. Comparsion of backpropagation learning (BP) with backpercolation (Perc) learning of neural networks,for 4-Bit parity classification. All networks have one hidden layer with 4 neurons.

learning parameters (λ, θ) don't have to be well adjusted for successful learning by backpercolation compared to quickpropagation.

Due to this experiences during a first study phase with neural networks, we decided to apply only the backpercolation learning for "real" classification problems like agricultural land use classification as described in the following sections.

3. APPLICATION TO LANDSAT-TM DATA

Since 1989 we are working together with the University of Stuttgart and Freiburg in a project to test the applicability of satellite remote sensing technique for agricultural land use classification in the State Baden-Württemberg (southwest of Germany). The mean agricultural field size in Baden-Württemberg is approximatly 1.5 ha which is very small in comparison to regions in Germany or Europe or even to the USA.

The small agricultural field size is one of the main difficulties in applying remote sensing techniques in this region. Therefore the rectification of the multitemporal data have to be very precise and for the classification there are a lot of non-uniform pixels due to borders between the fields with different land use.

land use class	nb. of pixels (original)	nb. of pixels (eroded)
winter wheat	21 465	10 409
rye	830	354
winter barley	8 021	4 201
summer barley	7 822	4 022
oats	891	223
spelt	106	—
pasture	1 882	643
pea	1 744	1 269
potatoes	128	—
beet	13 662	6 639
rape seed	1 962	833
sun flower	2 786	1 609
maize	7 222	2 819
orchard	1 022	626

Table 5. Table of agricultural land use classes with corresponding number of pixel of the ground true data (original) and by two pixels eroded fields (eroded) .

3.1 Classification

For the classifcation ground true data of fourteen different crops (Table 5) are used for the training of various neural network configurations. For the land use classification only the backpercolation algorithm for training of neural nets is used because during the test phase with other training schemes backpercolation get the best results with "artificial" benchmark tests (see section 3).

As input data two rectified LANDSAT-TM scenes (4000 X 3600 pixels) of the year 1990 of a region west of Stuttgart (county of Ludwigsburg) are used. The first one is from the beginning of the growing season (March 18) and the second one direct before harvest of grain (winter wheat, summer barley, winter wheat; July 24).

Table 5 list the different crops to be classified with the number of pixels of the ground true data which highly varies. For the training of the various networks a certain number of pixels of each class is randomly selected (training set) and the remaining (verification set) pixels are used for verification of classification results. For two classes (potatoes, spelt) only about 100 pixels available within the ground true data. Because of the use of 100 up to 500 pixels per class for the training of the neural nets we used the same pixels for these two classes in the verification data set.

Due to non-uniform pixels at the edges of fields we applied an erosion process for each class to remove 2 pixels at the edge of each field. This implies that for multitemporal data sets which we used, also an uncertainty of 0.5 pixels of the rectification of the data are removed too (this is approximately the failure for our rectification). After the erosion the small fields of the classes potatoes and spelt are totally eroded in the ground true data set (Table 5).

Experiments with the remaining 12 classes led to a much better convergence during the training of the various neural networks. This prove that the non-uniform pixels are mostly removed and that the failures due to rectification are neglicible.

3.2 Relaxation

Further improvements of the results of the classification can be achieved by a relaxation process we already applied for cloud detection [Kaifel, 1989] in AVHRR images and land cover classifications with SPOT data [Kaifel, 1990].

The relaxation is an iterative process inside a window (e.g. 5 x 5 pixels) for each pixel of a segmented image. This yields to a substantially improvement of cartographic quality of classified images and a more accurate classification on pixel basis (see next section).
During the iterative relaxation process the analog output data of a neural network inside a window are handled as probabilities of different classes and are weigthed with the corresponding number of pixels of each class in the window. After weigthing and addition of the probabilities for each class a new decision of the class for the central pixel of the window will be made corresponding to the highest probability.

This scheme is applied iterativly and yields to improvements of the overall classification rate compared to the first classification made by the neural network of up to 10 % on pixel basis.

3.3 Classification Scheme

For easy handling of the huge amount of data during learning of neural networks and to check the quality of classifcation, we developed a software package with different programs for each step of a classification (Fig. 1). It is easy to use and the different programs communicate by various data file.

Three typs of input data are neccessary for the program NEURALINPUT which extracts the different patterns to be learned (pattern file) by a neural net from the input images. The feature file contains the multispectral/multitemporal images, the mask file the ground true data and the additional filter file is used to mask out areas in images which should not be classified.

The pattern file is used as input for training a neural network (PERCLEARN) and a net file containing configuration, weights and threshold values of the learned network is the result. Optionally a net file can also be used as input file to adapt the network to new patterns or to continue learning with other parameters like learning rate etc..

If a network is successful adapted the image can be classified with the program PERCCLASSIFY which uses the output file for the segmented image and the possibility file for the analog output values of each neuron and each pixel. This file is used as input for the relaxation (see section 4.2)and the program CLASSQUAL perform a quality check and write the confusion matrix and some other relevant information on the quality file to check the segmentation by means of statistical values.

All programs write additional datas on a documentation file to get more information on parameters used during the run phase.

4. RESULTS

As described in section 4.1 and shown in table 5 the classification is made with two different training data sets. The first one are the original ground true data contain 14 different land use classes and the second one where the fields of the ground true data are eroded by two pixels containing the remaining 12 land use classes.

The feature file consists of two rectified LANDSAT-TM scenes with channel 1-5 and 7. Therefore we got 12 input features for each pixel which implies a neural network with 12 input neurons. In order to test the impact of network configuration and the number of patterns we trained various networks (one or two hidden layers) with both training datas set.

Table 6 show the various network configuration learned, used the two different training data sets. In the first column the network configuration is mentioned. The numbers (e.g. 12-50-14) means 12 neurons in the input layer corresponding to the number of classes, 50 neurons in the hidden layer and 14 neurons in the output layer.

The first three networks are trained with the first training data set (14 classes, see table 5) and the last three network

network configuration	maximum features per class	number of epochs	features learned [%]	classif. rate [%]	relax. 5x5*5 [%]	relax. 7x7*5 [%]	relax. 9x9*5 [%]
12–50–14	200	1821	94.5	65.2	74.4	76.6	77.4
12–100–14	200	1245	97.0	65.3	72.7	76.7	77.6
12–100–50–14	500	1030	96.3	69.4	72.8	75.5	77.5
12–40–12	500	2365	97.4	70.1	76.2	78.6	79.8
12–50–12	500	685	97.2	67.9	73.0	74.9	75.8
12–50–12	1000	512	98.2	70.3	74.1	75.1	75.9

Table 6. Comparison of classification rate of different network configuration trained with two sets of ground true data (Table 5). The last three columns show the classification rate after application of an iterative relaxation process with three different window sizes (5x5, 7x7, 9x9). The number of iterations is 5 for all three relaxation process.

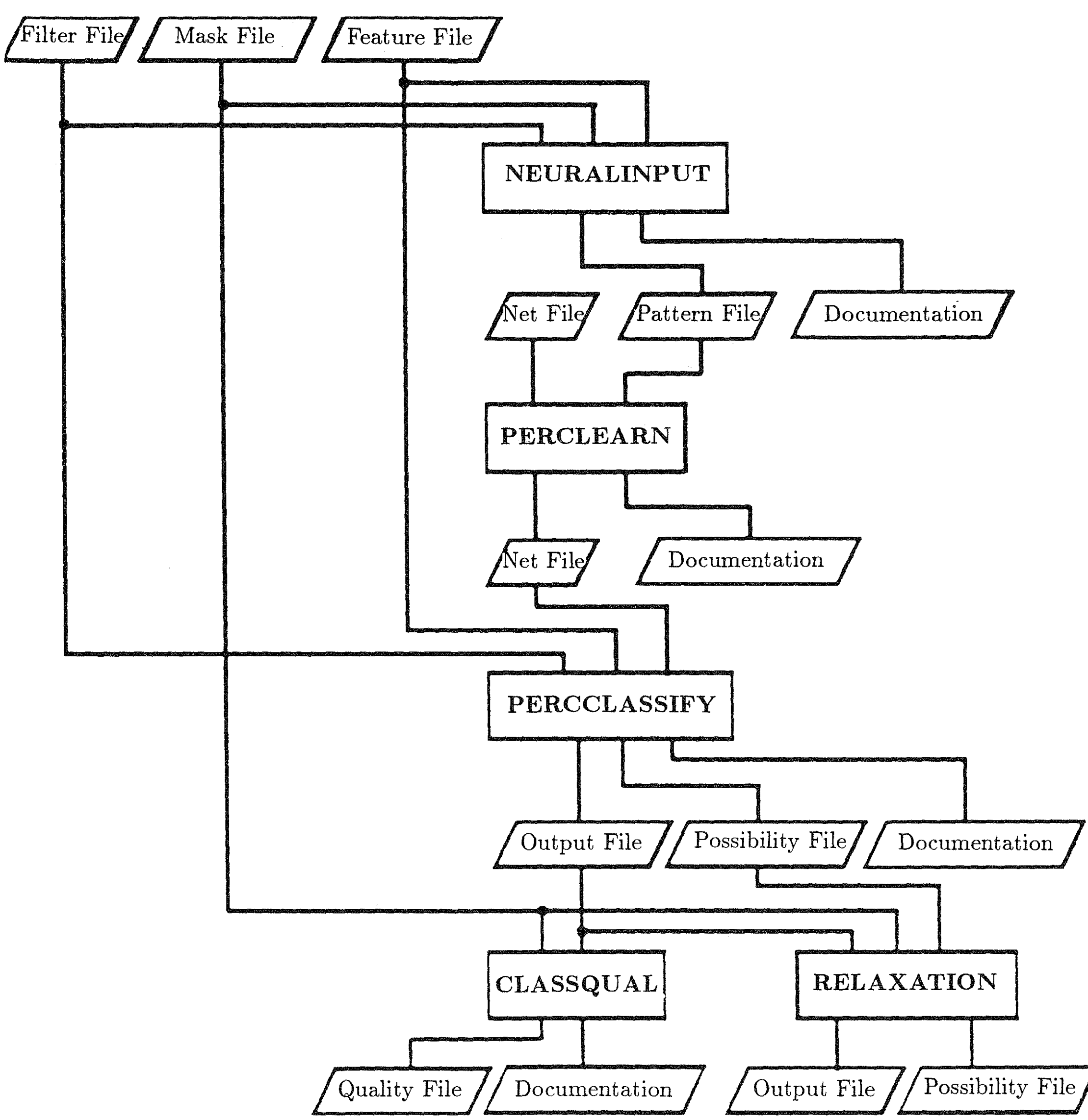

Fig. 1. Scheme and data flow of the software system for classification of image data by neural networks.

configurations used the second training data set with only 12 classes.

The network with 50 neurons in one hidden layer and 200 features of each class applied to training data with 14 classes had some difficulties to learn more than 95 % of the patterns because some neurons came in saturation so that the learning process convergence was very slowly. This makes the training uneffectiv and it wastes a lot of CPU-time.

The second network with 100 neurons in the hidden layer learned much faster and more patterns (97 %), but the classification rate is only some percent higher. This means that it is not neccessary to take more much more than 200 pixels per class for learning a neural network. The last network mentioned in Table 6 will be learned with up to 1000 pixels per class but the result is not much better as if we used only 500 pixels per class (Table 6).

The neurons of the network with two hidden layers are not in saturation especially in the first hidden layer. The training uses enormous computation time to learn the network. When we stopped learning it has more than 30 days CPU-time consumed on a VAX-Station 3100 M78, but the network still convergenced when we stopped it.

	classification error rate [%]			
network configuration	**12–100–50–14**		**12–40–12**	
land use	**relaxation**		**relaxation**	
classes	no	9x9*5	no	9x9*5
winter wheat	27.7	21.0	27.9	13.0
rye	37.1	29.5	42.7	33.7
winter barley	21.2	14.6	32.8	24.3
summer barley	37.6	31.9	29.0	21.0
oats	39.8	35.1	49.1	45.6
spelt	25.3	35.5	—	—
pasture	14.9	9.6	27.2	24.1
pea	18.3	12.6	14.4	11.0
potatoes	25.0	14.1	—	—
beet	26.4	22.2	32.0	26.1
rape seed	28.2	13.4	22.3	18.0
sun flower	25.3	27.1	34.2	31.5
maize	28.6	14.4	30.1	15.7
orchard	27.9	14.0	26.1	26.5

Table 7. Classification error rates for each class and one network configuration for the two different training data sets. The error rates before and after a relaxation process with a window size of 9 x 9 pixels and 5 iterations for each network are shown.

Fig. 2 shows the number of learned patterns during the training verus number of iterations (epochs). The total number of patterns to be learned is 6234.

The training data set two with 12 classes due to erosion of the ground true data fields didn't made a lot of problems. For learning of the first two networks we used the same but much more conservative parameters as for the training of the third one (Table 6).

All three networks are learned 97.2 % to 98 % of the patterns but the number of epochs are quite different. The overall classification result is about 70 % however if more epochs are used for the training the relaxation is more efficient.

Table 7 shows the classification error rate of each class for classification made by neural networks and after relaxation process. The networks mentioned corresponds to one configuration for each of two training data sets (Table 5). For several classes like winter wheat and maize with typical field size larger than the mean, the improvement by the relaxation is more than 15 %. For the class orchard with typical very small fields (less than 1 ha) there is no improvement by the relaxation.

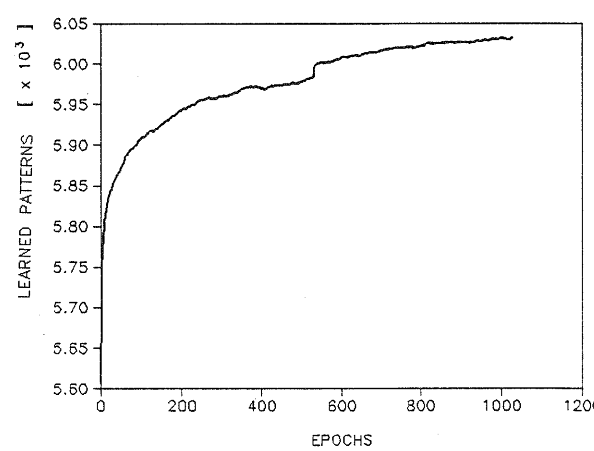

Fig. 2. Number of learned patterns from training data set for network with 12-100-50-14 configuration. The maximum number of patterns are 6234. After 1030 iterations 6035 pattern were learned. The step of the curve occured due to changes of some learning parameters when we stopped learning and restarted again.

5. CONCLUSION

The comparision of commonly used algorithms for training neural networks (backpropagation, quickpropagation) with backpercolation proved that training of neural networks with back percolation is much more efficient and research is still neccessary to opitimize and speed up learning.

Classification of agricultural land use with neural networks has great potential but the computation time for "real complex" applications with up to 15 and more classes is enormously high on commercial available workstations. More effort should be made to run such applications on highly parallel computers to shorten development cycles.

Recently we implemented the program package for training neural networks on a CRAY–2 with four CPU's and we obtained an increase in speed by a factor of 300. The disadvantage is that it is very expensive to use the main frame computers of the computation center of the University of Stuttgart.

With a following iterative relaxation process after the classification by a neural network further improvements of classification rate can be achieved. Future work will also be concentrated on testing spatial relaxation functions inside the relaxation window.

It is possible to obtain image classification results at least as good as can be obtained with statistical classification methods and even better if the data are distribution-free like multitemporal or multisource data. The results presented in this paper are encouraging to continue the research on neural network classification with remote sensing data and applying the available software package to other data in different regions.

6. ACKNOWLEDGEMENTS

This work was supported by the Ministry for Rural Landscape, Food, Agriculture and Forest of the State Baden-Württemberg under contract O-Nr. 41-89.41.

7. REFERENCES

Benediktsson, J.A., P.H. Swain, O.K. Ersoy, 1990. Neural networks approaches versus statistical methods in classification of multisource remote sensing data. IEEE Trans. on Geoscience and Remote Sensing, 28 (4), pp. 540-552.

Fahlmann, S.E., 1988. Faster Learning variations and backpropagation: An empirical study. In: T.J. Sejnowski, G.E. Hinton, D.E. Touretzky, Eds., Connection Models Summer School, San Mateo, CA, pp. 38-51.

Jurik, M., 1990. Backpercolation: Exploiting the duality between signals and weights. Jurik Research PO 2379, Aptos, CA.

Kaifel, A. K., Straub, B. J., 1990. Klassifikation von Satellitendaten mit Texturanalyse zur großflächigen Landnutzungsklassifikation, W. Pillmann, A. Jaeschke (Eds.), Informatik Fachberichte Bd. 256: Informatik für den Umweltschutz, 5. Symposium, Wien, Österreich, Sept. 1990, Springer-Verlag, 1990, pp. 315 - 323.

Kaifel, A. K., 1989. Cloud detection by texture analysis technique in AVHRR images, Proc. 5th International TOVS Study Conference, Toulouse, 1989.

Rummelhart, D.E., G.E. Hinton, and R.J. Williams, 1986a. Learning internal representation by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1, D.E. Rummelhardt, and J.L. McCelland, Eds., Cambridge, MA, MIT Press 1986, pp. 318-362.

Rummelhardt, D.E., G.E. Hinton, R.J. Williams, 1986b. Learning representations by backpropagation errors. Nature, 323, pp. 533-536.

8. APPENDIX

Due to the poor quality of reproduction of grey scale images and difficulties to recognize up to 15 different classes in a grey scale image we omitt showing results in form of classified images. If you want to get some coloured images or copies please write to the first autor and you will get some nice coloured images showing results according to the text.