



CLASSIFICATION OF SATELLITE IMAGES USING ANN

KALE S.N.* AND DUDUL S.V.

Dept. of Applied Electronics, SGBAU, Amravati, Maharashtra 444602

*Corresponding Author: Email- sujatankale@rediffmail.com

Received: July 28, 2012; Accepted: August 03, 2012

Abstract- This paper describes the MLP NN classifier performing optimally in classifying the different land types from Landsat data. In fact classification process is a compulsory step in any remote sensing research. Therefore, the main objective of this paper is to assess classification accuracy of classified lands on benchmark Landsat data from UCI machine learning repository. The six land type classes namely red soil, cotton crop soil, damp grey soil, soil with vegetation stubble, very damp grey soil can be identified. Result showed that overall classification accuracy is 87.57%, which is considered acceptable.

Results show that this new neural network model is more accurate than the other NN models. These results suggest that this model is effective for classification of satellite image data.

Keywords- Classification, MLPNN, SVM, RBF NN, Recurrent NN, Jordan Elman, PCA, Modular NN

Citation: Kale S.N. and Dudul S.V. (2012) Classification of Satellite Images Using ANN. International Journal of Machine Intelligence, ISSN: 0975-2927 & E-ISSN: 0975-9166, Volume 4, Issue 2, pp.-414-420.

Copyright: Copyright©2012 Kale S.N. and Dudul S.V. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original author and source are credited.

Introduction

The human ability to find patterns in the external world is ubiquitous. It is at the core of our ability to respond in a more systematic and reliable manner to external stimuli. Humans do it effortlessly, but the mathematics underlying the analysis and design of pattern-recognition machines are still in their immaturity. Humans perform pattern recognition through a learning process; so it is with neural networks. Pattern recognition is formally defined as the process whereby a received pattern/signal is assigned to one of a prescribed number of classes (categories). The goal of pattern-recognition is to build machines, called, classifiers, that will automatically assign measurements to classes. A natural way to make class assignment is to define the decision surface. The decision surface is not trivially determined for many real-world problems. The central problem in pattern-recognition is to define the shape and placement of the boundary so that the class-assignment errors are minimized. In classification problem the task is to assign new inputs to one of a number of discrete classes or categories. Here, the functions which we seek to approximate are the probabilities of membership of the different classes expressed as functions of the input variables.

Pattern recognition or classification is formally defined as the process whereby a received pattern/signal is assigned to one of a prescribed number of classes (categories). The goal of pattern-recognition is to build machines, called, classifiers, that will automatically assign measurements to classes. A natural way to make class assignment is to define the decision surface. The decision surface is not trivially determined for many real-world problems. The central problem in pattern-recognition is to define the shape

and placement of the boundary so that the class-assignment errors are minimized. In classification problem, the task is to assign new inputs to one of a number of discrete classes or categories. Here, the functions that we seek to approximate are the probabilities of membership of the different classes expressed as functions of the input variables.

In classification, it is accepted a priori that different input data may be generated by different mechanisms and the goal is to separate the data as well as possible into classes. Here, the input data is assumed to be multi-class and the purpose is to separate them into classes as accurately as possible. The desired response is a set of arbitrary labels (a different integer is normally assigned to each of the classes), so every element of a class will share the same label. Here, the functions that we seek to approximate are the probabilities of membership of the different classes expressed as functions of the input variables. Class assignments are mutually exclusive, so a classifier needs a nonlinear mechanism such as an all-or-nothing switch.

A neural network performs pattern recognition by first undergoing a training session, during which the network is repeatedly presented a set of input patterns along with the category to which each particular pattern belongs. Later, a new pattern is presented to the network that has not been seen before, but which belongs to the same population of patterns used to train the network. The network is able to identify the class of that particular pattern because of the information it has extracted from the training data. Pattern recognition performed by a neural network is statistical in nature, with the patterns being represented by points in a multidimensional decision space. The decision space is divided into regions, each one of

which is associated with a class. The decision boundaries are determined by the training process. The construction of these boundaries is made statistical by the inherent variability that exists within and between classes.

In generic terms, pattern-recognition machines using neural networks may take one of two forms:

The machine is split into two parts, an unsupervised network for feature extraction and a supervised network for classification. Such a network follows the traditional approach to statistical pattern recognition [1,2]. In conceptual terms, a pattern is represented by a set of m observables, which may be viewed as a point x in an m -dimensional observation (data) space. Feature extraction is described by a transformation that maps the point x into an intermediate point y in a q -dimensional feature space with $q < m$. This transformation may be viewed as one of dimensionality reduction (i.e., data compression), the use of which is justified on the grounds that it simplifies the task of classification. The classification is itself described as a transformation that maps the intermediate point y into one of the classes in an r -dimensional decision space, where r is the number of classes to be distinguished.

The use of neural networks in signal processing is becoming increasingly widespread, with applications in pattern recognition. Research on the rapidly expanding use of neural networks to identify, detect and classify patterns is still in its infancy. Thus, there has been ample scope for employing neural networks in the above-mentioned tasks. In the proposed research work, the benchmark land-satellite image data from UCI Machine learning repository has been used.

The Landsat satellite data [3] is one of the reliable sources of information available for a scene. The scene is interpreted by integrating spatial data of diverse types and resolutions including multi-spectral and radar data, maps indicating topography, land use etc. Such interpretation is expected to assume significant importance with the onset of an era characterized by integrative approaches to remote sensing. Existing statistical methods for classification are neither efficient nor precise.

One frame of Landsat MSS imagery constitutes four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and the other two are in the (near) infrared region. Each pixel is 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m x 80m. Each image contains 2340 x 3380 such pixels.

The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighbourhood of pixels which are completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighbourhood and a number indicating the classification label of the central pixel. The number is a code for the following classes as shown in Table 1.

In each line of data, the four spectral values for the top-left pixel are given first, followed by the four spectral values for the top-middle pixel and then those for the top-right pixel and so on with the pixels read out in sequence left-to-right and top-to-bottom. Thus, the four spectral values for the central pixel are given by attributes 17,18,19 and 20.

There are 4435 instances of training data set and 2000 that of test

data set. Number of attributes are 36 (= 4 spectral bands x 9 pixels in neighborhood). The attributes are quantitative, in the range 0 to 255. There are 6 decision classes: 1,2,3,4,5 and 7.

Table 1- Classification code of Landsat satellite data

Number	Class
1	red soil
2	cotton crop
3	grey soil
4	damp grey soil
5	soil with vegetation stubble
6	mixture class (all types present)
7	very damp grey soil

NB. There are no examples with class 6 in this dataset.

- In a satellite image, 3x3square neighbourhood of pixels is located and the corresponding multi-spectral values of pixels are computed.
- The task of classification is associated with the central pixel in each neighbourhood.

Remote sensing as a field of study has reached its adulthood; computer-assisted classifiers have been in development for more than two decades. The complexity of remote sensing classification has led to a variety of methods, some of them are based on artificial intelligence (AI). Artificial neural network (ANN) classifier is observed to perform very well in satellite images.

In the present research work different neural network architectures are trained and tested. It is observed that MLP NN based classifier with 19 processing Elements (PEs) performs satisfactorily for 5 classes out of six.

ANN based learning machine is able to solve the multivariable classification problem with regard to the determination of type of land. ANN approach is investigated to classify the type of land available into six classes. MLP NN classifies the given samples when trained properly and overall 87.57% average classification accuracy is achieved on testing. Except for damp grey soil class, all other classes exhibit 90-96 % classification accuracy. Consistently, the damp grey soil class has 75% classification accuracy, which is reasonably good.

In classification, the input data is assumed to be multi-class and the objective is to separate it into appropriate classes as accurately as possible. Different input data may be generated by different mechanisms and that the goal is to separate the data as well as possible into correct classes. The desired response is a set of arbitrary labels (a different integer is normally assigned to each one of the classes), so every element of a class will share the same label [4].

Performance of the classifier

In classification, the important parameter to test the performance is classification accuracy which is depicted in the confusion matrix.

Confusion Matrix

A confusion matrix is a simple methodology for displaying the classification results of a network. The confusion matrix is formed by labeling the desired classification on the rows and the predicted classifications on the columns. For each exemplar, a 1 is added to the cell entry defined by (desired classification, predicted classifica-

tion). The predicted classification should be the same as the desired classification, the ideal situation is to have all the exemplars end up on the diagonal cells of the matrix (the diagonal that connects the upper-left corner to the lower right).

The confusion matrix tallies the results of all exemplars of the last epoch and computes the classification percentages for every output vs. desired combination. It is used to determine the percentage of correctly classified exemplars for each output class.

Computer Simulation

In this dataset, there are total 6435 samples of satellite images of land. These samples are divided into training dataset and testing dataset. 70% (4504) samples are used for training and 30% (1931) for testing. Following NN architectures are exhaustively trained and tested and their performances are analyzed and compared.

The networks have been trained at least 5 times starting from different random initial weights so as to avoid local minima. NeuroDimension's NeuroSolutions (version 5) is specifically used for obtaining results. System with 1.6GHz CPU, 1GB RAM, 40GB hard disk and 2MB cache is used to carry out this simulation.

Multilayer Perceptron Neural Network (MLPNN)

MLP based NN model is used in this study because it has a strong theoretical foundation. MLPs are feedforward Neural Networks trained with the standard backpropagation algorithm [5,6]. They are supervised networks so they require a desired response to be trained.

A meticulous and careful experimental study has been carried out to determine the optimal configuration of MLP NN model. All possible variations such as number of hidden layers, number of PEs (processing elements) in each hidden layer, different transfer functions in the output layer, different supervised learning rules are investigated in simulation. MLP NN model having single hidden layer with 19 PEs gives superior performance as compared to other possible models.

Table 2 displays the variable parameters of MLP NN model

Table 2- Variable parameters of MLP NN Model

S.N.	Parameter	Typical Range	Optimal value
1	Hidden Layer	1 to 2	1
2	PE	1 to 25	19
3	Learning Rule Back propagation algorithm	Momentum, Conjugate gradient, Levenberg Marquardt, Step, Delta-bar-delta,	Momentum
4	Transfer Function in output layer	Linear, Lineartanh, Tanh	Tanh

Supervised learning epochs= 1000, Error threshold = 0.01, Transfer function in hidden layer= tanh, No. of Processing Elements (PEs) in input layer = 36, No. of PEs in output layer =6, No. of connection weights for 36-19-6 architecture (P) =853, Total no. of exemplars in training dataset (N) = 4504, N/P = 5.28

Table 3 depicts the confusion matrix which shows 96.75%, 95.98%, 80.17%, 90.13%, 87.4% % classification for 1,2,7,5, 3 classes respectively except class 4 i.e. damp grey soil. Damp grey soil is misclassified as grey soil and very damp grey soil. Therefore classification accuracy drops to 75%.

Table 4 displays the performance parameters of MLP NN model for various classes.

Table 3- Confusion Matrix for MLPNN architecture

Output / Desired	Red Soil (1)	Cotton Crop (2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	446	0	0	3	0	3
Cotton crop	0	215	0	6	1	1
Very damp grey soil	0	0	372	9	20	6
Soil with vegetation stubble	7	8	21	210	3	1
Damp grey soil	3	1	59	4	138	35
Grey soil	5	0	12	1	22	319

Table 4- Classification accuracy for MLPNN architecture

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.01088	0.00709	0.05653	0.02434	0.05441	0.03134
NMSE	0.05985	0.06912	0.30969	0.22938	0.63111	0.20445
MAE	0.03961	0.02828	0.12121	0.06735	0.12548	0.08342
r	0.97033	0.96514	0.83902	0.88007	0.65725	0.89552
Percent Correct	96.7462	95.9821	80.1724	90.1288	75	87.3973

Radial Basis Function (RBF)

RBF was first introduced in the solution of the real multivariate interpolation problem [7, 8]. The construction of a RBF network, in its most basic form, involves three layers. The input layer is made up of source nodes (sensory units) that connect the network to its environment or inputs. The second layer, the only hidden layer in the network, applies a nonlinear transformation from the input space to the hidden space. The output layer is linear, supplying the response of the network to the activation pattern (signal) applied to the input layer.

A rigorous experimental study has been undertaken to determine optimal performance of RBF NN model. Different learning rules, Transfer functions, cluster centers are varied. RBF NN architecture with tanh transfer function, momentum learning rule and 100 cluster centers gives maximum classification accuracy. Table 5 shows the confusion matrix for RBF NN. It is observed from the table 6 that except classes 4 and 5 all classes are reasonably classified.

Table 5- Confusion Matrix for RBF Model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	441	7	2	59	3	3
Cotton crop	0	198	0	5	0	0
Very damp grey soil	0	10	427	74	100	10
Soil with vegetation stubble	0	9	2	95	0	0
Damp grey soil	0	0	9	0	20	0
Grey soil	20	0	24	0	61	352

Table 6 displays the various performance parameters of RBF NN model for various classes.

Table 6- Classification Accuracy of RBF NN Model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.05684	0.01766	0.09168	0.07452	0.06709	0.046641
NMSE	0.31277	0.17220	0.50223	0.70230	0.77832	0.30426
MAE	0.20207	0.06367	0.24518	0.18001	0.16993	0.15369
r	0.91910	0.91608	0.76595	0.64262	0.59181	0.85942
Percent Correct	95.6616	88.3929	92.0259	40.7725	10.8696	96.4384

FFNN (Generalised Feed Forward Neural Network)

Generalized Feedforward is, in essence, the MLP plus additional layer-to-layer forward connections. It has additional computing power over standard MLP.

Generalized feedforward networks are a generalization of the MLP such that connections can jump over one or more layers. Generalized feedforward networks often solve the problem much more efficiently than MLP [9].

A rigorous and careful experimental study has been carried out to determine the optimal configuration of FFNN model. Number of hidden layers, number of PEs, transfer functions and learning rules are varied. Optimum performance is obtained for FFNN model having single hidden layer with 10 PEs and transfer function lintah, learning rule momentum.

Table 7 shows the confusion matrix for FF NN. It is seen that except class 4 and 7 all classes are sensibly classified.

Table 7- Confusion Matrix for FFNN model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	451	0	1	7	1	3
Cotton crop	1	213	0	6	1	1
Very damp grey soil	0	0	336	12	14	1
Soil with vegetation stubble	3	9	22	203	5	2
Damp grey soil	0	2	95	3	141	35
Grey soil	6	0	10	2	22	323

Table 8 displays the performance parameters of FF NN model for various classes.

Table 8- Classification Accuracy of FF NN Model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.01439	0.01084	0.07616	0.03234	0.06338	0.03417
NMSE	0.07921	0.10572	0.41719	0.30475	0.73519	0.22294
MAE	0.07428	0.06642	0.14803	0.10189	0.15179	0.10378
r	0.96726	0.96199	0.79594	0.84438	0.59892	0.89269
Percent Correct	97.8308	95.0893	72.4138	87.1245	76.6304	88.4932

Focused Time Lag Recurrent Neural Network (FTLRNN)

Time lagged recurrent networks (TLRNs) are MLPs extended with short term memory structures. Most real-world data contains information in its time structure, i.e. how the data changes with time. TLRNs are the state of the art in nonlinear time series prediction, system identification and temporal pattern classification.

Recurrent networks are neural networks with one or more feedback loops. The time delay neural network (TDNN) memory structure is simply a cascade of ideal delays (a delay of one sample). The gamma memory is a cascade of leaky integrators. The Laguarre memory is slightly more sophisticated than the gamma memory in that it orthogonalizes the memory space. This is useful when working with large memory kernels [10].

The input PEs of an MLP are replaced with a tap delay line. It is called the focused TDNN. The topology is called focused because the memory is only at the input layer [11].

The delay line of the focused TDNN stores the past samples of the input. The combination of the tap delay line and the weights that connect the taps to the PEs of the first hidden layer are simply

linear combiners followed by a static nonlinearity. The first layer of the focused TDNN is therefore a filtering layer, with as many adaptive filters as PEs in the first hidden layer.

FTLRNN model performs optimally with Laguarre memory and 4 PEs in single hidden layer. Table 9 shows the confusion matrix for FTLRNN. It is observed that classes 1,2 and 3 are reasonably classified and classes 4 and 5 are poorly classified.

Table 9- Confusion Matrix for FTLRNN model with Laguarre memory

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	438	3	1	20	2	1
Cotton crop	7	205	2	17	2	2
Very damp grey soil	0	0	370	33	40	5
Soil with vegetation stubble	11	15	17	157	8	0
Damp grey soil	0	0	59	4	102	19
Grey soil	5	1	15	2	30	338

Table 10 shows the performance parameters of FTLRNN model.

Table 10- Classification Accuracy of FTLRNN Model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.02174	0.02004	0.06693	0.0521	0.0667	0.03474
NMSE	0.1196	0.19543	0.36663	0.49106	0.77372	0.22666
MAE	0.07428	0.06667	0.1537	0.12354	0.17097	0.09562
r	0.93856	0.89879	0.79667	0.71617	0.49244	0.87943
Percent Correct	95.0108	91.5178	79.7414	67.3819	55.4348	92.6027

Recurrent NN

Recurrent networks have feedback connections from neurons in one layer to neurons in a previous layer. Different modifications of such networks have been developed and explored. A typical recurrent network has concepts bound to the nodes whose output values feed back as inputs to the network. So the next state of a network depends not only on the connection weights and the currently presented input signals but also on the previous states of the network. The network leaves a trace of its behavior; the network keeps a memory of its previous states.

There are two models of recurrent network. Fully recurrent networks feed back the hidden layer to itself. Partially recurrent networks start with a fully recurrent net and add a feedforward connection that bypasses the recurrency, effectively treating the recurrent part as a state memory. These recurrent networks can have an infinite memory depth and thus find relationships through time as well as through the instantaneous input space. Most real-world data contains information in its time structure. Recurrent networks are the state of the art in nonlinear time series prediction, system identification and temporal pattern classification [12].

Hidden layer, number of PE and transfer function variations are attempted. It is seen that optimal performance is achieved for partially recurrent NN with 10 PEs and tanh transfer function in output layer.

Table 11 shows the confusion matrix for Recurrent NN. It is observed that except classes 4 and 5 all classes are reasonably classified.

Table 12 displays various parameters for recurrent NN model.

Table 11- Confusion Matrix for Recurrent NN model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	455	1	0	13	1	1
Cotton crop	0	208	1	16	0	0
Very damp grey soil	1	1	391	49	59	10
Soil with vegetation stubble	1	7	22	138	6	1
Damp grey soil	0	4	37	12	83	12
Grey soil	4	3	13	5	35	341

Table 12- Classification Accuracy of Recurrent NN Model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.01205	0.01088	0.07872	0.06426	0.07186	0.03782
NMSE	0.06629	0.10614	0.43123	0.60559	0.83354	0.24672
MAE	0.05933	0.05148	0.16664	0.13499	0.1498	0.10317
r	0.96702	0.94956	0.75793	0.63589	0.43866	0.87475
Percent Correct	98.6985	92.8571	84.2672	59.2275	45.10869	93.4247

Jordan Elman Neural Network

Recurrent networks are neural networks with one or more feedback loops. The Recurrent networks are used as input-output mapping networks and also as associative memories [13]. By definition, the input space of a mapping network is mapped onto an output space, a recurrent network responds temporarily to an externally applied input signal. Recurrent networks can be considered as dynamically driven recurrent networks. Because of global feedback memory requirement reduces significantly.

Jordan and Elman networks extend the multilayer perceptron with context units, which are processing elements (PEs) that remember past activity. Context units provide the network with the ability to extract temporal information from the data.

Optimal performance is obtained for Jordan Elman NN model having integrated axon with 0.8 context unit and 10 PEs..

Table 13 displays the confusion matrix for Jordan Elman NN. It is seen that except classes 4 and 7 all classes are reasonably classified.

Table 14 displays various parameters for Jordan Elman NN model.

Table 13: Confusion Matrix for Jordan Elman NN model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	452	0	1	8	0	2
Cotton crop	0	215	0	6	1	1
Very damp grey soil	0	0	305	15	15	1
Soil with vegetation stubble	4	8	37	0.0199	4	2
Damp grey soil	3	1	112	4	147	45
Grey soil	2	0	9	1	17	314

Table 14- Classification Accuracy of Jordan Elman NN Model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.00914	0.00889	0.08723	0.03804	0.07144	0.03357
NMSE	0.05029	0.08666	0.47783	0.35851	0.82875	0.21899
MAE	0.04436	0.03305	0.16083	0.09135	0.15526	0.09196
r	0.97508	0.95638	0.75987	0.81067	0.59531	0.89243
Percent Correct	98.0477	95.9821	65.7328	85.4077	79.8913	86.0274

Principal Component Analysis (PCA) NN

Unsupervised PCA at input is followed by a supervised MLP. It projects high dimensional redundant input data onto smaller dimension. Resulting outputs are orthogonal. PCA is a very well known statistical procedure that has important properties. Suppose that we have input data of very large dimensionality then it is to be projected onto a smaller-dimensionality space and later feature extraction is carried out. Projection always distorts the data to some extent. Obviously, this projection is to be done to small dimensional space while maximally preserving the dispersion (variance from a representation point of view) about the input data. The linear projection that accomplishes this goal is the PCA.

When variations in various parameters are tested it is seen that PCA NN model performs optimally for sangersfull learning rule and principal component 7 with 10 PEs.

Table 15 shows the confusion matrix for PCA NN. It is observed that except class 4 all classes are reasonably classified.

Table 16 displays various parameters for PCA NN model

Table15- Confusion Matrix for PCA NN model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	456	0	0	13	1	2
Cotton crop	1	217	0	6	1	1
Very damp grey soil	0	1	393	25	39	9
Soil with vegetation stubble	2	5	14	186	7	0
Damp grey soil	0	1	45	2	101	10
Grey soil	2	0	12	1	35	343

Table 16: Classification Accuracy of PCA NN model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.01124	0.00738	0.05586	0.02919	0.05382	0.03089
NMSE	0.06183	0.07206	0.30598	0.27508	0.62426	0.20157
MAE	0.04264	0.03626	0.12807	0.07624	0.12104	0.07703
r	0.96873	0.96422	0.83537	0.85162	0.61329	0.89515
Percent Correct	98.9154	96.875	84.6983	79.8283	54.8913	93.9726

Support Vector Machine (SVM)

The Support Vector Machine (SVM) is implemented using the kernel Adatron algorithm. The kernel Adatron maps inputs to a high-dimensional feature space and then optimally separates data into their respective classes by isolating those inputs which fall close to the data boundaries. Therefore, the kernel Adatron is especially effective in separating sets of data, which share complex boundaries.

Table 17 shows the confusion matrix for SVM model. It seen that all classes are reasonably classified except class 4.

Performance parameters of SVM model are displayed in table 18

Table 17- Confusion Matrix for SVM NN model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	438	0	0	7	0	2
Cotton crop	9	222	0	12	4	2
Very damp grey soil	0	2	421	12	66	2
Soil with vegetation stubble	8	0	20	201	1	8
Damp grey soil	4	0	7	0	79	8
Grey soil	2	0	16	1	34	343

Table 18- Classification Accuracy of SVM NN model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	1.38838	3.27564	0.44623	0.79387	0.42445	0.55895
NMSE	7.63933	31.9433	2.44441	7.48205	4.92358	3.64627
MAE	1.01185	1.41287	0.56271	0.70368	0.49127	0.62068
r	0.86403	0.86579	0.67537	0.59319	0.25583	0.64076
Percent Correct	95.0108	99.1071	90.7327	86.2661	42.9348	93.9726

Modular NN

Modular feedforward networks are a special class of MLP. These networks process their input using several parallel MLPs and then recombine the results. In contrast to the MLP, modular networks do not have full interconnectivity between their layers. Therefore, a smaller number of weights is required for the same size network (i.e. the same number of PEs). This tends to accelerate training times and reduce the number of required training exemplars.

Modular NN model performs optimally for single hidden layer with 10 PEs. Table 19 shows the confusion matrix for Modular NN. It is noticed that classes 1, 2 and 3 are reasonably classified and classes 4,5 and 7 are poorly classified.

Performance parameters of Modular NN model are displayed in table 20.

Table 19- Confusion Matrix for Modular NN model

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil(7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
Red soil	454	0	0	21	1	2
Cotton crop	0	211	0	16	1	1
Very damp grey soil	0	0	337	14	14	0
Soil with vegetation stubble	1	13	19	175	6	1
Damp grey soil	0	0	98	5	141	43
Grey soil	6	0	10	2	21	318

Table 20- Classification Accuracy of Modular NN model

Performance	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MSE	0.01221	0.010729	0.06642	0.03818	0.06566	0.03451
NMSE	0.06718	0.10462	0.36385	0.35989	0.76171	0.22513
MAE	0.05589	0.04295	0.1449	0.10374	0.16702	0.0939
r	0.97049	0.94656	0.81295	0.80006	0.56859	0.88353
Percent Correct	98.4816	94.1964	72.6293	75.1073	76.6304	87.1233

Comparison of NN models

Table 21 illustrates the classification accuracy for all the classes of land type for each NN model.

Table 21- Classification Accuracy of Various NN Models

Output / Desired	Red Soil (1)	Cotton Crop(2)	Very Damp Grey Soil (7)	Soil with Vegetation Stubble(5)	Damp Grey Soil (4)	Grey Soil (3)
MLPNN 36-19-6	96.74	95.98	80.17	90.13	75	87.4
FFNN 36-10-6	97.83	95.09	72.41	87.12	76.63	88.49
RBF, Cluster centers=100	95.66	88.39	92.02	40.77	10.869	96.44
PCA 36-10-6	98.91	96.87	84.69	79.83	54.89	93.97
Modular NN,36-10-6	98.48	94.19	72.62	75.11	76.63	87.13
Jordan Elman NN 36-10-6	98.48	95.08	65.51	85.83	80.43	85.47
FTLRNN 36-4-6	95.01	91.51	79.74	67.38	55.43	92.60
Recurrent NN 36-10-6	98.69	92.85	84.26	59.22	45.11	93.42
SVM	95.01	99.11	90.73	86.27	42.93	93.97

With thorough inspection of the Fig. 1, better average classification accuracy and optimum performance of the NN models have been obtained for MLP NN model.

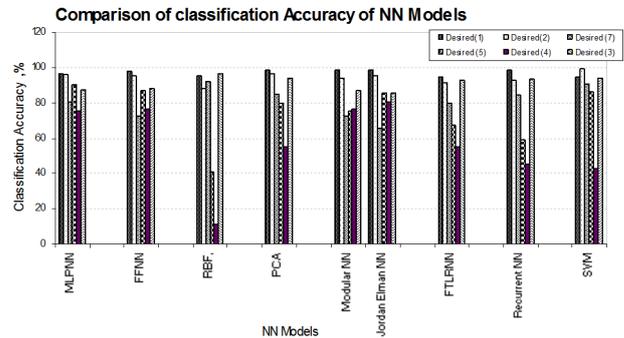


Fig. 1- Comparison of classification Accuracy of Various NN models

Table 22 displays the training time and average classification for each NN model. Though the performance of SVM model is comparable with MLP NN, very large time is required to train the SVM model. SVM NN model is implemented using kernel adatron algorithm and all other NN models are trained with backpropagation algorithm. Average classification accuracy is higher for MLP NN model than all other models.

Table 22- Time required for training the NN Models and Average Classification Accuracy

NN Model	No. of Epochs	Training time per epoch per exemplar in microseconds	Average Classification Accuracy (%)
MLPNN (36-19-6)	1000	39.9	87.57
FFNN (36-10-6)	1000	39.9	86.26
RBF, Cluster centers=100	1000	39.9	70.69
PCA (36-10-6)	1000	39.9	84.86
Modular NN, (36-10-6)	1000	39.9	84.03
Jordan Elman NN (36-10-6)	1000	39.9	85.13
FTLRNN (36-4-6)	1000	39.9	80.29
Recurrent NN (36-10-6)	1000	39.9	78.93
SVM	100	133	84.67

Conclusion

When all the NN models are exhaustively trained and tested it has been observed that consistently class 4 i.e. damp grey soil, is always misclassified in each NN model. Performance of MLP NN model is found superior to all other NN models. Overall 87.57% classification accuracy is achieved on testing. For red soil and cotton crop land the classification accuracy is achieved as 96.74% and 95.98% respectively. Class 5, soil with vegetation stubble and class 3, grey soil is classified to 90.13 % and 87.4%, respectively. Damp grey soil i.e. class 4 is poorly classified to 75%. SVM model performs equally well as compared to MLP NN model. The time required for training the NN models is almost same for all the NN models except SVM model. For SVM model training period required is 133 microseconds per epoch per exemplar whereas in other NN models it is about 39.9 microseconds per epoch per exemplar. This training period is relative with respect to computational speed and resources available. It is sure that MLP NN is outperforming amongst all the NN models.

References

- [1] Duda R.O. and Hart P.E. (1973) *Pattern classification and scene analysis*, Wiley, New York.
- [2] Fukunaga Keinosuke (1990) *Introduction to Statistical Pattern Recognition*, Academic Press.
- [3] UCI Repository, *Landsat Satellite Data*.
- [4] Mario Caetano (2009) *Image classification, ESA advanced training course on land remote sensing*, Prauge.
- [5] Demuth H. and Beale M. (1998) *Neural Network Toolbox: User's Guide Version 3.0*.
- [6] Cybenko G. (1989) *Approximation by superpositions of a sigmoidal function Mathematics of Control, Signals and Systems*, 2(4), 303-314.
- [7] Powell M.J.D. (1985) *IMA Conference on Algorithms for the Approximation of Functions and Data*, 143-167.
- [8] Light W. (1992) *Ridge Functions, Sigmoidal Functions and Neural Networks*," in E.W. Cheney, C.K. Chui and L L. Schumaker, eds, *Approximation Theory VII*, 163-206.
- [9] Principe J.C., Euliano N., Lefebvre W.C. *Interactive book Neural & Adaptive Systems: Fundamentals Through Simulations*.
- [10]Quin S.Z., Su H.T. and McAvoy T.J. *IEEE Transactions on Neural Networks*, 3(1), 122-30, 1992
- [11]Haykin S., *Neural Networks: A Comprehensive Foundation*.
- [12]Kasabov Nikola K., *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*.
- [13]Ham M. Fredric, Ivica Kostanic, *Principles of Neurocomputing for Science & Engineering*, Tata McGraw Hill.