Satellite Image Classification by Fuzzy Neural Network

Loay Kadam Abood
Fatima Bahjat Ibrahim
Department of Computer Science, College of Science, University of Baghdad, Baghdad-Iraq
Received: 3/16/2002
Accepted: 16/6/2004

Abstract
Implementation of Fuzzy Reasoning by structure of Neural Network method is used in this work. Weights of Neural Network is Fuzzy Reasoning. The FNN can automatically identify the Fuzzy Rules and tune the membership function by modifying the connection weights of Network using the back-propagation algorithm. Post-classification rules is used to reduce the number of isolated mis-classified pixels occur after the pixel-by-pixel classification principle.

1. Introduction
An advantage of using multisource spatial data is that additional features and spatial attributes can be incorporated in the classification. Ideally, each of the data sources will have unique information contributing to the classification process. A substantial difficulty for comprehensive analysis of multisource spatial data, however, arises from the conflicts and incompatibilities among the differences in measurement scales and feature distributions from the various data types. Therefore, a distribution-free and measurement scale-free classification technique is desirable for processing multisource spatial data.[4]

Neural network and fuzzy systems estimate input-output functions. Both are trainable dynamical systems. Unlike statistical estimators, they estimate a function without a mathematical model of how outputs depend on inputs. They are model-free estimators. They “learn from experience” with numerical and, sometimes, linguistic sample data.

Neural network and fuzzy systems encode sampled information in a parallel-distributed framework. Like brain, neural networks recognize patterns we cannot even define. We called this property recognition without definition. Recognition without definition characterizes much intelligent behavior. It enables systems to generalize [2].

The aim of this work is to adopt and design a Fuzzy Neural Network technique for remote sensing image classification. So the needs for an accurate, quick, and flexible method for image classification have led to the development of this work. Furthermore, post classification rules, which we have suggested have been used to reduce the mis-classified pixels for certain, classes, increase the accurate of classification.

In short, the specification, design, and implementation of this work are presented in a software package.

2. What Fuzzy is Logic?
Fuzzy theory holds that all things are matters of degree. It mechanizes our “folk psychology”. Fuzzy theory also reduces black-white logic and mathematics to special limiting cases of gray relationships. [2]

Fuzzy logic and neural networks have been combined in a variety of ways. In general, hybrid systems of fuzzy logic and neural networks are
often referred to as fuzzy neural networks. Fuzzy neural networks in the first category are basically fuzzy rule-based systems where fuzzy if-then rules are adjusted by iterative learning algorithm [3].

3. Implementation of Fuzzy Neural Network

Supervised-learning networks represent the main stream of the development in neural networks. Tow phases are involved in a supervised learning network: learning phase and retrieving phase, [7].

Classifying multsource remote sensing and spatial data requires the ability to match large volumes of input pattern data simultaneously to generate categorical information as output [5]. So the fuzzified features extracted from data implicitly in the network via its layers as describe below.

3.1 Network Architecture

Figure (1) shows the architecture of four layers; they are describe here:

Layer 1: (INPUT) the nodes in this layer just transmit input values to the next layer directly.

Layer 2: (FUZZIFICATION) if we use a single node to perform a simple membership function, then the output function of this node is this membership function. (e.g. bell shaped function like Gaussian Distribution Function); the weights here are the mean values, and standard deviation [6].

Layer 3: (INFERENCe RULE) this layer performs preconditon matching of fuzzy logic rules. Hence, the rule nodes should perform the fuzzy AND operation (mean the antecedent matching will be determine here).

Layer 4: (DEFUZZIFICATION) nodes here in down-up transmission mode, the link at layer four should perform the fuzzy OR operation (mean select the maximum value).

Referring to figure (1) there is n inputs, with n neurons (nodes) in the input layer, and c rules, with c neurons (nodes) in the inference and defuzzification layers. And there are n*c neurons(nodes) in the fuzzification layer. So the number of inputs n and number of classes c are determined the layout structure of the network[5].

3.1.1 Back-Propagation Algorithm:

The goal of Back-propagation algorithm is to minimize the error function.

\[ E(W) = \frac{1}{2} \sum_{i,c} (T_{i,c} - O_{i,c}(W))^2 \] ... (1)

Where

\[ O_{i,c}(w) : \text{Output of the jth node in input-output case c.} \]

\[ T_{i,c} : \text{Target of the jth node specified by the teacher.} \]

There are 2-passes in learning forward pass (we describe it above), and backward pass is used to compute \( \frac{\partial E}{\partial W} \) for all hidden nodes. Assuming that W is the adjustable parameter in a node (center of the membership function)[6]. The general learning rule is:

\[ \Delta W = -\frac{\partial E}{\partial W} \]

\[ W(t+1) = W(t) + \eta \left( \frac{\partial E}{\partial W} \right) \] ... (2)

where \( \eta \) is the learning rate, and

Layer 4: only error signals need to compute and propagate, no parameters adjust

\[ \delta_j = (t_j - O_j) \] where \( j = 1, 2, ..., c \) ... (3)

Layer 3: again only error signals need to compute and propagate, no parameters adjust

\[ \delta_i^3 = -\delta_i^4 \left( \frac{\sum_{j=1}^{c} O_j}{2} - O_i \right) \] ... (4)

Layer 2: the adjusted parameters here are mean and standard deviation of membership function by:

\[ m_i(t+1) = m_i(t) - \eta \delta_i^3 \exp(\text{net}_i) \ast 2 \frac{(x_i - m_i)}{\sigma_i^2} \] ... (5)

\[ s_i(t+1) = s_i(t) - \eta \delta_i^3 \exp(\text{net}_i) \ast 2 \frac{(x_i - m_i)}{\sigma_i^2} \] ... (6)

3.2 Post Classification:

After classification based on the principle of "pixel by pixel", the resulted image may contain some isolated mis-classified pixels; so we may suggest some rules to eliminate these mis-classified pixels; the classified image is stored, then we apply the rule to the classified image. These rules is specified for the chosen area and area has the same nature of this chosen area, example on these rules[5]:

251
• If $X_{o}$ is Tree AND ($X_{w_o}$ is deep water
OR $X_{w_o}$ is deep water) Then $X_{o}$ is shallow
water.

4. Implementation and Results

The work is tested using a LANDSAT-TM
image of Ramadi area recorded in 1989 of size
(512*512) pixels window of (8-bit) format for
bands (2-3-4). So the intensity of pixel is the
characteristical feature used.

The software is Graphical User Interface GUI
implementation for the theory described in this
work, it contain three main parts:

1. Feature extraction
2. Training
3. Testing (classification)

These steps run in sequence to get the
required output of this work (classified image), if
the user has an enough required information input
for any stage of the software, user can skip the
previous stage(s).

Feature extraction is the first subroutine
concerned with identifying the main input
parameters to the system, the parameters that
used to train the net, to classify the certain
image(s) there after in the program. Selection of
class will be visually by the mouse pointer, and
the output data file name will be chosen by the
user with the extension (.dat). The training sets
intensity range(0-255) should be normalized
between (0-1). This is necessary for neural
networks.

The software is also provided with the ability to
examine the displayed image (helping to
determine the number of classes) by clipping
(crop) part of it and viewing it's histogram,
changing it’s brightness, shrinking and stretching
it’s histogram, quantizing it's gray levels to
another gray levels, and visually subdividing
the image by a grid. In this stage the user must
determine the names of selected classes, that is
considered an important information in later part
of the system (post-classification).

We extract the characteristical features that
identify classes and name each class and save the
training sets in file with extension (.dat). Then
the training sets used to train the net, and feed the
net with the required input variables, for this
every they are:

- Learning rate = 0.001
- Momentum term = 0.95
- Number of bands = 3
- Number of classes = 7

The system reaches high accuracy about 95.238%
in about 183 iteration, this is high-speed
classification time, and good accuracy
classification. Although, another mechanism of
interest used here to reduce the classification
time, by skipping the phase of updating network
weight (fuzzy reasoning) for the corrected
classified vectors (training set)[5].

Learning rate represents updating steps toward
solution. Therefore, as the learning rate becomes
smaller the error updating is improved but in
small values. However, when the learning rate is
large, fluctuations in error could be seen for the
reason mentioned above. The criterion is to use
the total error of the network to change the
learning rate. When the total error decreases
relative to the total error of the cycle before, the
learning rate was kept constant. Otherwise, the
learning rate was decreased by a proper amount
defined by the user [1]. After training is finished
the weights of the net (mean and standard
deviation) is stored in a file with extension (.wtr);
these information is used later to classify
image(s). Classified image for this example is
shown in figure (2), it is also shows the
information of each class obtained from the
classification. Post-classification rules are applied
to enhance the result of classification (reduce the
number of isolated misclassified pixels). For this
part, we store the classified image with some
information in a file with extension (.cif), and
use this file to apply post-classification rules to
the classified image latter [5].
Figure (1) Fuzzy Neural Network Architecture

Figure (2) Classification
5. Conclusions
1. Using combination of fuzzy system and neural network which they are trainable dynamic systems that estimate input/output functions; i.e., estimate a function without any mathematical model and learn from experience with sample data, make the system flexible.
2. Also, the system is modular architecture with high flexibility, so it is able to change the number of bands and number of output classes without distributing the over all structure of the system.
3. Gaussian distribution function used is an efficient general distribution model; it is simple and optimal extraction of the data.
4. The system provide a user friendly interface software package, guiding the user step by step to examine the system, providing with help and error massage if the user make any mistake.

References
http://wwwsgi.ursus.maine.edu/gisweb/spatdb/ asem/