Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering Research

Computer Science and Engineering

Report Number: WUCS-93-17

1993-01-01

Supervised Competitive Learning with Backpropagation Network and Fuzzy Logic

Takayuki Dan Kimura, Thomas H. Fuller Jr., and Ce Wang

SCL assembles a set of learning modules into a supervised learning system to address the stability-plasticity dilemma. Each learning module acts as a similarity detector for a prototype, and includes prototype resetting (akin to that of ART) to respond to new prototypes. Here (Part I) we report SCL results using back-propagation networks as the learning modules. We used two feature extractors: about 30 energy-based features, and a combination of energy-based and graphical features (about 60). ACL recognized 96% (energy) and 99% (energy/graphical) of test digits, and 91% (energy) and 96% (energy/graphical) of test letters. In the accompanying paper (Part... Read complete abstract on page 2.

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Recommended Citation

Kimura, Takayuki Dan; Fuller, Thomas H. Jr.; and Wang, Ce, "Supervised Competitive Learning with Backpropagation Network and Fuzzy Logic" Report Number: WUCS-93-17 (1993). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/304

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

This technical report is available at Washington University Open Scholarship: https://openscholarship.wustl.edu/cse_research/304

Supervised Competitive Learning with Backpropagation Network and Fuzzy Logic

Takayuki Dan Kimura, Thomas H. Fuller Jr., and Ce Wang

Complete Abstract:

SCL assembles a set of learning modules into a supervised learning system to address the stability-plasticity dilemma. Each learning module acts as a similarity detector for a prototype, and includes prototype resetting (akin to that of ART) to respond to new prototypes. Here (Part I) we report SCL results using back-propagation networks as the learning modules. We used two feature extractors: about 30 energy-based features, and a combination of energy-based and graphical features (about 60). ACL recognized 96% (energy) and 99% (energy/graphical) of test digits, and 91% (energy) and 96% (energy/graphical) of test letters. In the accompanying paper (Part II), we report the results of SCL using fuzzy sets as learning moduels for recognizing handwritten digits.

Supervised Competitive Learning with Backpropagation Network and Fuzzy Logic

Takayuki Dan Kimura, Thomas H. Fuller, Jr. and Ce Wang

WUCS-93-17

April 1993

Laboratory for Pen-Based Silicon Paper Technology
Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130-4899
phone: (314) 935-6122; fax: (314) 935-7302
tdk@wucs1.wustl.edu

This research was partially supported by the Kumon Machine Project and by PenLab. The two papers included here were presented at ANNIE '92 (International Conference on Artificial Neural Networks in Engineering, St. Louis, MO, November 1992.)

Supervised Competitive Learning
Part I: SCL with Backpropagation Networks

Thomas H. Fuller, Jr.[2] and Takayuki D. Kimura
Department of Computer Science, Washington University in St. Louis

ABSTRACT

SCL assembles a set of learning modules into a supervised learning system to address the stability-plasticity dilemma. Each learning module acts as a similarity detector for a prototype, and includes prototype resetting (akin to that of ART) to respond to new prototypes. Here (Part I) we report SCL results using backpropagation networks as the learning modules. We used two feature extractors: about 30 energy-based features, and a combination of energy-based and graphical features (about 60). SCL recognized 98% (energy) and 99% (energy/graphical) of test digits, and 91% (energy) and 96% (energy/graphical) of test letters. In the accompanying paper (Part II), we report the results of SCL using fuzzy sets as learning modules for recognizing handwritten digits.

1. Introduction

When an adaptive learning system such as a backpropagation (BP) net is used to encode input patterns from an evolving environment, it suffers the the *stability-plasticity dilemma* formulated by Grossberg [Grossberg 1986] for the competitive learning paradigm: How can a learning system remain *plastic* in response to significant events and yet remain *stable* in response to irrelevant or routine events? How can it maintain previous knowledge while continuing to gain new?

An example application is handwritten character recognition. Suppose a system has been successfully trained to recognize the handwritten character "7" by a person who writes "7" consistently with two strokes (European style). Now, the same system is to be trained by another person who writes "7" with one stroke. After adapting to the one-stroke "7," it may not be able to recognize the two-stroke "7" as well as it used to. A similar problem arises when a system learns alphabetic characters after mastering numeric characters.

Adaptive resonance theory (ART) was proposed by Carpenter and Grossberg [Carpenter 1988] as a possible solution for the stability-plasticity dilemma in the competitive learning paradigm. It consists of two sets of processing nodes: the *attention* subsystem and the *orienting* subsystem. The nodes in the attention subsystem compete with each other when activated by an input pattern. The winning node represents the learned category of the input pattern and also carries the *prototype* (*attention*) pattern associated with the category. The orienting subsystem compares the prototype with the input, and if the two are significantly different, it resets (disables) the winning node for a new round of competition, with the assumption that the input pattern does not belong to a category represented by the current winner. If all prototype patterns in the attention subsystem are sufficiently different from the input, the input pattern itself becomes the prototype of a new node representing a new category. The degree of similarity is controlled by the *vigilance* parameter.

The ART model assumes no teaching input and performs unsupervised learning. It organizes itself to group "similar" input patterns into the same

2 Assistant Professor, Department of Mathematics and Computer Science, Principia College

category. Category proliferation is controlled by the vigilance parameter. An ART system with low vigilance will permit grouping of only grossly similar patterns, and a system with high vigilance will try to form separate categories for patterns that have only minor differences. In the ART2/BP network, Sorheim uses the ART2 [Carpenter 1987] model to build a supervised backpropagation network in his attempt to resolve the stability-plasticity dilemma [Sorheim 1991]. A simple backpropagation net is connected to each output unit of the ART2 subsystem. The competitive learning occurs in the ART2 subsystem, and no competition exists among the backpropagation nets.

We propose a scheme of compounding a set of learning modules into a supervised learning system called *Supervised Competitive Learning* (SCL). We use each learning module as a similarity detector for one prototype and adopt a prototype resetting mechanism (akin to that of ART) to create new prototypes. Any learning model can be used for component modules; backpropagation nets and pattern classification models based on fuzzy logic are two natural candidates.

Kohonen has advanced a related scheme in his topology-preserving maps in general [Kohonen 1982], and Linear Vector Quantization in particular [Kohonen 1988, 1990]. But LVQ processors are critically dependent on their neighbors to establish pairwise boundary surfaces, and consequently require more processing units (ten per category instead of the one to four prototypes typical of SCL). Also, SCL prototypes are topologically independent; a classification category may be represented by prototypes scattered widely over the feature space (cf. Section 4). In this respect our work is closer to that of Reilly *et al.* [1982], though employing different control mechanisms.

Our intended application for SCL is a handwriting recognition system for pen computers. Input patterns for such a system vary from alphanumeric characters to geometric shapes such as circles and rectangles. Users of pen computers also vary from young children to adults. Thus the system has to be open-ended; its implementation demands adaptive coding for a complex environment, embracing different character sets and different handwriting styles. The system is required to *remain* in the learning (training) mode. When the system mistakes the input pattern, the user is asked to enter the correct response and the system trains itself until it can recognize the same pattern consistently.

We assume that the environment consists of a set of categories (characters) and that each category has a set of subcategories (character prototypes). For example, a single stroke "7" and a double stroke "7" are subcategories of the category representing the numeral "7."

To demonstrate the utility of SCL, a simulator was constructed as a handwriting recognition system for a pen computer. In this work (Part I) we report the results of SCL simulation using backpropagation networks as the learning modules, SCL/BP. In the accompanying paper (Part II) we report the results of SCL simulation using fuzzy sets as learning modules, SCL/FZ, for the same problem of recognizing handwritten digits.

2. Supervised Competitive Learning (SCL) Model

The schematic definition of SCL is given in Figure 1. SCL receives the input pattern X and outputs the category name C. If the system fails to produce the correct category name, then the correct name is given to the system as the teaching value Y. The system learns the association between X and Y so that it may respond correctly to the input X next time.

An SCL system consists of a set of N (>0) *prototype units* (attention subsystem) and a *Selector* (orienting subsystem). Each prototype unit, n_i ($1 \le i \le N$), is responsible for identifying all the input patterns that belong to a particular subcategory (prototype). When the input pattern X is given, the output value, $n_i(X)$, from the unit n_i represents the *certainty* of X belonging to the subcategory of the unit. Or equivalently, it represents the *similarity* between the input pattern and the prototype pattern of that subcategory. We assume that the output of each prototype unit is normalized to [-1,1]; i.e., $-1 \leq n_i(X) \leq 1$.

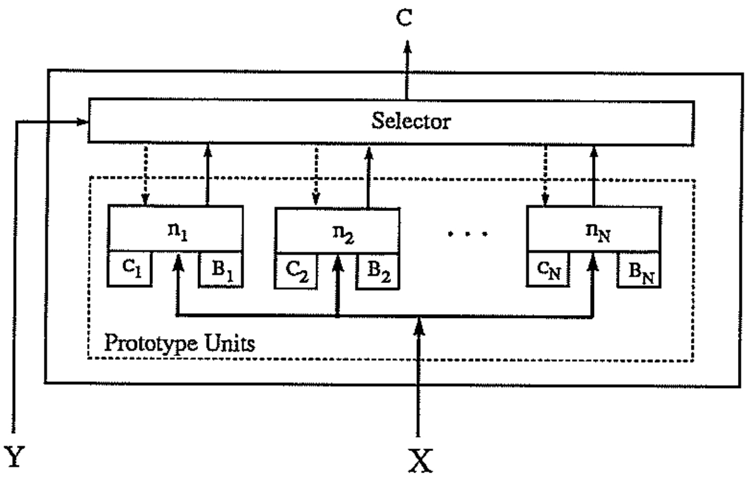

Figure 1: SCL Scheme

The Selector selects, as the winning unit, the unit whose output is larger than a threshold value and the largest among those above the threshold. Then it produces the name of the category, to which the winning unit belongs, as the output of the SCL system. If there is no winner, then the input X is considered to be a member of a new subcategory, and a spare (unused) prototype unit is assigned to represent it. If no spare unit is available, the unit winning least frequently, presumably representing the least significant subcategory of input patterns, will be assigned to represent the new subcategory.

If the winning unit is wrongly selected, the teaching value, Y, is used to train the prototype units as follows: The units of the category Y that respond to X with low outputs will be trained to increase their outputs for X up to σ_H. Those units representing categories other than Y that respond to X with high outputs will be trained to decrease their outputs down to σ_L. We assume that each unit is trainable to produce high output values for members of its subcategory and to produce low output values for non-members.

Associated with each prototype unit, n_i, the system maintains the following information: the name of the category, C_i, to which the subcategory belongs, a set of typical input patterns, B_i, that are selected by n_i, and the frequency, f_i, of n_i's winning the competition. Initially each unit has the *null* category name, Λ.

The algorithm for the Selector is given below:

Parameters: $-1 < \sigma_L < \rho < \sigma_H < 1$
Initialization: $C_i = \Lambda$, $f_i = 0$, $B_i := \phi$ (the empty set), for $1 \leq i \leq N$.

1. Get the input pattern X.
2. $K := \{ i \mid C_i \neq \Lambda \text{ and } n_i(X) > \rho \}$.
3. If $K = \phi$ then Produce Λ; Goto 7.
4. Find j such that $n_j(X) = \max\{ n_i(X) \mid i \in K \}$.
5. Produce C_j.
6. If accepted then Goto 1.
7. Get the correct category name Y.
8. $K := \{ i \mid C_i = Y \text{ and } n_i(X) > \rho \}$.
9. If $K = \phi$ then Goto 14.
10. Find j such that $n_j(X) = \max\{ n_i(X) \mid i \in K \}$.
11. Train n_j with $\{(X,1)\} \cup \{(u,1) \mid u \in B_j\}$ until $n_j(X) > \sigma_H$.

12. $B_j := B_j \cup \{X\}$; $f_j := f_j + 1$.
13. For all i such that $i \notin K$ and $n_i(X) \geq \sigma_L$,
    train n_i with $\{(X,-1)\} \cup \{(u,1) \mid u \in B_i\}$ until $n_i(X) < \sigma_L$; Goto 1.
14. $K := \{ i \mid C_i = \Lambda \}$.
15. If $K \neq \phi$ then Select $j \in K$; Goto 11.
16. Find j such that $f_j = \min\{ f_i \mid 1 \leq i \leq N\}$.
17. $f_j = 0$; $B_j := \phi$; Goto 11.

When the input pattern X is received, each unit predicts the certainty that X belongs to the unit's subcategory. The unit, n_j, with the largest output value greater than the *vigilance* ρ, is selected as the winner, and its category name C_j is given as the output. If the output is correct (Λ never being correct), no prototypes are changed. Otherwise, the system receives the correct category name Y. If there is no winner, but a unit remains with the empty name Λ, it becomes the winner with Y as its category. If there are no more units with Λ, then the unit with the smallest frequency count (of winning) will become the winner after re-initializing its settings. The winning unit updates its history set B_j by concatenating X to it and increments its frequency count f_j. Then the winning unit is positively trained on all the patterns in B_j until the output value $n_j(X)$ becomes greater than the *high confidence value*, σ_H, for the input pattern X. Those losing units n_i whose category name is not Y and whose output value $n_i(X)$ is greater than σ_L, get positively trained on all the patterns in B_i as well as negatively trained on X until $n_i(X)$ becomes less than the *low confidence value* σ_L. Losing units with category name Y are not trained.

3. SCL/BP: SCL with Backpropagation Nets

The first task for SCL/BP was to recognize handwritten digits, 0 - 9, collected on a pen-based Lombard computer from GO Corporation. The data is captured by the **x** and **y** movement of the pen on a digitizing tablet. A series of points is combined into a *stroke*, and a series of strokes is combined into a *scribble*. Details of the energy-based feature abstraction algorithm are in [Fuller 1992].

For example, the raw data for a five scribble is sampled by the tablet as:

```
108  5 2
15   103,56 103,51 102,47 102,41 102,37 107,40 116,40 119,39
     121,37 123,35 121,27 117,23 113,21 107,21 102,21
4    104,60 107,61 109,63 115,63
```

This scribble has the tag 108, value 5, and consists of 2 strokes. The first stroke consists of 15 points and the second one of 4 points. The velocity is divided into 7 sections and the acceleration is divided into 6 sections. The average X and Y components of velocity and acceleration for each section are divided by the maximum values to generate 26 floating point features, and the stroke count is used as the 27th feature.

SCL/BP is limited to 40 prototype units (N=40). Each unit is an acyclic backpropagation net [Kimura 1990] consisting of 3 layers; typically, 27 input units, 2 hidden units, and 1 output unit. The input layer is fully connected to both the hidden layer and the output layer. The hidden layer is fully connected to the output layer. There are no lateral connections, that is, no connections within the same layer. We used the activation and error functions of Kalman and Kwasny [1991, 1992], namely,

$$a = \lambda(x) = (1 - e^{-x}) / (1 + e^{-x}) \qquad \text{(activation function)}$$

$$e = \Sigma (c^2 / (\lambda'(x)) = \Sigma (c^2 / (1 - a^2) \qquad \text{(error function)}$$

c is the difference between the *training value* (1 if correct, -1 if incorrect) and the actual activation value of the unit. Note that the output value of the prototype unit ranges from -1 to 1. The learning rate and the momentum value (η and α in the table below) were typically fixed to 0.0005 and 0.4, respectively.

In SCL/BP each prototype unit keeps a set of randomly selected input patterns other than the members of its history set, to be used for smoothing the learning progression. Training of the prototype units, n_j , (Steps 11 and 13 of the Selector algorithm) utilizes the history set, B_j , the input pattern X, and the set of randomly selected input patterns, R_j, as negative exemplars.

4. Experiments

Early experiments used *only* energy-based features in SCL/BP to recognize 1400 handwritten digits (600 test digits after *Trials* training on 800 training digits, collected from 20 subjects) collected on the Lombard computer. A trial is one backpropagation for one prototype unit. In the first experiment below, for example, since 19 prototypes in total were generated, each unit was trained an average of 22,000 times. After the training iterations, the system was tested by 600 digit patterns obtaining 92.7% recognition. The column *Time* is training time on a NeXTstation (25 MHz MC68040-based).

Several later experiments using a combination of 57 to 63 real-valued features (stroke count, 20 to 26 energy-based features, and 36 static features based on position and orientation of scribble segments). Recent experiments with our actual pen-based prototype (Kumon Machine, or KM) used 1600 digits (1280 train, 320 test) and 4160 letters (3328 train, 832 test). Improvements to the preprocessing of feature extraction and the reduction to two hidden units significantly sped up learning convergence. The 63 features, training set, and testing set are identical to those used for SCL fuzzy logic digit training reported in "Supervised Competitive Learning, Part II: SCL with Fuzzy Logic," permitting comparison of SCL under the two regimens.

SCL Parameters η	σ_H	ρ	σ_L	Hidden units	Prototypes created	Trials	Time (min)	Recognition
(GO-collected digits -- Energy only)								
.0005	0.5	-0.10	-0.5	4	19	211,761	<20	92.7%
.0005	0.5	-0.10	-0.5	4	26	2,791,989	<250	94.8%
(GO-collected digits -- Energy/Graphical)								
.0010	0.3	0.05	-0.5	4	35	648,287	32	97.5%
(KM-collected digits -- Energy only)								
.0010	0.9	0.10	-0.5	2	39	712,466	9	98.4%
(KM-collected digits -- 63 Energy/Graphical features)								
.0010	0.8	0.10	-0.5	2	29	65,921	2	97.5%
.0010	0.9	0.05	-0.5	2	35	906,315	19	99.7%
(KM-collected letters -- Energy only)								
.0010	0.9	0.10	-0.5	2	144	3,383,493	<200	91.1%
(KM-collected letters -- Energy/Graphical)								
.0010	0.9	-0.10	-0.5	2	139	2,079,554	139	95.2%
.0010	0.9	-0.05	-0.5	2	143	3,271,052	181	95.7%

Importantly, the system exhibits a high correlation between "confidence" (difference between the first and second highest output activations divided by the maximum, 2.0) and correctness. This confidence is greater than 0.35 for about half of the test set of handwritten lower case letters. For these items, SCL/BP is 99.0% correct. Confidence is greater than 0.2 for 78% of the test data, and the system is 98.3% correct for these.

Following are several examples of members of the training set selected by various prototype units as being "similar".

One unit selected single-stroke fives; another assembled double-stroke fives:

One prototype unit gathered single-stroke eights, and another garnered double-stroke eights with an interesting exception. Eights that began in the middle of the two loops, but that were drawn with a single continuous stroke were also collected by this adventurous unit.

5. Conclusions and future work

The experiments (99% for digits and 96% for letters, and comparable results for gestures) demonstrate the success of SCL in negotiating the stability/plasticity dilemma. SCL/BP adapts to new environments without losing the recognition capability obtained in the old environment. We are currently using SCL/BP with on-line training of unrecognized characters. This adaptability leads to desirable collaboration between the user and the pen computer. The next phase of our SCL evaluation is to continue training SCL/BP to recognize alphabetic characters and various gesture command sets for pen computers.

6. References

Carpenter, G. A. and Grossberg, S. (1987). ART2: Self-Organizing of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics*, 4919-4930.

Carpenter, G. A. and Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer* 21:3, 77-88.

Fuller, T. (1992). Energy-related Feature Abstraction for Handwritten Digit Recognition. *Proceedings of the Fourth Midwest Artificial Intelligence and Cognitive Science Conference*, May 1992, 71-76.

Grossberg, S. (1986). Competitive Learning: From Interactive Activation to Adaptive Resonance. *Cognitive Science* 11, 23-63.

Kalman, B. L. and Kwasny, S. C. (1991). A Superior Error Function for Training Neural Nets. *Proceedings of the International Joint Conference on Neural Networks*, Seattle, Washington, 1991, Vol. 2, 49-52.

Kalman, B. L. and Kwasny, S. C. (1992). Why Tanh: Choosing a Sigmoidal Function. *Proceedings of International Joint Conference on Neural Networks*, Baltimore, MD, June 1992.

Kimura, T.D. (1990). An Algebraic Proof for Backpropagation in Acyclic Neural Networks. *Proceedings of International Joint Conference on Neural Networks 1990*, San Diego, CA, June 1990, 554-557.

Kohonen, Teuvo (1982). Self-Organized Formation of Topologically Correct Maps. *Biological Cybernetics*, Vol 43, 59-69.

Kohonen, Teuvo (1988). An Introduction to Neural Computing. *Neural Networks*, Vol.1, 3-16.

Kohonen, Teuvo (1990). The Self-Organizing Map. *Proceedings of the IEEE*, 78:9, 1464-1478.

Reilly, D. L., L. N. Cooper, and C. Elbaum. A Neural Model for Category Learning. *Biological Cybernetics* Vol. 45, 1982, 35-41.

Sorheim, E. (1991). ART2/BP Architecture for Adaptive Estimation of Dynamic Processes. *Advances in Neural Information Processing Systems*, Vol. 3 ed. by D. Touretzky, 169-75.

Supervised Competitive Learning
Part II: SCL with Fuzzy Logic

Ce Wang, Takayuki D. Kimura, and
Thomas H. Fuller, Jr.[3]

Department of Computer Science, Washington University in St. Louis

ABSTRACT

Supervised Competitive Learning (SCL) is described in an accompanying paper [1]; SCL assembles a set of learning modules into a supervised learning system to address the stability-plasticity dilemma. That paper reported results using backpropagation networks as the learning modules (SCL/BP). Here (Part II) we report SCL results using learning modules based on fuzzy logic (SCL/FZ). Although its learning algorithm is very different from that of backpropagation networks, fuzzy logic also suffers the stability-plasticity dilemma. A simulator on handwritten digit and gesture recognition was constructed to demonstrate the utility of SCL/FZ; it recognized 98% of test digits, and 91% of test gestures. In this paper, we also compare SCL/BP with SCL/FZ for recognizing handwritten digits.

1. Introduction

A major problem in developing pen-based computers is handwriting recognition. Pen-based recognition technology is generally based on pattern matching, and its performance is not yet satisfactory for practical uses. Therefore, systems based on other technologies, such as fuzzy logic and neural networks, are currently studied.

A vector-based handwritten character recognition system takes a sequence of pen-location points as input. The great variations of handwritten characters makes it difficult to create a set of standard reference characters. Some automatic feature abstraction algorithms provide the way to find a set of features necessary to classify each input patterns. But in general, these features are *fuzzy* in nature.

Fuzzy logic is a mathematical approach to deal with fuzziness, i.e. fuzzy features and fuzzy decision rules. Its statistical methods also deal with inexact information, but they rely on probability density functions some of whose parameters must be estimated from large experimental data sets. Offhand assumptions within these functions would not provide a proper empirical basis for our application. The fuzzy logic approach serves as a numerical model-free estimator. They estimate a function without requiring a mathematical description of how the output functionally depends on the input, that is, they "learn from samples." The freedom from models is the key advantage over the traditional statistical approaches. It is felt that fuzzy logic provides a useful approach to pattern classification, particularly, in problems having imprecisely defined input patterns and a small number of samples, and where statistical independence can not be assumed.

Like some other adaptive learning systems, such as a backpropagation neural networks, a fuzzy learning system also suffers the stability-plasticity dilemma [Grossberg, 1986]: How can the learning system prevent new learning from washing away prior learned knowledge? A compounding system of Supervised Competitive Learning that resolves the stability-plasticity dilemma is explored in

[3] Assistant Professor, Department of Mathematics and Computer Science, Principia College

the accompanying paper (Part I). Based on the same learning scheme, instead of using the backpropagation networks, we propose an SCL algorithm using fuzzy sets in the learning modules (SCL/FZ). We adaptively update prototypes according to sample data. For each unknown input pattern, the system applies a membership function to measure a set of feature similarities to the existing prototypes. Then the system maps this array of membership function values to a character similarity measure and so classifies the input pattern. If the input pattern is not found similar to any existing prototypes, the system creates a new subclass with the label specified by the user.

Due to the variations in writing styles, two characters belonging to the same character class may have very different features. For example, some people write the numeral "5" with a single stroke while some others write it with two strokes. So we need to divide each character class into several subclasses. Even for each subclass, a single pattern may not be sufficient to describe that subclass. Multiple patterns are sometimes necessary. But how many writing styles are there for each character class? We don't know in advance. Although we could sample a large set of handwriting data, it is impossible to exhaust all the possibilities. In general, prototypes are generated based on feature clusters. Deterministic clustering algorithms generate class partitions such that each pattern is assigned to exactly one cluster. However, a pattern is often "between" clusters. It could belong to one cluster nearly as well as to another. Only fuzzy clustering methods can yield more accurate representations of real data objects. A fuzzy clustering algorithm (from Ruspini and Bezdek) is discussed in [Pao, 1989]. This algorithm requires us to input the number of current cluster centers. But we are dealing with an evolving environment. No number of cluster centers can be preconceived. SCL/FZ provides an approach to determine feature clusters by similarity measures and we use the ensemble average of the many values obtained for a particular feature of a particular cluster.

Another challenge in fuzzy systems is the design of membership functions. Fuzzy logic serves as a similarity detector. Basically it maps an input pattern to the degree of membership of a class by means of the membership function. The membership function is defined as $\mu_A: X \rightarrow [0, 1]$, where X is a collection of input patterns. But what is its exact formula? Some applications such as the fuzzy system for handwritten Chinese character recognition [Cheng et al., 1989] use exponential functions. Another typical formula of the membership function is

$$\mu = (1 + | (p_i - p) / E |^F)^\alpha \qquad (1\text{-}1)$$

where p represents an input pattern, p_i represents the corresponding value of the ith prototype in a fuzzy recognition system, and E, F, and α are three parameters to be determined. Notice that no matter what formula is used, there are some parameters to be determined. Is there a principle for the parameter determination? A method based on the class separability measure is presented in [Cheng et al., 1989]. This method again needs to employ some probability density functions; for example, the class conditional probability density functions. We select the membership functions in the form of equation (1-1) and determine the parameters E, F, and α empirically.

This work is organized into three major parts. In the next section we describe the SCL/FZ recognition system. In section 3, we report the results of experiment on handwritten character recognition. In the last section, we compare the performances of two learning systems: SCL/FZ and SCL/BP in terms of recognition rate and computational complexity.

2. SCL/FZ: SCL with Fuzzy Logic

The SCL model is defined in the accompanying paper (Part I). Here we briefly review it to clarify later discussion. The basic idea of the compounding system of

SCL is to use each learning module as a similarity detector for a prototype and adopt the prototype resetting mechanism [Carpenter 1988] to create new prototypes. In the SCL/FZ system, we use the learning scheme shown in Figure 2.1. Now, instead of using the backpropagation networks, we use fuzzy sets as learning modules. Due to the difference between the fuzzy logic training and the neural network training, the algorithm for the selector is slightly different. We rewrite the algorithm to relate it to the fuzzy system below.

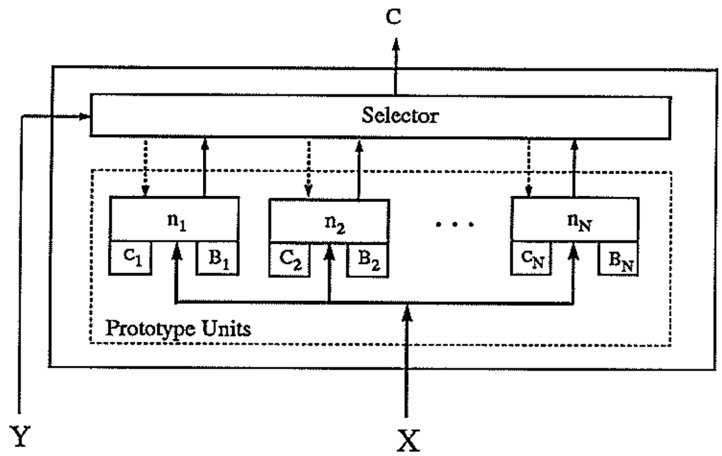

Figure 2.1 SCL Scheme

Note that B_i here is defined a little differently from that in the accompanying paper (Part I). In addition to a set of typical patterns, B_i carries an array of cluster centers of these patterns. We consider B_i the prototype maintained by prototype unit n_i to represent subcategory C_i. The algorithm is:

Parameters: $0 < \rho < 1$
Initialization: $C_i = \Lambda$, $f_i = 0$, $B_i := \phi$ (the empty set), for $1 \le i \le N$.

1. Get the input pattern X.
2. $K := \{ i \mid C_i \ne \Lambda$ and $n_i(X) > \rho \}$.
3. If $K = \phi$ then Produce Λ; Goto 7.
4. Find j such that $n_j(X) = \max\{ n_i(X) \mid i \in K\}$.
5. Produce C_j.
6. If accepted then Goto 1.
7. Get the correct category name Y.
8. $K := \{ i \mid C_i = Y$ and $n_i(X) > \rho\}$.
9. If $K = \phi$ then Goto 13.
10. Find j such that $n_j(X) = \max\{ n_i(X) \mid i \in K\}$.
11. Modify B_j by adding X to it; $f_j := f_j + 1$.
12. For all i such that $i \notin K$ and $n_i(X) \ge \rho$, modify B_i such that $n_i(X) < \rho$; Goto 2.
13. $K := \{ i \mid C_i = \Lambda\}$.
14. If $K \ne \phi$ then Select $j \in K$; Goto 11.
15. Find j such that $f_j = \min\{ f_i \mid 1 \le i \le N\}$.
16. $f_j = 0$; $B_j := \phi$; Goto 11.

Because it is not clear how to modify B_i to let $n_i(X) < \rho$, the algorithm currently in use skips step 12 (add "Goto 2" to the end of step 11). The negative training (step 12) remains an interesting unsolved problem. Also, in current experiments, we are not using the variable f_i to record the occurrence frequency of the pattern belonging to subcategory C_i. When the system exhausts the capacity to adopt new patterns, it simply exits the learning process.

Fuzzy Logic Learning Module: The digitizing tablet collects the number of strokes followed by a sequence of x,y coordinates when a character is drawn on its writing surface. We apply the feature abstraction algorithm to the pen data to get the feature vector of the input character and use it as the input pattern X to the recognition system. When X is compared with the existing prototypes, we first calculate the feature similarities of X to each prototype and then map them to a character similarity measure.

The framework of a speech identification system [Pal and Majunder, 1977,1978] underlies the design of our fuzzy learning modules. The issue is how to weight the total effect of an array of membership function values to make a fuzzy decision on pattern classification. Their system demonstrated that the pattern classification can be done by a similarity measure defined somewhat arbitrarily in terms of a ratio of the membership functions of a pattern and of cluster prototypes.

Now the pattern in question is X in the form of an array of features. Assume that X is an L-dimensional vector, the size of which should be much smaller than that of the raw data. First, we transfer X to an array of the membership function values, that is,

$$P_j(X) = (p_{1j}, p_{2j}, ..., p_{Lj}) \tag{2-1}$$

where the values of p_{ij} lie in the interval [0,1]. $P_j(X)$ can be considered as the feature similarity measure of X to the prototype B_j. And each element of $P_j(X)$ is obtained by the membership function formulated as

$$p_{ij} = (1 + | (x_i - a_{ij}) / E |^F)^{\alpha} \tag{2-2}$$

where,

p_{ij} denotes the degree to which the feature i is possessed by the prototype B_j,
x_i is the value of feature i in pattern X,
a_{ij} is the cluster center value of feature i of the prototype B_j,
E and F are positive constants to be determined, and,
α is a constant, usually with the value -1.

Assume that there are N subcategories C_1, C_2, ..., C_j, ..., C_N in the recognition system, and B_j carries H_j patterns for each subcategory C_j. Let $B_j^{(h)} = (b_{1j}^{(h)}, ..., b_{Lj}^{(h)})$ be the hth pattern in B_j, where h = 1, ..., H_j. We transfer each of these patterns into the fuzzy references represented in terms of membership function values, i.e.

$$R_j^{(h)} = (r_{1j}^{(h)}, r_{2j}^{(h)}, ... , r_{Lj}^{(h)}) \tag{2-3}$$

where $r_{ij}^{(h)}$ denotes the degree to which the property i is possessed by the hth pattern of B_j, and

$$r_{ij}^{(h)} = (1 + | (b_{ij}^{(h)} - a_{ij}) / E |^F)^{\alpha} \tag{2-4}$$

In both (2-2) and (2-4), we need to know the value a_{ij}. As we have mentioned, the system prototypes are determined by the feature clusters. But it may not be possible for us to obtain these feature clusters by applying the fuzzy clustering algorithm mentioned in section 1 because no number of cluster centers can be preconceived. So in the design of our system, we may consider the above a_{ij} as the ensemble average of the many values obtained for the ith feature of the prototype B_j, i.e.

$$a_{ij} = (1 / H_j) \sum_{h=1}^{H_j} b_{ij}^{(h)} \tag{2-5}$$

The data for the computation of $P_j(X)$ and $R_j^{(h)}$ are tabulated in Table 2.1.

Table 2.1 The data for the computation of $P_j(X)$ and $R_j^{(h)}$.

B_j:	$B_j^{(1)}$:	$b_{1j}^{(1)}$	...	$b_{ij}^{(1)}$	...	$b_{Lj}^{(1)}$
	$B_j^{(2)}$:	$b_{1j}^{(2)}$	...	$b_{ij}^{(2)}$	...	$b_{Lj}^{(2)}$
	...	...	...	...	...	...
	$B_j^{(Hj)}$:	$b_{1j}^{(Hj)}$	...	$b_{ij}^{(Hj)}$	...	$b_{Lj}^{(Hj)}$
	X:	x_1	...	x_i	...	x_L

The similarity vector $S_j(X)$ for the pattern X with respect to the subcategory C_j is defined to be

$$S_j(X) = (s_{1j}, ..., s_{Lj}) \tag{2-6}$$

where

$$s_{ij} = (1/H_j) \sum_{h=1}^{H_j} s_{ij}^{(h)} \tag{2-7}$$

The values $s_{ij}^{(h)}$ are obtained from two membership function values p_{ij} and $r_{ij}^{(h)}$ through the relationship

$$s_{ij}^{(h)} = (1 + w_i \mid 1 - p_{ij} / r_{ij}^{(h)} \mid) ^{-2} \tag{2-8}$$

The numerical value of s_{ij} denotes the degree of similarity of the ith feature of X with that of C_j. In (2-8) the w_i are positive constants that can be individually tailored to indicate the relative sensitivity of the classification to deviations from the prototype values.

The similarity value of the input pattern X to the prototype B_j representing the subcategory C_j is defined to be the length of $S_j(X)$. We compute the normalized similarity values, which reside in [0,1], from each similarity vectors by

$$\mid S_j(X) \mid =((1/L) \sum_{i=1}^{L} s_{ij}^2)^{1/2} \tag{2-9}$$

The closed form of $\mid S_j(X) \mid$ is

$$\mid S_j(X) \mid = ((1/L) \sum_{i=1}^{L} ((1/H_j) \sum_{h=1}^{H_j} (1+w_i \mid 1- p_{ij}/r_{ij}^{(h)} \mid) ^{-2z})^2)^{1/2} \tag{2-10}$$

where j = 1, ..., N. We use $\mid S_j(X) \mid$ as the $n_j(x)$ for the SCL/FZ and select the winning subcategory C_j for X by the decision rule,

$$\mid S_j(X) \mid = \max(\mid S_i (X) \mid \mid i = 1, ..., N) \tag{2-11}$$

3. Experiments

The experimental results on SCL/FZ for handwritten digit recognition are tabulated in Figure 3.1.

Parameters α	E	F	z	Confidence Value ρ	Number of Prototypes	Training Time (Min.)	Recognition Rate Train(800)	Test(600)
-1	1	1	1	0.8	88	4	96.62%	95.67%
-1	1	1	1	0.85	88	4	96.88%	96.83%
-1	1	1	1	0.9	157	11	99.50%	97.17%

Figure. 3.1 Handwritten Digit Recognition Results: SCL/FZ

4. Comparison and Conclusion

The comparison of performances of the SCL/FZ and SCL/BP is illustrated in Figure 4.1. There is no significant difference in computational complexity between the two systems. The SCL/BP system achieved higher recognition rates because of negative training. The experimental results from both systems demonstrate that the SCL is a powerful technique for resolving the stability-plasticity dilemma. Our next research is to introduce negative training into SCL/FZ. We believe that it will significantly improve the performance of the fuzzy recognition system.

System	Number of Prototypes	Training Time (Min.)	Recognition Rate (Testing Data)
SCL/BP	37	4	98.1%
SCL/BP	35	19	99.7%
SCL/FZ	88	4	96.8%
SCL/FZ	157	11	97.2%

Figure 4.1 Comparison on SCL/FZ and SCL/BP

5. References

[1] Thomas H. Fuller, Jr. and Takayuki Dan Kimura, "Supervised Competitive Learning Part I: SCL with Backpropagation Networks," Submitted to ANNIE'92, St. Louis, MO, 1992.

[2] Grossberg, S. "Competitive Learning: From Interactive Activation to Adaptive Resonance. Cognitive Science 11, 1986, 23-63.

[3] Yoh-Han Pao, "Adaptive Pattern Recognition and Neural Networks," Addison-Wesley Publishing Company, Inc., 1989.

[4] Bart Kosko, "Neural Networks and Fuzzy Systems," Prentice-Hall, Inc., 1992.

[5] Fang-Hsuan Cheng, Wen-Hsing Hsu, and Chien-An Chen, "Fuzzy Approach to Solve the Recognition Problem of Handwritten Chinese Characters," Pattern Recognition, Vol. 22, No. 2, 1989, pp. 133-141.

[6] L. A. Zadeh, "Fuzzy Sets," Information and Control, Vol. 8, 1965, pp. 338-353.

[7] L. A. Zadeh, "Fuzzy Logic," Computer, Vol. 21, No. 4, April, 1988, pp. 83-93.

[8] S. K. Pal and D. D. Majunder, "Fuzzy Sets and Decisionmaking Approaches in Vowel and Speaker Recognition," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, August, 1977, pp. 625 ~ 629.

[9] S. K. Pal and D. D. Majunder, "On Automatic Plosive Identification Using Fuzziness in Property Sets," IEEE Transactions on Systems, Man, and Cybernetics, Vol.8, 1978, 302-7.