Report Number: WUCS-93-43

1993

Supervised Competitive Learning Part I: SCL with Backpropagation Networks

Thomas H. Fuller Jr. and Takayuki D. Kimura

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). 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... **Read complete abstract on page 2.**

Supervised Competitive Learning Part I: SCL with Backpropagation Networks

Thomas H. Fuller Jr. and Takayuki D. Kimura

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). 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 moduels for recognizing handwritten digits.

Supervised Competitive Learning Part I: SCL with Backpropagation Networks

Thomas H. Fuller, Jr. and Takayuki D. Kimura

WUCS-93-43

September 1993

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis, Missouri 63130-4899

Reprinted from Intelligent Engineering through Artificial Neural Networks, Volume 2. New York: ASME Press, 1992, C. Dagli et al ed., pages 185-190. This work supported by the Kumon Machine Project.

Supervised Competitive Learning
Part I: SCL with Backpropagation Networks

Thomas H. Fuller, Jr. and Takayuki D. Kimura

WUCS-93-43

Reprinted from *Intelligent Engineering through Artificial Neural Networks, Volume 2. New York: ASME Press, 1992.*
Editor: C. Dagli et al.
(pages 185-190)

November 1992

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis, MO 63130-4899

This work was supported by the Kumon Machine Project.

Supervised Competitive Learning
Part I: SCL with Backpropagation Networks

Thomas H. Fuller, Jr.[1] **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 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). 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

---

[1] 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., $-1 \leq n_i(X) \leq 1$.

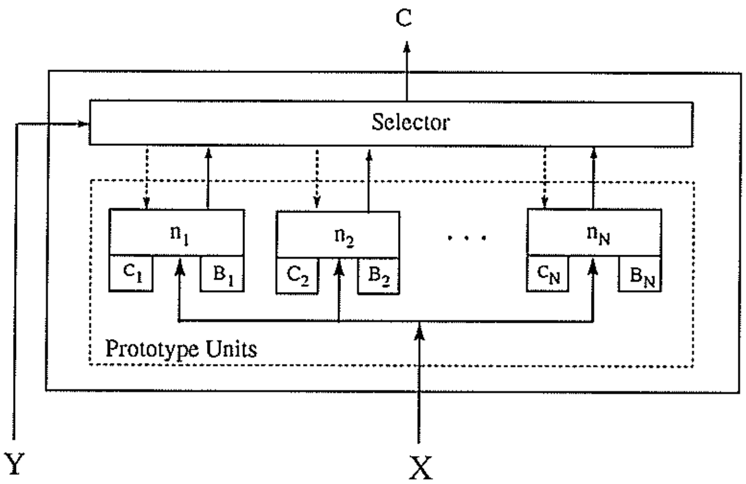


Figure 1: SCL Scheme

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.

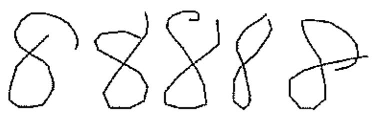
SCL Parameters η	σ_H	ρ	σ_L	Hidden units	Prototypes created	Trials	Time (min)	Recognition
(GO-collected digits -- Energy only)								
.0005	0.5	-0.10	-0.5	4	19	211,761	<20	92.7%
.0005	0.5	-0.10	-0.5	4	26	2,791,989	<250	94.8%
(GO-collected digits -- Energy/Graphical)								
.0010	0.3	0.05	-0.5	4	35	648,287	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.