음성 인식을 위한 신경회로망 접근과 동향.

°김 순 협, 이 성 관
광운대학교 컴퓨터 공학과

Neural Network Approaches and Trends for Speech Recognition

Soon-Hyob Kim, Seong-Kwon Lee

Computer Engineering Dept. KwangWoon Univ.

ABSTRACT

We proposed the approach method of neural network for Signal Processing, especially speech signal processing and reviewed the algorithms for several neural networks which are used for many alppication field in speech processing. Finally, investigated the trends in neural network method through 3 confernce jounal and the ASK (AcousticalSociety of Korea) jounal in 1994.

1. Introduction

Neural network have become a very popular field of research in cognitive science, neurobiology, computer engineering/science, signal processing, optics, and physics. They present a very broad range of neural processing models. An artificial neural network is an abstract simulation of a real nervous system that contains a collection of neuron units communicating with each other via axon connections. The first fundamental modeling of neural nets was proposed in 1943 by McCulloch and Pitts in terms of a computational model of "nervous activity". The McCulloch-Pitts neuron is a binary device and each neuron has a fixed threshold, thus performing simple threshold logic. The McCulloch-Pitts model lead the works of John Neumann, Marvin Minsky, Frank Rosenblatt, and many others. Hebb postulated that the neurons were appropriately interconnected by self-organization and that "an existing pathway strengthens the connections between the neurons". The neural models can be divided into two categories: 1) The biological type. 2) The application-driven.

1.1 Biological-Type Neural Networks

Tha main objective of biological-type neural nets is to develope a synthetic element for verifing hypotheses concerning biological systems. A simplified sketch of a natural neural network is illustrated in Figure 1. There are three parts in a neuron : (1) a neuron cell body (2) branching extensions called dendrites for receiving input, and (3) an axon that carries the neuron's output to the dendrites of other neurons. The synapse represents the junction between an axon and a dendrite. The process of neurons is often modeled as a propagation rule represented by a net value u(·), cf.

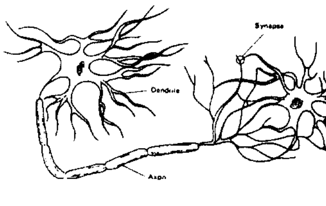

Figure 1. A simplified sketch of biological neurons

Figure 2(a). The neuron function can be modeled as a simple threshold function $f(\cdot)$.

1.2 Application-Driven Neural Networks

In general, neurons and axons are mathematically modeled by activation functions and net functions(or basis function) respectively, cf. Figure 2(b). The selection of these functions often depends on the applications the neural models are for. The strength of application-driven neural networks hinges upon three main characteristics: (1) Adaptiveness and self-organization (2) Nonlinear network processing (3) Parallel processing. These characteristics have played an important role in neural network's applicabilities to signal processing and analysis.

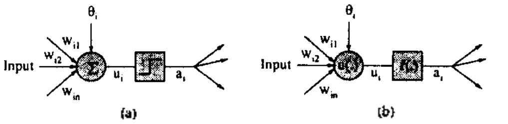

Figure 2. (a) A simplfied neural model witth linear net function and a threshold neuron function.
(b) A general neural model
Net function $u(\cdot)$ Activation function $f(\cdot)$

2. Applications, Algorithms and Architectures

In order to have an integrated understanding on neural networks, we review applications, algorithm and architecture.

2.1 Application Paradigms of Neural Models

The application domains of neural nets can be roughly divided into the following categories: (1) association / clustering / classification (2) pattern completion (3) regression/generalization (4) optimization. Their mathematical formulations are summarized in Table 1.

2.1.1 Association, Clustering and Classification

In this paradigm, input static patterns or temporal signals are to be classified or recognized as shown in Figure 3.

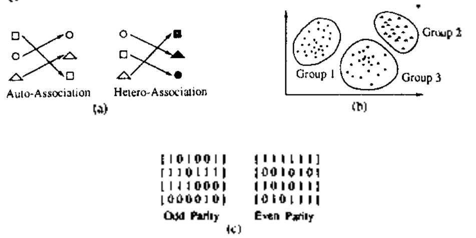

Figure 3. Classification :
(a) association : auto-association and hetero-association
(b) unsupervised classification
(c) supervised classification

(1) Association : Of special interest are the two association formulations auto-association and hetro-association, shown in Figure 3(a). The auto-association problem is to retrieve the complete pattern, given partial information of the desired pattern. The hetro-association is to retrieve a corresponding pattern in set B, given a pattern in set A. The weights in associative networks are often predetermined based on a Hebbian-type formulation.
(2) Unsupervised Clustering : The synaptic weights of the network are trained by an unsupervised learning rule, that is, the network adapts the weights and verifies the result based exclusively on the input patterns.
(3) Supervised Classification : In many classification applications, for example, speech recognotion, the training data consist of pairs of input/output patterns. In this case, it is more advantageous to adopt supervised networks such as the well-known back-propagation network.

2.1.2 Pattern Completions

There are two kinds of pattern completion problems: temporal and static. Most conventional multilayer nets, Boltzmann machines, and Hopfield nets are for static pattern completion, whereas Markov models and time-delay dynamic networks are for temporal pattern completion and recognition. The proper use of contextual information is key to successful recognition.

2.1.3 Regression and Generalization

Linear or nonlinear regression provides a smooth and robust curve fitting to training patterns, as shown in Figure 4(a).The objective generation is to yield a correct output response to an input stimulus to which it has not been trained before.

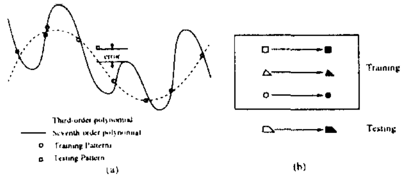

Figure 4. (a) A regression example
(b) A generalization example

2.1.4 Optimization

Neural nets offer an appealing tool for optimization applications, which usually involve finding a global minimum of an energy function. A major difficulty associated with the optimization problem is the high possibility of a solution converging to a local optimum instead of the global optimum.

2.2 Algorithmic Study on Neural Networks

In order to have a systematic approach to the design of neural models it is important to clearly identify the design criteria and factors. In order to achieve a through algorithmic study, formal and theoretical treatments of the neural models will be indispensable.

2.2.1 Taxonomy of Neural Network Design

The design of application-driven neural networks hinges upon the choice of energy function. Many training mechanisms are governed by minimization of the energy function. Various application paradigms based on such a formulation are diplayed in Table 1.

The following design factors could be equally critical in characterizing the neural models.

Supervised and unsupervised models;
Basis functions and activation functions;
Neural network structures;
Mutual and individual training strategies;
Static and temporal patternnn recognitions;
Decision and approximation/optimization formulations.

2.2.2 Formal Theory for Neural Models

Theoretical analyses provide an indispensable means to affirm the capability of the neural models. They provide a reliable basis for measuring the performance of a model, which cannot be support a variety of neural models.

2.3 Architectures of Neural Networks

Most neural algorithms are computationally intensive and iterative in nature. Most applications also demand very high throughput, especially in a real-time processing environment. For this, massively parallel processing represents a very natural and desirable solution.

Network Type	Application Paradigms	Training-Phase Formulation	Retrieving-Phase Formulation
Supervised training	Classification	Given x_i, target symbol (s_i), find W, s.t. $y = s_i$.	Given x_i, W, determine symbol s_i.
	Approximation	Given x_i, target value (t_i), find W, s.t. $\sum_i [t_i - \Phi(x_i, W)]^2$ is minimum.	Given x_i, W, find the value of $y = \Phi(x_i, W)$
	Regularization	Given x_i, target value (t_i), find W and Φ, s.t. $\sum_i [t_i - \Phi(x_i, W)]^2 + \lambda P(\Phi(x, W))$ is minimum.	Given x_i, W, find the value of $y = \Phi(x_i, W)$.
Unsupervised training	Classification	VQ or clustering, or competitive learning techniques.	Given x_i, determine the group to which it belongs
Fixed-weight	Association	Weight predetermined (Hebbian rule, sometimes)	Given W, x', find x, the "local" minimum of $E(x, W)$.
	Optimization	Weight predetermined via the energy function	Given W, find x, the "global" minimum of $E(x, W)$.

Table 1. The training and retreving formulations for various application paradigm.

2.4 Total Information Processing Systems

In order to holistically analyze a total system for neural information processing, it is important to clearly identify the role of each of its subsystems. A total recognition system involves the mappings between several different spaces.

- Instantiation Space : During the instantiation process, a symbol is instantiated into a physical object. The instantiation space contains all actual occurrences of objects.
- Feature Space : The object is descrived in terms of a set of primitives. The mapping from instantiation space to feature space is called feature extraction. Moreover, this mapping represents a data-compression stage.
- Symbol Space : The symbol space contains the symbols representing classes of objects. The mapping from feature space to symbol space is called classification.

2.5 Representation and Feature Extraction

Feature extraction and representation are indispensable to a neural information processing system. The power of neural networks lies in the details of representation and coding of pattern vectors. It is essential that a representation can provide concise, invariant, and/or intelligible information of the input patterns. They dictate the ultimate performance of the system. The following are general criteria for measuring the quality of a feature representation.

(1) Data compression : to extract vital representations or features.

(2) Invariance : to reduce the dependency of the features on imaging conditions.

(3) Fidelity : to best preserve intelligibility of the features.

As shown in **Figure 5.**, the representations can be classified into four categories:

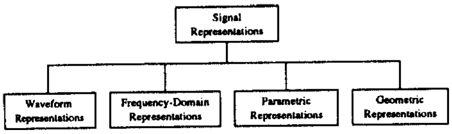

Figure 5. Signal represennntations can be classified into four group.

Pattern Type	Static		Temporal	
Training Scheme	Individual	Mutual	Individual	Mutual
• Model (Energy or Discriminant Functions)	• OCON (LSE) • OCON (Likelihood Function)	• BP (LSE) • DBNN (LBP RBF)	• HMM (Likelihood Function) • PBIT (Prediction Error)	• TDNN • DBNN (Prediction, Likelihood, or DTW)

Table 2. A possible taxonomy of supervised neural models, characterized by the network structure, energy function, temporal property, training strategy, and neuron/basis function.

	OCR	TEXTURE CLASSIFICATION	SPEECH RECOGNITION
Instantiation process	Handwritten characters	Texture samples	Time-warped words
Feature extraction	Structural coding	Cooccurrence/ reduced spectrum	FFT, LPC, Walsh coding
Neural network	HMM network	DBNN	Multilayer perceptron

Table 3. The process od several different application examples.

3. Taxonomy of Neural Networks

There are also two phases in neural information processing. They are the learning phase and the retrieving phase.

Retrieving Phase : Various nonlinear systems have been proposed for retrieving desired or stored patterns. The results can be either computed in one shot or updated iteratively based on the retrieving dynamics equations. The final neuron values represent the desired output to be retrieved.

Learning Phase : A salient feature of neural networks is their learning ability. They learn by adaptively updating the synaptic weights that characterize the strength of the connections. The weights are updated according to the information extracted from training patterns. Usually, the optimal weights are obtained by optimizing certain "energy" functions.

Real-world applications may face two very different kinds of real-time processing requirements. One requires real-time retreving but off-line training speed.

The other demands both retrieving and training in real-time. These two lead to very different processing speeds, which in turn affect the algorithm and hardware adopted. A possible taxonomy of neural networks is displayed in Table 2. The following are critical factors for a systematic design of neural models: supervised and unsupervised models;basis functions and activation functions; structures of neural networks; mutual and individual training strategies; static and temporal pattern recognitions; and decision-based and optimization formulations.

3.1 Supervised and Unsupervised Networks

As shown **Figure 6.** ,the neural networks are commonly categorized in terms of their corresponding training algorithms: fixed-weight networks, unsupervised networks, and supervised networks.

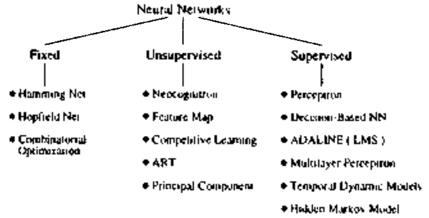

Figure 6. Neural networks can be classified into fixed-weights networks, unsupervised networks, and supervised networks

3.1.1 Supervised Learning Rules

Superrrvised learning networks have the mainstream of neural model development. The training data consist of many pairs of input/output training patterns. Therefore, the learning will benefit from the assistance of the teacher. cf. Figure 7.(a). As an example of supervised training, Figure 7. shows that the decision boundaries are linear hyperplanes specified by the synaptic weights w_{ij}. Given a new training pattern, (m+1)th, the weights may be updated as follows:

$$w_{ij}^{(m+1)} = w_{ij}^{(m)} + \Delta w_{ij}^{(m)}$$

Note that the classification performance is gradually improved.

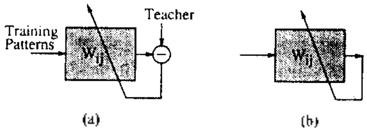

Figure 7. Schematic diagrams training synaptic weights (a) supervised learning (b) unsupervised learning

3.1.2 Unsupervised Learning Rules

For an unsupervised learning rule, the training set consists of input training patterns only. Therefore, the network is trained without benefit of any teacher, as shown in Figure 7(b). Typical examples are the Hebbian learning rule and the competitive learning rule.

3.2 Basis Function and Activation Function

A basic neural model is illustrated in Figure 2(b). It can be characterized by the functional descriptions of the connection network and neuron activation. The net value u_i will then be further transformed by a nonlinear activation function f to yield a new activation value a_i. The final output y can usually be expressed as function of the input and the weights $y = \varphi(x, W)$.

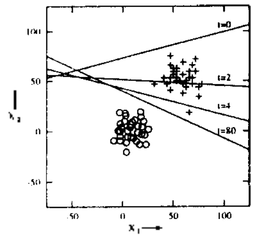

Figure 8. When a linear basis function is adopted in a training process, linear hyperplanes are adjusted to best classify one group from another group.

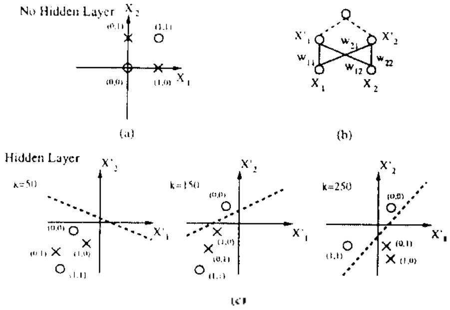

Figure 9. In this simulation of a two-layer BP network for the XOR problem (a) XOR problem (b) a two-layer network (c) The change of weights gradually adjusted the cordinates.

3.2.1 Basis Function (Net Function)

The connection networks are mathematically represented by a basis function u(w,x), where w stands for the weight matrix, and x for the input vector. The basis function has two common forms (cf. Figure 10.)

1. Linear-basis function (LBF) is a hyperplane-type function.

$$u_i(w,x) = \sum_{j=1}^{n} w_{ij} x_j$$

2. Radial-basis function (RBF) is a hypersphere-type function.

$$u_i(w,x) = \sqrt{\sum_{j=1}^{n} (x_j - w_{ij})^2}$$

3.2.2 Activation Function (Neuron Function)

The net value as expressed by the basis function, u(w,x), will be immediately transformed by a nonlinear activation function of the neuron.

$$\text{Sigmoid function :} \quad f(u_i) = \frac{1}{1+e^{-u_i/\sigma}}$$

$$\text{Gaussian function :} \quad f(u_i) = ce^{-u_i^2/\sigma^2}$$

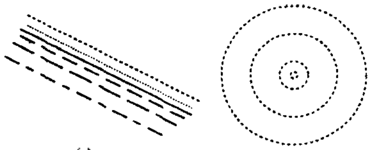

Figure 10. (a) Linear-basis function (b) Radial-basis function

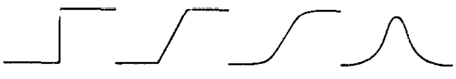

Figure 11. (a) step func. (b) ramp func. (c) sigmoid func. (d) Gaussian func.

3.3 Structures of Neural networks

The major structure factors are connection structures, network size, and ACON(All-Class-One-Network) versus OCON(One-Class-One-Network) approaches.

3.3.1 Interlayer and Intralayer Connection Structures

As shown Figure 12, a neural network comprises the neuron and weight building blocks. There are three types of neuron layers: input, hidden, and output layers. Two layers of neurons communicate via a weight connection network. There are four types of weighted connections: feedforward, feedback, lateral, and time-delayed connections.

(1) Feedforward Connections
(2) Feedback Connections
(3) Lateral Connections
(4) Time-Delayed Connections

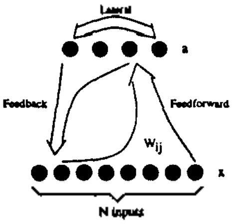

Figure 12. A basic network structure comprises of neurons andweights layers.

3.3.2 Size of Neural Networks

The size of neural networks depends on the number of layers and the number of hidden-units per layer.

(1) Number of layers : The number of layers is very often counted according to the number of weight layers(instead of neuron layers)
(2) Number of hidden-units : For the best network performance(e.g.generalization), an optimal number of hidden-units must be properly determined.

3.3.3 ACON Versus OCON Approaches

The issue is how many networks should be used for multi-category classification. Typically, one output node is used to represent one class. Given an input pattern in the retreving phase, the winner(i.e., the class that wins the recognition) is usually the output node that has the maximum among all the output values. Two plausible network structures are ACON (All-Class-One-Network) and OCON(One-Class-One-Network). The ACON and OCON differ significantly in size and speed, that is the total numbers of synaptic weights and the training time. Emprical results confirm that the convergence rate of ACON degrades drastically with repect to the network size because the training is influenced by conflicting signals from different teachers. Also experimental results based on some speech application suggests that 3-5 hidden units are all it needs per subnet.

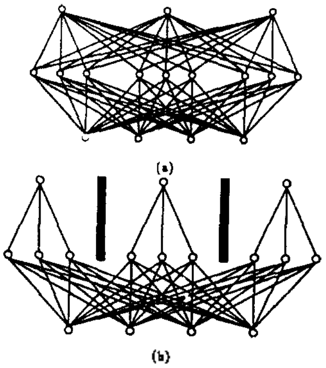

Figure 13. (a) The structure for an ACON model (b) An OCON structure viewed as a resulttt of partitioning a single supernet into many small subnet

3.4 Mutual and Individual Training Stratrgies

The training strategies may be divided according to the presence or absence of cross reference between different outputs. In a mutual(individiual) training rule, cross-references among the outputs can (cannot) be used to assist the training . More precisely , in mutual training, the training of all the weights is influenced by all the output values.

3.4.1 Mutual Training Stratrgy

It allows two neighboring categories to "negotiate" their mutual decision boundary.This leads to more acute boundaries.

3.4.2 Individual Training Stratrgy

The training of an individual subnet is strictly trained by the outputs of itself. The advantage is its ovious simplicity. Individual training Stratrgy may be further diviiided according to the training data involved.

- ○ Discriminative Training : The strategy uses all the training patterns(both the positive and negative examples) to train each subnet.
- ○ Independent Training : The alternative is the totally independent training in which each subnet is trained by its coressponding positive examples only.

3.4.3 Hierarchical Training Strategy

For higher performance the mutual training should be used. For simpler and faster training , the individual traininiig very attractive.

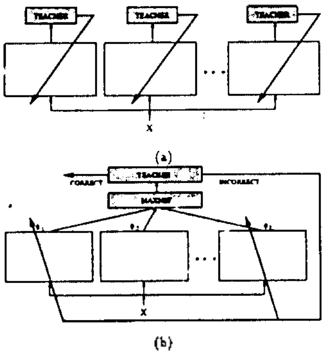

Figure 14. (a) The subnet are individually trained in the first stage (b) The network is mutually trained and only thr "critical" subnets will be updated trained

3.5 Static and Temporal Pattern Recognition.

Two types of patterns are recognized : (a) static pattern and (b) temporal patterns. Static patterns are not oder-sensitive. Temporal patterns have strong order sensitivity. Temporal patterns have two networks. One is Deterministic temporal networks. The other is Stochastic temporal networks.

3.6 Decision and Approximation/Optimization Formulations

The formulations of supervised neural networks can be divided into desicion-based and optimization-based categories.

3.6.1 Decision-Based Formulation

In a decision based neural network(DBNN) , the teacher only tells the correctness of the classification for each training pattern. The objective of the training is to find a set of weights which yields a correct classification. It is very important to identify a proper discriminant function in order to best distinguish each class in the presence of other classes. The selection of such discriminant functions varies siginificantly between the static and temporal recognition.

3.6.2 Approximation/Optimization Formulation

The most popular function is the minimum-squares -error between the teacher and the actual response. Obviously, the exact teacher's values have to be available as a refernece at the output. However, the exact values will not be required if likelihood function is used as the training criterion. The key is to identify a proper training criterion to be optimized, which depends on the application(s) intended and on whether static or temporal patterns are involved.

4. Back-Propagation Networks

Th Back-Propagation(BP) algorithm offers an effective approach to the computation of the gradients. This can be applied to any optimization formulation (i.e.,any type of energy function) as well as the DBNN formulation. A linear basis function (LBF) multilayer network is characterized by the following dynamic equations:

$$u_i(l) = \sum_{j=1}^{N_{l-1}} w_{ij}(l)\, a_j(l-1) + \theta_i(l)$$

$$a_i(l) = f(u_i(l)) \quad 1 \le i \le N_l \; ; \; 1 \le l \le L$$

where the input units are represented by $x_i = a_i(0)$, the output units $y_i = a_i(L)$, and the L is the number of layers. The activation function is very often a sigmoid function :

$$f(u_i) = \frac{1}{1+e^{-u_i/\sigma}}$$

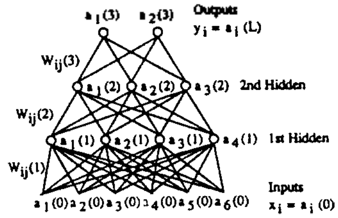

Figure 15. The multiple-layer network shown is a net with three weight layers.

The BP algorithm, independently proposed by Werbos, Parker, and Rumelhart, offers an efficient computational speed-up for training multilayer networks. The objective is to train the weights wij so as to minimize E. The basic gradient-type learning formular is

$$w_{ij}^{(m+1)}(l) = w_{ij}^{(m)}(l) + \Delta w_{ij}^{(m)}(l)$$

with the mth training pattern, $a^{(m)}(0)$, and its corresponding teacher $t^{(m)}$, m=1,2,...,M, presented. The derivation of the BP algorithm follows a chain-rule technique:

$$\Delta w_{ij}^{(m)}(l) = -\eta \frac{\partial E}{\partial w_{ij}^{(m)}(l)}$$

$$= -\eta \frac{\partial E}{\partial a_i^{(m)}(l)} \frac{\partial a_i^{(m)}(l)}{\partial w_{ij}^{(m)}(l)}$$

$$= \eta\, \delta_i^{(m)}(l)\, f'(u_i^{(m)}(l))\, a_j^{(m)}(l-1)$$

where the error signal $\delta_i^{(m)}(l)$ is defined as

$$\delta_i^{(m)}(l) = -\frac{\partial E}{\partial a_i^{(m)}(l)}$$

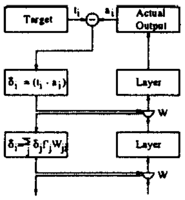

Figure 16. The schematic diagram for the back propagation process

5. Time Delay Neural Network

It was proposed by Waibel et al. to expand the hidden layer by adding multiple(orthogonal) delay-lines, one delay-line for each of the original hidden units. As shown in Figure 17. , this leads to a structure with 2-dimensional delay arrays. Such a network structure is called Time-Delay Neural Network(TDNN). The TDNN is non-recurrent and copes with time alignment by explicitly delaying the signal waveform by a fixed time span. The conventional BP(Back Propagation) algorithm can be adopted to train the weights. In terms of real-world applications, the TDNN suffers from several critical drawbacks. The complexity of the structure usually requires a time consuming training process. The prefixed span of the time-delay-lines renders it less suitable for heavily warped (speech) signals. Moreover, the TDNN is quite sensitive to ambient noise, causing significant performance degradation when speech signals arerr noise corrupted. The theoretical analysis and emprical remedy pertaining to the TDNN structure remain open research subjects.

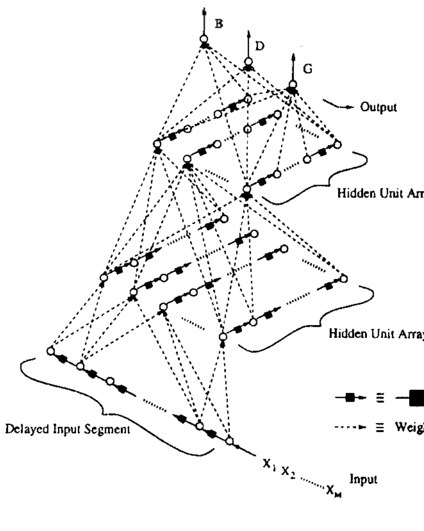

Figure 17. The network structure for a TDNN phoneme recognizer for three phoneme "B", "D", and "G".

6. Self-Organized Map (SOM) Algorithm

One of the representative unsupervised learning model is SOM(Self Organizing Map) model proposed by T.Kohonen. SOM algorithm is a feature classifier representing the probability distribution of input pattern as topological maps to the two dimensional output layer through competitive learning and lateral inhibition. SOM learning consists of two processes, the selecting a similar cell to input pattern and the modifying weight vectors of the selected cell. Inner product or Euclidean distance measures their similarity. Since all cells compete against other cells for being learned, the learning in SOM is called "competitive learning". The best matching vector, a winner , has the chance to be learned. In this point the Kohonen network is a "winner-take-all fashion". The learning algorithm adjusts the weight vectors in the vicinity of the winning neuron according to learning rules. It is as follows;

$$M_i(t+1) = \begin{cases} M_i(t) + \alpha(t)[\, X(t) - M_i(t)\,] & \text{if } i \subset N_c(t) \\ M_i(t) & \text{if } i \notin N_c(t) \end{cases}$$

where , $X(t)$ is input pattern, M_i is weight vector, $N_c(t)$ is a neighbor cell which can be learned with winner cell. $\alpha(t)$ is adaptation gain, $0 < \alpha(t) < 1$, learning rate. $\alpha(t)$ is decreased monotonically with time. We use the lateral inhibition which modifies the weights of neighbooor cells as well as that of the winner cell. Kohonen's algorithm creates topological organized map of various features of input patterns. It is analogous to the basic functions of the biological neuron. The effect of lateral inhibition is similar to that of Maxican hat function. The neighborhood starts with large number, and is gradually decreased with time. Eventually, the neighborhood includes only the nearest cells around the winner cell.

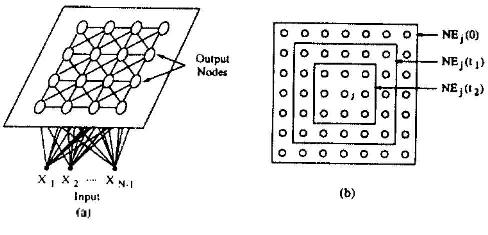

Figure 18. (a) A network for a self-organizing feature map (b) Once such a correlation is established, the size of a neighborhood can be decreased gradually, baaased onnn the desire of having a stronger identity of individual nodes.

7. Recurrent Neural Network

A major alternative to TDNN is to incorporate delay feedbacks into temporal dynamic models, making them recurrent. This leads to the so-called recurrent neural network(RNN). For example, a multi-layer network may be made recurrent by introducing time-delay loops to the input, hidden, and/or output layers. Another way is to route delay connections from one layer to another. As a result of such a structural change, the gradient computation for the RNNs involves a complex back-propagation rule through both time and space. A very important class of RNNs is represented by single-layer recurrent models such as those shown Figure 19 (a). This model serves to reduce the network complexity. The state-space representation again proves very useful for RNNs. The states can be either fully or sparsely interconnected. Some direct connections between the neurons (without delay) may also be accommodated. This is shown in Figure 19 (b). In this case, the model can be approxiamted by "unfolding" the network over a finite time period into a multi-layer network. Then the conventional back-propagation leasning rule can be applied with the restriction that all the unfolded weights must be uniformed.

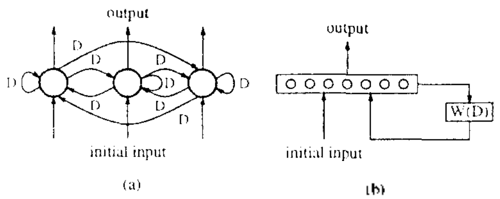

Figure 19. Two forms for state-space represenation of a single-layer RNN

8. Speech Recognition Trends by Neural Networks and Conculsion.

In present, the method using neural networks for speech recognition are MLP(MultiLayer Perceptron), TDNN, RNN, SOFM, Hybrid method with HMM(Hidden Markov Model), and conventional neural network with many variations, we investiagtes the neural network method in the ICSLP, ICASSP, and ICNN confer-ence journal in 1994, ICSLP in 1995 , and the jounal of the acoustical society of Korea in 1994. They are shown in Table 4 (a),(b),(c).

As shown in Table 4, the hybrid method(NN+HMM) is studying briskly. Also a variety of methods in neural networks are researching parallelly.

Table 4. (a) The jounal of the Acoustical Society of Korea in 1994

Type	Nation	Unit	SD/SID	Recognition
MLP	Korea	Word		89.6%
FSNN Frequency State NN	Korea	Phoneme	Algorithm	96%
Hybrid(Hebb + BP)	〃			
TDNN	〃	Phoneme		98.33%
Dynamically Localized DLSOFM	〃	Digit		93%
PNN Rediction	〃	Digit		94%
HCNN Hidden Control	〃	Digit	SID	97.35%

Table 4. (b) ICASSP in 1995

Type	Nation	Unit	SD/SID	Recognition
LVQ	Iran		SID	Error rate 2.9%
Hybrid (Neural Tree Network + HMM)	USA	Isolated Phoneme		Error rate 0.36%
		Isolated Word		Error rate 0.17%
NP(Neural Prediction) System	France	Phoneme	SID	77.4%
RNN (Recurrent NN)	Hongkong	Single Word		95.3%
		Multi Word (10 digit)		87.7%
NTN (Neural Tree Network)	USA	Digit	SID	88.7%
		Phoneme		42.9%
FRNN (Fully Recurrent NN)	Germany	Word	SID	97.1%
		Digit		98.2%

Table 4. (c) ICSLP, ICNN, and ICASSP in 1994

Type	Nation	Unit	SD/SID	Recogniton
TDNN	Korea	phoneme	SD	93.7%-95.2%
Hybrid(HMM+MLP)	USA	word	SD	error rate 1.7%
PNN(predictive NN)	Korea	word	SID	95.8%
NN	English	phoneme	SID	53-57%
MLP	English	word	SID	11.3%
MSGCNN (mulit state gaussian competitive NN)	China	syllable	SD	98.52%
Modular WNN (window based NN)	Australia	word	SD	82.8%
Hybrid(HMM+ANN)	USA	word	SD	96%
SDNN (state deletator NN)	China	phoneme		98.5%
NN	India	word		98%
Hybrid(HMM-MLP)	USA	telephone speech word		error rate 9.1%
NTN (neural tree net.)	USA	TIMIT DB		
NN	France	phoneme	SD	75.6%
MLP	Belgium	word		91.2%
Hybrid (TDNN+HMM)	Germany	phoneme	SD	error rate 22.2%
TDNN	Singapole	syllable	SID	95.3%
Hybrid(HMM+NN)	Italy	word	SID	97%
TDNN	USA	phoneme	multi SP	64.6%
			single SP	67.08%
DNN(dynamic NN)	USA	phoneme	SID	88%
MLP	France	vowel	SD	90-97%
TSRN(time slice recurrent NN)	Japan	phoneme	SD	95.8%
LVQ(learning VQ)	Japan	word		99.2%
Fuzzy NN	USA	vowel		89.6%

REFERENCE

1. S.Y.KUNG "Digital Neural Network " pp1-39. 1993
2. J.A.Anderson "Neorocomputing - Paper collection" MIT Press 1988. pp 2,76
3. S.Katagiri, C.H.Lee, and B.H. Jong "Deterministic multi-layer feed-forward networks" NNSP, proceeding 1991. pp127.

4. T.Ash "Dynamic node creation in back-propagation networks" Connection Science 1989. p183.
5. P.Baldi and K.Hornik. "Neural networks for principal component analysis: learning from examples without local minima" 1989. NN
6. H.Bourland and C.J. Wellelkens "Multilayer perceptron and automatic speech recognition" ICNN,1987, p266
7. P.C. Chang and B.H.Jung "Discriminative template training for dynamic programming speech recognition" IEEE Trans. SP p183
8. J.Connor, L.E.Atlas, and D.R. Martin. "Recurrent networks and NARMA modeling" Advances in NIPS,p301-308, 1992
9. G.cybenko. "Approximation by superpositions of a sigmoidal function" Math of control,signal, and systems,1989. p169
10. J.L Flanagan. "Computers that talk and listen : Man-machine communication by voice" Proceedings IEEE 1976, p21
11. K.S.Fu "Syntactical Pattern Recognition and Applications" Prentic Hall,1982. p24.
12. K. Fukushima. "Cognitron: A self-organizing multilayered neural network" Biological Cybernetics 1975, p73,91
13. J.Hertz, A. Krogh, and R.G. Palmer "Introduction to the theory of Neural Computation" Addison Wesley,1991, p317
14. W.Huang and R.Lippmann "Comparison between neural network and conventional classifiers" ICNN,1987, pp242,258
15. T.Kohonen "Self-Organized foramtion of topologically correct feature map" Biological Cybernatics, 1982, pp 74, 85
16. Y.H.Pao "Adaptive Pattern Recognition and Neural Network" Addison Wesley,1989, p79
17. J.S. Taur and S.Y.Kung "Prediction-based networks with ecg application" ICNN 1993, p225
18. K.Diamantaras and S.Y.Kung "An unsupervised neural model for oriented principal component extraction" ICASSP, 1991, 0303
19. A.Waibel,T.Hanazawa, G.Hinton,K.Shikano, and K.Lang "Phoneme recognition using time-delay neural networks" ICASSP, 1989, p217
20. R.E. Kalman "A new approach to linear filtering and prediction problems" Jounal of Basic Enginnering, 1960, p233
21. A.Barr and K. Feigenbaum "The handbook of Artificial Intelligence" Kaufmann. 1981, p259