

Radial Basis 함수를 이용한 동적-단기 전력수요예측 모형의 개발

민 준 영[†] · 조 형 기^{††}

요 약

전력수요의 예측은 예측기간에 따라 중장기 전력수요 예측과 단기 부하 예측으로 구분할 수 있다. 기존의 단기 부하예측은 주로 역전파 알고리즘(backpropagation algorithm) 다층퍼셉트론을 이용하여 예측을 하였으나 이는 학습시간이 많이 걸릴 뿐만 아니라 학습도중에 지역최소점(local minima)에 빠져 학습이 계속되지 못한다는 문제가 있다. 본 논문은 이러한 역전파 알고리즘의 문제점을 해결할 수 있는 방법으로 Radial Basis 함수(Radial Basis Function)를 이용하여 동적 단기부하 예측 모형을 제안한다. Radial Basis 함수는 하나의 은닉층(hidden layer)을 갖고 있으며, 전방향(feed-forward) 학습을 한다는 특징이 있다. 본 논문에서 제안한 단기 부하 예측모형은 학습을 하기 위하여 시간대별 부하량을 클러스터링 하고, 이 클러스터의 중심값을 Radial Basis 함수의 은닉층으로 하여 학습을 한 다음 예측하고자 하는 패턴을 한 단위로 하여 시간대별로 예측하였다. 기존의 연구에서의 클러스터링 방법으로는 통계학의 K-Means 방법이나 Kohonen의 LVQ(Learning Vector Quantization)을 주로 이용하였으나 본 논문에서는 패턴의 분류에 있어서 다른 알고리즘보다 편차가 작은 Pal, et. al.의 GLVQ(Generalized LVQ) 알고리즘을 이용하였다. 본 논문에서 이용한 데이터는 1995년 3월 1일-3일, 6월 1일-3일, 7월 1일-3일, 9월 1일-3일, 11월 1일-3일의 72시간 데이터를 입력하여 월별 4일의 24시간의 예측기간으로 예측하였다. 실험결과 월별 1일과 3일까지의 학습데이터로 1시간 후의 부하량을 24시간동안 예측한 결과 1.3795%의 평균오차율로 예측하였다.

The Development of Dynamic Forecasting Model for Short Term Power Demand using Radial Basis Function Network

Joon Young Min[†] · Hyung Ki Cho^{††}

ABSTRACT

This paper suggests the development of dynamic forecasting model for short-term power demand based on Radial Basis Function Network and Pal's GLVQ algorithm. Radial Basis Function methods are often compared with the backpropagation training, feed-forward network, which is the most widely used neural network paradigm. The Radial Basis Function Network is a single hidden layer feed-forward neural network. Each node of the hidden layer has a parameter vector called center. This center is determined by clustering algorithm. The treatments of class-

※ 이 논문은 1996년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

† 정 회 원: 상지대학교병설전문대학 전자계산과 조교수

†† 정 회 원: 아주대학교 산업공학과 박사과정수료, 상명대학교 강사

논문접수: 1997년 2월 17일, 심사완료: 1997년 5월 26일

ical approaches to clustering methods include theories by Hartigan(K-means algorithm), Kohonen(Self Organized Feature Maps: SOFM and Learning Vector Quantization: LVQ model), Carpenter and Grossberg(ART-2 model).

In this model, the first approach organizes the load pattern into two clusters by Pal's GLVQ clustering algorithm. The reason of using GLVQ algorithm in this model is that GLVQ algorithm can classify the patterns better than other algorithms. And the second approach forecasts hourly load patterns by radial basis function network which has been constructed two hidden nodes. These nodes are determined from the cluster centers of the GLVQ in first step. This model was applied to forecast the hourly loads on Mar. 4th, Jun. 4th, Jul. 4th, Sep. 4th, Nov. 4th, 1995, after having trained the data for the days from Mar. 1th to 3th, from Jun. 1th to 3th, from Jul. 1th to 3th, from Sep. 1th to 3th, and from Nov. 1th to 3th, 1995, respectively. In the experiments, the average absolute errors of one-hour ahead forecasts on utility actual data are shown to be 1.3795%.

1. 서 론

전력수요의 예측은 예측기간에 따라 1) 장기전원개발계획, 송배전 설비계획 및 발전기 보수계획등을 위한 중장기 전력수요 예측과 2) 발전기 가동정지, 주간발전계획, 일일발전계획, 경제급전등의 계통운용등을 위한 단기부하예측으로 구분되어진다[1]. 장기 부하 예측에 있어서의 가장 영향을 주는 요소로는 경제성장률이나 인구증가율등이 있으나[2] 단기 부하 예측은 일일 또는 일주간의 전력부하를 예측하는 것으로써 기온 및 습도, 시간대별 사용량에 의하여 예측을 하였다[3]. 본 논문은 단기부하예측만을 고려하였다. 이전까지 연구되어진 단기 부하예측으로는 지수평활화(exponential smoothing)방법, 다중회귀(multiple regression)방법, Box-Jenkins방법, State Estimation, Kalman-Filtering방법등이 있고, 실제로 전력회사에서 가장 널리 적용하고 있는 부하예측 모형은 ARMA, ARIMA 모형을 토대로 하고 있다. 예를 들어 Vemuri, et. al.[4]은 기존의 Box-Jenkins 모형보다 단순화되고 효과적인 결과를 제시해 주는 ARMA 모형을 제시하였다. 본 논문에서는 일일 부하량을 예측하면서 예측오차인 RMS의 범위가 3.66%-5.28%이며, 이때 일일부하가 189-849Mw였다고 하였다.

최근에 와서는 신경망 이론을 중심으로한 부하예측이 활발히 연구되어지고 있다. D. C. Park, et. al.은 역전파학습(backpropagation)알고리즘을 이용하여 1시간과 24시간의 후의 부하예측을 각각 1.40%와 2.06%의 오차율로 예측하였으며[5], 이 결과는 이전의 오차율 4.22%보다 향상된 알고리즘을 제안하였다. T. M. Peng, et. al.은 각 요일을 5개의 그룹(월요일, 화요일-목요일, 금요일, 토요일, 일요일)으로 나누고, 예측하

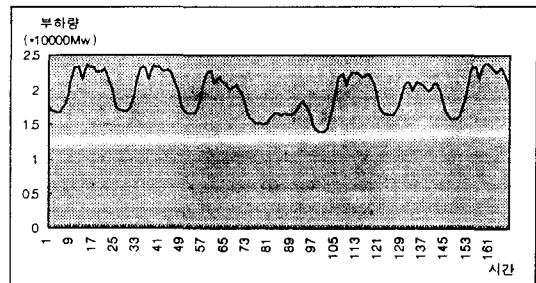
고자 하는 일자의 요일을 고려하여 과거 1년(약 52주) 동안의 데이터를 학습데이터로 추출하였고, 그 다음 이 데이터 가운데 거리를 계산하여 유사성이 있는 데이터를 다시 추출하여 네트워크에 입력시킴으로써 학습할 수 있는 데이터의 수를 줄여 학습에 소요되는 비용을 절감시키는 알고리즘을 제안하였다[6]. K. Y. Lee, et. al.은 부하패턴을 주말(토요일, 일요일, 월요일)과 평일의 패턴으로 구분하여, 역전파알고리즘에 적용하였다. 적용방법으로는 정적접근방법(static approach)과 동적접근방법(dynamic approach) 두가지를 제안하였는데 그 결과 정적접근방법일 경우 1.885%, 동적접근방법일 경우 1.834%의 오차율을 보였다[7]. S. T. Chen, et. al.은 다층퍼셉트론의 네트워크 구성(topology)을 완전연결(fully connected)이 아닌 부분연결로 구성하여 학습하는 데 소요되는 시간을 단축시킨 알고리즘을 제안하였는데 적용 결과 1.12%의 오차율로써 통계적인 방법보다는 효과적인 예측 알고리즘을 제안하였다[8]. 또한 K. L. Ho, et. al.은 기존의 역전파 알고리즘을 개선하여 기존의 학습시간을 단축시킨 Adaptive Learning Algorithm을 제안하였다[9]. 그러나 신경망을 이용한 예측모형들을 살펴보면, 1992년까지는 주로 교사학습(supervised learning)을 중심으로 학습시간의 단축을 위한 알고리즘의 제안으로 국한 되어지는 것을 알 수 있다. 이와는 달리 Y. Y. Hsu와 C. C. Yang은 비교사 학습(unsupervised learning)인 Kohonen의 SOFM(Self Organizing Feature Maps)알고리즘을 이용하여 유사성이 있는 부하패턴을 분류하는 알고리즘을 제안하였다[2]. 그 결과 부하패턴을 일요일, 공휴일이 제외된 평일, 월요일, 토요일, 특수일(현충일등), 공휴일(현충일 다음날)의 6가지 패턴으로 분류하였다.

본 논문은 기존의 단기 전력부하 예측 모형으로 주로 적용했던 역전파학습 알고리즘이 아닌 교사학습과 비교사 학습의 복합 모델인 Radial Basis 함수를 이용한 예측 모델을 제안한다. 또한 Radial Basis 함수를 사용하는 데 있어서 off-line으로 Kohonen 알고리즘이나 ART-2 등의 클러스터링 알고리즘을 이용한 모델이 대부분이나 본 논문에서는 클러스터링의 편차가 적은 GLVQ 알고리즘을 적용하여 예측하였다. Radial Basis 신경망을 적용하기 위하여 1995년 시간대별 부하량을 입력패턴으로 하여 학습을 시켰다. Radial Basis 함수 네트워크 구성은 하루 24시간의 부하를 입력하는 입력 노드와 이를 주말패턴(토요일, 일요일, 월요일)과 평일패턴으로 구분하는 2개의 은닉층 노드, 예측하고자 하는 1개의 출력노드로 구성하였다. 은닉층 노드는 일반적으로 일주일 단위의 부하패턴은 주말과 평일로 구분되어질 수 있기 때문에 2개로 하였으며[7], 이때 Pal, et. al[10]의 GLVQ(Generalized Learning Vector Quantization) 알고리즘을 이용하여 클러스터링을 하였다. 본 논문에서의 예측기간은 D. C. Park et. al.[5]이 예측한 예측기간인 24시간 단위로 예측을 하였다. 본 논문에서 이용한 데이터는 한국전력공사에서 제공한 1995년 일별, 시간대별로 전국 평균 전력수요량이다.

2. 부하량의 특징 및 분류

전력 부하량 패턴의 형태를 살펴보면, 평일패턴과 토요일, 일요일 패턴, 그리고 공휴일 패턴이 구분될 수 있다. (그림 1)은 1995년 6월 1일에서 14일까지 2주 동안의 부하를 시간대별 그래프로 나타내고 있다. 그림에서 73-87 시간대별이 1995년 6월 4일로 일요일이고, 121-145 시간대별이 6월 6일 현충일의 패턴이다. 패턴의 특징을 보면, 휴일인 경우와 평일인 경우의 패턴이 다르다는 것을 알 수가 있다. 특히, 6월 6일은 현충일로서 그 부하량이 평일의 패턴과는 다른 패턴을 이루고 있으며, 토요일의 부하패턴도 저녁시간대에 부하량의 감소추세를 그리고 있다. 이러한 부하패턴을 보다 자세히 분류한 방법으로 K. Y. Lee와 J. H. Park[7]의 부하패턴 분류로써 1987년 2월 8일-21까지의 한국전력공사의 부하패턴을 분석하였다. 일주일간의 부하패턴을 크게 두가지로 분류하였는데 하나

는 평일일 경우(화요일-금요일)과 다른 하나는 주말의 부하패턴(토요일-월요일)의 구분하였다. 이중 주말의 패턴인 경우 토요일의 부하패턴은 오후에 상대적으로 낮게 나타났고, 또한 일요일인 경우에도 한달 중 첫째와 셋째 일요일이 둘째와 넷째 일요일의 부하보다 낮게 나타났는데 그 이유는 한국에서의 산업 및 상업활동에 기인한 것으로 분석하고 있다. 이러한 현상은 다음날인 월요일 오전의 부하에도 영향을 미치는 것으로 나타났다. 본 논문에서는 이러한 분석내용을 근거로 본 예측모델에서 크게 두개의 클러스터로 나누어 Radial Basis 함수에 입력패턴으로 하였다.



(그림 1) 부하량의 특징 그래프
(Fig. 1) Hourly load curve over one week

3. 단기부하예측을 위한 적용 알고리즘

3.1 Radial Basis 함수 네트워크(Radial Basis Function Networks)

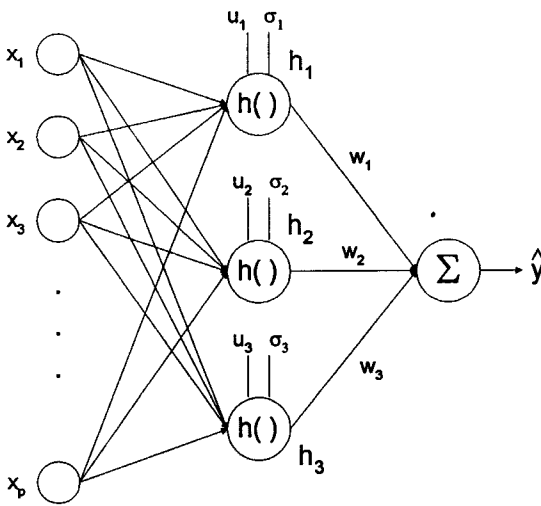
Radial Basis Function Network(RBFN)는 역전파 신경망 모형과(Backpropagation Neural Network: BPN)과 같이 Feed-Forward Network 구조를 가지며, 학습시 BPN이 갖는 지역최소점에 빠지는 등의 단점을 갖지 않고, 학습시간이 BPN에 비하여 월등히 빠르다는 장점을 가지고 있는데, 이는 BPN이 입력층과 은닉층 및 출력층간 모든 학습이 비선형인데 반하여 RBFN은 입력층과 은닉층이 비선형이며, 은닉층과 출력층이 선형성을 가지고 있기 때문이다. 그러나 RBFN의 단점은 학습이 완료 후 클러스터링하는 데 많은 연산이 요구된다는 점이다.

Radial Basis 함수 네트워크는 (그림 2)와 같이 1개의 은닉층으로 구성되어 있는 데 이 은닉층은 기본적인

으로 Gaussian 함수를 이용하고 있다. 각 은닉층은 (식 1)과 같은 지수함수를 이용하여 계산 되어진다[11].

$$h_i = \exp(-D_i^2/2\sigma^2) \tag{1}$$

여기서, $D_i^2 = (\mathbf{x} - \mathbf{u}_i)^T (\mathbf{x} - \mathbf{u}_i)$
 \mathbf{u}_i : 클러스터 중심벡터.
 \mathbf{x} : 입력 패턴벡터



(그림 2) Radial Basis 함수 네트워크
 (Fig. 2) Radial Basis Function network

은닉층에서의 \mathbf{u} 는 입력 데이터의 전체 영역에서의 클러스터 중심 벡터를 의미하며, 이 중심벡터는 통계적 또는 신경망에서의 클러스터링 방법을 이용하여 미리 구할 수가 있다. 일반적으로 중심벡터를 구하는 방법으로는 통계학의 K-means방법이나, 또는 Kohonen의 자기조직형상(Self Organization Feature Maps: SOFM), LVQ(Learning Vector Quantization)방법으로 구할 수가 있다. 본 논문에서는 클러스터링의 중심벡터가 출력에 영향을 미치는 점을 고려하여 보다 패턴의 분류가 잘 되는 GLVQ(Generalized LVQ)알고리즘을 이용하였다. 또한 출력층은 (식 2)의 방법으로 은닉층과 출력층에 있는 연결강도를 곱하여 그 합으로 구할 수가 있다.

$$\hat{y} = \sum_i h_i w_i \tag{2}$$

여기서, \hat{y} : Radial Basis Network 출력값
 h_i : i 번째 Hidden 노드 출력값 (식 1).
 w_i : i 번째 Hidden 노드와 출력노드 사이의 연결강도(weight)

Radial Basis의 학습 방법은 다음과 같다.

- [단계 1] 패턴벡터 \mathbf{x} 를 RBFN 네트워크에 입력
- [단계 2] (식 1)에 의하여 은닉층의 출력값을 계산.
- [단계 3] 최종 출력값 \hat{y} 를 계산하고 목표값(target) \mathbf{t} 벡터와 비교하여 허용오차의 범위를 넘을 경우 (식 3)에 의하여 연결강도(weight)를 조정.

$$w_{ij}(n+1) = w_{ij}(n) + \eta(t_j - y_j) x_k \tag{3}$$

여기서, w_{ij} : i 번째 hidden node j 번째 output node 사이의 weight
 n : 반복횟수
 t_j : j 번째 output의 목표값(target) 벡터
 y_j : j 번째 출력값 벡터
 x_k : k 번째 입력벡터
 η : 학습률($\eta < 1.0$)

- [단계 4] 단계 1에서 단계 3까지 반복.
- [단계 5] 목표값 \mathbf{t} 와 출력값 \hat{y} 와의 차이가 허용오차의 범위를 넘으면 단계 1에서 단계 4를 계속 반복하고, 그렇지 않으면 학습 종료.

3.2 GLVQ(Generalized Learning Vector Quantization)알고리즘

Kohonen은 자기조직형상(Self Organization Feature Maps: SOFM)의 개념을 입력형태의 특성들과 학습벡터양자화(Learning Vector Quantization: LVQ)에 기초를 둔 학습알고리즘을 제안하였다[13]. 실제로 LVQ는 클러스터링을 목적으로 한 알고리즘이 아니라 데이터의 프로토타입을 설계하기 위하여 데이터 중에 가장 가까운 프로토타입(nearest prototype)들을 그룹화 하는 알고리즘이다[10]. 이 알고리즘은 기본적으로 입력벡터들과 클러스터의 중심벡터들간의 평균제곱오차(mean square error)를 최소화 하는 것으로 학습방법은 MSE에 대한 목적함수를 최소화 시킴으로써 양질의 패턴 분류가 가능하도록 한 것이다. 여기서의 제곱오차는 유클리디안 거리와 동일한 개념으로 취

급될 수 있다. LVQ의 학습규칙은 (식 4)와 같다.

$$v_{i,t} = v_{i,t-1} + \alpha_t(x_k - v_{i,t-1}) \quad (4)$$

여기서, $v_{i,t}$: 반복 t 에서 승자노드(winner node)와 연결된 연결강도.

$$1 < i < c (= \text{number of clusters}),$$

$$1 < t < T (= \text{max.iteration})$$

x_k : k 번째 패턴벡터. ($1 < k < N$)

α_t : 반복 t 에서의 학습률. ($= \alpha_0(1 - t/T)$),

α_0 = 초기 학습률

그러나 LVQ는 다음과 같은 몇가지 문제점이 있다[10].

첫째, LVQ의 결과가 연결강도의 초기값에 민감한 영향을 받는다. 즉, 클러스터의 중심(연결강도)에 대한 초기값이 N 개의 입력패턴들과 멀리 떨어진 상태(out of convex hull)에서 결정되었을 경우 여러개의 클러스터 중 한개의 클러스터만 승자노드가 되어 반복이 계속되더라도 승자노드를 제외한 다른 노드들은 초기의 연결강도 값을 그대로 유지하게 된다.

둘째, 초기 학습률이 0에 가깝게 주어졌다면, 반복이 계속되면서 α_t 는 감소하게 되어 0에 수렴하게 되므로 연결강도의 변화량이 허용오차 안에 들어오지 않았는데도 불구하고, (식 2)에서 두번째 항인 $\alpha_t(x_k - v_{i,t-1})$ 의 값이 0으로 수렴하므로 학습이 이루어지지 않는다. 따라서 학습이 도중에 끝나게 되기 때문에 오분류되는 패턴이 증가할 수가 있다. 이러한 문제점을 극복하기 위하여 GLVQ알고리즘이 제안되었다[10, 14, 15]. 이 방법은 LVQ의 승자노드 학습에 비승자노드(non winner node)의 학습을 부가하여 학습하는 방법으로 승자노드의 벡터와 입력벡터와 완전히 일치할 경우에는 LVQ와 GLVQ는 같은 학습을 하게 된다.

GLVQ는 (식 5)의 손실함수(loss function) L_x 를 최소화 시키는 학습방법을 적용하였다.

$$L_x = \sum_{r=1}^c g_{ir} \|x - v_r\|^2 \quad (5)$$

$$\text{여기서, } g_{ir} = \begin{cases} 1 & , \text{ if } r = i \\ \frac{1}{\sum_{j=1}^c \|x - v_j\|^2} & , \text{ if } r \neq i \end{cases}$$

여기서 승자노드일 경우에는 L_x 를 v_i (winner class center)로 미분하고, 승자노드가 아닐 경우에는 v_j (non winner center)로 미분하여 (식 6)으로 학습을 한다.

$$v_{i,t} = v_{i,t-1} + \alpha_t(x_k - v_{i,t-1}) \frac{D^2 - D + \|x_k - v_{i,t-1}\|^2}{D^2}$$

승자노드인 경우

$$v_{r,t} = v_{r,t-1} + \alpha_t(x_k - v_{r,t-1}) \frac{\|x_k - v_{i,t-1}\|^2}{D^2}$$

승자노드가 아닌 경우

$$\text{여기서, } D = \sum_{r=1}^c \|x_k - v_r\| \quad k = 1, 2, \dots, N$$

각 단계별 GLVQ 클러스터링 알고리즘은 다음과 같다.

[단계 1] 패턴벡터 $X = \{x_1, x_2, \dots, x_N\} \subset R^p$,

$c, T, \epsilon (> 0)$ 를 결정한다.

여기서, c : 클러스터의 수.

T : 최대 반복횟수

ϵ : $\|v_t - v_{t-1}\|$ 의 허용오차.

[단계 2] 초기 연결강도 $V_0 = \{v_{1,0}, v_{2,0}, \dots, v_{c,0}\}$ 와 초기 학습률 α_0 를 결정한다.

[단계 3] 반복 $t = 0, 2, \dots, T$: 반복횟수를 0번부터 T 번까지 반복한다.

반복횟수마다 학습률 $\alpha_t = \alpha_0(1 - t/T)$ 계산을 한다.

반복 $k = 1, 2, \dots, N$: k 개의 패턴을 네트워크에 하나씩 입력시킨다.

a. 승자노드를 다음 식에 의하여 계산하고,

$$\|x_k - v_{i,t-1}\| = \min_{1 \leq j \leq c} \|x_k - v_{j,t-1}\|$$

b. 연결강도를 다음 식에 의하여 수정한다.

$$v_{i,t} = v_{i,t-1} + \alpha_t(x_k - v_{i,t-1})$$

$$\frac{D^2 - D + \|x_k - v_{i,t-1}\|^2}{D^2}$$

$$v_{r,t} = v_{r,t-1} + \alpha_t(x_k - v_{r,t-1}) \frac{\|x_k - v_{i,t-1}\|^2}{D^2}$$

$k = k + 1$

[단계 4] 전체 오차를 계산한다.

$$E_t = \sum_{k=1}^N \sum_{r=1}^c |v_{rk,t} - v_{rk,t-1}|$$

[단계 5] if $E_t \leq \epsilon$ 이면, STOP; 아니면 $t = t + 1$

$$u_{LVQ,i} = \begin{cases} 1 & \|x_k - v_i\| \leq \|x_k - v_j\|, 1 \leq j \leq c, j \neq i \\ 0 & \text{otherwise} \end{cases}$$

$U_{LVQ} = [u_{LVQ,i}]$: $c \times N$ 행렬로써 x_k 패턴이 클러스터 c 에 포함되어 있는 상태를 나타내는 행렬.

실제로 Pal, et. al.은 Fisher의 IRIS 데이터를 가지고 반복횟수와 학습률을 변화시켜 가면서 LVQ와 GLVQ알고리즘을 비교한 결과 GLVQ는 150개의 IRIS 데이터 중에서 오분류의 갯수가 17개로 일정한 반면에 LVQ인 경우에는 100개에서 17개의 오분류를 나타났음을 제시한 바 있다.

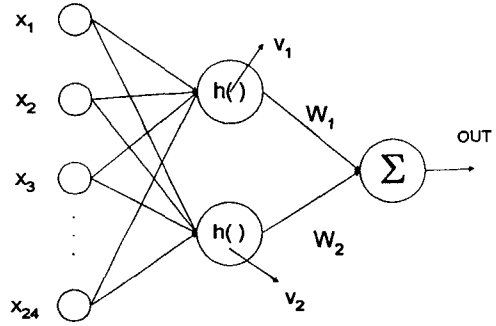
4. 단기부하예측을 위한 클러스터링

본 논문에서 단기부하 예측을 위하여 Radial Basis 함수 네트워크를 적용하였을 때 Off-Line으로 부하의 특징을 클러스터링은 2개의 클러스터로 나누었으며, 이 두개의 클러스터가 Radial Basis함수 네트워크의 은닉층 노드의 수로 결정하였다. 본 논문의 네트워크 모델은 (그림 3)과 같다. 클러스터링은 Pal et. al.의 GLVQ알고리즘을 이용하였는데 입력패턴은 (식 7)과 같으며, 클러스터링된 결과는 <표 1>과 같다.

<표 1>은 1995년 7월 1일에서 7월 3일까지의 부하량을 24시간 단위로 시간대별로 입력패턴으로 하였을 경우, GLVQ 클러스터링을 한 결과로써 (식 7)과 같이 예측을 하고자 하는 일자의 t 시간의 24시간 전의 부하노드 y_{t-24} 부터 y_{t-1} 까지의 입력 벡터 48개의 2개의 클러스터로 나누었을 때 각 클러스터의 중심벡터와 각 클러스터에 포함되는 패턴의 레이블을 나타낸 것이다.

$$x_i = [y_{i-1}, y_{i-2}, \dots, y_{i-24}]^T \quad (7)$$

여기서, y_i : i 시간대의 부하.



v_1, v_2 are determined by GLVQ algorithm

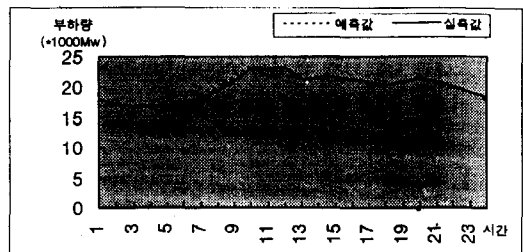
(그림 3) 단기부하예측 네트워크
(Fig. 3) Short-term load forecasting network

<표 1> GLVQ 클러스터 결과
(Table 1) The result of GLVQ clustering

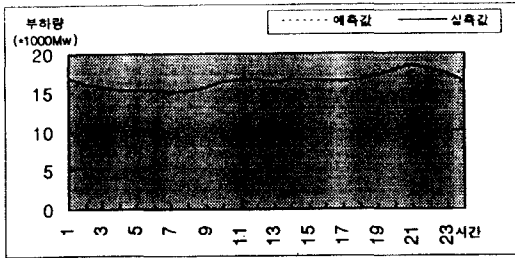
Cluster	클러스터 중심값					포함된 패턴
1	20.99144	21.23984	21.53364	21.86548	22.14044	$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$
	22.28444	22.28569	22.04160	21.60680	21.11085	$x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}$
	20.59183	20.15133	19.67068	19.17002	18.70035	x_{17}, x_{18}
	18.31338	18.01540	17.79408	17.55406	17.27343	
	17.04617	16.88080	16.77140	16.77424		
2	17.96680	17.74338	17.50432	17.24230	17.01979	$x_1, x_2, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}$
	16.86609	16.79036	16.84265	17.02282	17.25309	$x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}$
	17.50231	17.71883	17.95391	18.21088	18.47450	$x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36}, x_{37}$
	18.71415	18.91765	19.10239	19.29423	19.50644	$x_{38}, x_{39}, x_{40}, x_{41}, x_{42}, x_{43}, x_{44}$
	19.68375	19.82590	19.92624	19.97335		$x_{45}, x_{46}, x_{47}, x_{48}$

5. 실험결과

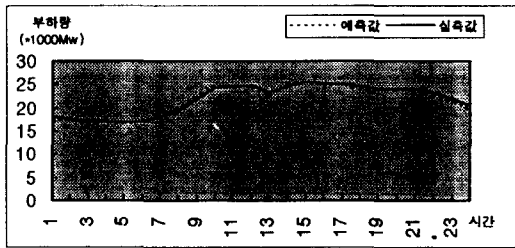
본 논문에서 이용한 데이터는 1995년 3월 1일-3일, 6월 1일-3일, 7월 1일-3일, 9월 1일-3일, 11월 1일-3일의 72시간 데이터를 입력하여 매월 4일의 24시간의



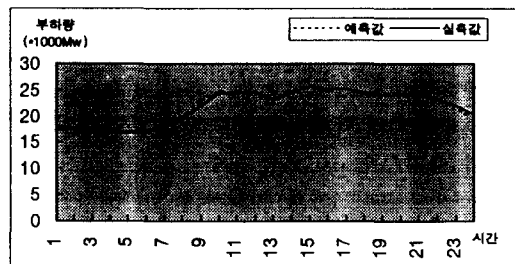
(그림 4) 3월 1일-3일 데이터로 3월 4일 24시간 예측그래프
(Fig. 4) Hourly load forecasting and actual load on Mar. 4th (24-hour load forecast)



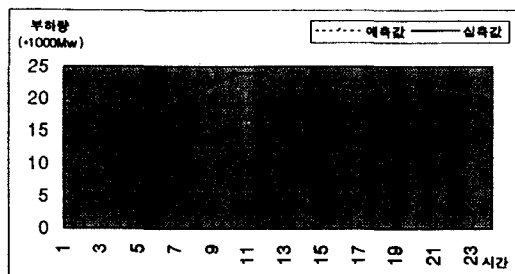
(그림 5) 6월 1일-3일 데이터로 6월 4일 24시간 예측그래프
(Fig. 5) Hourly load forecasting and actual load on Jun. 4th (24-hour load forecast)



(그림 6) 7월 1일-3일 데이터로 7월 4일 24시간 예측그래프
(Fig. 6) Hourly load forecasting and actual load on Jul. 4th (24-hour load forecast)



(그림 7) 9월 1일-3일 데이터로 9월 4일 24시간 예측그래프
(Fig. 7) Hourly load forecasting and actual load on Sep. 4th (24-hour load forecast)



(그림 8) 11월 1일-3일 데이터로 11월 4일 24시간 예측그래프
(Fig. 8) Hourly load forecasting and actual load on Nov. 4th (24-hour load forecast)

예측기간으로 예측하였다. 본 논문에서 Radial Basis 함수를 이용하는 데의 파라미터는 학습률 0.7로 하였다. 실험결과는 각각 (그림 4)에서 (그림 8)까지 본 바와 같이 기존의 예측률보다 높은 예측을 할 수가 있었다.

각 예측일의 오차율은 <표 2>와 같으며, 그 산출 방법은 (식 8)과 같다.

<표 2> 예측일의 오차율
<Table 2> Error rate of hourly load forecasting

예측일	최대오차율(%)	최소오차율(%)	평균오차율(%)
3월 4일	1.066176	0.034968	1.3795
6월 4일	0.957795	0.0003267	0.9088
7월 4일	0.851666	0.015163	1.34288
9월 4일	0.887426	0.000751	1.483743
11월 4일	1.189106	0.05888	1.29744
평균	0.990434	0.049899	1.3795

$$\text{Error Rate} = \frac{|L_{Actual} - L_{Forecasted}|}{L_{Actual}} \times 100 \quad (8)$$

여기서, L_{Actual} : 실제 부하량

$L_{Forecasted}$: 예측된 부하량

상기 <표 2>의 결과는 D. C. Park et. al.의 역전과 알고리즘을 기반으로 적용한 평균예측 오차보다 0.02%의 향상된 예측결과임을 보여주고 있다. 또한 본 연구에서 적용한 동일한 기초입력자료를 이용한 M. H. Park et. al.의 Kalman Filter Technique(KFT) 기반의 Dynamic Model[16]과 ARMA Model과 비교시 평균 상대오차 측면에서 KFT가 본 연구의 결과보다 약 0.356%가 더 우수한 것으로 나타났으며, ARMA 보다는 약 0.06%가 더 향상된 것으로 나타났다. 그러나 상대오차율의 변동폭을 비교시 KFT의 최대 오차가 9.743%로서 본 연구의 결과보다 약 8%의 변동폭을 가지고 있어서, 결국 본 연구의 적용모형이 KFT모형 혹은 ARMA 모형보다는 상대적으로 변동폭이 적고 안정된 결과를 나타냈다.

6. 오차분석

본 연구에서 단기수요전력을 예측하기 위하여 적용된 모형은 신경망 모형이었다. 일반적으로 예측모

〈표 3〉 적합도검증 결과
 〈Table 3〉 The result of goodness-fit test

	Lower Limit	Upper Limit	Observed Freq.	Expected Freq.	χ^2
at or below					
		-0.61429	9	6.1	1.3630
	-0.61429	-0.50000	0	5.0	5.0339
	-0.50000	-0.38571	3	7.6	2.7736
	-0.38571	-0.27143	7	10.4	1.1003
	-0.27143	-0.15714	17	12.9	1.3125
	-0.15714	-0.04286	22	14.5	3.8500
	-0.04286	.07143	14	14.9	0.0491
	.07143	.18571	20	13.8	2.7982
	.18571	.03000	9	11.6	0.5896
	.03000	.41429	7	8.9	0.3994
	.41429	.52857	3	6.2	1.6253
above	.52857		9	8.2	0.0850

$\chi^2=20.98$ with 9 d.f. sig. Level=0.0129397

형은 실측치와 예측치간 잔차분석이 필수적으로 요구되는데, 이는 예측치와 실측치간 발생오차의 백색잡음(white noise)에 대한 기본가정을 만족시켜야 하기 때문이다. 백색잡음은 정규분포 $N(\mu, \sigma_e^2)$ 을 전제로 하며, 만약 잔차가 칼라잡음(color noise)인 경우에는 비록 상대오차율이 작다고 할지라도 좋은 예측모형이라 할 수 없게 된다.

따라서 본 연구에서는 이를 검증하기 위하여 잔차에 대한 정규분포 적합도검증(Goodness-Fit Test)을 수행하였다. 이러한 통계검증은 Statgraphics PC version 4.0에서 수행되었으며, 목적인 검증을 위하여 수

립한 귀무가설과 대립가설은 H_0 : Residual Data = Normal Distribution vs H_1 : Residual Data \neq Normal Distribution이다. 이에 대한 분석결과는 다음 〈표 3〉과 같으며, (그림 9)는 백색잡음에 대한 정규분포 확률도를 나타내고 있다.

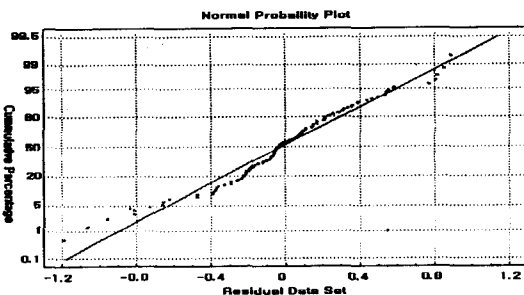
상기 〈표 3〉에 제시되어 있는 바와 같이 적합도 검증결과 유의수준 0.01로서 귀무가설을 기각시킬 수 없는 것으로 분석되었다. 따라서 본 연구에서 적용한 RBFN에 의한 예측모형은 평균 오차율 1.3795%를 갖는 우수한 단기수요전력 예측모형으로 판정할 수 있다.

7. 결 론

본 연구에서 적용한 RBFN은 기존에 단기전력수요예측을 위하여 적용된 BPN과 비교시 학습 및 예측결과 그리고 신경망 구조적 단순성 등 다양한 측면에서 우수성을 갖는다. 본 연구에서 목표로한 1시간단위의 단기전력수요예측 일일반전 및 경제급전등의 계통운영체계의 실제 상황에서 당일 24시간내에서의 첨두전력수요량을 예측할 수 있는 매우 중요한 정보가 된다.

본 연구의 특징은 다음과 같다.

첫째, 기존의 부하 예측 알고리즘은 대부분 역전파



(그림 9) 정규분포 확률도
 (Fig. 9) Normal Probability plot

학습 알고리즘으로 예측되어 왔으나 본 연구에서는 교사학습과 비교사 학습의 복합 모델인 Radial Basis 함수로 예측을 하여 효과적인 예측결과를 얻었다.

둘째, Radial Basis 함수 모델에 있어서 off-line으로 은닉층의 중심값을 결정하는 방법에는 Kohonen의 SOFM 알고리즘 또는 LVQ 알고리즘과 그리고 Carpenter와 Grossberg의 ART 모델이 주로 이용되었으나 본 논문에서는 보다 클러스터링 하는 데 있어서 편차가 적은 GLVQ 알고리즘을 이용한 모델을 적용하였다.

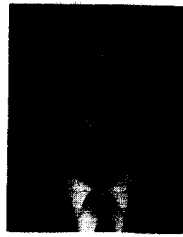
그 결과 평균 오차율 1.38%로 본 연구에서 적용한 동일한 기초입력자료를 이용한 M. H. Park et. al.의 Kalman Filter Technique(KFT) 기반의 Dynamic Model과 ARMA Model과 비교시 평균 상대오차 측면에서 KFT가 본 연구의 결과보다 약 0.356%가 더 우수한 것으로 나타났으며, ARMA보다는 약 0.06%가 더 향상된 것으로 나타났다. 그러나 상대오차율의 변동폭을 비교시 KFT의 최대 오차가 9.743%로서 본 연구의 결과보다 약 8%의 변동폭을 가지고 있어서, 결국 본 연구의 적용모형이 KFT모형 혹은 ARMA 모형보다는 상대적으로 변동폭이 적고 안정된 결과를 나타냈다.

본 연구는 RBFN을 적용시 전력사용 패턴을 평일과 휴일로 단순히 구분하여 적용하였으나, 향후 이를 더 세분화하고 RBFN에서 요구되는 local 및 spread 모수를 자동 적용시킴으로써 더욱 예측력이 높은 모형 적용이 가능할 것으로 판단된다. 그러나 본 연구에서는 단지 24시간의 전력사용 패턴을 기초로 1시간단위의 단기예측을 하였기 때문에 주별 및 월별의 급격한 전력수요의 변화에 적용하여 효과적으로 예측하는 데는 문제가 있을 것으로 판단된다. 따라서 향후의 연구에서는 예측 기본단위를 24시간단위 또는 주단위 및 월단위 예측모형을 기반으로 장기 및 중기 그리고 단기의 각각의 예측모형을 결합한 복합형 예측 모형으로 구축하기 위한 지속적인 연구가 요구된다.

참 고 문 헌

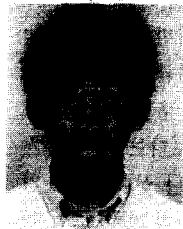
- [1] 박문희, 조형기, 정근모, 최기련, "실시간 단기부하예측을 위한 동적모형," 에너지공학, Vol. 4, No. 3, Dec., 1995, pp. 387-393.
- [2] Yuan-Yih Hsu, Chien-Chun Yang, "Electronic Load Forecasting," Applications Neural Networks, A. Murry(ed), 1995, pp. 157-189.
- [3] Ho, K. L., Hsu, Y. Y., Chen, C. F., Lee, T. E., Liang, C. C., Lai, T. S. and Chen, K. K., "Short Term Load Forecasting of Taiwan Power System using a Knowledge-based Expert System," *IEEE Trans. on Power Systems*, Vol. 5, No. 4, Nov., 1990, pp. 1214-1221.
- [4] S. Vemuri, et. al., "On-Line Algorithms for Forecasting Hourly Loads of Electric Utility," *ibid.*, Vol. PAS-100, 1981, pp. 3775-3784.
- [5] D. C. Park, M. A. El-Sharkawi, R. J. Marks II, L. E. Atlas and M. J. Damborg, "Electric Load Forecasting using an Artificial Neural Network," *IEEE Transaction on Power Systems*, Vol. 6, No. 2, May, 1991, pp. 442-449.
- [6] T. M. Peng, N. F. Hubele and G. G. Karady, "Advanced in the Application of Neural Networks for Short-Term Load Forecasting," *IEEE Transaction on Power Systems*, Vol. 7, No. 1, Feb., 1992, pp. 250-257.
- [7] K. Y. Lee, J. H. Park, "Short-Term Load Forecasting using an Artificial Neural Network," *IEEE Transaction on Power Systems*, Vol. 7, No. 1, Feb., 1992, pp. 124-131.
- [8] S. T. Chen, D. C. Yu, A. R. Moghaddamjo, "Weather Sensitive Short-Term Load Forecasting using Non Fully Connected Artificial Neural Network," *IEEE Transaction on Power Systems*, Vol. 7, No. 3, Aug., 1992, pp. 1098-1105.
- [9] K. L. Ho, Y. Y. Hsu and C. C. Yang, "Short-Term Load Forecasting using a Multilayer Neural Network with an Adaptive Learning Algorithm," *IEEE Transaction on Power Systems*, Vol. 7, No. 1, Feb., 1992, pp. 141-149.
- [10] Nikhil R. Pal, James C. Bezdek, Eric C. K. Tsao, "Generalized Clustering Networks and Kohonen's Self-Organizing Scheme," *IEEE Trans. on Neural Networks*, Vol. 4, No. 4, Jul., 1993, pp. 549-557.

- [11] Phillips D. Wasserman, "Advanced Methods in Neural Computing," Van Nostrand Reinhold, 1993.
- [12] K. L. Ho, Y. Y. Hsu, C. F. Chen, T. E. Lee, C. C. Liang, T. S. Lai and K. K. Chen, "Short-Term Load Forecasting of Taiwan Power System using a Knowledge-Based Expert System," *IEEE Transaction on Power Systems*, Vol. 5, No. 4, Nov., 1990, pp. 1214-1221.
- [13] T. Kohonen, "The Self-Organizing Map," *Proc. of IEEE*, Vol. 78, No. 9, 1990, pp. 1464-1480.
- [14] 조형기, 민준영, 최종욱, "클러스터링 방법을 이용한 차종인식모형," *정보처리학회 논문지* Vol. 3, No. 2, 1996. 3., pp. 369-379.
- [15] 민준영, "신경망클러스터링의 성능평가," 성균관대학교 박사학위논문, 1995.
- [16] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Mar., 1960, pp. 35-45.



민 준 영

1982년 아주대학교 산업공학과 졸업(학사)
 1989년 성균관대학교 경영대학원 정보처리학과 졸업(석사)
 1995년 성균관대학교 대학원 전산통계전공 졸업(박사)
 1993년 10월~1996년 3월 상지대학교병설전문대학 전자계산과 전임강사
 1996년 4월~현재 상지대학교병설전문대학 전자계산과 조교수
 관심분야: 신경회로망, 패턴인식



조 형 기

1987년 아주대학교 산업공학과 졸업(학사)
 1989년 아주대학교 대학원 산업공학과 졸업(석사)
 1994년~현재 아주대학교 박사과정 수료
 1991년~현재 아주대학교 교통연구소 전임연구원
 1993년~현재 상명대학교 정보과학과 강사
 관심분야: 신경회로망, IVHS