변압기의 내부 구조 격자화와 신경망을 이용한 부분방전 위치추정 연구

(A Study on The Estimation of Partial Discharge Location Using Division of Internal Structure of Transformer and Neural Network)

이양진* · 김재철* · 김용성* · 조성민*
(Yang-Jin Lee · Jae-Chul Kim · Young Sung Kim · Sung-Min Cho)

Abstract

This paper suggests the method for estimating a partial discharge (PD) location using divide of the inside transformer as a grid. The PD location is found swiftly and economically compared with the typical method detecting a PD. The reason is that the location of PD is detected in the section. The estimation of PD location is trained using the Neural Network. JavaNNS(Java Neural Network Simulator) and SNNS(Stuttgart Neural Network Simulator) are used for searching the location of PD. The simulation procedure is following. The transformer is assumed that the case is a regular hexahedron. The sensor is installed in a proper location. A section of PD location is set as a target, and training set is studied with several PD locations in the inside of the transformer. As a result of training process, the learning capability of neural network is excellent. The PD location is detected by division of internal structure of transformer and application of neural network.

1. 서 론

변압기 내부에는 core, frame, copper, conductors 등의 구조물이 있다. 그 내부 구조물로 인하여 PD로부터 전자기파의 발산은 많은 반사와 굴절을 포함한 다수 경로를 통해 센서에 도달한다. 전자기적 신호가 전파하지 못할 지역의 주요한 내부 요인을 설명하는 것에 최근 연구가 집중되고 있다[1].

이러한 추세에서 본 연구에서는 내부 구조에 큰 영향을 받지 않기 위해 변압기 구조를 격자 화시켜 신경망을 적용하여 패턴을 나누어 훈련시켜 PD발생구간을 빨리 찾기 위한 시뮬레이션을 시행하였다. 시뮬레이션 프로그램은 JavaNNS와 SNNS를 사용하였다. SNNS는 Stuttgart Neural Network Simulator의 약자로서 Unix 기반의 신경망 시뮬레이션이고, Tubingen에서 개발한 JavaNNS는 SNNS에 JAVA GUI를 적용한 비쥬얼한 시뮬레이션이다. 본 연구에서는 JavaNNS와 SNNS를 이용하여 Batch file을 만들어서 시뮬레이션을 하였다.

2. 본 론

2.1. 오차 역전파(back-propagation)를 사용한 신경망의 개념

2.1.1. 신경망의 기본 고려사항

신경망은 많은 옵션들을 제공한다. 그 옵션은 다양한 PD 패턴인식의 업무에 유용하다. 신경망은 또한 패턴을 학습하는데 짧은 시간을 필요로 한다. 일단 phase 학습이 완료되면, 그것을 알려지지 않은 입력 패턴들에 대한 정당한 일반화를 시키는 것에 적용이 가능하다. 게다가 신경망은 다른 통계학적 분포에서 자신들을 적응시키는 능력이 다른 분류방법보다 더 우세하다. PD 패턴인식을 처리할 수 있는 신경망 시스템을 획득하기 전에 고려해야 할 사항이 2가지가 있다. 그것은 특징 추출과 분류이다. 특징추출의 목적은 특색 있는 각 PD소스를 캡쳐하는 것이다. 분류의 주요 개념은 공통원소를 가지고 있지 않은 특징 공간을 나눔으로써 결정을 줄이는 것이다. 그것은 여러 다른 신경망 패러다임으로 알려져 있다. 이 패러다임들은 그들의 능력에 따라 다르며 각각 자신의 구조와 학습 순서를 가진다. 지도학습방법(supervised learning method)과 비지도 학습방법이 있다. 특별한 신경망 모델의

선택은 요구되는 적용 방법에 의존한다. 본 논문에서는 PD 인식 수행을 위해 다층 퍼셉트론(multi-layer perceptron: MLP)을 쓴다. 다층 퍼셉트론의 주요 이점은 대단히 비선형적인 클러스터 모양을 만드는 능력이다. 그것은 다른 신경망 패러다임과 함께 경계를 비교하는 결정의 근사치를 취득할 수 있다 그러므로 PD 패턴들 사이에 더 정밀한 구별을 위해 다층 퍼셉트론을 사용했다.

다음 그림은 신경망의 기본적인 뉴런 모델이다.

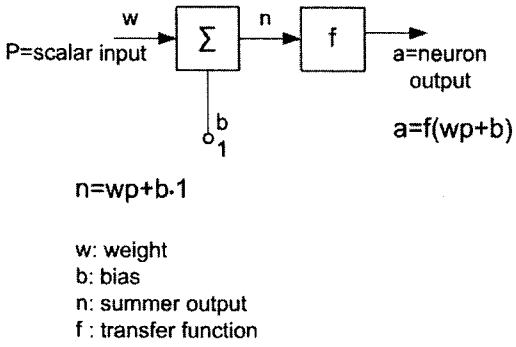

그림 1. 단일-입력 뉴런
Fig. 1. Single-Input Neuron

그림 1.에서는 단일-입력 뉴런을 보여준다. 스칼라 입력 p는 스칼라 웨이트 w에 의해 곱해지고, 다른 입력인 바이어스와 1의 곱이 더해진다. 그 합의 출력인 n은 전달함수 f로 들어간다. 전달함수를 거쳐 스칼라 뉴런 출력 a를 생산한다.

2.1.2 다층 퍼셉트론의 구조와 학습계획

다층 퍼셉트론은 전방향 네트워크(feed-forward network)이다. 뉴런이라 불리는 상호 연결된 처리요소(processing element)로 조직된 위상기하학의 구성이다.

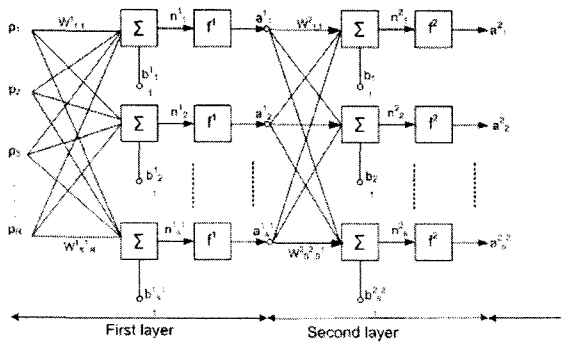

그림 2 다층 퍼셉트론
Fig. 2. Multi-layer perceptron

위의 그림 2. 에서는 다층 퍼셉트론의 구성을 보여준다. 다층 퍼셉트론은 입력층(input-layter), 출력층(output-layer), 하나 이상의 은닉층(hidden-layer)으로 구성된다. 신경망에서 학습은 요구되는 출력으로 적응시킨다. 이 적응은 웨이트(weight)에서 변화를 통해 실행되고, 각각 입력 벡터는 정확한 출력을 얻는 쪽으로 차차 수렴한다. 이 논문에서 다층 퍼셉트론의 학습계획은 오차 역전파이다. 이 학습 알고리즘의 기본 원리는 다음과 같다. 입력층의 각 유니트에 입력패턴을 주면, 이 신호는 각 유니트에서 변환되어 중간층에 전달되고 최후에 출력층에서 신호를 출력하게 된다. 이 출력값과 기대값을 비교하여 차이를 줄여나가는 방향으로 연결강도를 조절하고, 상위층에서 역전파하여 하위층에서는 이를 근거로 다시 자기층의 연결강도를 조정해 나간다. 지도학습에서는 입력 및 원하는 출력(목표출력) 패턴(벡터)이 네트워크에 제시된다. 네트워크는 입력층에 주어진 입력패턴이 출력층에 전파되면서 변한 출력패턴을 목표패턴과 비교한다. 네트워크에서 출력된 패턴이 목표패턴과 일치하는 경우에는 학습이 일어나지 않는다. 그렇지 않은 경우는 얻어진 출력패턴과 목표패턴의 차이를 감소시키는 방향으로 네트워크의 연결강도를 조절하여 학습을 한다.[7]

2.2. 시뮬레이션을 위한 기본 이론

전력용 변압기 내에서 발생한 부분방전의 위치 예측을 위한 기존 알고리즘의 대다수는 수학적인 해석을 통하여 처리하였다. 그러나 비선형 특성이 포함되어 있고, 부분 방전에 따라 검출된 초음파 신호 자체에 잡음이 포함되어 오차를 가지고 있다면 수학적인 처리가 곤란하다. 그래서 검출된 초음파 신호에 이동 평균과 해밍창의 적용으로 오차를 최소화 시킬 수 있다. 그 후 신경 회로망을 도입하여 부분 방전의 발생 위치를 예측한다. 신경 회로망은 지도 학습의 일종인 다층 구조의 오차 역전파 알고리즘을 사용한다.[5]

부분 방전시 발생되는 초음파 신호는 초음파 센서를 통해 검출된다. 그리고 계측 시스템에 의하여 A/D 변환된 신호 처리가 수행된 후 컴퓨터에 전송된다. 컴퓨터는 전송된 초음파 신호로서 상호상관 기법 등과 같은 신호처리 이론을 이용한다. 그리고 초음파 센서쌍에서 검출한 초음파 신호들의 시간차를 구한 후, 각 시간차를 거리차로 변환한다.

이 때 신경 회로망은 부분 방전 발생 위치에 따른 초음파 센서쌍에서의 거리차로 학습된 오차 역전파 알고리즘을 사용한다. 신경 회로망은 각 초음파 센서쌍의 거리차를 입력으로 사용하고, 전력용 변압기 내의 부분 방전 발생 위치를 출력으로 나타낸다. 부분 방전 발생

위치를 예측하기 위하여 3쌍(6개)의 초음파 센서를 전력용 변압기 외벽에 부착한다. 신경 회로망의 훈련 예제는 다음 식으로 계산한다.

$$d_{i(1-2)} = \sqrt{(x_p-x_{i1})^2+(y_p-y_{i1})^2+(z_p-z_{i1})^2} - \sqrt{(x_p-x_{i2})^2+(y_p-y_{i2})^2+(z_p-z_{i2})^2} \quad (1)$$
$$(1 \le i \le 3)$$

$d_{i(1-2)}$: 부분방전 발생 위치로부터 초음파 센서쌍 i의 AS_{i1}과 AS_{i2}의 거리차(m)

x_p, y_p, z_p : 부분 방전 발생 위치의 3차원 좌표

x_{i1}, y_{i1}, z_{i1} : 초음파 센서쌍 i의 AS_{i1}의 3차원 좌표

x_{i2}, y_{i2}, z_{i2} : 초음파 센서쌍 i의 AS_{i2}의 3차원 좌표

즉, 부분방전의 발생 위치를 온 라인으로 예측하기 위한 시스템의 전체적인 블럭도는 다음 그림과 같다.

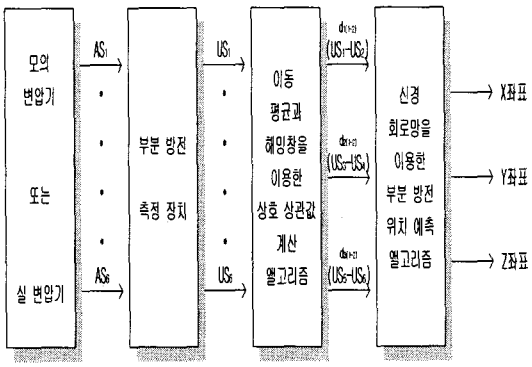

그림 3. 부분방전 발생 위치의 온 라인 예측을 위한 시스템 구성도

Fig. 3. System diagram for on-line estimation of partial discharge location

2.3. JavaNNS와 SNNS를 이용한 Neural Network 시뮬레이션

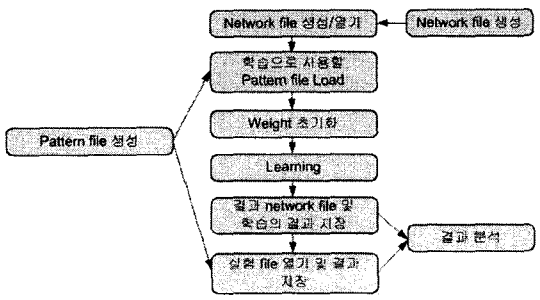

그림 4. JavaNNS의 블럭선도

Fig. 4. Block-diagram of JavaNNS

그림.4 는 JavaNNS의 블럭선도를 나타낸다. JavaNNS, SNNS는 위와 같이 입력 pattern file를 생성 후 network를 생성한다. 그리고 학습으로 사용할 pattern file을 불러들인다. 그 후 옵션을 통하여 weight를 초기화 시킨 후 cycle과 학습률을 입력한 후 learning을 시킨다. 그리고 학습 data를 저장하고 결과를 분석하여 원하는 목표값에 얼마나 적은 오차로 근접해 가는지를 확인한다. 만약 수렴하지 않을 경우 weight와 bias를 변화시키거나 학습 반복수, hidden layer등 네트워크를 조정하여 원하는 목표치를 얻는다.

JavaNNS, SNNS 프로그램을 이용하여 다음과 같은 모의 변압기를 시뮬레이션 하였다.

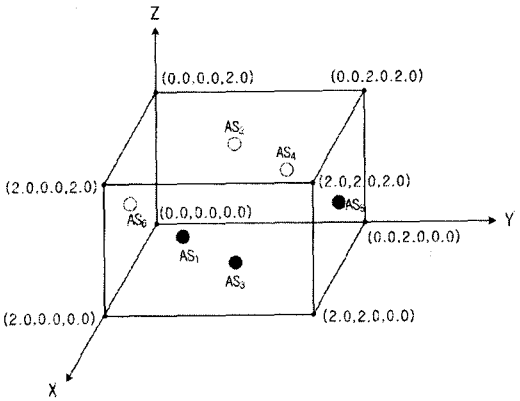

그림 5. 모의 변압기

Fig. 5. Structure of the model transformer.

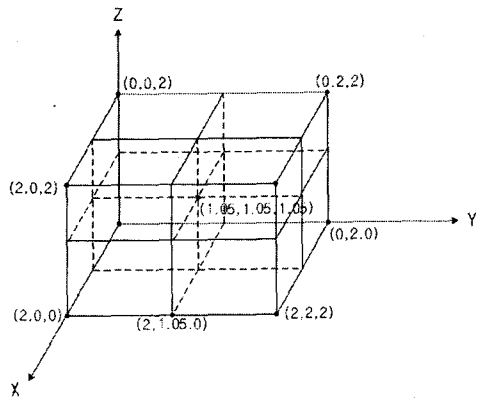

그림 6. 8등분으로 격자화 시킨 모의 변압기

Fig. 6. Model transformer that is divided by 8 classification.

위의 그림과 같이 시뮬레이션을 하기위하여 다음의 순서로 계획을 잡았다.

1. 모의 변압기 외함 크기 결정
2. 센서의 위치 선정(좌표)

3. 격자의 크기(범위) 선정
4. 목표값 결정(번지)
5. 목표값에 오차가 적은 방향으로 weight와 bias선정

간단한 시뮬레이션을 하기위해 단위는 지정하지 않고 0~2의 수치를 사용하겠다. 표 1.은 등분된 모의 변압기 범위를 나타낸다. 기본적으로 변압기를 8개의 격자로 분류하였다. 그리고 각각의 격자에 번지(address)를 부여 하였다. 시뮬레이션 목표 출력값을 8개 격자의 번지로 하였으며 입력값은 PD의 가상 좌표와 센서쌍간의 거리차를 각각 해보았다. 그리고 만약 PD의 위치가 격자의 경계면에 위치 할 경우 패턴 구분이 어려우므로 경계를 우측과 위쪽으로 0.05 이동하여 정하였다. 즉 좌표가 (1, 1, 1)인 경우 번지 (0, 0, 0)에 속하는 것으로 하였다. PD의 위치는 각 x, y, z좌표 각각 0, 0.5, 1, 1.5 순서로 정하였고 훈련예제는 125개를 하였다.

표 1. 등분된 모의 변압기 범위
Table 1. The range of dividend test transformer

좌표	범위
X 좌표	0 ~ 2
Y 좌표	0 ~ 2
Z 좌표	0 ~ 2

센서쌍의 위치는 다음과 같다.

AS_1 = (2, 0.5, 1.5) AS_2 = (0, 0.5, 1.5) AS_3 = (2, 1.5, 0.5)
AS_4 = (0, 1.5, 0.5) AS_5 = (1, 2, 1) AS_6 = (1, 0, 1)

먼저 입력이 PD의 x, y, z좌표를 정규화(normalization) 한 값으로 하고 출력은 8개의 격자의 번지로 한 시뮬레이션을 살펴보기로 하자. javaNNS, SNNS를 사용하기 위해서는 입·출력값을 normalization을 해 주어야 한다. 여기서 normalization은 다음의 식을 사용하였다.

$$\frac{x-Min}{|Max-Min|}\cdot|\max-\min|+\min \qquad (2)$$

x : 원래의 값
Max, Min : 원래 값의 최대 최소값
max, min : 바꾸고 싶은 최대 최소값

각 입력 x, y, z좌표의 0, 0.5, 1, 1.5, 2의 값은 0과 1 사이의 0.05, 0.275, 0.5, 0.725, 0.95의 값으로 normalization 된다. 출력은 8개의 격자로 나누어진 번지이다. 그 번지는 각 x, y, z를 1과 2를 사용하여 111, 112, 121, 122, 211, 212, 222로 되었고 이는 001, 010, 011, 100, 101, 110, 111로 normalization 된다. 만약 격자가 더 늘어나면 x, y, z를 1, 2, 3 ... n 으로 해서 normalization시킨 후 javaNNS에 적용시키면 된다.

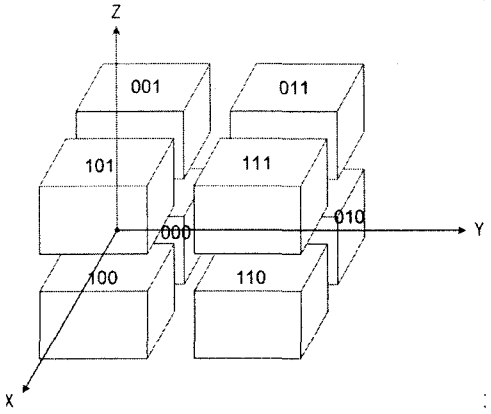

그림 7. 변압기에서 분류된 번지
Fig. 7. Address of division in the transformer

위와 같이 입·출력을 결정하고 JavaNNS로 시뮬레이션을 한다.

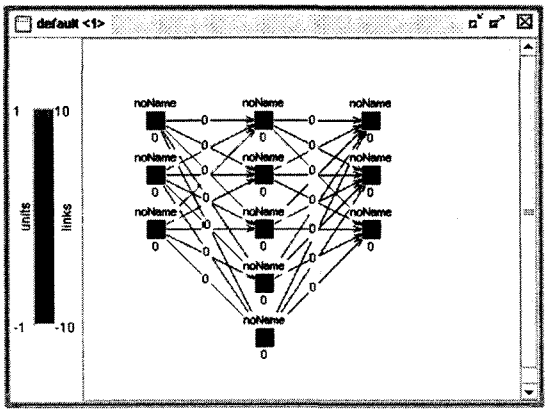

그림 8. JavaNNS의 네트워크
Fig. 8. The network of JavaNNS

위 그림은 학습 network으로 입력층 3개, 은닉층 5개, 출력층 3개로 시뮬레이션을 하였다. 이 외에도 네트워크 은닉층을 1개 2개 3개로 변형해가며 시뮬레이션을 하였으나 은닉층이 적을 때에는 오차가 많이 났다. 또한 출력층을 1개로 하여 0.5~0.95를 8등분하여 normalization 시켜서 시뮬레이션 해 보았으나 구분하는 수의 크기가 작아서 오차가 생겼다. 그래서 출력층을 3개로 해서 구분하였다.

초기치 함수는 random weight로 해 주었으며 weight parameter는 -1에서 1까지, learning function은 back-propagation으로, 학습률은 0.2로 그리고 cycle은 100000으로 시뮬레이션을 하였다

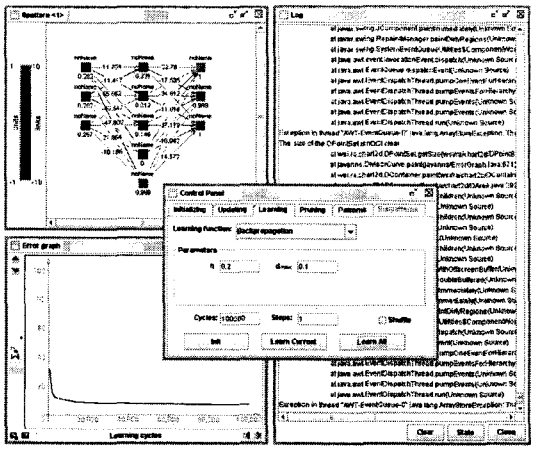

그림 9. JavaNNS 시뮬레이션화면
좌측상단: network, 좌측하단: error graph, 우측: log
중앙: control panel
Fig. 9. Display of JavaNNS simulation
upper of left plane: network, down of left plane: error graph, right plane: log, center: control panel

위의 그림 9.는 시뮬레이션 화면이며 결과는 다음과 같다. 표 2.는 그림 7.에 대한 PD 실 좌표의 시뮬레이션을 한 값이다. 목표 출력 과 비교하여 시뮬레이션 값은 0.5보다 크면 1로 생각하고 반대이면 0으로 해석한다. 오차가 매우 작은 것을 볼 수 있다.

표 2. 실 좌표 입력 시뮬레이션 결과
Table 2. Input simulation result of real coordinates

번호	PD 발생 위치 좌표			목표 출력	시뮬레이션 값		
	x	y	z	x y z	x	y	z
1	0.05	0.275	0.95	0 0 1	0.00015	0.00179	0.98641
2	0.05	0.725	0.95	0 1 1	0.0001	0.90079	0.98947
3	0.275	0.05	0.95	0 0 1	0.00077	0.00059	0.98971
4	0.725	0.05	0.95	1 0 1	0.90174	0.00224	0.99868
5	0.725	0.95	0.95	1 1 1	0.90039	0.99703	0.99853
6	0.05	0.05	0.05	0 0 0	0.00098	0.00046	0.00066
7	0.05	0.725	0.275	0 1 0	0.00057	0.90169	0.00199

다음 표 3은 PD발생 지점과 3개의 센서쌍 간의 거리차를 normalization한 것을 입력으로 넣어 목표출력으로 학습한 시뮬레이션 결과를 나타낸다.

표 3. 센서쌍과 PD위치(좌표)와 거리차를 입력으로 한 시뮬레이션 결과
Table 3. Simulation result of input distance between PD location and sensor pairs

번호	PD위치와 센서간의 거리차			목표 출력	시뮬레이션 값		
	AS_{2-1}	AS_{4-3}	AS_{6-4}	x y z	x	y	z
1	0.851	0.700	0.626	0 0 1	0.001	0	0.991
2	0.763	0.725	0.373	0 1 1	0	0.999	0.998
3	0.678	0.594	0.763	0 0 1	0.001	0	0.869
4	0.321	0.405	0.763	1 0 1	1	0	0.884
5	0.382	0.382	0.236	1 1 1	0.915	1	0.992
6	0.717	0.717	0.732	0 0 0	0	0	0.001
7	0.732	0.95	0.354	0 1 0	0.001	0.913	0

표 3.은 PD의 위치와 센서쌍간의 거리차를 시뮬레이션한 결과이다. 오차가 매우 적어지고 목표결과 값으로 수렴하는 것을 볼 수 있다. 이와 같이 실 변압기를 적절한 수의 격자로 나누어 pattern화 시킨 후 적절한 단위로 부분방전 점을 결정하여 트레이닝 시켜 신경망 시뮬레이션을 이용하면 PD 위치추정을 할 수 있을 것이다.

앞의 기본 8격자를 확장하여 다음으로는 (3 x 3 x 3)으로 27개의 격자 화를 시켰다.

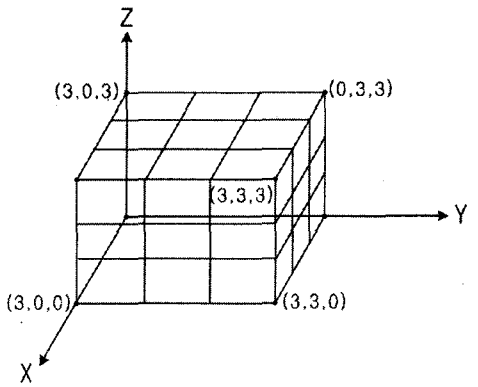

그림 10. 27등분으로 격자 화시킨 모의 변압기
Fig. 10. The model transformer of 27 equal part grid

그림 10.에서 각 27격자의 번지는 정규화되어 결정되었다. 모의 변압기의 크기는 (가로 × 세로 × 높이)

각 3이다. PD의 위치는 각 x, y, z 축으로 0.5씩 증가한다. 입력 x, y, z 좌표의 각 값들(0, 0.5, 1, 1.5, 2, 2.5, 3)은 0에서 1까지의 범위의 각 값들(0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95)로 normalization된다. 출력은 27개의 격자로 나누어지고 번지가 부여된다. 각 x, y, z는 (0.3, 0.3, 0.3)에서 (0.9, 0.9, 0.9)로 번지(address)가 주어진다. 그림 11.은 27가지 격자로 분류된 모의 변압기를 보여준다.

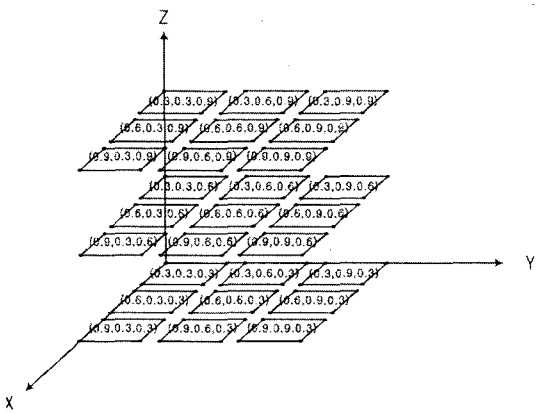

그림 11. 27가지 격자로 분류된 변압기
Fig. 11. Transformer that is classified by 27 lattices

경계면위의 PD점은 일단 고려하지 않기 위해서 경계면은 앞의 모델 변압기 8 격자 화와 마찬가지로 0.05씩 우측과 위쪽으로 이동시켰다. hidden layer는 5개 8개 10개로 변경하면서 시뮬레이션을 하였다. 5개 정도가 적정하였으며 세 경우 다 큰 차이를 보이지는 않았다.

표 4. 실 좌표 입력 시뮬레이션 결과
Table 4. Input simulation result of real coordinates

번호	PD 발생 위치 좌표	목표 출력	시뮬레이션 값
	x_{nor} y_{nor} z_{nor}	x_{nor} y_{nor} z_{nor}	x y z
1	0.2 0.2 0.95	(0.3 0.3 0.9)	0.283 0.282 0.891
2	0.2 0.5 0.8	(0.3 0.6 0.9)	0.280 0.526 0.813
3	0.35 0.5 0.65	(0.3 0.6 0.6)	0.394 0.535 0.693
4	0.65 0.05 0.95	(0.6 0.3 0.9)	0.685 0.216 0.895
5	0.65 0.65 0.95	(0.6 0.6 0.9)	0.686 0.697 0.904
6	0.95 0.5 0.05	(0.9 0.6 0.3)	0.890 0.521 0.214
7	0.95 0.8 0.65	(0.9 0.9 0.6)	0.894 0.817 0.689

nor : 정규화(normalization)

각각 Input pattern(:PD좌표), Output pattern(:격자구간), 시뮬레이션 값은 모두 0과 1사이의 숫자로 normalization된 값들이다. 여기서 트레이닝은 343개를 하였고 cycle는 100,000번을 시뮬레이션 하였다. 대체적으로 잘 수렴한다.

3. 결 론

위의 모의 변압기 모형 시뮬레이션 결과에서 보여주듯이 javaNNS, SNNS 프로그램 등을 이용하여 부분방전 위치추정을 할 수 있다. 이를 실 변압기에 적용하여 외함의 크기에 따라 적절한 격자화를 시켜 입력값(PD 발생 좌표, PD 발생 좌표 ~ 3개 센서쌍 간의 거리차)을 training 시킨다면 좀 더 빠른 부분방전 위치추정이 가능 할 것이다.

아직까지는 변압기 내부의 구조물내의 PD에 대한 정확한 위치추정은 어려운 실정이다. 그러므로 각 격자간 변압기 내부 구조물에 의한 패턴도 분석해야 한다. 이것을 트레이닝 시키고 시뮬레이션 한다면 더 정확한 부분방전 위치추정을 할 수 있을 것이다.

참 고 문 헌

[1] Matin D. JUDD, Li Yang, Ian B.B. Hunter, "Partial Discharge Monitoring for Power Transformers Using UHF Sensors Part1 : Sensors and Signal Interpretation", IEEE Electrical Insulation Magazine, Vol.21, No.2, pp5-14, March/April 2005

[2] Martin T. Hagen, Howard B. Demuth, Mark Beale, "Neural Network Design", PWS Publishing Company, 1996

[3] Paul A. Lynn, Wolfgang Fuerst, "Introductory Digital Signal Processing with Computer Applications", John Wiley & Sons, Inc, pp231-pp386, 1998

[4] SNNS(Stuttgart Neural Network Simulator), User Manual, Version 4.2

[5] Yong-Han Yoon, "A Monitoring and Diagnostic Expert System for Power Transformers", Soongsil University, Ph.D. graduation thesis 1996.12

[6] S.J.Lim, Y.J.Jun, S.Y.Yun, "A Study on On-Line Estimation of Partial Discharge Location in Power Transformer using Ultrasonic - Ultrasonic Method", Report of Soongsil University to KERI, 2000.10.31

[7] S. H. Park, K. W. Lee, K. J. Lim and S. H. Kang, "Classification of External and Internal PD signals Generated in Model Transformer by Neural Network", Proceeding of the 7th International Conference on Properties and Applications of Dielectric Materials, pp. 463-466, June 1-5 2003 Nagoya.