

# 신경망을 위한 해석적 결정경계 특징추출 알고리즘

고 진 욱(高 纘 穉), 이 철 희(李 哲 熙)  
연세대학교 전기-컴퓨터 공학과  
전화: (02) 361-2779 팩스: (02) 312-4584

## Analytical Decision Boundary Feature Extraction for Neural Networks

Jinwook Go and Chulhee Lee  
Dept. Electrical and Computer Engineering, Yonsei University  
Email: chulhee@yonsei.ac.kr

### Abstract

Recently, a feature extraction method based on decision boundary has been proposed for neural networks. The method is based on the fact that all the features necessary to achieve the same classification accuracy as in the original space can be obtained from the vectors normal to decision boundaries. However, the normal vector was estimated numerically, resulting in inaccurate estimation and a long computational time. In this paper, we propose a new method to calculate the normal vector analytically. Experiments show that the proposed method provides a better performance.

### I. 서론

신경망은 패턴 인식, 동적 시스템 제어, 통신 시스템 등과 같은 많은 비선형 신호처리 문제에 성공적으로 적용되었다 [1]. 그러나 신경망을 복잡한 문제에 적용하기 위해 학습시킬 경우 그 신경망을 구성하는 연결 강도(connection weight)의 수가 상당히 많다. 신경망의 연결 강도는 신경망 입력 벡터의 차원 증가에 따라 상당히 증가할 수 있으므로, 입력의 차원을 줄이면서도 동일한 학습 성능을 얻을 수 있는 특징추출(feature extraction) 과정이 요구된다. 그러나 신경망에 적용할 수 있는 특징추출 알고리즘은 다소 부족한 형편이다. 기존의 특징추출 알고리즘인 principal component analysis(PCA) 방법과 canonical analysis(CA) 방법을 신경망의 특징추출에 적용할 수 있지만, 이러한 방법들은 복잡한 결정경계(decision

boundary)를 형성하는 신경망의 특징을 이용하지 않으므로 적절한 신경망 특징추출 알고리즘이라고 볼 수 없다.

최근 발표된 결정경계 특징추출(DBFE) 알고리즘에서는 특징추출 과정 전의 원래 데이터로부터 얻을 수 있는 분류 정확도와 동일한 분류 정확도를 얻기 위한 특징벡터(feature vector)는 분류기(classifier)가 정의하는 결정경계에 대한 수직벡터(normal vector)로부터 구할 수 있다는 이론을 제시하였다 [3]. 특히, DBFE 알고리즘은 결정경계로부터 직접 특징벡터를 추출하기 때문에 확률밀도함수(probability density function)의 가정 없이 임의의 결정경계를 형성하는 신경망의 중요한 특징을 충분히 이용한다 [2]. 이러한 DBFE 알고리즘을 신경망의 특징추출에 적용한 결과 기존의 특징추출 알고리즘과 비교하여 우수한 성능 향상을 보여주었다. 그러나 DBFE 알고리즘은 신경망의 결정경계의 수직벡터를 구하기 위해 결정경계의 기울기(gradient) 근사 방법을 사용하였다. 그 결과 결정경계의 수직벡터를 부정확하게 구하는 경우가 발생할 수 있고 계산 시간이 길어지는 단점이 존재한다. 본 논문에서는 이러한 결정경계의 수직벡터를 해석적으로 계산하는 방법을 제안하고 계산에 필요한 식을 유도한다. 이러한 해석적 결정경계 특징추출(ADBFE)을 통해 DBFE 알고리즘의 특징추출에 필요한 계산 시간을 반정도로 줄일 수 있고 또한 더 향상된 특징추출 성능을 얻을 수 있다.

### II. 신경망 결정경계 특징추출 알고리즘

결정경계 특징추출 알고리즘은 패턴 클래스간의 결정경계로부터 패턴 분류에 유용한 정보를 제공하는 특징(feature)을 추출하는 방법이다 [3]. 이러한 결정경계 특징추출 알고리즘은 순방향 다층 신경망의 특징

추출에 성공적으로 적용되었다 [2].

결정경계 특징추출 알고리즘을 신경망에 적용하기 위해 신경망의 결정경계 상의 점(point)들을 구하고, 이러한 점들에 대해 기울기 근사 (gradient approximation) 방법을 통해 결정경계에 수직인 벡터들을 구한다. 이와 같이 구한 수직벡터들로부터 결정경계 특징 행렬(decision boundary feature matrix)을 다음 식을 통해 추정한다.

$$\Sigma_{EDBFM} = \frac{1}{L} \sum_i N_i N_i' \quad (1)$$

이때 L은 결정경계 상의 점들의 개수를 나타내고,  $N_i$ 는 결정경계 수직벡터를 나타낸다.

결정경계 특징 행렬이 결정경계에 수직인 성분들로 이루어져 있으므로 결정경계 특징 행렬의 고유벡터(eigenvector)를 구함으로써 패턴 클래스를 분류할 수 있는 특징벡터를 얻을 수 있다. 그리고 이렇게 구한 고유벡터에 대한 고유치(eigenvalue)의 크기가 클수록 상응하는 고유벡터는 클래스를 더 효율적으로 분류할 수 있으며 고유치의 크기가 0인 고유벡터는 클래스를 분류하는데 필요치 않은 벡터이다. 즉 식 (1)의  $\Sigma_{EDBFM}$ 로부터 구한 고유치의 크기가 0이 아닌 고유벡터를 새로운 특징벡터로 사용한다.

### III. 신경망을 위한 해석적 결정경계 특징추출 알고리즘

본 논문에서는 시그모이드(sigmoid) 함수를 신경망 활성화 함수(activation function)로 갖는 순방향 신경망에 대한 특징추출을 고려한다. 또한 클래스 결정 규칙은 가장 큰 출력값을 갖는 출력 뉴런(neuron)에 대응되는 클래스를 선택하는 것으로 가정한다.

순방향 다층 신경망에서 두 클래스간에 정의되는 결정경계는 공간상에서 두 클래스에 상응하는 각 출력 뉴런의 크기가 같은 점들의 궤적이다. 즉, 두 패턴 클래스의 결정경계는 다음과 같이 정의된다.

$$OUT_1(\mathbf{X}) = OUT_2(\mathbf{X})$$

이때  $OUT_i(\mathbf{X})$ 는  $i$ 번째 출력 뉴런의 출력값을 나타내고,  $\mathbf{X}$ 는 신경망의 입력 벡터이다.

신경망 출력 뉴런의 출력값간의 차를  $h(\mathbf{X})$ 라 하면 결정경계는 다음과 같이 나타낼 수 있다.

$$\{\mathbf{X} | h(\mathbf{X}) = OUT_1(\mathbf{X}) - OUT_2(\mathbf{X}) = 0\} \quad (2)$$

식 (2)를 순방향 3층 신경망에 대해 표현하면 다음

식과 같다.

$$\{\mathbf{X} | F(\mathbf{W}_1^1 \mathbf{F}(\mathbf{W}_1 \mathbf{X})) - F(\mathbf{W}_2^1 \mathbf{F}(\mathbf{W}_1 \mathbf{X})) = 0\}$$

이때  $\mathbf{W}_1$ 는 입력층과 은닉층간의 연결 강도 행렬이고  $\mathbf{W}_1^1$ ,  $\mathbf{W}_2^1$ 는 은닉층과 출력층간의 연결 강도 벡터이며  $F$ 는 신경망 활성화 함수를 나타낸다. 그러므로 신경망 입력층의 뉴런 개수가 N개일 때, 입력벡터  $\mathbf{X}$ 에 대한 결정경계 수직벡터는 다음 식과 같이 나타낼 수 있다.

$$\nabla h(\mathbf{X}) = \frac{\partial h}{\partial x_1} x_1 + \frac{\partial h}{\partial x_2} x_2 + \frac{\partial h}{\partial x_3} x_3 + \dots + \frac{\partial h}{\partial x_N} x_N \quad (3)$$

최근 발표된 결정경계 특징추출 알고리즘에서는  $\nabla h(\mathbf{X})$ 를 구하기 위해 다음 식과 같은 기울기 근사 방법을 이용하였다 [2].

$$\nabla h(\mathbf{X}) \approx \frac{\Delta h}{\Delta x_1} x_1 + \frac{\Delta h}{\Delta x_2} x_2 + \frac{\Delta h}{\Delta x_3} x_3 + \dots + \frac{\Delta h}{\Delta x_N} x_N$$

본 논문에서는 식 (3)의 수직벡터를 해석적으로 구하기 위해 시그모이드 함수의 단조증가(monotonically increasing) 특성을 이용한다. 즉, 신경망의 학습이 종료된 경우 출력 뉴런의 시그모이드 함수의 유무는 분류 정확도에 영향을 미치지 못한다. 그러므로 식 (3)과 동일한 결정경계 식을 다음과 같이 얻을 수 있다.

$$\{\mathbf{X} | \mathbf{W}_1^1 \mathbf{F}(\mathbf{W}_1 \mathbf{X}) - \mathbf{W}_2^1 \mathbf{F}(\mathbf{W}_1 \mathbf{X}) = 0\} \quad (4)$$

은닉층 뉴런의 개수가 M개라고 가정할 때, 식 (4)는 다음과 같이 나타낼 수 있다.

$$\frac{w_{1,1}^h - w_{2,1}^h}{1 + e^{-\Phi_1^T \mathbf{X}}} + \frac{w_{1,2}^h - w_{2,2}^h}{1 + e^{-\Phi_2^T \mathbf{X}}} + \dots + \frac{w_{1,M}^h - w_{2,M}^h}{1 + e^{-\Phi_M^T \mathbf{X}}} = 0$$

이때  $\Phi_j$ 는  $N \times 1$  입력층과  $j$ 번째 은닉층 뉴런간의 연결 강도 벡터이고  $w_{k,j}^h$ 는  $j$ 번째 은닉층 뉴런과  $k$ 번째 출력층 뉴런간의 연결 강도이다.

$j$ 번째 은닉층 뉴런의 전체 입력은 다음 식과 같다.

$$z_j = \Phi_j^T \mathbf{X} = \phi_{j,1} x_1 + \phi_{j,2} x_2 + \dots + \phi_{j,N} x_N \quad (j = 1, \dots, M)$$

이때  $\phi_{j,i}$ 는  $i$ 번째 입력층 뉴런과  $j$ 번째 은닉층 뉴런간의 연결 강도이다. 또한  $j$ 번째 은닉층 뉴런의 출력은 다음 식과 같다.

$$z_j = \frac{1}{1+e^{-z_j}} = \frac{1}{1+e^{-\phi_{j,i}x}} \quad (5)$$

식 (5)를  $i$  번째 입력  $x_i$  에 대해 편미분하면 다음과 같다.

$$\begin{aligned} \frac{\partial z_j}{\partial x_i} &= \frac{\partial z_j}{\partial z_j} \frac{\partial z_j}{\partial x_i} = \frac{\partial z_j}{\partial x_i} \phi_{j,i} x_i = \frac{e^{-z_j}}{(1+e^{-z_j})^2} \phi_{j,i} x_i \\ &= \frac{1}{1+e^{-z_j}} \left(1 - \frac{1}{1+e^{-z_j}}\right) \phi_{j,i} x_i \\ &= z_j(1-z_j)\phi_{j,i}. \end{aligned}$$

최종적으로  $i$  번째 입력  $x_i$  에 대한  $h(\mathbf{X})$  의 편미분은 다음 식과 같다.

$$\frac{\partial h}{\partial x_i} = \sum_{j=1}^M \phi_{j,i} z_j(1-z_j)(w_{1,j} - w_{2,j}).$$

#### IV. 실험 및 고찰

제안된 해석적 결정경계 특징추출 알고리즘의 성능을 평가하기 위해, 정규(normal) 분포를 갖는 발생 데이터와 실제 원격탐사(remote sensing) 데이터를 사용해 실험을 하였다. 비교를 위해 PCA, CA, DBFE 알고리즘을 사용하였다. 모든 실험에서 신경망의 은닉층 뉴런의 개수는 입력벡터 차원의 3 배를 사용하였고, 학습 알고리즘은 오류 역전파 알고리즘을 사용하였다.

첫번째 실험으로 발생한 데이터에 대한 성능을 평가하기 위해 다음에 주어진 평균벡터와 공분산 행렬을 갖는 3 개의 클래스로부터 정규분포를 가정하여 각 클래스 당 2000 개의 샘플을 발생시키고 500 개를 학습 데이터로 사용하고 나머지를 시험 데이터로 사용하였다.

$$\begin{aligned} \text{클래스 1: } \mathbf{M}_1 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \Sigma_1 &= \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix} \\ \text{클래스 2: } \mathbf{M}_2^1 &= \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} & \Sigma_2^1 &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix} \\ & \mathbf{M}_2^2 = \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix} & \Sigma_2^2 &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix} \\ \text{클래스 3: } \mathbf{M}_3^1 &= \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} & \Sigma_3^1 &= \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix} \\ & \mathbf{M}_3^2 = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix} & \Sigma_3^2 &= \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix} \end{aligned}$$

표 1 과 2 는 3 클래스 분류 문제에 대한 4 가지 특징추출 방법의 성능을 보여준다. 표 1 과 2 에서 DBFE 알고리즘은 원래 데이터로부터 얻을 수 있는 분류 정확도를 2 개 특징으로부터 획득할 수 있음을 보여준다. 즉 2 개의 특징을 사용한 경우 DBFE 알고리즘은 학습 데이터의 경우 84.2%의 분류 정확도를 보인 반면, canonical analysis 는 65.8%, principal component analysis 는 59.3%의 성능을 보여주고 있다. 이는 기존의 특징추출 방법이 찾지 못하는 우수한 특징들이 존재함을 암시한다. 그리고 DBFE 방법과 ADBFE 방법간의 분류 정확도의 차가 거의 없음을 알 수 있다. 이러한 사실은 이 문제의 경우엔 기울기 근사 방법을 통해 결정경계의 수직벡터가 정확하게 추정되었음을 의미한다.

Table 1. 특징추출 성능 비교 (학습 데이터).

No. Feature	분류 정확도 (%)			
	PCA	CA	DBFE	ADBFE
1	34.5	52.5	62.8	62.7
2	59.3	65.8	84.1	84.1
3	84.1	84.4	84.2	84.2

Table 2. 특징추출 성능 비교 (시험 데이터).

No. Feature	분류 정확도 (%)			
	PCA	CA	DBFE	ADBFE
1	34.2	51.7	63.6	63.4
2	59.6	66.6	85	85
3	84.7	84.8	84.8	84.8

다음 실험에서는 농업 지역에서 채취한 고차원 다중스펙트럴(multispectral) 데이터인 실제 원격탐사 데이터를 사용하였다 [4]. 표 3 에서 보여주듯이 문제는 4 개의 클래스로 구성되고, 각각의 클래스는 2 개의 부분 클래스의 데이터의 합성으로 이루어진다. 각 클래스 스로부터 500 개의 학습 데이터를 무작위로 선택하고 나머지 데이터를 시험 데이터로 사용하였다. 신경망은 수렴을 보장할 수 있도록 10000 번을 학습시켰다.

그림 1 과 2 는 각각 학습 데이터와 시험 데이터에 대한 4 개의 특징추출 알고리즘의 성능을 보여준다. 원래 데이터의 차원은 17 차원이고, 17 개의 특징을 사용하였을 때 학습 데이터와 시험 데이터의 분류 정확도는 각각 98.6%와 94.2%이다. 그림 1 과 2 에서 볼 수 있듯이 4 개 이상의 특징이 사용된 경우 제안된 ADBFE 알고리즘이 나머지 특징추출 알고리즘들과 비교하여 우수한 성능을 보여준다. 특히 시험 데이터의 경우 ADBFE 알고리즘은 7 개의 특징을 사용하여 원래의 테스트 데이터로부터 얻을 수 있는 분류 정확도와 유사한 92.5%의 분류 정확도를 얻었다. 반면 DBFE, PCA, CA 알고리즘은 각각 88.2%, 85.7%.

80.8%의 분류 정확도를 나타낸다. 또한 ADBFE 알고리즘과 DBFE 알고리즘만을 비교하면, ADBFE 알고리즘이 보다 우수한 성능을 보인다.

표 3. 클래스 정보.

클래스	Subclass	No. samples	No. total samples
클래스 1	Winter Wheat May 3, 1977	657	1300
	Summer Fallow June 26, 1977	643	
클래스 2	Winter Wheat March 8, 1977	691	1206
	Spring Wheat July 26, 1978	515	
클래스 3	Winter Wheat June 26, 1977	677	1146
	Spring Wheat Sep. 21, 1978	469	
클래스 4	Winter Wheat Oct 18, 1977	660	1101
	Spring Wheat Oct 26, 1978	441	

### V. 결론

본 논문에서는 신경망의 특징추출 알고리즘인 기존의 결정경계 특징추출 알고리즘에서 결정경계의 수직 벡터를 해석적으로 계산하는 방법을 제안하였다. 수직 벡터를 해석적으로 계산함으로써 결정경계 특징추출 알고리즘의 성능을 향상시키고 특징추출 시간을 절반으로 줄일 수 있다. 실험 결과 제안된 해석적 결정경계 특징추출 방법이 결정경계 특징추출 방법에서의 부정확한 수직벡터의 추정 문제를 해결할 수 있음을 확인하였다.

### 참고문헌

[1] S. Haykin, *Neural Networks*, New York: Macmillan, 1994.  
 [2] C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks," *IEEE Trans. Neural Networks*, vol. 8, no. 1, January 1997.  
 [3] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundary," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 388-400, 1993.

[4] L. L. Biehl et. al., "A crops and soils data base for scene radiation research," in *Proc. Machine Processing Remotely Sensed Data Symp.*, W. Lafayette, IN, 1982.

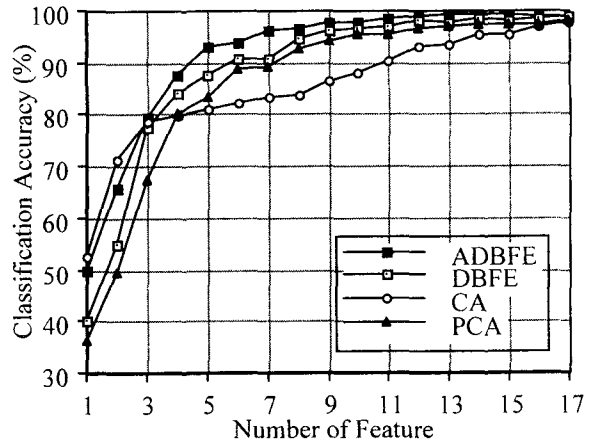


그림 1. 성능 비교 (학습 데이터, PCA: Principal Component Analysis, CA: Canonical Analysis, DBFE: Decision Boundary Feature Extraction, ADBFE: Analytical Decision Boundary Feature Extraction).

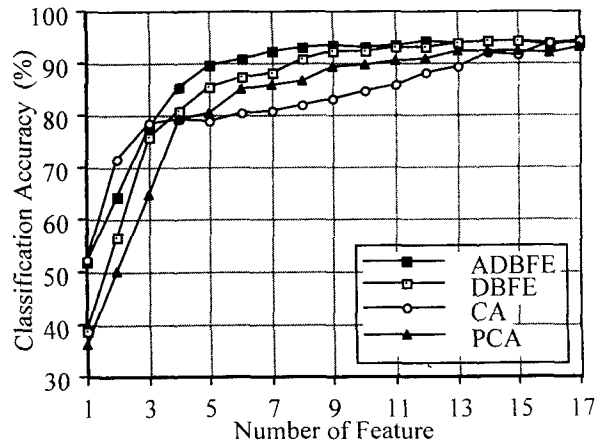


그림 2. 성능 비교 (시험 데이터, PCA: Principal Component Analysis, CA: Canonical Analysis, DBFE: Decision Boundary Feature Extraction, ADBFE: Analytical Decision Boundary Feature Extraction).