

## Neural Network를 기반으로 한 color 분석 시스템

이성근, 김미정, 남정확, 한상일, 황규석\*  
 부산대학교 화학공학과  
 (kshwang@pusan.ac.kr\*)

## Color Analysis System Based on a Neural Network

Sung Gun Lee, Mi Jung Kim, Jung Hwak Nam, Sang il Han, Kyu Suk Hwang\*  
 Department of Chemical Engineering, Pusan National University, Busan, Korea  
 (kshwang@pusan.ac.kr\*)

## 서론

21세기 산업경쟁력의 핵심요소는 성능과 함께 디자인으로 소비자의 감성을 만족시키지 못하면 제품이 팔리지 않는다. 한국 색채 연구소에 따르면 인간의 오감 중 시각부분의 영향은 83~87%수준이고 시각 중 색채가 차지하는 비중이 70%에 이른다. 지금까지 알려진 색만 해도 800만여 가지로 빨간색 하나만도 100만 가지에 이른다. 하지만 이런 다양한 색을 관리해주는 현재의 color matching system은 매우 고가이며, 사용하고 있는 database는 거의 수입에 의존하고 있다. 따라서 이러한 단점을 극복하기 위해 본 논문에서는 Neural network를 이용하여 color의 CIELab를 구함으로써 색을 평가하는 system을 구축하고자 한다.

## 본론

본 연구의 Neural network는 Matlab program으로 구축하였으며 두 개의 neural network로 구성된 MLP(Multi - Layer Perceptron)모델로 구성하였다. color배합비율을 학습시키기 위하여 제작한 panel은 paint를 이용하여 primitive color(red, blue, yellow)를 다양한 비율로 배합하여 제작, 임의로 training pattern과 target pattern으로 나누었다. Neural network를 사용하여 color prediction과 color matching을 하기 위해서는 color를 객관적이고 수치적으로 나타내는 CIE color order systems을 사용한다. 이는 크게 CIEXYZ, CIXLab, CIRLch로 나누어지는데, 본 연구에서는 CIEXYZ, CIXLab를 사용하였다. Multi-layer perceptron은 입력층과 출력층 사이에 하나 이상의 중간층이 존재하는 신경망이다. 각 층의 전달함수는 sigmoid 함수를 사용하였다.

1)CIELab data normalize : sigmoid함수를 전달함수로 채택하였으므로 color data는 0과 1사이의 값을 가져야 한다.

$$f(x) = \frac{x - \text{Min}(x)}{\text{Max}(x) - \text{Min}(x)}$$

$$x = f(x) * (\text{Max}(x) - \text{Min}(x)) + \text{Min}(x)$$

$$f(x) = \frac{1}{1 + \exp(-x)}$$

2)MLP(Multi-Layer Perceptron) model : back-propagation 학습방식을 채택한다.

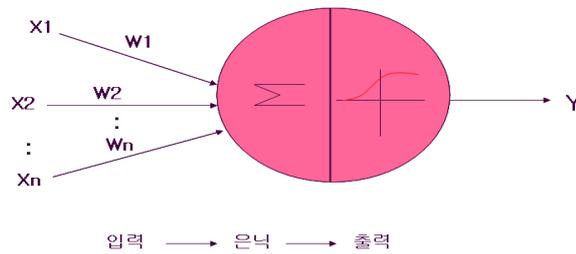


그림 1. neuron model.

Back-propagation(BP) - 역전파 알고리즘은 만일 어떤 신경세포의 활성이 다른 신경세

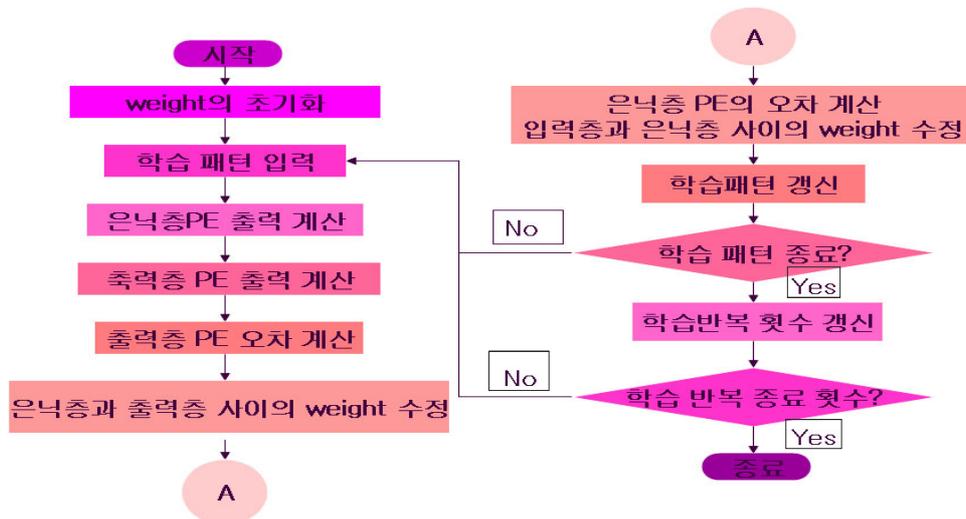


그림 2. back propagation 모식도.

포가 잘못된 출력을 일으켰다면 두 신경간의 연결 가중치가 그것에 비례하여 조절된다. BP는 다층 perceptron에서 노드 사이를 연결하고 있는 가중치들을 층 별로 오차 신호에 따라 조정된다. 먼저 다층 퍼셉트론의 가중치들이 임의의 작은 값에 의해 초기화 된 후, 학습 패턴이 입력되면 전방향 계산에 의해 출력 노드의 값들이 계산된다. 그 후 출력층의 가중치들은 출력 노드의 오차 신호에 따라 변경되는데, 출력 노드의 오차 목표 값과 실제 값의 차이에 sigmoid 함수의 미분값이 곱해져서 역전파 된다. 이처럼 BP는 전방향 계산과 역방향 가중치의 변경의 반복에 의해 multi-layer perceptron을 학습시킨다.

3)Neural network algorithm : i개의 input layer, j개의 hidden layer , k개의 output layer로 이루어진 신경망 알고리즘이다. 정지조건은 n회 이상 학습 횟수로 정하였다. Transfer function은 sigmoid 함수이다.

$$f(x) = \frac{1}{1 + \exp(-x)}$$

step 1 : 재현 가능성을 위해 random으로 weight(연결강도) 초기화.

step 2 : 학습패턴의 수  $\square$  n(학습 횟수) 이상 반복.

**step 3** : 각 학습패턴에 대해 step 4 - 9 반복.

**step 4** : i개의 input layer에서 input signal  $X_i$ 를 받아 hidden layer에 보낸다.

**step 5** : j개의 hidden layer에서 input signal과 weight( $w_{ij}$ )의 곱을 합한다.

$$z\text{-input}_j = \sum_i x_i w_{ij}$$

hidden layer의 output signal을 계산하는 transfer function을 적용

$$z_j = f(z\text{-input}_j)$$

계산된 signal을 output layer로 보낸다.

**step 6** : k개의 output layer에서 weight 조절된 input signal과 합한다.

$$y\text{-input}_k = \sum_j z_j w_{jk}$$

이 output signal을 적용하기 위해 transfer function을 적용.

$$y_k = f(y\text{-input}_k)$$

**back-propagation of error:**

**step 7** : 각 output layer에서 input의 training pattern과 일치하는 target pattern을 받는다.

$$\delta_k = (t_k - y_k) f'(y\text{-input}_k)$$

hidden layer와 output layer사이의 weight 교정항 계산.

$$\Delta w_{jk} = \alpha \delta_k z_j$$

다음 단계로 layer unit에  $\delta_k$ 전송.

**step 8** : 각 hidden layer 는 delta input을 합한다.

$$\delta\text{-input}_j = \sum_k \delta_k w_{jk}$$

error information term을 계산하기 위해 transfer function의 미분값을 곱한다.

$$\delta_j = (\delta\text{-input}_j) f'(z\text{-input}_j)$$

input layer와 hidden layer 사이의 weight 교정항 계산.

$$\Delta w_{ij} = \alpha \delta_j x_i$$

**Update weights:**

step 9 :  $w_{ij} = w_{ij} + \Delta w_{ij}$        $w_{jk} = w_{jk} + \Delta w_{jk}$

step 10 : 정지 조건(학습 횟수 n회 이상)의 만족 여부를 확인한다.

아래의 결과 그래프를 살펴보면 weight는 자동 지정이며, input data는 normalize 시킨 CIELab 값으로 (0.78759 0.56582 0.83645)이고 target 값은(0.56892 0.89526 0.45689)이다. 아래의 왼쪽의 그래프는 학습횟수(epoch)를 20000번 수행 시킨 모습이고 오차가 0에 도달하지 않았지만, epoch를 20000번으로 지정했기 때문에 학습이 정지된 상태의 그래프이다.

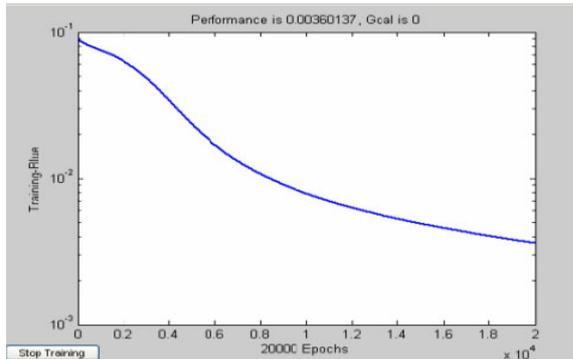


그림 3. Neural network의 학습수행 결과.

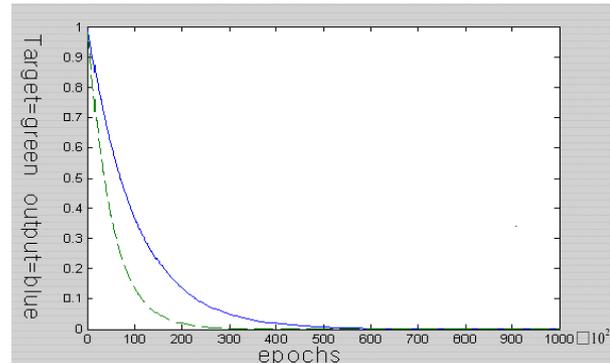


그림 4. output과 target사이 오차 그래프.

오른쪽 그래프는 epochs 100000번의 target과 output의 모습을 그래프이다. epoch가 60000회 정도부터 target과 output이 거의 일치하여 오차가 0에 도달한다.

## 결론

본 연구의 일차적 목표는 color의 CIELab 값을 구하여 색을 평가하는 것이다. 연구를 통한 결론은 아래와 같이 정의할 수 있다.

- 1) Neural network의 모델 식을 제시하고 색과의 관계식을 만들었다.
- 2) Back-propagation 학습 중 오차를 수정함으로써 학습 시간을 감소시켰다.
- 3) color order system중 CIELab를 사용함으로써 우수한 결론을 도출하였다.
- 4) 전달함수를 sigmoid함수를 사용하여 학습 시간을 단축하였다.

## 참고문헌

1. 李相一 人工신경망을 이용한 finished paint의 CIELab-colorimetric matching. 2000.8
2. 崔理相 유전자 알고리즘과 뉴로-퍼지를 이용한 안료 배합비율의 예측 모델링. 1999.2
3. [www.mathworks.com](http://www.mathworks.com) : Matlab toolbox-neural network , neural network toolbox4
4. [www.sciencedirect.com](http://www.sciencedirect.com) : on-line adaptation of neural networks for bioprocess control
5. 金三洙 digital 색상의 원리와 응용. 2002. 2