

# 영상압축을 위한 코넨네트워크

손형경\* · 이영식\* · 배철수\*

\*관동대학교

## KOHONEN NETWORK FOR ADAPTIVE IMAGE COMPRESSION

Hyeong-Kyeong Son\* · Yong-Sik Lee\* · Cheol-Soo Bae\*

\*Kwandong University

E-mail : sonsandy@hotmail.com

### 요 약

본 논문에서는 코호넨 네트워크를 이용한 효과적인 적응 코딩 방법을 제안한다. 신경망을 응용한 압축법 분석을 통해 설명되는 코딩방법은 압축률을 높이기 위해서 우선 영상을 8×8 부영상으로 나누고, 나뉜 모든 부영상은 DCT로 변형한다. 이들 DCT 부블럭들은 코호넨 네트워크로 N(4) 등급으로 나누어지게 되고, 비트들은 DCT 부블럭의 변수에 따라 분류된다. 그래서 N(4)비트 할당 행렬을 얻었다. 실험 결과는 시뮬레이션으로 나타내었고, 제안한 방법이 신경네트워크에서의 AC 에너지에 의해 부영상을 분류하는것보다 우수하다는 결론을 얻을 수 있었다.

### ABSTRACT

In our paper, We propose an efficient adaptive coding method using kohonen neural network. An efficient adaptive encoding method using Kohonen net work is discribed through the analysis of those compression methods with the application of the neural network. In order to increase the compression ratio, a image is first divided into 8\*8 subimages, then all subimages are transformed by DCT. These DCT sub-blocks are divided into N(4) classes by Kohonen network. Bits are distributed according to the variance of the DCT sub-block. Thus we get N(4)bit allocation matrices. Excellent performance is shown by the computer simulation. so we found that our proposed method is better then classifing subimages by AC energy.

### 키워드

Kohonen network, Neural network, DCT, AC energy

### 1. 서 론

최근 신경망은 병렬 분배 성질로 인해 정보처리분야와 자기구성에 폭넓게 이용되고 있고, 영상 데이터 압축 방법에 대해서도 여러 가지 방법을

적용하고있다. BP 네트워크를 이용한 영상 데이터의 압축은 전형적인 방법으로 입력층의 뉴런 단위들의 수는 출력층의 뉴런 단위 수와 같고, 은폐층의 뉴런 단위 수는 입력과 출력층의 뉴런 숫자보다 작다. 영상 데이터는 은폐층에서 압축된다. 하지만, 네트워크에 대해 불충분한 일반화 성

질을 지닌 이 방법은 제한된 영상에 대해서만 실행된다. 그러므로 임의의 영상에 대해서는 압축결과가 좋지 않은데, 주 원인은 은폐층의 출력이 적정한 특성으로 볼수가 없기 때문이다. 사실 모든 부영상에는 DC와 AC 에너지가 존재하고, 영상 스펙트럼은 최상의 특징을 갖는다. 부영상은 높은 DC 에너지와 낮은 AC 에너지를 갖는 것으로 잘 알려져 있다. 그러므로 DCT에 의해 종속되어진 신경망이 압축 영상에 대한 올바른 방법이다. 본 논문에서 하나의 영상은 우선 8×8 부영상의 열로 분할되며, 이때 모든 부영상은 8×8 행렬식인 DCT로 변형된다. 예를 들어 256×256영상이 있다고 하면, 1024(32×32) 부영상과 (32×32) DCT 부블럭을 얻을 수가 있게 된다. (32×32)행렬들은 코호넨 네트워크에 의해 N(4) 등급으로 분할되고, 이때 비트는 모든 등급의 변수에 의해 분류된다. 위에서 기술한 방법은 자기 다른 분류에 대해서 자기 다른 비트 할당 행렬을 주게된다.

## II. 본 론

### DCT 부블럭의 통계 특성

256×256 크기의 영상에서 영상은 8×8 부영상으로 분할되고, 부영상의 수는 (32×32)가 되고 부영상은  $f_{ij}(m,n)$ 으로 표현한다.

여기서  $m,n=0,1,2,\dots,7$ ,  $i,j=0,1,2,\dots,31$ 이다.  $f_{ij}(m,n)$ 의 값은  $(8 \times i + m, 8 \times j + n)$ 에서 영상과 똑같이짐을 쉽게 알 수가 있다. 함수  $f_{ij}$  ( $i,j=0,1,\dots,31$ )의 DCT는 다음과 같이 정의되어진다.

$$F_{i,j}(s,t) = (1/4)c(s)c(t) \sum_{m=0}^7 \sum_{n=0}^7 f_{i,j}(m,n)$$

$$\cos(\text{pie}(2n+1)t/16) \cos(\text{pie}(2m+1)s/16)$$

여기서:  $\text{pie}=3.14156$ ,  $s=0$  일때  $c(s)=1/\sqrt{2}$ ,  $s=1,2,\dots,7$ 일 때  $c(s)=1$ 이 된다.

DCT가 수행된 후, 변형 부블럭  $F_{ij}$ 는 N(4) 등급으로 분할된다.  $g_{k,L}(s,t)$ ,  $t=1,2,\dots,N_k$ 은 k번째 등급을 나타내고,  $N_k$ 는 k번째 등급에 속해 있는 부영상의 수를 나타낸다. 위에서 가정했던 것에 따라 식  $\sum_{k=0}^N N_k = 32 \times 32$ 를 얻을 수 있다. k번째 등급의 영상 픽셀이 평균  $m_k$ 를 갖는 랜덤과정의 샘플이라고 가정하자. 이때 k번째 DCT 부블럭의 예상 값은 다음과 같이 표현된다.

$$E[ g_k(s,t) ] = (1/4)c(s)c(t)m_k \sum_{m=0}^7 \sum_{n=0}^7$$

$$\cos(\text{pie}(2n+1)t/16) \cos(\text{pie}(2m+1)s/16)$$

위 식에서 다음 식을 얻을 수 있다.

$$E[ g_k(0,0) ] = 8m_k$$

$$E[ g_k(s,t) ] = 0; (s,t) \neq (0,0)$$

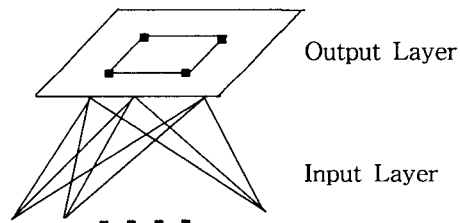
위에서 나타난 평균으로 k번째 DCT 부블럭의 변수는 다음으로부터 계산될 수 있다.

$$\delta_k^z(0,0) = E[ g_k^z(0,0) ] - 64 \times m_k^z$$

$$\delta_k^z(s,t) = E[ g_k^z(s,t) ]; (s,t) \neq (0,0)$$

### 코넨 네트워크

DCT 부블럭은 코넨 네트워크로 분류할 수 있다. 코넨 네트워크는 자기형상화 네트워크(self-organization feature mapping network)이다. 코넨 네트워크의 학습은 같은 입력에 반응하는 활동적인 공간상의 이웃 뉴런 셀과 가장 잘 매칭하는 것을 찾는 것이다. 이런 집단적이고 협력적인 학습은 습득된 네트워크에 관해 몇몇 특징을 대동하게 정의함으로써 네트워크를 변화시킨다. 학습 후에 각 입력들은 입력의 우세특성을 나타내는 뉴런들의 배열위에 위치하는 제한된 응답을 발생한다. 특성 맵핑은 특성들을 나타내는 뉴런 배열위의 입력 패턴 공간의 비선형 투영으로 간주된다. 그래서 각각의 뉴런 공간 위치는 입력의 한 개의 등급을 나타낸다. 두 개의 층을 지닌 코넨 네트워크는 그림 1로 나타낸다. 여기서 출력층은 2차원 배열이다.



$$X_n = \{ x_1 x_2 \dots x_{63} \}$$

그림 1. 코넨네트워크

쉽게 설명하기 위해서  $F_{i,j}$ 는 행에 따라  $X_n$  ( $n=i \times 32 + j$ )이 되도록 쌓이게 된다. 코넨 네트워크의 학습 알고리즘은 다음과 같다: 코넨 네트워크가 63개의 입력 노드와 N(4) 출력 노드를 갖는다고 가정한다. DC 아이템은 단지 부영상의 가벼운 레벨을 나타내기 때문에 분류에 연결되어 있지 않다. 벡터  $W_m$  ( $m=1,2,\dots,N(4)$ )는 임의 N(4) DCT 부블럭에 의해서 초기화된다. 임의 입력  $X_n$ 에서  $X_n$ 과  $W_m$ 간의 거리 계산은

$$d_{nm} = \|X_n - w_m\|, \quad m=1,2,\dots,N(4) \quad \text{이다.}$$

최적 매칭노드 r은

$$\|X_n - W_r\| = \min(\|X_n - W_m\|) \quad \text{으로 선택된}$$

다. 코넨 네트워크의 가중 업데이트 규칙은 다음과 같이 정의된다.

$W_m(t+1) = W_m(t) + a(t)(X_n - W_m(t)), m \in NEr$   
 여기서 NEr은 뉴런 r의 공간상의 이웃뉴런이다. NEr의 반지름은 습득 과정으로 줄어들게 된다. 이 반지름은 가중치의 초기 구형 층에 필요하기 때문에 학습 시작처럼 매우 클 수 있다. 습득의 끝부분에서 이웃 뉴런은 중심부의 우세 뉴런과는 다른 뉴런을 포함하지 않을 것이다. a(t)는  $0 < a(t) < 1$ 인 양수 값 학습함수이다 a(t)는 학습과정처럼 줄어들게 된다.

**비트 분할**

각 등급의 DCT 부블럭에 대한 비트할당 행렬은 그것의 평균 변수에 따라 계산된다. 각 등급의 (s,t) 변수는 다음 공식에 따라 계산된다.

$$\delta_k^2(0,0) = (1/N_k) \sum_{i=0}^{N_k} g_k \cdot \frac{2}{1} (0,0) - 64 \times m_k^2$$

$$\delta_k^2(s,t) = (1/N_k) \sum_{i=0}^{N_k} g_k \cdot \frac{2}{1} (s,t), (s,t) \neq (0,0)$$

각 등급의 변수가 계산되고, 이 등급에 대한 비트 할당 행렬은 결정될 수가 있다. 각 비트 할당 행렬에 대한 비트 할당  $NB_k(s,t)$ 는 윌콕슨 이론에 기반을 둔다. 비트수는 다음과 같다.

$NB_k(s,t) = 0.5 \log_2 [\delta_k^2(s,t)] - \log_2 D$  여기서 D는 파라미터이다. 비트는 초기화 D에 의해 할당 되고, 설정된 비트의 총 수가 될 때까지 반복적으로 계산한다. 원래 이 함수는 Chen 시스템에서 제안되었다. 실험에서 좋은 비트할당을 구하기는 어려웠다. 그리고 압축율은 낮았다. 높은 압축율과 DC 아이템 주변의 이런 요소들에 대한 좀더 많은 비트들을 분할하기 위해서 비선형 처리는 홀뿌려진 영역에서 비트 분할을 막기 위해서 실행된다. 이것을 그림 2에 나타내었다. 여기서 na는 실험하는 동안에 조절할 수 있다.

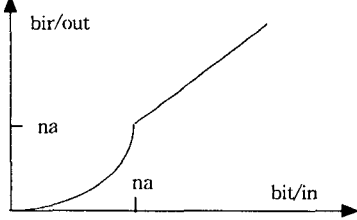


그림 2. 비선형처리

**영상 데이터 압축의 전 과정**

영상 데이터압축의 전 과정은 두 부분으로 나눌 수 있는데 습득 부분과 압축 부분이다. 습득 부분은 다음과 같다: 크기가  $256 \times 256$ 인 영상을 크기가  $8 \times 8$ 인  $32 \times 32$  부영상  $f(m,n)$ 으로 분할한다. 모든 부영상에 대해서 그 자체의 DCT 부블럭을 얻을 수 있다. 그런데 DCT 부블럭의 수는  $32 \times 32$

이다. 행에 따라서 벡터  $X_n, n=0,1,\dots,(32 \times 32 - 1)$ 이 되도록 이들 부영상을 쌓는다. 이 모든 벡터  $X_n$ 을 습득 코넨 네트워크에 이용한다. 자체 구성 과정을 끝낸 후 가중치 벡터  $W_m, m=1,2,\dots,N(4)$ 는 고정된다. 위에서 언급한 처리에 근거해서 비트할당 행렬이 계산된다. 습득 부분은 그림 3에 블록 다이어그램으로 나타내었다. 압축 부분은 다음과 같다: 영상을 크기가  $8 \times 8$ 인  $32 \times 32$  부영상으로 분할한다. 모든 부영상에 대해서 DCT 부블럭을 계산한다. DCT 부블럭은 코넨 네트워크에 의해서 분류되어진다. 등급에 속해있는 것에 따라서 비트들을 분할하기 위해 비트할당 행렬을 이용한다.

본문에 삽입될 그림은 적당한 에디터로 작성하시어 원고 파일에 삽입하여 주시고, 파일로 작성할 수 없는 그림이나 사진이 포함되는 경우에는 저자가 "Camera-ready form"을 만들어 논문 전체를 Hard copy로 제출할 수 있습니다. 이 때에도 논문의 편집양식은 그대로 지켜야 합니다.

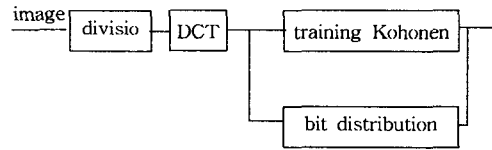


그림 3. 습득 시스템

**실험 결과**

코넨 네트워크는 자동 수집을 이용한다. 부영상이 DCT로 변형된 후 "LENA"의 분류 맵이 그림 5에 있다. 0.5 비트/픽셀(D=3)인 평균 코드비트를 갖는 "LENA"에 대한 비트할당 행렬은 그림 4에 있고 0.5 비트/픽셀인 평균 코드비트를 갖는 복원된 영상은 그림 6에 나타내었다. Chen과 Kong이 실험한 것과 비슷하거나 더 높은 압축율을 갖는 본 논문의 복원 영상은 이들 복원 영상과 비교할 때 높은 선명도를 갖는다. 완전히 다른 성질의 능동 분할 영상을 변형하기 위해서 똑같은 가중치 벡터는 절충안에 이르기 위해서 여러 번 이들 영상을 학습시켜야 한다. 그러므로 가중치 벡터는 좀더 보편성을 갖도록 해야 한다.

85421000	84200000	85422100	84300000
44221000	54201000	44244000	44101000
32210000	52000000	32110000	30000000
00000000	30100000	00100000	00000000
00000000	30101000	00000000	00001000
00000000	20000000	00000000	01000000
00000000	20001000	00000000	00001000
00000000	10000000	00000000	00000000

그림 4. 비트할당행렬

