Simulated Annealing 방법을 이용한 온라인 벡터 양자화기 설계

송 근 배[†]·이 행 세[††]

요 약

벡터 양자화기 설계는 다차원의 목적함수를 최소화하는 학습 알고리즘을 필요로 한다. 일반화된 Lloyd 방법 (GLA)은 벡터 양자화기 설계를 위해 오늘날 가장 널리 사용되는 알고리즘 이다. GLA 는 일괄처리(batch) 방식으로 코드북을 생성하며 목적함수를 단조 감소시키는 강하법(descent algorithm)의 일종이다. 한편 Kohonen 학습법(KLA)은 학습벡터가 입력되는 동안 코드북이 갱신되는 온라인 벡터 양자화기 설계 알고리즘 이다. KLA는 원래 신경망 학습을 위해 Kohonen에 의해 제안되었다. KLA 역시 GLA와 마찬가지로 강하법의 일종이라 할 수 있다. 따라서 이들 두 알고리즘은, 비록 사용하기 편리하고 안정적으로 동작을 하지만, 극소(local minimum) 점으로 수렴하는 문제를 안고있다. 우리는 이 문제와 관련하여 simulated annealing(SA)방법의 응용을 논하고자 한다. SA는 현재까지 극소에 빠지지 않고 최소(global minimum)로 수렴하면서, 해의 수렴이 (통계적으로) 보장되는 유일한 방법이라 할 수 있다. 우리는 먼저 GLA에 SA를 응용한 그 동안의 연구를 개괄한다. 다음으로 온라인 방식의 벡터 양자화기 설계에 SA 방법을 응용함으로써 SA 방법에 기초한 새로운 온라인 학습 알고리즘을 제안한다. 우리는 이 알고리즘을 OLVQ-SA 알고리즘이라 부르기로 한다. 가우스-마코프 소스와 음성데이터에 대한 벡터양자화 실험 결과 제안된 방법이 KLA 보다 일관되게 우수한 코드북을 생성함을 보인다.

On-line Vector Quantizer Design Using Simulated Annealing Method

Geun-Bae Song[†] · Haing-Sei Lee[††]

ABSTRACT

Vector quantizer (VQ) design needs an algorithm to minimize a multidimensional objective function. The generalized Lloyd algorithm (GLA) is one of the most well-known such algorithms today. The GLA is a kind of descent algorithm which decreases monotonically an objective function and generates codebooks in a batch processing mode. The Kohonen learning algorithm (KLA) is an on-line VQ design algorithm, where the codebook is designed (or updated) while training data arrives. The KLA was proposed originally by Kohonen as a learning algorithm for neural networks. The KLA can be also considered as a kind of descent algorithm just as the GLA. Therefore, these two algorithms, although convenient to use, get entrapped into local minima for complex high-dimensional VQ design problems. To solve this entrapment issue we discuss the simulated annealing (SA) method which searches for a solution in a non-deterministic manner. The SA is the only method which is statistically guaranteed to yield globally optimal performance. We first review the previous work on the general formulation of the SA applied to batch processing VQ design, and then propose a new design algorithm based on the SA for on-line VQ design. We call this new algorithm OLVQ-SA. Experimental results for Gauss-Markov sources and real speech demonstrate that the proposed algorithm can consistently provide better codebooks than the KLA.

키워드 : 일반화된 Lloyd 방법(generalized Lloyd algorithm), Kohonen 학습법(Kohonen learning algorithm), simulated annealing

1. 서 론

벡터 양자화기는 신호의 압축, 코딩, 패턴인식 등에 널리 쓰인다. 이는 다수의 입력벡터들을 유한개의 벡터(코드벡터)들로 사상시킨다. 이러한 유한 개의 코드벡터들의 집합을 코드북이라 부른다. 벡터 양자화기를 설계함에 있어서의 목표는 평균 양자화 오차를 최소화하는 코드북을 생성하는 것이다. 그러므로 벡터 양자화기 설계의 문제는 다차원의 복잡한 목적함수(평균 양자화 오차 함수)의 최적화 문제라 할 수 있다. 벡터 양자화기 설계에 있어서 현재 가장 널리 쓰이는 알고리즘은 일반화된 Lloy 방법(GLA) 이다[1]. GLA는 목적함수를 단조 감소시키는 쪽으로 코드북을 반복적으로 갱신시켜나가는 알고리즘이다. 만약 오차함수가 convex 형태일 경우 이러한 학습법은 좋은 결과를 가져온다. 그러나 일반

† 정 회 원 : 아주대학교 대학원 전자공학과
†† 정 회 원 : 아주대학교 교수
논문접수 : 2000년 11월 13일, 심사완료 : 2001년 4월 26일

적으로 양자화 문제는 nonconvex 문제임으로 GLA는 자주 최적이 아닌 코드북을 생성하게 된다.

Self-organizing map(SOM) 알고리즘은 신경망 학습을 위해 Kohonen에 의해서 제안된 일종의 클러스터링 알고리즘이다[2]. SOM은 또한 벡터 양자화기 설계를 위해서도 사용되었다[3]. GLA는 모든 학습벡터가 입력된 뒤에 코드북이 수정되는 일괄처리(batch) 알고리즘인데 반해 Kohonen의 학습 알고리즘(KLA라 부르기로 한다.)은 학습벡터가 입력될 때마다 코드북을 수정해나가는 온라인 알고리즘 이다. 또한 KLA는 일종의 nonconvex 형태의 목적함수에 적용된 least mean squares(LMS) 알고리즘 [4]이다. 결론적으로 KLA는 온라인 방식으로 GLA는 일괄처리 방식으로 동작하는 gradient descent 알고리즘(GD) [4]이라 할 수 있다. 이에 대한 간단한 유도와 수렴 특성에 대한 논의는 각각 [2]와 [5]를 참조할 수 있다. 수렴 특성에 있어서는 GLA 보다 KLA 혹은 KLA형태의 알고리즘이 약간 더 우수한 코드북을 생성하는 것으로 알려져 있다 [3,6]. 그러나 이는 약간의 개선이며 GLA와 KLA 모두 복잡한 양자화 문제에 대해 최소(global minimum)점이 아닌 극소(local minimum)점으로 수렴하는 문제를 안고 있다.

최적화 문제와 관련하여 simulated annealing (SA) [7-9] 방법은 현재까지 극소에 빠지지 않고 최소로 수렴하면서, 해의 수렴이 (통계적으로) 보장되는 유일한 방법이라 할 수 있다. 최적화 문제에 온도와 SA 개념을 도입하는데 선구적인 역할을 한 사람은 Cerny [8]와 Kirkpatrick *et al.* [9]이다. Kirkpatrick은 VLSI설계의 문제를 가상 물리 계의 최소 에너지 상태 탐색의 문제로 환원하였으며 각 온도(순차적으로 감소하는)에서 정상 상태에 도달하도록 하기 위한 방법으로서 Metropolis 알고리즘 [10]을 사용하였다. 이 경우의 정상 상태란 시스템의 가능한 상태가 *Gibbs*(혹은 *Boltzmann*) 분포를 나타내는 상태를 말하는데, 열역학 분야에서는 이 상태에 도달한 물리 계에 대해 그 온도에서 '열평형 상태'에 도달하였다고 말한다. 곧이어 Geman *et al.* [11]은 *Gibbs sampler*라는 것을 정의하였는데, 이는 병렬적으로 상태 변환을 해나가는 것이 가능한 알고리즘으로서 순차적 상태변환 알고리즘인 Metropolis 알고리즘에 비해 빨리 열평형 상태에 도달할 수 있다는 장점을 가진다. Geman은 또한 [11]에서, 까다로운 조건 없이 그리고 초기 상태에 상관없이, 주어진 온도에서 시스템이 정상상태로 수렴하는 것을 Gibbs sampler가 보장함을 증명하였다.

벡터 양자화 문제는 복잡한 목적 함수의 최적화 문제이다. 따라서 최소점 탐색과 관련하여 기존의 알고리즘 대신 혹은 그들의 보완으로서 SA 개념의 도입을 생각해 볼 수 있다. 우리는 먼저 벡터 양자화기 설계에 SA개념의 도입과 관련한 그 동안의 연구들을 개괄한다. 이와 관련한 논문으로는 [12-16]을 들 수 있다. 특히 Zeger *et al.*는 [16]에서 벡터 양자화기 설계에 SA 개념의 응용과 관련하여 다양한 방법의 종합과 정형화를 시도하고 있다. 이러한 논의들은 모두 일괄처리 알고리즘(GLA)과 관련한 논의들이다. 이에 대한 보다 자세한 사항들은 3절에서 언급하기로 한다. 우리는 그 다음으로 이러한 검토의 바탕 위에 '온라인 방식의 벡터 양자화기 설계에 있어서 SA개념'의 도입을 시도한다. 그리하여 '새로운 온라인 벡터 양자화기 설계 알고리즘'(OLVQ-SA로 부르기로 한다.)을 정의한다. 이 방법은 충분한 학습 시간이 주어진다면 초기값에 상관없이 최소의 양자화 오차를 가지는 코드북을 생성할 것이다.

2. 벡터 양자화의 기본 개념과 알고리즘들

2.1 벡터 양자화의 기본 개념

임의의 입력벡터를 $x(\in R^N)$, 임의의 코드 벡터를 $W_i = [w_{i1}, w_{i2}, \cdots, w_{iN}]^T \in R^N$라 하자. 여기서 아래 첨자 i는 $i \in \{1, \cdots, M\}$으로 M은 코드북의 크기를 나타낸다. 벡터 양자화기는 R^N Euclid 공간의 임의의 벡터 x를 코드북 내의 임의의 코드벡터 W_i로 사상시키는 함수를 말한다. 이때 할당되는 코드벡터는 Euclidean 거리 측도 d의 관점에서 입력 벡터 x에 가장 가까운 코드벡터 W_c로 선정된다. 여기서 c는 식 (1)에 의해 결정된다.

$$c = \arg\min_i d(x, W_i) \tag{1}$$

각 입력 벡터를 각각의 대응하는 코드벡터로 근사 시킴에 따라 오차가 생겨나게 되는데 이것을 양자화 오차라고 부른다. 그리고 주어진 입력 벡터 전체에 대한 오차의 평균을 생각할 수 있는데, 이를 평균 양자화 오차라고 부른다. x의 확률밀도 함수를 $p(x)$라 하면 평균 양자화 오차함수는 다음과 같이 정의된다.

$$\hat{E} = \int [d(x, W_c)]^2 p(x) dx \tag{2}$$

식 (2)에서 현실적으로 입력벡터 x의 확률밀도함수 $p(\cdot)$를 사전에 알기 어렵다. 따라서 $p(\cdot)$를 근사적으로 구하거나 혹은 실험으로 추출된 충분한 수의 샘플벡터들로 대신하여 미지의 $p(\cdot)$에 대한 양자화기를 설계하게 된다. 후자의 경우 식 (2)의 오차함수는 식 (3)의 형태로 수정되며 이는 식 (2)의 샘플평균을 의미한다 할 수 있다.

$$\hat{\hat{E}} = \frac{1}{L} \sum_x [d(x, W_c)]^2 \tag{3}$$

여기서 L은 총 학습벡터의 수를 나타낸다. 벡터 양자화 문제는 바로 이 오차함수 $\hat{E}$ (혹은 $\hat{\hat{E}}$)을 최소화하는 유한개의 코드벡터들의 집합, 즉 코드북 $W = \{W_i\}_{i=1}^M$을 찾는 문제이다. 그러나 식 (2)나 식 (3)의 최적 해를 닫힌 형태로

풀어내기는 어렵다. 따라서 현실적으로 GD 방법으로 이 문제를 우회한다. GLA와 KLA는 각각 일괄처리 방식과 온라인 방식의 GD라 할 수 있다. GD는 반복과정을 통하여 목적함수를 단조 감소시키며 극소점에로 수렴을 보장한다. Nearest neighbor (NN) 조건과 centroid 조건은 최적의 양자화기가 만족시켜야하는 잘 알려진 두 가지 필요조건이다 [1]. GD는 반복과정을 통하여 이 두 조건을 만족하는 극소점에로 수렴해 가게 된다.

2.2 일반화된 Lloyd 방법

Linde *et al.* [1]은 스칼라 양자화기 설계 알고리즘인 Lloyd 알고리즘 [17]을 벡터 양자화기 설계 알고리즘으로 확장하였다. 이 알고리즘은 일반화된 Lloyd 방법(간단히 GLA 혹은 LBG)라 불리어 진다. GLA는 일괄처리 방식의 알고리즘이다. 즉 매회 주어진 학습벡터가 모두 입력된 뒤에 학습벡터 전체를 사용하여 코드북을 갱신시킨다. 구체적인 알고리즘은 (그림 1)과 같다. 여기서 m은 학습횟수를, $W_i^{(m)}$은 m회 학습에서의 i번째 코드벡터를, $R_c^{(m)}$는 c번째의 분할 영역(혹은 클러스터)을, 그리고 $|R_c^{(m)}|$은 그 영역에 속한 벡터들의 수를 나타낸다. 그리고 $\hat{\hat{E}}_m$은 양자화 오차를 나타낸다.

1) 초기화 :

$$W_1^{(1)}, \cdots, W_M^{(1)}, \quad m=1, \quad \hat{\hat{E}}_0=\infty.$$

2) NN Repartition : $(1 \le j \le L)$

$$c = \arg\min_i d(x_j, W_i^{(m)}),$$

$$Put \ x_j \rightarrow R_c^{(m)},$$

$$\hat{\hat{E}}_m = \hat{\hat{E}}_m + d(x_j, W_c^{(m)})^2.$$

3) Stopping Criterion :

$$If \ \frac{(\hat{\hat{E}}_{m-1} - \hat{\hat{E}}_m)}{\hat{\hat{E}}_m} < \varepsilon, \quad Stop.$$

$$m = m+1.$$

4) Centroid 계산 : $(1 \le i \le M)$

$$W_i^{(m)} = \frac{1}{|R_i^{(m)}|} \sum_{x_j \in R_i} x_j$$

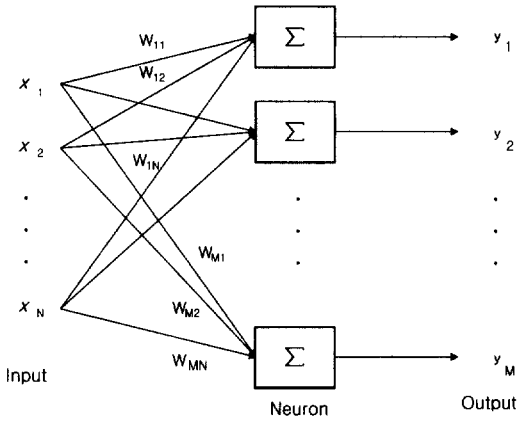
5) Goto 2)

(그림 1) 일괄처리 코드북 설계를 위한 GLA

2.3 Kohonen 학습법

Kohonen 학습법(KLA)은 신경망의 자기 조직화 학습을 위해 Kohonen에 의해서 제안되었다[2]. 보통 Self-organizing map(SOM) 알고리즘이라 하면 복수 승자 방식의 KLA를 일컫는다. 즉 SOM은 승자 출력 뉴런 뿐 아니라 그 뉴런으로부터 공간적으로 일정 거리 안에 드는 이웃뉴런에게까지 학습의 기회를 준다. 이렇게 함으로써 학습과정을 통해 인접한 뉴런들이 서로 닮게 되는 결과를 가져오는데, 이는 고차원 입력벡터의 복잡하고 비선형적인 관계를 저차원의 단순하고 기하학적인 관계로 보존하여 표현하여 줌으로써 고차원 데이터를 시각화(visualization)하고 정보를 함축(abstract)하는 효과를 가져올 수 있다[2].

SOM은 또한 벡터 양자화기 설계를 위해서도 사용되었다[3]. (그림 1)은 벡터 양자화기로 이용될 수 있는 단층 신경망 구조를 나타낸 것이다. 그림에서 출력 뉴런 $y_1, y_2, \cdots, y_M$은 코드인덱스에, 가중치 벡터 $W_i = [w_{i1}, w_{i2}, \cdots, w_{iN}]^T$는 코드벡터에 각각 대응된다. 벡터 양자화의 관점에서 보면 복수승자 방식은 하나의 휴리스틱이 될 수 있다. 일반적으로 입력벡터는 벡터 공간에서 특정 지역에 편중되어 분포되어 있다. 따라서 균일 분포의 랜덤 값으로 코드벡터들을 초기화시키는 것은 그리 좋은 방법이 아니다. 학습 초기에 승자뿐만 아니라 넓은 범위의 이웃 뉴런들에게까지 학습기회를 주는 것은 전체 코드 벡터들을 입력 벡터의 분포 지역으로 이동시켜줄 것이다. 그 뒤 학습이 진행됨에 따라 이웃뉴런의 범위를 줄여나가면 각 뉴런에 의해 세밀한 클러스터링을 할 수 있을 것이다. 그러나 이러한 전략의 효과는 불분명하며 일관되지 않은 결과를 가져오는 것으로 알려져 있다[18]. 이 문제와 관련한 다른 휴리스틱들은 [19, 20]을 참조할 수 있다. 본 논문에서는 문제를 간단히 하기 위해 단일 승자 방식의 SOM 알고리즘에 대해서만 논의하기로 하며 이를 간단히 'KLA'라 부르기로 한다.



(그림 2) 단층 신경망

KLA는 온라인 벡터 양자화기 설계 알고리즘이다. KLA는 GLA와 달리 연속적인 데이터의 흐름 속에서 순간 순간 코드북을 갱신시켜 나간다. 따라서 데이터의 통계가 변해갈 경우 이를 실시간으로 코드북 설계에 반영할 수 있다는 장점이 있다. KLA는 원칙적으로 전체 입력벡터들이 사전에 주어지지 않는 온라인 환경에서 동작하므로 이를 반영하기 위해 입력벡터 및 오차함수의 표기에 시간변수 n를 도입한다. 즉,

$x=x(n)$. 여기서 n는 이산 시간을 나타내며, $x(n)$는 시간 n에 따라 변화하는 벡터 랜덤 프로세스이다. 그리고 오차 함수는 일괄처리 방식의 식 (2)나 식 (3)이 아닌 '순시오차 함수' 식 (4)를 사용하며 이 기준에 따라 순간 순간의 코드북 갱신이 이루어진다[2].

$$E(n)=[d(x(n), W_c(n))]^2 \tag{4}$$

여기서 $W_c(n)$는 시간 n 에서 입력벡터 $x(n)$에 가장 가까운 코드벡터를 나타낸다. n시점에서의 승자코드벡터 $W_c(n)$의 갱신은 KLA에 의해 다음과 같이 정의된다.

$$\begin{aligned} W_c(n+1) &= W_c(n)+\Delta W_c(n) \\ &= W_c(n)+\alpha(n)[x(n)-W_c(n)], \\ &\quad (0<\alpha(n)<1) \end{aligned} \tag{5}$$

$$W_i(n+1)=W_i(n), \quad \text{그 밖의 } i \neq c. \tag{6}$$

여기서 $\alpha(n)$는 학습률(혹은 스텝사이즈)을 나타내며 0과 1 사이에서 시간에 따라 단조 감소하는 임의의 함수로 정의한다.

3. SA를 이용한 일괄처리 벡터 양자화

3.1 Simulated Annealing

SA [7-9]는 최적화 문제를 금속의 담금질 공정과 관련지은 기술이다. 최적화 문제에 적용된 SA방법의 본질은 평형 특성에 있다. 먼저, 최적화 하고자 하는 시스템의 상태벡터(보통 큰 차원의 벡터)를 W라하고, 그 상태에 대해 에너지 함수(혹은 목적함수) $E(W)$가 정의되어 있다고 하자. 또한 정상상태로의 수렴을 보장하는 연산자(가령 *Metropolis 알고리즘*)가 정의되어 있어 이를 사용하여 시스템이 상태천이를 해 나간다고 하자. 이러한 동작을 하는 시스템은 연산자에 의해 결정되는 상태천이 확률을 가지는 일차 Markov chain 모델이라 할 수 있다[11]. 이러한 전개는 충분한 상태천이 후에 결국 정상상태(즉 열 평형상태)에 도달하게 될 것이다. 이때 그 시스템의 정상상태분포가 다음의 Gibbs 분포를 띤다고 가정하자.

$$P(W)=\frac{1}{Z}e^{-\frac{E(W)}{T}} \tag{7}$$

여기서 Z는 $\sum_W P(W)=1$이 되도록 정한 정규화 상수이다. $T(>0)$는 열역학적 비유에 의해 온도라 불리어지는 변수이다. T를 고정시킨 상태에서 식 (7)을 관찰하여 보면 이 함수는 $E(W)$에 따라 지수 함수적으로 감소하는 함수이다. 즉 어떤 시스템이 식 (7)과 같은 정상상태분포를 나타낸다고 할 때 그 시스템이 어느 특정 상태 W에 있을 확률은 그 상태의 에너지 $E(W)$와 관련되며 낮은 에너지 상태로 존재할 확률이 높은 에너지 상태 보다 높다는 것을 알 수 있다. 이제 온도를 변화시켜 보면 높은 온도 즉, $T\to\infty$일 때에는 $\frac{1}{T}E(W)\to 0$ 임으로 식 (7)은 균일 분포의 형태를 닮아간다. 반대로 $T\to 0$일 때는 식 (7)은 낮은 에너지 쪽에 크기가 집중된 임펄스 함수의 형태를 띠게 된다. 따라서 식 (7)과 같은 정상상태 시스템은 높은 온도에서 평형상태에 도달할 경우 에너지에 무관하게 모든 상태의 존재 확률이 균등해 지며 반면, 낮은 온도에서는 시스템이 낮은 에너지 상태에 존재할 확률이 급격히 증가하게 된다.

모든 최적화 문제는 상태벡터 W와 $E(\cdot)$에 의해 정형화 될 수 있다. 그리고 이때 $E(W)$를 최소로 하는 W를 찾는 것이 탐색의 목적이 된다. 문제의 정형화가 이루어졌을 경우 Gibbs 분포로의 수렴을 보장하는 연산자에 의해 상태천이를 해나간다면 시스템은 열 평형상태에 도달하게 될 것이고 이때 일반적으로 시스템이 낮은 에너지 상태에 있을 확률이 높은 에너지 상태에 있을 확률보다 크게 된다.

Metropolis 알고리즘 [10]은 열 평형상태의 분포가 식 (7)과 같이 되는 것을 보장하는 상태천이 연산자이다. 이 알고리즘을 간단히 설명하면 다음과 같다. 시스템의 현재상태 $W(n)$가 주어져있을 경우 새로운 상태 η를 $\eta=\pi(W(n))$의 규칙에 따라 결정한다. 여기서 π는 $W(n)$으로부터 어떤 통계적 규칙에 따라 새로운 상태 η를 생성하는 함수로 'perturbation 함수'라 부른다. 다음으로 에너지 변화 $\Delta E=E(\eta)-E(x(n))$와 식 (8)의 양을 계산한다.

$$q=\frac{P(\eta)}{P(x(n))}=e^{-\Delta E} \tag{8}$$

여기서 $P(\cdot)$는 식 (7)에 정의된 바, Gibbs 분포를 나타낸다. 만약 $q>1$ 이면, 새로운 상태 η로 천이 한다. 즉 $W(n+1)=\eta$. 반대로 만약 $q\le 1$ 이면, 상태천이는 q의 확률로 일어난다. 0과 1 사이에서 균일 분포의 난수 γ를 발생시켜 만약 $\gamma\le q$ 이면 $W(n+1)=\eta$로 $\gamma>q$ 이면 $W(n+1)=W(n)$로 설정한다.

최저 에너지 상태의 시스템을 얻기 위한 직접적인 방법은 0에 가까운 낮은 온도에서 시스템이 열 평형에 도달하도록 시도하는 하는 것이다. 그러나 주어진 초기 상태에서 열 평형 상태에 도달하기 위해 필요한 상태천이의 수는 낮은 온도일수록 기하급수적으로 증가한다[7]. 따라서 낮은 온도에서 열 평형 상태에 도달하기 위한 가장 빠른 방법은 열 평형에 빨리 도달할 수 있는 높은 온도에서 시작하여 서서히 온도를 낮추어 나가는 것이다. 각 온도에서는 충분한 상태천이 연산으로 열 평형에 도달된 뒤에 다음 온도로 온도를 낮추어야 한다. 각 온도에서 열 평형에 도달하기 위해 시도하는 상태천이의 수는 온도 강하 폭에 의해 좌우된다. 만약 너무 빨리 온도를 낮춘다면 평형을 회복하기 위해 그만큼 더 많은 시간이 필요하다. 온도를 낮추어 가는 계획표를 온

도(혹은 냉각) 스케줄이라 한다. Geman [11]은 최적 점으로의 수렴을 보장하기 위해 온도 T를 시간 n(상태천이 시간)에 따라 식 (9)와 같이 내리면 되는 것을 제시하였다.

$$T(n) = \frac{C}{\log n} \tag{9}$$

여기서 C는 문제에 따라 결정되는 상수이다. 현실적으로 이러한 스케줄은 매우 느리다. 그래서 Kirkpatrick [9]은 식 (10)과 같은 준최적의 스케줄을 추천하였다.

$$T(n) = T_0 K^n \tag{10}$$

여기서 T_0는 초기온도를 나타내며 K는 0과 1 사이의 상수(가령 0.9) 이다. 일반적으로 지수 함수적으로 감소하는 온도 스케줄을 사용한다. 그러나 구체적인 모양은 응용에 따라 실험적으로 결정한다.

3.2 SA를 이용한 일괄처리 벡터 양자화기 설계

SA방법을 이용한 일괄처리 방식의 벡터 양자화기 설계에 관하여 그 동안 많은 앞선 연구들이 있어 왔다[12-16]. 특히 Zeger는 [16]에서 이 주제와 관련하여 그간의 연구들을 종합하고 정형화하였다. [16]에서 정리된 바, SA방법을 사용한 벡터 양자화기 설계 방법을 간략히 복습하면 다음과 같다.

우선 벡터 양자화기의 기호를 정리하면, 코드북을 W로 나타낸다. 즉, $W=\{W_i\}_{i=1}^{M}$가 된다. 여기서 W_i는 개개의 코드벡터를, M은 코드북 크기를 나타낸다. 그리고 목적함수 식 (2)의 오른쪽 식의 코드벡터 W_c의 c는 식 (1)에서의 정의에 의해 임의의 입력벡터 x와 코드북 W의 함수이다. 따라서 식 (2)의 $\hat{E}$는 $\hat{E}=\hat{E}(W)$와 같이 W의 함수로 나타낼 수 있다. 벡터 양자화기 설계의 문제는 $\hat{E}(W)$를 최소로 하는 코드북 W를 찾는 문제다. 이를 SA와 관련지어보면, 코드북 W를 상태벡터에, 목적함수 $\hat{E}(\cdot)$을 에너지 함수에 대응시킬 수 있다. 그리하여 벡터 양자화 문제에 SA방법을 응용하기 위한 준비가 끝난다. 이제 3.1절에서 설명한 Metropolis 알고리즘을 사용하여, 높은 온도에서 낮은 온도로 천천히 열 평형을 유지하며 $W(n)\rightarrow W(n+1)$의 상태천이를 해나간다면 결국 최소 에너지 상태 즉, 우리가 찾는 해 $W_{opt}=\arg\min_W \hat{E}(W)$에 도달하게 될 것이다. 상태천이를 하는 과정에서 에너지 함수 식 (2)(혹은 식 (3))를 계산하게 된다. 이는 전체 학습벡터를 대상으로 하는 식이다. 따라서 이 알고리즘은 일괄처리 알고리즘이 된다.

여기서 사용되는 perturbation 함수 π는 다음과 같이 정의된다.

$$\pi(W(n)) = W(n) + \xi(T(n)) \tag{11}$$

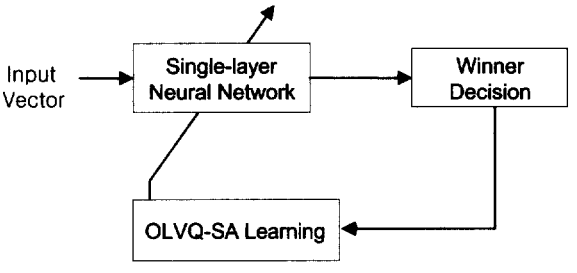
여기서 ξ는 랜덤벡터(가령 Gaussian)로서 분산이 온도 T에 의해 결정된다. Zeger는 [16]에서 잡음 ξ의 분산을 조절하는 온도스케줄로서 식 (10)과 식 (12) 등을 사용하였다.

$$T(n)=\sigma_W^2\,(1-\frac{n}{I})^3 \tag{12}$$

여기서 σ_W^2은 코드벡터 성분들의 분산을, I는 미리 정해진 총 학습회수를 나타낸다. 따라서 식 (12)의 경우에 초기온도 $T(0)$는 초기 코드벡터 성분들의 분산 σ_W^2이 된다. 그 밖의 다른 온도스케줄링 방법에 대해선 [16, 21]등을 참조할 수 있다.

4. SA를 이용한 온라인 벡터 양자화

이 절에서는 'SA를 이용한 온라인 벡터 양자화기 설계 방법(OLVQ-SA)'을 제안하고자한다. 먼저 SA를 응용하기 위해서는 최적화 하고자하는 시스템(양자화기)의 상태벡터를 정의해야 한다. 3.2절에서 언급된 바와 같이 전체 코드벡터들의 집합, 즉 코드북 W로 시스템 상태벡터를 정의하기로 한다.



(그림 3) OLVQ-SA 벡터 양자화기 설계 블럭도

다음으로 시스템 상태에 대한 에너지함수(즉 목적함수)를 정의해야한다. OLVQ-SA는 KLA와 마찬가지로 순시오차함수 식 (4)를 에너지함수로 정의하기로 한다. 이와 같이 하면 n시점에서의 에너지함수 $E(n)$은 n시점의 상태벡터 $W(n)$와 더불어 다음과 같이 나타내질 수 있다.

$$E(W(n))=[d(x(n), W_c(n))]^2 \tag{13}$$

식 (13)의 오른쪽 식의 코드벡터 $W_c(n)$의 c는 식 (1)에서 정의된 바와 같이, n시점에서의 입력벡터 $x(n)$과 그 시점에서의 모든 코드벡터, 즉 코드북 $W(n)$의 함수이다. 따라서 순시오차함수를 식 (4)의 $E(n)$ 대신 식 (13)처럼 $E(W(n))$으로 표기할 수 있다.

이상과 같이 상태벡터와 에너지함수가 정의가 일단락 되면 SA방법의 응용은 직접적이고 간단하다. 이에 대한 간단한 절차는 (그림 4)와 같다.

1) 초기화 :

$n=1,\ T=T_0\ ,\ W_1^{(1)},\cdots,W_M^{(1)}.$

2) Nearest Neighbor 탐색 :

$c=\arg\min_i d(x(n),\ W_i(n))$

$E_1=d(x(n),\ W_c(n))\ .$

3) Centroid Perturbation :

$W_{c\,new}=W_c(n)+\alpha\, Koh+\xi(T(t))$

$E_2=d(x(n),\ W_{c\,new})$

4) 에너지 증분 ΔE 계산 :

$\Delta E=E_2-E_1\ .$

5) Acceptance 결정 :

If $q=\exp[-\frac{\Delta E}{T}]>1$,

$W_c(n+1)=W_{c\,new}.$

Else

Gernerate *uniform RN* $\gamma\in(0,1).$

If $\gamma\le q$

$W_c(n+1)=W_{c\,new}$

Else

$W_c(n+1)=W_c(t).$

6) 열 평형 :

If Thermal Equilibrium, *Goto* 6).

Else

$n=n+1,$

Goto ②.

7) 온도 냉각 :

If $T(t)<$ *End Temperature, Stop*

Else

$T=T(t+1),$

Goto ②.

(그림 4) 온라인 코드북 설계를 위한 OLVQ-SA 방법

(그림 4)에서 단계 3의 *Koh*는 식 (5)에 정의되어 있는 Kohonen 학습량을 뜻한다. SA를 온라인 양자화기 설계에 응용함에 있어 이와 같은 KLA와의 결합은 필요조건이 아니다. 그러나 KLA와 결합된 학습법은 수렴속도($O(10^1)$ 이상)나 수렴의 안정도 면에서 상당한 개선을 가져다준다. 따라서 기본적 선택사항으로 볼 수 있다. 온도 $T(t)$는 단계 3)의 Gaussian 잡음 $\xi(\cdot)$의 분산을 결정하는 역할과 단계 5에서 제안된 perturbation의 채택 여부를 결정하는 변수로 작용한다. 높은 온도는 잡음의 영향을 크게 하고 목적함수를 나빠지게 하는 상태천이가 채택될 확률을 높인다. 온도가 낮아짐에 따라 잡음의 영향은 줄어들며 목적함수를 개선시키는 천이만 받아들이는 경향을 띄게 된다. 이러한 과정으로 SA 방법은 점차로 KLA에 근사해 간다. 온도는 각각의 온도에서 시스템이 열 평형에 도달할 때마다 서서히 감소하는데 이때의 스케줄에 따라 수렴속도나 성능이 영향을 받게된다. 사용되는 온도스케줄 $T(n)$은 실험적으로 결정된다. 본 연구에서 사용한 곡선은 식 (10)과 같다.

시스템이 열 평형에 도달하였는가 하는 것을 판단하는 것은 어려운 문제다. Kirkpatrick [9]은 주어진 온도에서 상태천이가 (시도된 회수가 아닌)받아들여진 회수로 이를 추정하였다. 대략 이 지침에 따라, $200\times M$(여기서 M은 코드북 크기) 정도의 상태 천이가 일어난 시점을 그 온도에서 열 평형에 도달한 시점으로 보고 온도를 낮추었다.

5. 실험 및 결과분석

제안된 방법 즉, OLVQ-SA와 KLA의 성능을 비교하기 위해 두 가지 실험을 하였다. 첫째는 가우스-마코프 소스에 대한 양자화 실험이다. 이는 벡터 양자화기 성능실험에 흔히 사용되는 소스로 실제 데이터(음성이나 화상)에 대한 간단한 확률적 모델로서 사용될 수 있다. 둘째로는 실험실에서 채집된 실제 음성데이터에 대한 양자화실험이다.

5.1 가우스-마코프 소스에 대한 양자화 문제

본 실험은 상관계수 ρ가 각각 0, 0.5, 0.9인 세 가지 경우의 1차 가우스-마코프 소스에 대해 조사하였다. 이들 소스는 식 (14)에 의해 생성된다.

$$x_{i+1}=\rho x_i+w_i \tag{14}$$

여기서 w_i는 평균 0, 분산 1인 i.i.d. Gaussian 잡음이다. 이렇게 생성된 x_i는 평균과 분산이 각각 0과 $1/(1-\rho^2)$인 Gaussian 소스가 된다. 각 상관계수에 대해 총 16384(2^{14}) 샘플을 채집하여 벡터를 구성한 뒤(2차원일 경우 9182개 4차원일 경우 4096개) 순차적으로 인가하며 학습에 사용하였다. KLA의 최대 epoch 수는 실험 결과를 토대로 200회로 설정하였다. 그리고 제안된 방법과 형평을 기하기 위해 KLA의 점수는 초기조건을 바꿔가며 30회 재시행을 실시하여 최고점으로 기록하였다(GLA에 대해서도 마찬가지이다.). 이는 OLVQ-SA의 계산량이 KLA의 대략 30배정도 되도록 설정한데 따른 회수다.

각 소스에 대해 벡터의 차원과 코드북의 크기를 변화시켜가며 제안된 방법과 KLA의 신호 대 잡음비 (SNR) 성능을 비교한 것이 <표 1>에 주어져 있다. <표 1>에서 보면, 대체로 상관계수, 벡터 차원, 코드북의 크기가 증가함에 따라 제안된 방법과 KLA 간의 성능차이가 커짐을 알 수 있다. OLVQ-SA는 KLA에 대해 최대 3.12(dB)의 성능우위를 나타내고 있다. 이는 코드북 성능이 SNR의 관점에서 약 2배정도 개선되었음을 뜻한다. 그러나 코드북 크기가 64인 경우에는 상관계수나 벡터차원에 상관없이 제안된 방법과 KLA의 성능에 별 차이가 없었다. 이는 간단한 양자화 문제일 경우 KLA도 좋은 성능을 보이며 적절한 해법이 될

수 있음을 시사하는 것이다. <표 1>은 또한 동일한 데이터에 대해 GLA에 의해 설계된 코드북의 성능에 대해서도 보여주고 있다. 앞선 연구 [3,6]의 결과와 마찬가지로 GLA는 가장 낮은 성능을 나타내었다.

〈표 1〉 가우스-마코프 소스에 대한 KLA와 OLVQ-SA의 SNR 성능비교. 단위는 dB 이다.

ρ	벡터 차원	코드북 크기	GLA	KLA	OLVQ-SA
0	2	64	15.63	**16.46**	15.67
		128	17.94	**18.96**	19.04
		256	19.59	**21.78**	22.65
	4	64	7.91	**7.95**	8.06
		128	9.62	**9.77**	9.97
		256	11.39	**12.03**	12.27
0.5	2	64	15.60	**16.96**	16.40
		128	18.29	**19.47**	19.63
		256	19.75	**21.93**	23.27
	4	64	8.78	**8.88**	8.93
		128	10.30	**10.63**	10.82
		256	11.88	**12.88**	13.15
0.9	2	64	17.56	**19.68**	19.45
		128	20.25	**22.11**	22.62
		256	20.97	**23.11**	26.14
	4	64	12.85	**13.32**	13.35
		128	13.78	**14.98**	15.11
		256	14.37	**16.02**	17.46

〈표 2〉 음성 소스에 대한 KLA와 OLVQ-SA의 SNR 성능비교. 단위는 dB 이다.

벡터 차원	코드북크기	GLA	KLA	OLVQ-SA
2	64	20.93	**22.46**	22.72
	128	21.95	**24.10**	26.24
	256	22.83	**24.89**	30.23
4	64	15.95	**17.21**	18.15
	128	16.94	**18.39**	20.46
	256	17.65	**18.96**	23.35

5.2 음성 소스에 대한 양자화 실험

음성은 SB-64 사운드카드로 모노채널, 11kHz 샘플링 주파수, 16bit 양자화 레벨로 총 16384 샘플을 채집하였다. 이렇게 채집된 샘플들을 다시 평균 0, 분산 1로 표준화한 뒤에 양자화기 설계를 위한 학습데이터로 사용하였다.

<표 2>에서 보면 KLA에 비해 제안된 방법의 우수성이 <표 1>에서 보다 분명히 드러남을 알 수 있다. 이는 패턴의 분포가 복잡해짐에 따라 통계적 학습이 효과를 발휘하는 것으로 볼 수 있다. 대체적으로 여기서도 벡터 차원, 코드북의 크기가 증가함에 따라 제안된 방법들과 KLA 간의 차이가 커지고 있다. OLVQ-SA는 KLA에 비해 최대 5.52에서 최저 0.81까지의 성능 우위를 나타내고 있다. 이를 코드북의 크기와 관련지어 생각해 볼 때, KLA에 비해 코드북의 크기를 절반으로 줄여도 제안된 방법이 보다 우수한 코드북을 생성할 수 있음을 보여주는 것이다. 또한 이는 1 bit 이상의 전송률 이득이 가능하다는 것을 뜻한다. KLA는 코드북의 크기가 128에서 256으로 증가할 때 SNR의 증가율이 급격히 둔화되고 있다. 이는 문제가 복잡해짐에 따라 KLA에 의한 극소점 근방으로의 접근이 더욱 어려워지고 있다는 것을 시사한다. 반면 제안된 방법은 코드북 크기에 대해 거의 등 간격의 SNR 개선을 보이고 있다. 이는 복잡해지는 문제에 대해서 잘 대응하고 있음을 시사한다. 이 문제에 대해서도 GLA는 가장 낮은 성능을 나타내었다.

6. 결 론

우리는 온라인 방식의 벡터 양자화기 설계에 있어서 극소점 문제를 해결하기 위해 기존의 KLA에 SA 방법을 응용하였다. 가우스-마코프 소스와 음성 데이터에 대한 벡터 양자화 실험 결과 제안된 방법이 KLA에 비해 일관되게 우수한 성능의 코드북을 생성함을 보였다. 따라서 온라인 양자화기 설계에도 SA 방법이 성공적으로 응용될 수 있음을 보였다.

참 고 문 헌

[1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, Vol. COM-28, pp. 84-95, Jan. 1980.

[2] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, 1995.

[3] N. M. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organization feature maps," in *Proc. 2nd ICNN Conf.*, Vol.I, pp.101-105, 1988.

[4] B. Widrow and S. D. Sterns, *Adaptive Signal Processing*, Englewood Cliffs, NJ : Prentice-Hall, 1985.

[5] Eyal Yair, Kenneth Zeger and Allen Gersho, "Conjugate Gradient Methods For Designing Vector Quantizers," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, Vol.1, pp.245-248, 1990.

[6] P. -C. Chang and R. M. Gray, "Gradient algorithms for designing predictive vector quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol.ASSP-34, pp.679- 690, Aug. 1986.

[7] P. J. M. van Laarhoven and E. H. L. Aarts, *Simmulated Annealing : Theory and Applications*, Boston : D. Reidel Publishing, 1987.

[8] V. Cerny, "A thermodynamical approach to the traveling salesman problem : an efficient simulation algorithm," preprint, Inst. Phys. & Biophys., Comenius Univ., Bra-tislava, 1982.

[9] S. Kirkpatrick, C. D. Gellatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol.220, pp.671-680, May 1983.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, Vol.21, pp.1087-1091, 1953.

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Baysien restoration of images," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol.PAMI-6, pp.721-741, Nov. 1984.

[12] J. Vaisey and A. Gersho, "Simulated annealing and codebook design," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pp.1176-1179, Apr. 1988.

[13] J. K. Flanagan, D. R. Morrell, R. L. Frost, C. J. Read, and B. E. Nelson, "Vector quantization codebook generation using simulated annealing," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pp.1759-1763, May. 1989.

[14] A. E. Cetin and V. Weerackody, "Design vector quantizers using simulated anneaing," *IEEE Trans. Circuit Syst.*, Vol.CAS-35, Dec. 1988.

[15] K. Zeger and A. Gersho, "A stochastic relaxation algorithm for improved vector quantizer design," *Electron. Lett.*, Vol.25, No.14, pp.896-898, July. 1989.

[16] K. Zeger, J. Vaisey and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. Signal Proc.*, Vol.40, No.2, Feb. 1992.

[17] S. P. Lloyd, "Least squares quantization in PCM,"IEEE Trans. Inform. Theory, Vol.21, pp.129-137, Mar. 1982.

[18] P. D. Wasserman, *Neural Computing*, NY : Van Nostrand Reinhold, 1989.

[19] D. DeSieno, "Adding a conscience to competitive learning," in *Proc 2nd ICNN Conf.*, Vol.I, pp.117-124, 1988.

[20] H. Ritter and K. Shulten, "Kohonen's self-organizing maps : Explorering their computational capabilities," in *Proc 2nd ICNN Conf.*, Vol.I, pp.109-116, 1988.

[21] E. Yair, K. Zeger and A. Gersho, "Competitive Learning and soft competition for vector quantizer design," *IEEE Trans. Signal Proc.*, Vol.40, No.2, Feb. 1992.

송 근 배

e-mail : geunbae@madang.ajou.ac.kr

1992년 아주대학교 전자공학과(학사)

1995년 아주대학교 전자공학과(공학석사)

1995년~1996년 현대전자 소프트웨어 연구소

1996년~현재 아주대학교 전자공학과 (박사과정 재학)

관심분야 : 음성신호처리, 음성인식, 인공지능 및 신경망

이 행 세

e-mail : haingsei@madang.ajou.ac.kr

1966년 전북대학교 전기공학과(학사)

1972년 서울대학교 전자공학과(공학석사)

1984년 고려대학교 전자공학과(공학박사)

1968년~1970년 해군 사관학교 교관

1982년~1983년 Columbia Univ.n.y. 객원 교수

1987년~1988년 INRIA PARIS 객원교수

1992년~1994년 거제전문대학장

1973년~현재 아주대학교 교수

관심분야 : 음성신호처리, 음성인식, 인공지능 및 신경망