

## 再初期化法を用いた適応ベクトル量子化

西田 健<sup>†</sup>      黒木 秀一<sup>†</sup>      佐伯 知則<sup>†</sup>

Adaptive Vector Quantization Using Re-initialization Method

Takeshi NISHIDA<sup>†</sup>, Shuichi KUROGI<sup>†</sup>, and Tomonori SAEKI<sup>†</sup>

あらまし 画像データや音声データなどの大量の信号のデジタルコーディングに有用なベクトル量子化 (VQ) において、従来手法の多くは入力信号が時不変な確率分布をもつと仮定していた。しかし実際の場面では、入力信号やセンサの特性などの環境の確率的性質が時間的に変化する場合がありますため、近年、このような入力信号の変化に適応可能ないくつかの VQ の手法が提案されている。しかしそれらには、局所解に陥る、適応が遅いなどの問題がある。そこで本論文では再初期化法と呼ぶ手法を用いることにより、従来手法よりも入力信号の変化に対する適応が速い適応 VQ の手法を提案する。本論文ではまず、この配法の局所解問題を解決するための再初期化法を示し、その後、2次元ベクトルと高次元ベクトルに対する VQ について従来手法と提案手法の性能比較実験を行う。その結果、提案手法は、入力信号の確率分布の時間変化に対して従来手法よりも高い適応能力をもつことが確かめられた。

キーワード 適応ベクトル量子化, この配法の局所解問題, 競合学習, 再初期化法

### 1. ま え が き

ベクトル量子化 (VQ: Vector Quantization) とは、連続信号若しくは大量の離散ベクトル信号である標本データを有限個の代表ベクトルを用いて量子化することで情報伝送における情報量や保存データ量の圧縮を行おうとするものであり、画像データや音声データなどの大量の信号のデジタルコーディングに有用であることが示されている [1]。しかし従来の VQ の手法の多くは入力信号の確率分布が一定であることを前提としているため、実際の多くの応用における時間変化する入力信号の確率分布に対して、直接これらの手法を用いることは困難であった。したがって、近年、非定常入力信号に適応することを目的とした VQ アルゴリズム (適応 VQ アルゴリズム) が多く提案されている。現在までに提案されている適応 VQ の手法は、バッチ処理による手法とオンライン処理による手法に大きく分類される [1], [2]。前者の手法は、一定時間区間ごとに一定の入力信号をバッファに蓄え、これに対する最適な VQ を順次行うものであり、入力信号の変化への適応は各一定時間ごとに行われる。代表的

なものとして LBG (Linde-Buzo-Gray) 法 [3] を拡張した Gersho らの手法 [4] や CSL (Competitive and Selective Learning) [5] などがある。これに対し後者の手法は Grossberg [6] や Kohonen [7] による競合学習を発展させたものであり、各時刻の入力信号に対して逐次的に適応し VQ を行う手法である。この手法はすべての入力データをバッファに蓄えておく必要がないため、バッチ処理手法では一度に扱うことができないような多くの入力信号の VQ に用いることができ、またバッチ処理手法の代替として用いることもできる。現在までに提案されているオンライン処理型の適応 VQ アルゴリズムには OPTM (Optimal Adaptive K-Means Algorithm) [8] や DOCL-II (Diversity Oriented Competitive Learning) [9] などがある。OPTM は時間変化するひずみ変数を導入し、これを用いてこの配法の学習率を逐次的に計算することで適応 VQ を可能にしている [8]。しかし OPTM による VQ は入力信号の変化によっては局所解に陥ることが芳澤ら [9] によって指摘されており、この問題を解決するための手法として DOCL-II が提案された。DOCL-II は発火回数に基づくエントロピーを導入し、代表ベクトルの数が十分多い場合の最適解条件 (漸近的最適解条件) の近似条件である等発火回数条件を満たす解に近づける工夫によって適応 VQ を達成しようとするものであ

<sup>†</sup>九州工業大学工学部, 北九州市  
Faculty of Engineering, Kyushu Institute of Technology,  
Kitakyushu-shi, 804-8550 Japan

る．しかし DOCL-II は決定すべきパラメータの数が多く，入力信号の時間変化に対する適応が遅いという問題があり，更に等発火回数条件による VQ は漸近的最適性を保証しないことが Ueda ら [5] によって示されている．そこで，本論文では漸近的最適解のための条件を与える等ひずみ原理 [5] を理論的背景とする再初期化法と，局所最適化を行うこう配法を統合したオンライン処理型の適応 VQ の手法を提案する．提案手法は入力信号の逐次学習が可能であるニューラルネットワークを用いて実現され，従来手法よりもアルゴリズムが比較的簡潔であると同時に，迅速に入力信号の変化に適応することができることに特徴がある．

以下では 2. で時間変化する入力信号の確率分布に適応するためのこう配法について述べる．3. では提案手法の理論的背景となる等ひずみ条件を説明した後，競合学習に組み込む再初期化法について述べる．4. では 2 次元ベクトルに対する VQ について，入力が入力一定時間区で切り換わる場合と連続的に変化する場について提案手法と DOCL-II，OPTM の性能比較を行う．5. では画像データのコーディングと高次元ベクトル VQ における提案手法の有効性についての検証を行い，6. で結論を述べる．

## 2. こう配法と局所解問題

ニューラルネットワークを用いた VQ を行う場合には，標準データを入力ベクトルに，代表ベクトルを各セルのもつ荷重ベクトルにそれぞれ対応付けし，入力ベクトルに最近隣の荷重ベクトルのみをこう配法によって更新するという競合学習法を用いるのが一般的である．

本論文では  $x \in \mathbb{R}^k$  を確率密度関数  $p(x)$  をもつ  $k$  次元の入力ベクトルとし，荷重ベクトル  $w_i \in \mathbb{R}^k$  ( $i \in I = \{1, 2, \dots, N\}$ ) をもつセルからなるニューラルネットワークを考える．このニューラルネットワークの学習においては入力ベクトル  $x$  と荷重ベクトル  $w_i$  のひずみ

$$D = \frac{1}{k} \sum_{i=1}^N \int_{V_i} \|x - w_i\|^2 p(x) dx = \sum_{i=1}^N D_i \quad (1)$$

を最小化することを目的とする．ここで  $D_i$  は  $w_i$  に対するポロノイ領域  $V_i$  におけるひずみ（部分ひずみ）を表し， $V_i$  は以下の式で与えられる．

$$V_i = \{x \mid \|x - w_i\| \leq \|x - w_j\| \text{ for each } j \neq i\} \quad (2)$$

このひずみ  $D$  の最小化を行うためのこう配法による荷重ベクトルの更新法は，確率近似法 [12] より

$$w_c(t+1) := w_c(t) + \alpha(t)(x(t) - w_c(t)) \quad (3)$$

で与えられる．ただし  $w_c(t)$  は離散時刻  $t$  における入力  $x(t)$  に最近隣の荷重ベクトルであり“:=”は代入を意味する．また  $\alpha(t)$  は刻み幅であり学習率とも呼ぶ．しかしこう配法のみによる荷重ベクトルの配置は確率密度関数で重み付けされた各ポロノイ領域の重心

$$w_i = \frac{\int_{V_i} xp(x)dx}{\int_{V_i} p(x)dx} \quad (4)$$

に収束するため，大域的最適解ではなく局所最適解にしかならない [10]．すなわち，この手法のみによる荷重ベクトルの最終配置は初期配置に大きく依存し，例えば一度も競合に勝てないセルが存在するような状況が生ずる場合があることが知られている．またこの式を荷重ベクトルの更新に用いる多くの VQ アルゴリズムでは収束のために学習率  $\alpha(t)$  を  $t \rightarrow \infty$  で  $\alpha(t) \rightarrow 0$  とするが，これでは本研究で考えるような時間変化する入力信号の確率分布に対しては適用できない．すなわち，長い時間が経過し  $\alpha(t)$  が 0 に非常に近い値となると  $w_c(t)$  はほとんど更新されなくなり，新たな確率分布をもった入力には対応できなくなる．そこで本研究ではこう配法の局所解問題を解決するための再初期化法を導入し，また時変入力分布に適応するため学習率を学習の収束の度合によって逐次的に決定する（学習率の計算法の詳細は 3.3 (1) で述べる）．

## 3. 再初期化法を用いた適応 VQ

### 3.1 等ひずみ条件

最適な荷重ベクトル配置の条件は式 (1) で定義されたひずみを用いて以下のように与えられる [5], [11]；入力ベクトル  $x$  の確率密度関数  $p(x)$  は連続であり，荷重ベクトル  $w_i$  ( $i \in I$ ) の数  $N$  は十分大きく，すべての  $x \in V_i$  とすべての  $i \in I$  について近似  $p(x) \simeq p(w_i)$  が成り立つとする．このとき  $D$  の下限若しくは最小値は  $i \in I$  においてすべての  $D_i$  が互いに等しくなったときに得られる．以上の最適 VQ に関する条件を等ひずみ条件という．

### 3.2 再初期化法

まず  $w_i$  をもつセルは各離散時刻  $t$  における入力ベクトルに対するひずみを加算したひずみ変数  $d_i(t)$  を

$$d_i(t+1) := \begin{cases} \eta d_i(t) + \|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2 & \text{if } i = c \\ \eta d_i(t) & \text{otherwise} \end{cases} \quad (5)$$

として蓄えることとする．ここで  $\eta$  ( $0 < \eta < 1$ ) は忘却率であり,  $\mathbf{w}_c(t)$  は  $\mathbf{x}(t)$  の最近隣荷重ベクトルである．また  $d_i(t)$  ( $i \in I$ ) の初期値は非常に小さな値とする．式 (5) より  $\mathbf{w}_i(t)$  ( $i \in I$ ) の平衡状態における  $d_i(t)$  は,

$$\begin{aligned} d_i(t) &= \frac{1}{1-\eta} \int_{V_i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2 p(\mathbf{x}) d\mathbf{x} \\ &= \frac{kD_i}{1-\eta} \end{aligned} \quad (6)$$

に収束するため, 等ひずみはひずみ変数  $d_i(t)$  がすべての  $i \in I$  で等しいときに達成されるといえる．そこで, 等ひずみに対する全体的なひずみ変数の状態を評価する指標としてひずみ変数  $d_i(t)$  のエントロピー  $H(t)$  を導入する． $H(t)$  は以下のように計算される．

$$H(t) = - \sum_{i=1}^N \frac{d_i(t)}{\sum_{j=1}^N d_j(t)} \ln \left( \frac{d_i(t)}{\sum_{j=1}^N d_j(t)} \right) \quad (7)$$

これを正規化して

$$I(t) = \frac{H(t)}{\ln(N)} \quad (8)$$

とし  $I(t)$  を学習収束度と呼ぶことにする．この値は  $0 \leq I(t) \leq 1$  であり, すべてのひずみ変数  $d_i(t)$  が等しいときのみ 1 となり, ひずみ変数  $d_i(t)$  の値のばらつきが大きいくほどこの値は小さくなる．そこで学習収束度のしきい値  $I_{th}$  を 1 よりも小さなある定数で定め, 時刻  $t$  において

$$I(t) < I_{th} \quad (9)$$

を満たす場合は, 全体的な荷重ベクトルの配置は等ひずみの状態から遠いと判断することにする．

さて式 (9) とは対照的に, 以下のような条件によっても各セルのひずみ変数の等ひずみからの偏りを逐次的かつ容易に判断することができる．すなわち, 競合学習では各時刻で入力ベクトルに最近隣である荷重ベクトルをもつセルを探索するので, そのセルのひずみ変数が

$$d_c(t) > d_{th} d_{\text{mean}}(t) \quad (10)$$

を満たすとすると, このひずみ変数が等ひずみを満たさない要因となっていると判断することにする．ここで  $d_{\text{mean}}(t) = \sum_{i=0}^N d_i(t)/N$  であり,  $d_{th}$  は 1 より大きな定数である．以下では  $d_{th}$  をひずみ変数のしきい値と呼ぶ．

最後に, これら 2 種類の等ひずみに対するひずみ変数の判断基準を以下のように統合する．すなわち式 (9) と式 (10) を同時に満たす場合にはすべてのセルの中で最小のひずみ変数  $d_l(t)$  をもつ第  $l$  番目のセルのもつデータと競合に勝つセルのひずみ変数  $d_c(t)$  を以下のように更新する．

$$\begin{aligned} \mathbf{w}_l(t+1) &:= \mathbf{x}(t) \\ d_i(t+1) &:= \begin{cases} \eta d_{\text{mean}}(t) & \text{if } i = c \text{ or } i = l \\ \eta d_i(t) & \text{otherwise} \end{cases} \end{aligned} \quad (11)$$

これを第  $l$  番目のセルを再初期化するという．すなわちこの処理は, ひずみ変数が小さいセルのデータを初期化し, 大きなひずみ変数をもつセルの近傍に再配置することを意味する．また, この処理によってひずみ変数が平均化されることでひずみ変数のエントロピー  $H(t)$  が上昇し,  $I(t)$  も上昇することになる．なお, この処理を行った直後のひずみ変数は正しい値をもたないが, 時間が十分経過すると式 (5) より式 (6) が成立するようになり, 正しい値をもつようになると考えられる．

以上をまとめると以下のアルゴリズムとなる；各時刻  $t$  における入力  $\mathbf{x}(t)$  に対して, 式 (9) と式 (10) を満たす場合は式 (11) により第  $l$  番目のセルを再初期化する．その他の場合は式 (3) により  $\mathbf{w}_c(t)$  を修正し, 式 (5) により  $d_i(t)$  を更新する．以下, このアルゴリズムを CRL (Competitive and Re-initialization Learning) と呼ぶことにする．

### 3.3 各パラメータの決定法

#### (1) 学習率

前述したように, 式 (3) の学習率  $\alpha(t)$  を単に学習回数や学習時刻に対して減少させるようなアルゴリズムでは入力分布の変化に対応できない．理想的には, 学習率はベクトル配置が収束するに従って減少し, もし入力分布に変化が生じ, 入力信号に対する荷重ベクトルのひずみのばらつきが大きくなった場合には増加するべきである．したがって, 学習収束度を用いて学習率を以下のように決定することにする．

$$\alpha(t) = \frac{\ln(N) - H(t)}{\ln(N)} = 1 - I(t) \quad (12)$$

これは文献 [8] において提案された手法と同様のものであるが  $H(t)$  の導出法は異なる．この式によって逐次的に求められる学習率は学習の進行状態によって自動的に調節される．

### (2) 学習収束度のしきい値とひずみ変数のしきい値

学習収束度のしきい値  $I_{th}$  とひずみ変数のしきい値  $d_{th}$  の設定のために、まず各時刻におけるひずみ変数  $d_c(t)$  と学習収束度  $I(t)$  の値の変動の性質について述べる．

まずひずみ変数  $d_i(t)$  ( $i \in I$ ) の値は、初期時刻若しくは入力分布が変化してしばらくは荷重ベクトル  $w_i(t)$  の更新に応じて変動し、時間経過とともに式 (4) を満たす解に収束していく．ここで各  $d_i(t)$  の収束値は入力分布と荷重ベクトルの数に依存して決まり、一般にはそれぞれ異なる値に収束する．例えば、入力分布に対して等ひずみを満たす荷重配置が存在せず、 $d_i(t)$  を等しい値に収束させることが原理的にできない場合があり (4.1 参照)、このような場合には最適な荷重配置でも  $d_i(t)$  の収束値はばらつく．したがって、各時刻において競合に勝つ荷重ベクトルのひずみ変数  $d_c(t)$  の値は確率的に入力されるベクトルに対応して  $d_i(t)$  の中から選択されるため、各  $d_i(t)$  の収束後も変動する．これに対し  $I(t)$  はすべての  $d_i(t)$  のばらつきを計る尺度であるので、 $d_i(t)$  が収束すると  $I(t)$  も収束する．より詳しくいうと、初期時刻や入力分布が変化してしばらくは  $d_i(t)$  がかなりばらついていてのでかなり低い値になるが、 $d_i(t)$  の収束に対応して時間とともにあまり変動しなくなり、ある程度高い値になっていく．

さて以上の  $I(t)$  と  $d_i(t)$  の変動の性質を考慮して、入力信号が一定時間区間ごとに变化する場合と連続的に变化する場合の  $I_{th}$  と  $d_{th}$  の値の設定について考える．まず前者の場合には、同一の入力信号の学習を一定時間行うことができるため、より高い最適化が可能であり、各処理の後半には再初期化はなるべく起こらない方がよい．そこで、 $d_c(t)$  は学習収束後もかなり変動するが  $I(t)$  はあまり変動しなくなることに注意して、式 (9) と式 (10) のしきい値の値を具体的に 4.1 で用いる値  $d_{th} = 1.4$ ,  $I_{th} = 0.985$  とすることを考える．すると式 (9) の  $I(t) < I_{th} = 0.985$  という条件は、 $d_i(t)$  が一様にばらついているとすると

$d_i(t)/d_{\text{mean}}$  の値のばらつく範囲がおおよそ  $1 \pm 0.53$  以上であることを示しており、この条件が満たされるときには式 (10) の  $d_c(t)/d_{\text{mean}} \geq d_{th} = 1.4$  という条件も高い頻度で満たされることになる．したがって、これらのしきい値を用いて式 (9) と式 (10) を同時に満たすときにはのみ再初期化を行うことにすると、初期時刻付近あるいは入力分布が変化してしばらくは式 (9) と式 (10) が同時に満たされて高い最適性を達成するように再初期化が起こる．しかし時間が十分経過すると  $I(t)$  が  $I_{th} = 0.985$  よりも高い値に収束していくことにより再初期化が起こらなくなり、この配法だけで最適化が行われる．なお  $d_{th}$  を小さくしすぎると再初期化すべきでない荷重も再初期化し、 $I_{th}$  を小さくしすぎると入力分布の変化が検知できなくなることに注意しなければならない [13]．次に入力信号が連続的に变化する場合には、高い最適性や少ない再初期化よりも適応速度の方が重視される．したがって収束の遅い  $I(t)$  を用いる式 (9) の条件は用いず (あるいは  $I_{th} = 1.000$  とし)、式 (10) の  $d_{th}$  による条件だけを用いるとこの配法と再初期化により適応の速さが保たれる．

### (3) 忘却率

忘却率  $\eta$  は過去の入力ベクトルを徐々に忘却し、入力分布の変化に適応するために 1 に近い値で与えられるパラメータである．各荷重ベクトルが更新されるのに要する平均時間は荷重ベクトルの数  $N$  によって異なるので、忘却率を

$$\eta = \exp\left(\frac{-1}{N\tau}\right) \quad (13)$$

で与えることにすると、 $\eta$  は  $N\tau$  時刻後にもとの値を  $e^{-1}$  倍に減衰させる効果をもち、 $\tau$  でこの忘却率を決定できるので直観的に理解しやすい．この  $\tau$  は、入力分布の変化する場合などに過去の入力分布の忘却を行う必要があるならば小さくし、逆に入力分布があまり変化しないような場合には大きくするといった調節が必要である．

## 4. 2次元ベクトルへの適用

### 4.1 一定時間区間ごとに变化する入力分布

まず一定時間区間ごとに变化する入力分布に対する CRL の性能の検討を行う．ここでは比較のために文献 [9] と同様のデータセットを用い (図 1(a) 参照)、これらは以下のようにして生成した．まずガウス分布  $N(0, 0.0025)$  に従う 2000 のベクトル

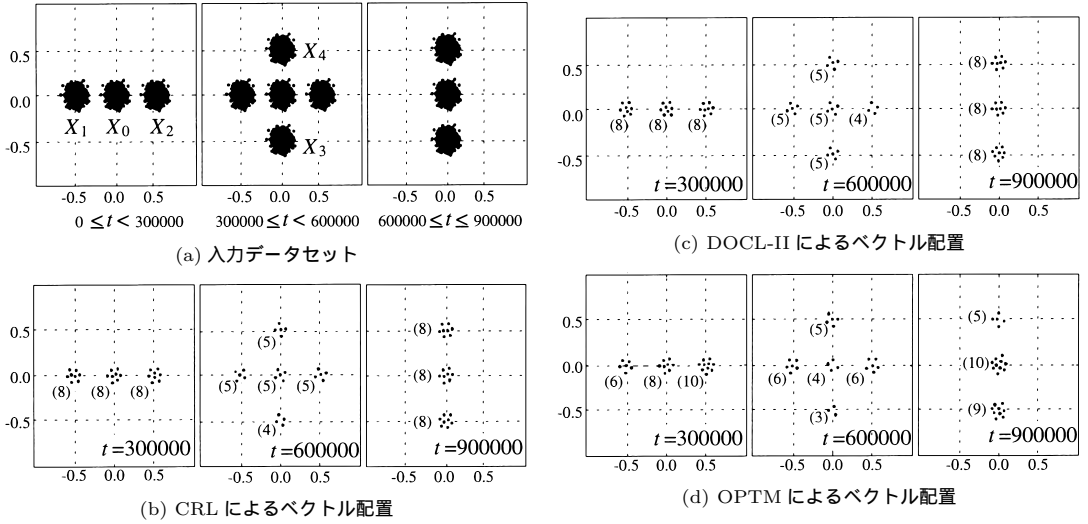


図1 一定時間区間で変化する入力分布に対する VQ の結果  
 Fig.1 Result of VQ for abruptly changing input distributions.

$x_{0j} = (x_{0j}, y_{0j})^T$  ( $j = 1, 2, \dots, 2000$ ) を生成し、その集合を  $X_0$  とする。次にこれらを  $x_{ij} = m_i + x_{0j}$ ,  $m_i = (a_i, b_i)$  として平行移動し生成される入力ベクトルの集合をそれぞれ  $X_1, \dots, X_4$  とした。ただし  $(a_i, b_i) = (0, \pm 0.5), (\pm 0.5, 0)$  であり、以下ではこれらのベクトル集合をクラスタと呼ぶ。これらのクラスタの組合せを 300000 回の学習回数ごとに図 1 (a) のように切り換えて入力し、CRL, DOCL-II, OPTM の 3 種類のアゴリズムを用いて VQ を行った。荷重ベクトル数は  $N = 24$  とし、学習に必要なパラメータの値は OPTM, DOCL-II においてはそれぞれの文献に挙げてあるものを用い、CRL においては  $I_{th} = 0.985$ ,  $d_{th} = 1.4$ ,  $\tau = 300$  とした。各荷重ベクトルの各要素の初期値は  $-1$  から  $1$  の間のランダムな値とした。図 1 (b) ~ (d) にそれぞれ 300000 回、600000 回、900000 回の学習が終わった後の各アルゴリズムによる荷重ベクトルの配置を示し、図 2 に入力データセットと荷重ベクトルの 2 乗誤差の平均 (MSE: Mean Squared Error) の推移を示す。ただし  $MSE = kD/N$  であり、図 1 (b) ~ (d) における各クラスタ付近の数字は配置されたベクトルの数を表している。まず図 1 (d) より、OPTM では同図 (b) の CRL や (c) の DOCL-II とは異なり、各クラスタに対する荷重ベクトル配置に偏りが生じ局所解に陥っていることがわかる。また、各クラスタに対する荷重ベクトル配置がうまくいっている DOCL-II においても、図 2 では最も低い MSE に収

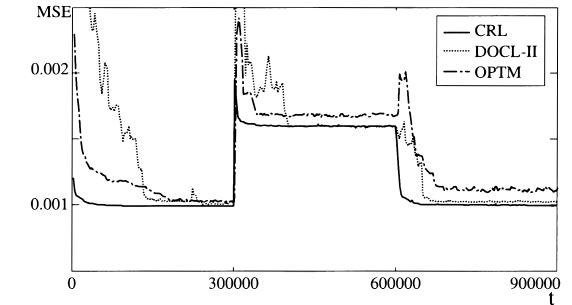


図2 MSE の推移  
 Fig.2 Transition of MSE.

束するまでに多くの学習回数を必要としていることがわかる。これらの結果より、CRL は局所解に陥ることなく、変化する入力分布に最も速く適応していることがわかる。

#### 4.2 連続的に変化する入力分布

次に連続的に変化する入力分布に対する CRL の性能について検討する。連続的に変化する入力分布として図 3 (a) に示すようなデータセットを用いた。すなわち、入力信号は初期時刻に中心がそれぞれ  $(-0.5, 0.5)$ ,  $(-0.5, 0.0)$ ,  $(-0.5, -0.5)$  である三つのクラスタからなり、これらは水平方向の移動によって 50000 回の学習の後にそれぞれのクラスタ中心が  $(0.5, 0.5)$ ,  $(0.5, 0.0)$ ,  $(0.5, -0.5)$  に移り、更に次の 50000 回の学習の後に水平方向の移動によって再びもとに戻る。この入力信号に対して CRL, DOCL-II, OPTM を用いて VQ を

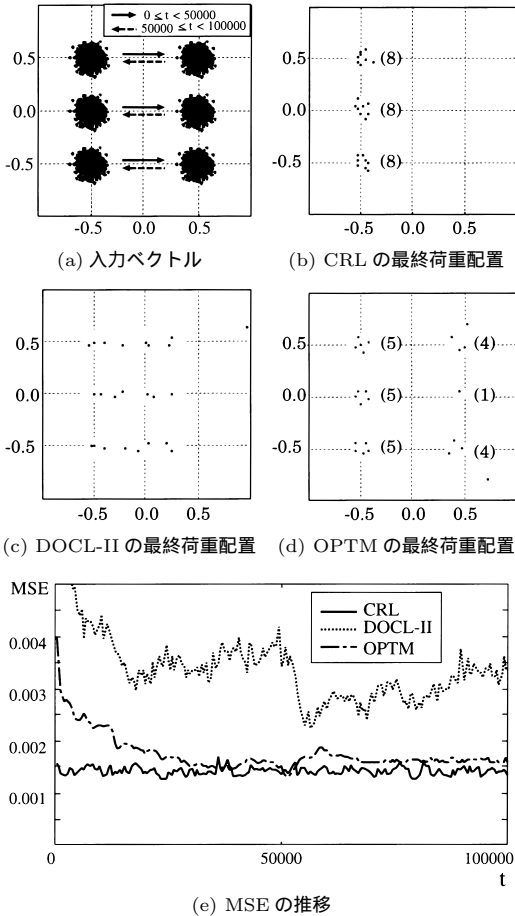


図3 連続的に変化する入力分布に対するVQ結果  
 Fig.3 Result of VQ for constantly changing input distributions.

行った。このとき3.3(2)(3)の議論より、連続的に変化する入力分布に対する適応を考慮してCRLのパラメータを $I_{th} = 1.000$ ,  $d_{th} = 1.4$ ,  $\tau = 50$ とし、 $I_{th}$ による再初期化の発動条件を用いず、また $\tau$ を小さめに定めた。他のアルゴリズムのパラメータは4.1と同様とした。100000回の学習後の荷重ベクトルの配置を図3(b)~(d)に示し、MSEの推移を同図(e)に示す。まず図3(b)~(d)よりDOCL-IIやOPTMでは荷重ベクトルが最終時刻には全く入力のない領域にとどまっているが、CRLの荷重ベクトルの配置は入力分布の変化に適応できていることがわかる。次に同図(e)の結果よりCRLによるMSEの推移が三つのアルゴリズムの中で最良のものとなっていることがわかる。それに対しDOCL-IIでは入力信号の変化に適

応が追いつかず大きなMSEの値で推移しており、またOPTMにおいてもMSEが低い値に推移するまでには多くの学習回数が必要であることがわかる。

#### 4.3 異なるパラメータ値に対する性能

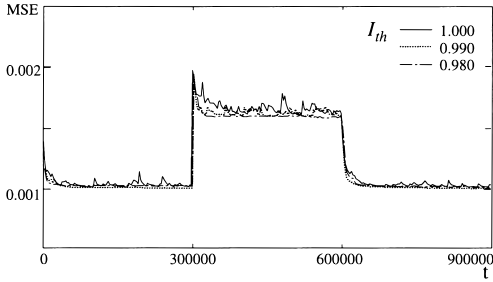
ここではCRLのパラメータ $I_{th}$ ,  $d_{th}$ ,  $\tau$ によって、その性能がどのように変化するかを検討する。まず、4.1の一定時間区間で入力信号が切り換わる場合についてそれぞれ $I_{th}$ と $d_{th}$ について値を変化させシミュレーションを行ったときのMSEの推移を図4(a), (b)に示す。ただし同図(a)のシミュレーションでは $d_{th} = 1.4$ , (b)では $I_{th} = 0.985$ とし、他の条件は4.1と同様とした。図4(a)より、 $I_{th} = 0.980$ とした場合には図2の $I_{th} = 0.985$ の場合とほぼ同様にMSEが推移したが、 $I_{th} = 0.990, 1.000$ とした場合にはMSEが振動的に推移していることがわかる。このMSEの振動は過度の再初期化が生じたことを示している。しかし図2の結果と比較して、最も大きく振動した $I_{th} = 1.000$ においてもOPTMと同等以下のMSEを維持していることがわかる。また図4(b)より、 $d_{th} = 1.2$ の場合は図2の $d_{th} = 1.4$ とした場合とほぼ同様にMSEが推移したが、 $d_{th} = 1.6, 1.8$ の場合には学習回数600000回以後にMSEが高い値にとどまり局所解に陥っていることがわかる。これは $d_{th}$ の値が大きすぎたため、各クラスタに対する荷重ベクトルの配置の偏りによって生ずる各セルのひずみ変数のばらつきが許容され、局所解に陥ったのだと考えられる。このような場合には $d_{th}$ を更に低く設定する必要があるといえる。

次に4.2の連続して変化する入力分布に対して、それぞれ $d_{th}$ と $\tau$ を変化させてシミュレーションを行ったときのMSEの推移を図4(c), (d)に示す。ただし同図(c)のシミュレーションでは $\tau = 50$ , (d)では $d_{th} = 1.4$ とし、また $I_{th}$ の値はいずれの場合も1.000とした。この結果より、いずれのMSEの推移も振動的ではあるがCRLによるMSEは図3(e)のDOCL-IIよりも小さく、またOPTMと同等程度の値で推移していることがわかる。以上より、CRLのパラメータ値の設定にはある程度の許容範囲があり、望ましい値と多少異なっていてもDOCL-IIやOPTMと同等以上の性能が望めることがわかる。

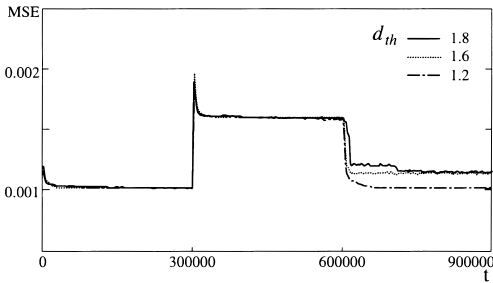
### 5. 高次元ベクトルへの適用

#### 5.1 画像データのコーディング

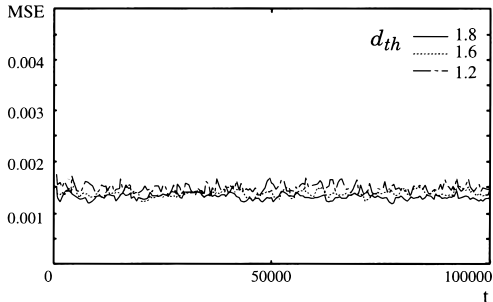
ここでは画像データのコーディングにおけるCRL



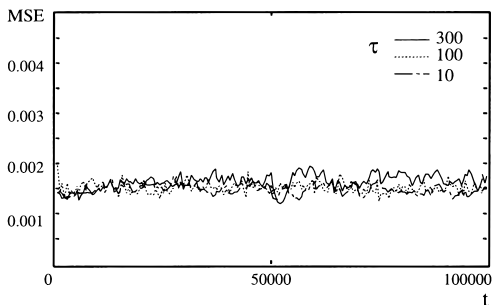
(a) 一定時間区間で切り換わる入力分布に対する  $I_{th}$  の変化による MSE の推移の比較



(b) 一定時間区間で切り換わる入力分布に対する  $d_{th}$  の変化による MSE の推移の比較



(c) 連続的に変化する入力分布に対する  $d_{th}$  の変化による MSE の推移の比較



(d) 連続的に変化する入力分布に対する  $\tau$  の変化による MSE の推移の比較

図 4 異なるパラメータ値に対する性能

Fig. 4 Performances for different parameter values.

の性能の検討を行った．比較のために，文献 [8] と同様の “Lena” 画像 ( 図 5 (a) 参照 ) を用いた．これは



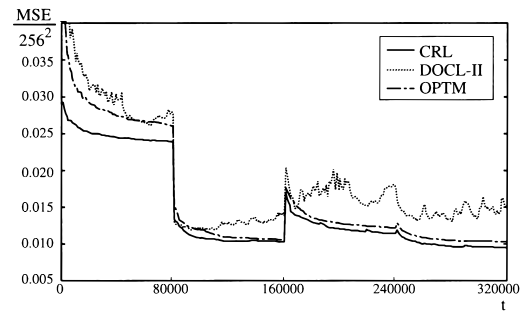
(a) “Lena” 画像

(b) CRL PSN=18.5988



(b) DOCL-II PSN=17.2885

(c) OPTM PSN=18.3000



(d) 正規化 MSE の推移

図 5 画像 “Lena” の VQ 結果

Fig. 5 Result of VQ for the “Lena” image.

512 × 512 画素からなるグレースケールの画像であり，1 画素は 8 bit ( 256 階調 ) で表現される．この画像に対する VQ は文献 [8] と同様の手順により，まず画像を 4 × 4 画素の方形ブロックに分割し，それぞれを 16 次元の一つの入力ベクトルとした．更にこの図形の中心を原点として画像を 256 × 256 画素の 4 象限に分割し，それぞれ左下，右上，左上，右下の画像の順にパッチ的な VQ を行っていくものとした．それぞれの象限の VQ においては，入力ベクトルをランダムに選択し 80000 回の学習を行った．この回数は各象限に含まれる 4096 のベクトルを約 20 回学習することに相当する．32 個の荷重ベクトルを用意し，初期の荷重ベクトルはランダムに生成した．またここでも CRL, DOCL-II, OPTM についての性能比較を行った．CRL のパラメータは  $I_{th} = 0.985$ ,  $d_{th} = 1.4$ ,

$\tau = 300$  とし、他はそれぞれの文献の高次元 VQ を行うためのパラメータの値を用いた。これらの結果を図 5 に示す。また同図 (b) ~ (c) には、静止画像の品質測度として用いられるピーク信号対雑音比 (PSN 比) を示した。ただし PSN 比は以下のように計算される。

$$PSN = 10 \log_{10} \left( \frac{256^2}{MSE} \right) \quad (14)$$

これらの結果より CRL は他のアルゴリズムよりも高い品質で画像のコーディングを達成しており、各象限の画像の VQ においても、最も速く低い MSE への推移が達成されていることがわかる。

### 5.2 回転変換パターンの学習

ここでは回転変換に不変なパターン認識を行うニューラルネットワーク [14] を用いて回転変換されたパターン集合の学習を行い、高次元 VQ における CRL の性能の検討を行った。このネットワークへの入力ベクトルは図 6 に示す 2 値 (0,1) パターンを回転したものをを用いた。これらのパターンは  $60 \times 60$  画素の背景の中央に Times New Roman Bold フォントの数字 '0', '1' をはめ込み回転したものであり、5.1 の入力ベクトルよりも更に高次元である 3600 次元のベクトルとして 1 パターンが扱われる。以下のシミュレーションでは 36 個の荷重ベクトルを用いた。このニューラルネットワークの訓練は、図 6 のパターンに  $0^\circ$  から  $360^\circ$  までの間でランダムな回転変換を施すと同時に正規化し、各時刻に 1 パターンずつ入力することで以下のように行った。まず訓練パターンを '0' として訓練を 10000 回繰り返した後、訓練パターンを '1' へと変化させ、再び 10000 回の訓練を行った。ここでも比較のために OPTM, DOCL-II について同様のテストを行った。各アルゴリズムのパラメータは 5.1 と同様の値を用いた。

まず MSE の推移を図 7 に示す。これはテストパターンとして  $3^\circ$  ごとに 120 通りの回転変換パターンを生成し、各時刻における荷重ベクトルとテストパターンとの MSE を求めたものである。この結果より

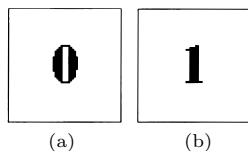


図 6 (a) パターン '0' と (b) パターン '1'  
Fig. 6 (a) Pattern '0' and (b) pattern '1.'

CRL による VQ が最も迅速に変化した入力分布に適應しており、DOCL-II や OPTM による VQ には収束のために、より多くの学習回数が必要であることがわかる。次に 10000 回、20000 回の学習が終わった時点での荷重ベクトルをそれぞれ図 8 (a) ~ (f) に示す。ま

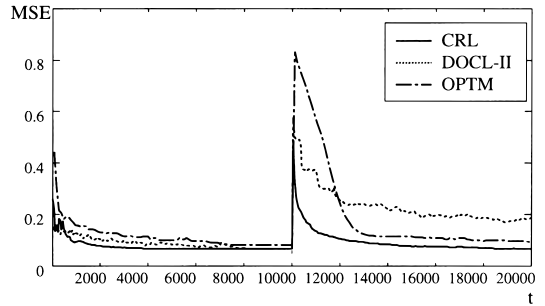


図 7 MSE の推移  
Fig. 7 Transition of MSE.

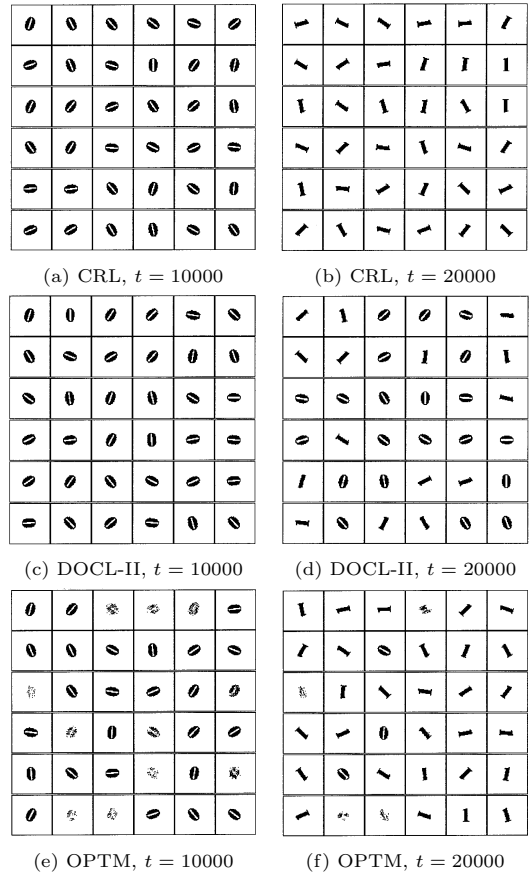


図 8 学習後の荷重ベクトル  
Fig. 8 Weight vectors after learning.



ず図 8 (a), (b) より CRL による VQ ではすべての荷重ベクトルにおいて入力信号の変化に適応できていることがわかる。それに対し DOCL-II は同図 (c) においては学習が収束しているが, 入力信号が切り換わった後の図 (d) では学習が収束しておらず, 10000 回の学習回数では入力信号の変化に適応しきれていない。また OPTM は学習にうまく利用されていない荷重ベクトルが存在し, 入力信号が変化した後も適応できない荷重ベクトルが多く存在することが同図 (e), (f) よりわかる。以上より, 非常に高い次元における VQ に対しても CRL は有効であることがわかる。

## 6. む す び

本論文では入力信号の確率分布の変化に適応可能な VQ アルゴリズム, CRL を提案し, 従来の適応 VQ アルゴリズムよりも高い性能をもつことを示した。CRL による適応 VQ は, 適応が迅速である, 設定するパラメータが少ないなどの特徴がある。また CRL による VQ は局所解に陥らず, 高次元ベクトルに対する応用が十分に行えることをシミュレーションにより確認した。

## 文 献

- [1] R.M. Gray, "Vector quantization," IEEE ASSP Mag., pp.4-29, April 1984.
- [2] J.E. Fowler, "Adaptive vector quantization for the coding of nonstationary sources," SPANN Laboratory Technical Report, TR-95-05, The Ohio State University, April 1995.
- [3] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., vol.COM-28, no.1, pp.84-95, Jan. 1980.
- [4] A. Gersho and M. Yano, "Adaptive vector quantization by progressive codevector replacement," Proc. International Conference on Acoustics, Speech, and Signal Processing, pp.4.6.1-4.6.4, 1985.
- [5] N. Ueda and R. Nakano, "A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers," Neural Networks, vol.7, no.8, pp.1211-1227, 1994.
- [6] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," Biol. Cybern., vol.23, pp.121-134, 1976.
- [7] T. Kohonen, Self-organization and associative memory, Springer-Verlag, New York, 1984.
- [8] C. Chinrungrueng and C.H. Séquin, "Optimal adaptive K-means algorithm with dynamic adjustment of learning rate," IEEE Trans. Neural Networks, vol.6, no.1, pp.157-169, 1995.
- [9] 芳澤伸一, 道木慎二, 大熊 繁, "変化する入力データに

対する学習の定常特性と過渡特性を改善する新しいベクトル量子化アルゴリズム," 信学論 (D-II), vol.J82-D-II, no.4, pp.807-817, April 1999.

- [10] M. Iri, K. Murota, and T. Ohya, "A fast Voronoi-diagram algorithm with applications to geographical optimization problems," Lecture Notes in Controls and Information Science, vol.59, System Modelling and Optimization, ed. P. Thoft-Christensen, pp.273-288, Springer-Verlag, 1984.
- [11] A. Gersho, "Asymptotically optimal block quantization," IEEE Trans. Inf. Theory, vol.28, no.2, pp.157-166, 1979.
- [12] K.I. Diamanaras and S.Y. Kung, Principal component neural networks, John Wiley & Sons, 1996.
- [13] S. Kurogi and T. Nishida, "Competitive learning using gradient and reinitialization methods for adaptive vector quantization," Proc. NNSP2000, pp.346-355, 2000.
- [14] 西田 健, 黒木秀一, "多層競合ネットによる座標変換に不変なパターン認識," 日本神経回路学会誌, vol.7, no.4, pp.106-114, 2000.

(平成 12 年 9 月 11 日受付, 12 月 25 日再受付)



西田 健 (学生員)

平 10 九工大・工・設計生産卒。平 11 同大学院博士前期課程了。現在, 同大学院博士後期課程に在学中。主にニューラルネットによるパターン認識の研究に従事。日本神経回路学会, 計測自動制御学会各会員。



黒木 秀一 (正員)

昭 55 九工大・工・電気卒。昭 60 東工大大学院博士課程了。同年より九工大・制御・助手を経て平 3 同大・助教授。工博。主にニューラルネットの研究に従事。日本神経回路学会, 計測自動制御学会各会員。



佐伯 知則

平 11 九工大・工・設計生産卒。現在, 同大学院博士前期課程に在学中。ニューラルネットによるベクトル量子化の研究に従事。