要素和エントロピー最小化原理により学習する神経回路モデルの研究

宮崎大学 工学部 情報システム工学科
76050460 馬場 一暢

1 はじめに

ものを見たときの情報は，脳内でどの様に表現されているのだろうか．このように外界からの情報を脳内に表現する情報表現の可能性を示す一つの方法として，神経回路モデルを使った研究がある．本研究では，外界からの入力信号に応じた情報表現を自動的に獲得する能力をもつ自己組織化の回路について考えた．具体的には，要素和エントロピー最小化原理により学習が進む回路について実験をおこない，神経回路モデルの性能を評価した．

2 要素和エントロピー最小化原理

2.1 神経回路モデルの構造

外界から入力 $\vec{x}$ が次々に観測され，これを層状の神経回路モデルで情報を処理することを考える．回路は，m 個の信号 $\vec{x} = (x_1, x_2, ..., x_m)$ を受け取り，n 個の信号 $\vec{a} = (a_1, a_2, ..., a_n)$ を出力する．典型的な例では，入力 $\vec{x}$ の次元 m は画像のように大きく例えば 1000 次元，出力 $\vec{y}$ の次元 n は 10 次元のような外界の次元が高い場合，つまり $m >> n$ の状況を考える．通常，外界の情報は高次元であるが，実際に埋め込まれている情報量は少ない．目的は入力 $\vec{x}$ の規則性を反映した表現 $\vec{a}$ を作り，高次元の入力を情報の損失なく低次元で表現することである．

2.2 入出力信号の情報量とエントロピー

今，神経回路が入力 $\vec{x}$ を受け取り，$\vec{a}$ を出力したとする．入力信号 $\vec{x}$ のエントロピーを $H(\vec{x})$，出力信号 $\vec{a}$ のエントロピーを $H(\vec{a})$ と書く．このとき，入力信号 $\vec{x}$ のエントロピー $H(\vec{x})$ と出力信号 $\vec{a}$ のエントロピー $H(\vec{a})$ から $\vec{x}$ と $\vec{a}$ の相互情報量を

$$I(\vec{a};\vec{x}) = H(\vec{a}) - H(\vec{a}|\vec{x}) \tag{1}$$

と書く．入力信号の情報が出力に損失なく伝わっている状況を考えよう．その場合，出力 $\vec{a}$ について不確実な点は一切なくなる．このため条件付きエントロピーは $H(\vec{a}|\vec{x}) = 0$ であり

$$I(\vec{a};\vec{x}) = H(\vec{a}) \tag{2}$$

となる．つまり，出力のエントロピーを最大化すれば，変換の効率を最大化できる．

出力のエントロピー $H(\vec{a})$ は各出力素子の値 a_i のエントロピー $H(a_i)$ の総和からすべての $i, j, i \neq j$ について a_i, a_j 間の相互情報量を指し引いたものであり

$$H(\vec{a}) = \sum_{i=1}^{n} H(a_i) - \sum_{i=1}^{n} I(a_i; a_{i-1}, \ldots, a_1) \tag{3}$$

と書ける．もし入力情報が損失なく出力に伝わっているなら

$$H(a) = \sum_{i=1}^{n} H(a_i) \tag{4}$$

であり，これよりも $H(\vec{a})$ を小さくすることはできない．これに応じて，$\vec{a}$ の確率分布を $p(\vec{a})$ と書くとき，同時分布は

$$p(\vec{a}) = \prod_{i=1}^{n} p(a_i) \tag{5}$$

と各要素の確率分布の掛け算で書け，各 a_i のとる値が独立であることを意味する．

2.3 要素和エントロピー最小化

要素和エントロピー最小化とは，教師なし学習の自己組織化モデルで，出力素子のとる値のエントロピー

を測り，そのエントロピーを最小にすることである．

以上をまとめて図で表してのが図 1 である．重要

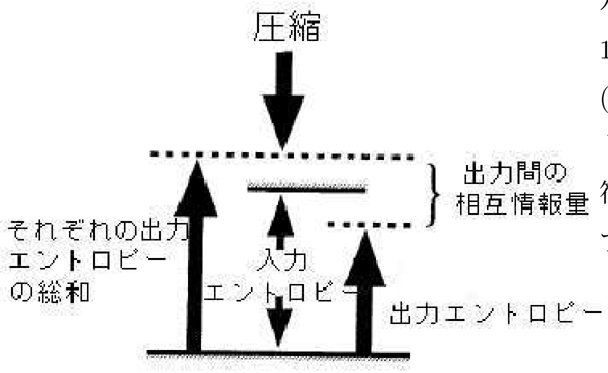

図 1: 情報処理過程におけるエントロピーのそれぞれの値

なのは，すべての出力のエントロピーの $H(\vec{a})$ の処理であり，個々のエントロピーの総和 $\sum_{i=1}^{n} H(a_i)$ がそれぞれに最小化するとき，出力間の冗長性も最小化される．以上より，2 つの一般的な符号化に関する目標をに従うことにした．

- 出力と入力間の相互情報量の最大化する．
- 1 の条件を達成したら，個々の出力のエントロピーの総和を最小化する．

3　学習アルゴリズム

ここで用いるネットワークを図 2 に示す．m 個の入力 $\vec{x} = (x_1, x_2, ..., x_m)$ は，入力と出力層からのフィードバック結合の総和からなるベクトル $\vec{r} = (r_1, r_2, ..., r_m)^{\mathrm{T}}$ とつながっている．第 1 層から出力層へのフィードバック結合は，$\vec{w} = (w_{11}, w_{12}, ..., w_{nm})^{\mathrm{T}}$ で表される．フィードフォアード結合は，フィードバック結合と同じ値を使い，逆符号にして使う．このとき出力と第 1 層の結合を表す式は，

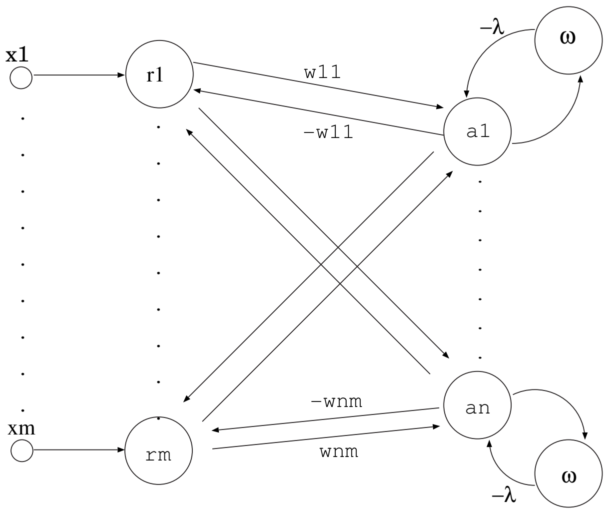

図 2: 神経回路モデル

$$\vec{r} = \vec{x} - \sum_{k=1}^{h} a_k \vec{w}_k \tag{6}$$

となる．$\vec{w}$ は，ヘブ学習によって学習され，

$$\triangle \vec{w}_i = \eta a_i \vec{r} \tag{7}$$

という規則に従い更新される．η は，学習係数であり，この値を変化させる事で学習の頻度を変化させることができる．ヘブ学習とは，結合同士が正の値なら結合を強め合い，負の値なら結合を弱め合う学習法である．

n 個の出力は，$\vec{a} = (a_1, a_2, ..., a_n)$ で表し，$\vec{a}$ は，初期値を 0 とする．このとき，次の式を使い，$\vec{a}$ の値を更新させる．

$$\triangle a_i = \mu[\vec{r}^T w_i - \lambda \omega'(a_i)] \tag{8}$$

係数 μ は，$0 < \mu < 1$ となるような値をとる．

出力 $\vec{a}$ の誤差を調節する疎性を得るための制御関数である $\omega(a)$ は，制御関数としてよく使われる式から選ぶこととした．制御関数には a^2 や $log(1 + a^2)$, $\sqrt{|a|}$ などがあるが，本研究では，計算する際に扱いやすい $\omega(a) = |a|$ を制御関数として使用した．

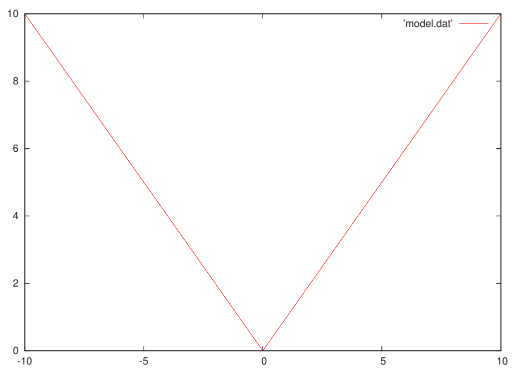

図 3: 制御関数：$\omega(a) = |a|$

4 計算機実験

4.1 2 次元のデータを入力とした場合

Weibull 関数で作成した乱数を使い，入力を作成しそのとき基底関数 w_i を観測した．Weibull 関数は，

$$p(y) \quad = \quad \alpha^{-\beta}\beta|y|^{(\beta-1)}\exp\left[-\left(\frac{|y|}{\alpha}\right)^{\beta}\right] \qquad (9)$$

で表される関数で，それぞれ $\alpha = 1$，$\beta = 2/3$ とした．入力 $\vec{x}$ は，

$$\vec{x} \quad = \quad \begin{pmatrix} \sigma_1\cos\theta_1 & \sigma_2\cos\theta_2 \\ \sigma_1\sin\theta_1 & \sigma_2\sin\theta_2 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (10)$$

で表す関数を用いる．それぞれ係数は，$\sigma_1 = 1.0$，$\sigma_2 = 2.0$，$\theta_1 = 16°$，$\theta_2 = 60°$ とし，500 回計算する毎に学習する事を 80 回繰り返した．学習に必要な係数は，$\lambda = 0.1$，$\mu = 0.00001$，$\eta = 0.005$ とした．基底関数は図では青点で，入力は赤点で表して

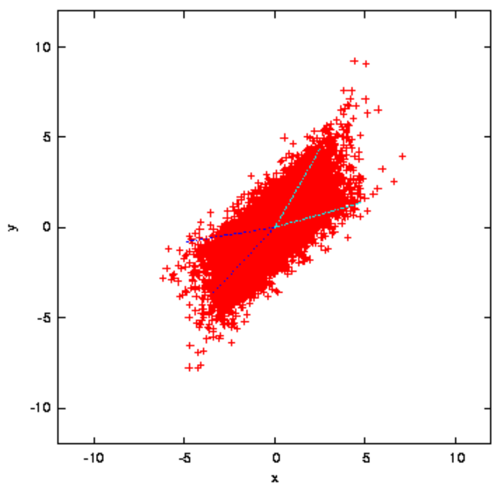

図 4: 入力：赤い点，基底：青点，回転角度：水色の点を表している

おり，図 4 から基底関数が入力を表現できている事がわかった．

次に，先ほどは，入力を 2 次元，出力も 2 次元なので上手く行ったが，出力を 3 次元に拡張させた場合どうなるか調べた．学習回数は，先ほどと同じとし，学習に必要な係数を $\sigma_1 = 0.7$，$\sigma_2 = 2.0$，$\sigma_3 = 0.2$，$\theta_1 = 18°$，$\theta_2 = 60°$，$\theta_3 = 125°$ とした．このとき，図 5 より 3 次元に拡張した場合でも，概ね入力を表現することができているのがわかる．

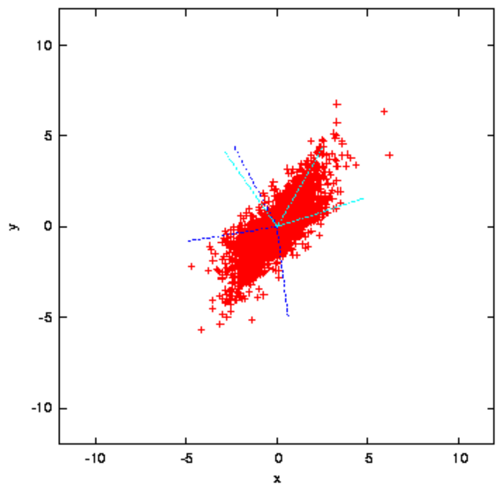

図 5: 入力：赤い点，基底：青点，回転角度：水色の点を表している

4.2 縦線と横線の組み合わさった 64 次元のデータを入力とする場合

先ほどの実験より 2 次元という低次元で表す際，概ね再現できたので，今度は入力 64 次元のデータを出力 16 次元で表すことができるか実験を行った．図 6 のような多数の 8×8 の画像に縦線や横線の組み合わせを一様乱数で作成し，それを入力とした．学習に必要な係数は，$\lambda = 0.4$，$\mu = 0.00001$，$\eta = 0.2$ と，学習は，1000 回に 1 回行うことを 10000 回繰り返した．

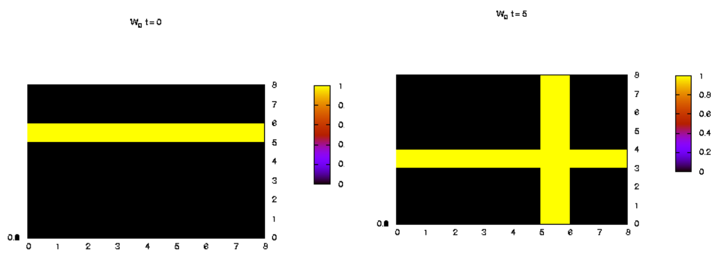

図 6: 入力画像

図 7，8 は，図 5 のような画像を入力としたときの基底関数の画像である．基底画像の一マスは，それぞれそのとき入ってきた入力画像一マスの値に対して学習させた値を示す．左側は，第 1 基底、右側は第 2 基底で，上から初期状態，2500 回目，5000 回目，8510 回目，9650 回目，10000 回目の基底画像である． 参照論文から予想した結果では，それぞれ 16 の基底画像内で縦線や横線が一本づつ表示された

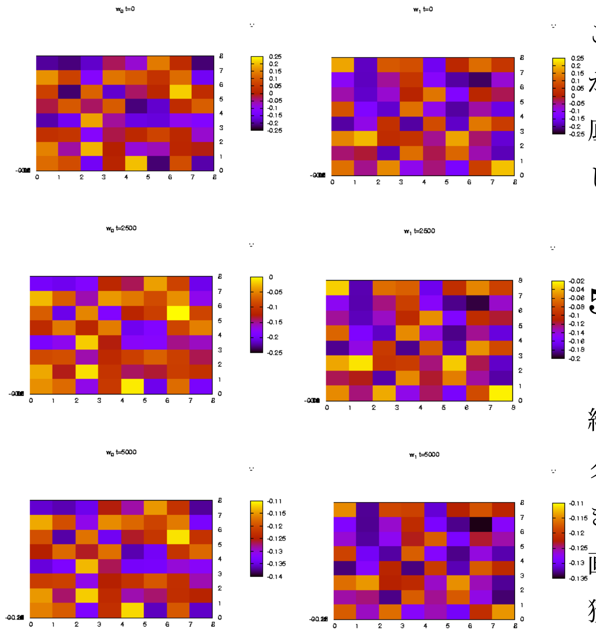

図 7: 基底画像：上から初期値，2500 回目，5000 回目

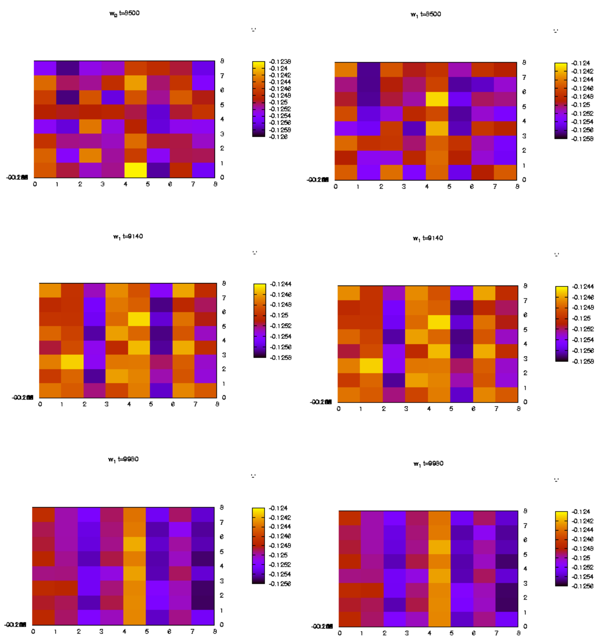

図 8: 基底画像：上から 8500 回目，9650 回目，10000 回目

画像が出るだろうと思っており，図 8 からも分かるように，9000 回目を過ぎた辺りから不完全だが徐々に縦線を見る事ができた．しかし，基底は確定することなく学習回数を重ねるごとに画像内で様々な線が代わる代わる表示された．その上，それぞれの基底が学習回数を重ねるごとに同じような画像を表示した．

5 まとめ

要素和エントロピー最小化原理により学習する神経回路モデルについて性能を評価した．2 次元のデータを入力したときの基底関数は，ほぼ入力を再現できる事が分かったが，64 次元からなる縦棒や横棒の画像データを 16 次元に収縮する場合の基底画像の獲得はできなかった．実験の過程で学習係数を色々変化させてみたが，基底関数を表現する画像に描かれる線は確定しなかった．考えられる原因としては，学習係数の値が適切でなかった事や，プログラム内の数式のミスや書き方が考えられる．

参考文献

[1] B. A. Olshausen, D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, vol. 381, pp.607-609, 1996.

[2] G Harpur, R Prager, 2000, Experiments with low-entropy neural networks, in R Baddeley, P Hancock, P Földiák (eds.) Information Theory and the Brain, Cambridge University Press, pp 84-100.

[3] 甘利 俊一：”神経回路網モデルとコネクショニズム”, 東京大学出版会（2008）.

[4] 村田 昇：”入門 独立成分分析”，東京電気大学出版局（2004）.