G-07

遺伝的アルゴリズムによるオートエンコーダの進化的学習

Evolutionary Training of Autoencoders by Genetic Algorithm

岡田 英彦
Hidehiko Okada

1. はじめに

中間層を多数に積み重ねたディープニューラルネットの研究が盛んに行われている．ディープニューラルネットの一種であるスタックトオートエンコーダ（Stacked Auto-encoders）は，オートエンコーダを多層に積み重ねたものである．オートエンコーダは階層型ニューラルネットの一種であり，その学習は一般的に誤差逆伝播法を用いて行われている．一方で，ニューラルネットの学習に進化的アルゴリズムを用いる方法も研究されており，誤差逆伝播法に対する優位性も報告されている．このため，オートエンコーダの学習においても進化的アルゴリズムの有用性が期待される．進化的アルゴリズムのなかでは遺伝的アルゴリズムが最もよく知られている．本研究では，遺伝的アルゴリズムを用いたオートエンコーダの進化的学習の試みについて報告する．

2. オートエンコーダ

オートエンコーダ（Autoencoders: AE）とは階層型ニューラルネットの一種であり，出力が入力と一致するように学習させることを特徴としている[1]．本研究における AE の構造を図 1 に示す．中間層の数は 1 であり，出力層ユニット数は入力層ユニット数と同一である．

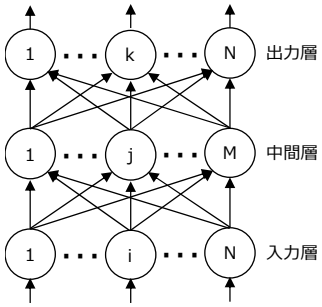

図 1: 本研究におけるオートエンコーダの構造

図 1 の AE の入出力関係を式(1)～式(5)に示す．この入出力関係は一般的な 3 階層パーセプトロンと同一である．

入力層（第 1 層）：

$$out_i^{(1)} = x_i \tag{1}$$

中間層（第 2 層）：

$$in_j^{(2)} = \theta_j^{(2)} + \sum_i w_{i,j}^{(2)}\, out_i^{(1)} \tag{2}$$

$$out_j^{(2)} = f(in_j^{(2)}) \tag{3}$$

出力層（第 3 層）：

$$in_k^{(3)} = \theta_k^{(3)} + \sum_j w_{j,k}^{(3)}\, out_j^{(2)} \tag{4}$$

$$out_j^{(3)} = f(in_k^{(3)}) \tag{5}$$

式(1)～式(5)における記号の意味は次の通りである．

x_i	i 番目の入力層ユニットへの入力値
$out_i^{(1)}$	i 番目の入力層ユニットからの出力値
$in_j^{(2)}$	j 番目の中間層ユニットへの入力値
$w_{i,j}^{(2)}$	i 番目の入力層ユニットから j 番目の中間層ユニットへの結合強度
$\theta_j^{(2)}$	j 番目の中間層ユニットのしきい値
$out_j^{(2)}$	j 番目の中間層ユニットからの出力値
$in_k^{(3)}$	k 番目の出力層ユニットへの入力値
$w_{j,k}^{(3)}$	j 番目の中間層ユニットから k 番目の出力層ユニットへの結合強度
$\theta_k^{(3)}$	k 番目の出力層ユニットのしきい値
$out_k^{(3)}$	k 番目の出力層ユニットからの出力値

式(3)および式(5)における関数fはユニット内部関数であり，本研究ではシグモイド関数$f(x) = 1/(1+e^{-x})$を用いる．

この AE の学習用データとして，N 次元実数ベクトルで表現されたデータが D 個得られているものとする．式(6)の$\boldsymbol{X}$が学習用データの集合を表しており，個々のデータ$\boldsymbol{x}_d$は式(7)のように N 個の実数$x_{d,1}, x_{d,2}, \ldots, x_{d,N}$で構成されるベクトルである．

$$\boldsymbol{X} = \{\boldsymbol{x}_d\}, d = 1,2,\ldots,\mathrm{D} \tag{6}$$

$$\boldsymbol{x}_d = (x_{d,1}, x_{d,2}, \ldots, x_{d,N}) \tag{7}$$

$x_{d,1}, x_{d,2}, \ldots, x_{d,N}$のそれぞれの値が入力層ユニット 1～N にそれぞれ入力されたとき，この入力に対する出力層ユニット 1～N の出力がそれぞれ$x_{d,1}, x_{d,2}, \ldots, x_{d,N}$とできるだけ一致するように，結合強度$w_{i,j}^{(2)}, w_{j,k}^{(3)}$およびしきい値$\theta_j^{(2)}, \theta_k^{(3)}$を学習させる．つまり，AE の学習は出力の N 次元ベクトルの値が入力の N 次元ベクトルの値を復元するように行われる．したがって，誤差逆伝搬法（Back Propagation: BP）[2]を用いて AE を学習させる場合には，入力した$x_{d,i}$の値と出力された$out_i^{(3)}$の値のずれを誤差関数として定式化すればよい．この例を式(8), 式(9)に示す．個々のデータ$\boldsymbol{x}_d$に対する誤差の 1 次元あたりの平均がe_dであり，その誤差のデータ平均が e である．式(9)において，$x_{d,i}$は

京都産業大学, Kyoto Sangyo University

$0.0 \leqq x_{d,i} \leqq 1.0$ を満たす実数値であり，$out_i^{(3)}$ とeの値は $0.0 < out_i^{(3)}, e < 1.0$ を満たす実数値である．$e \cong 0.0$のとき，AE の出力はすべての$\boldsymbol{x}_d$の入力をほぼ完全に復元していることになる．

$$e = \frac{1}{D}\sum_{d=1}^{D} e_d \tag{8}$$

$$e_d = \frac{1}{N}\sum_{i=1}^{N}(out_i^{(3)} - x_{d,i})^2 \tag{9}$$

3. 遺伝的アルゴリズムによる進化的学習

本研究では，前章に記載の AE の学習に BP ではなく遺伝的アルゴリズム（GA）[3,4]を適用する． BP は単点探索法であるのに対して GA は多点探索法である．この違いから，誤差関数 e の最小化問題の解法として両者を比較すると，GA は BP よりも広域探索能力が高く，不良な局所最適解を回避してよりよい準最適解を発見しやすいと期待できる．ニューラルネットの学習に BP を用いた場合より進化的アルゴリズムを用いた場合のほうが良好な結果が得られた例もこれまでに報告されている[5].

進化的アルゴリズムを用いたニューラルネットの進化的学習はニューロエボリューション（NE）と呼ばれている[6]．本研究の試みはAEを対象としたNEと言える．なお，NE は大きく 2 種類に分けることができ，(A)ニューラルネットの構造（階層数やニューロン数など）は事前に決定しておいて結合強度としきい値の値だけを学習させる手法と，(B)ニューラルネットの構造自体も学習によって最適化する手法がある．本研究で用いる NE は前者(A)の手法である．図 1 の構造を持つ AE には，結合強度がMN + MN = 2MN個，しきい値がM + N個含まれており，合わせると2MN + M + N個含まれている．これを2MN + M + N次元の実数ベクトルとして表現し，GA における個体の遺伝子形（genotype）として扱う．個体の表現形（phenotype）は図 1 の AE である．つまり，AE に含まれる合計2MN + M + N個の実数パラメータに対して GA による遺伝的操作を適用し，それらの実数パラメータの最適化を図る．各個体の評価値には式(9)に示した誤差関数を用い，式(6)に示した学習用データの集合$\boldsymbol{X}$を用いて評価値を求める．式(9)の誤差関数の値は小さいほど良いため，GA においては評価値が小さい個体ほどより良い個体と評価する．本研究における GA の手続きは次の通りである．

Step1: 初期化
Step2: 個体の評価
Step3: 終了判定
Step4: 次世代集団生成
Step5: Step2 に戻る

Step1 では，あらかじめ決められた数の個体のそれぞれについて genotype の値（2MN + M + N次元実数ベクトルの値）を初期化する．本研究では genotype の要素は AE の結合強度やしきい値であることを踏まえて，各要素の定義域をあらかじめ決めておく．

Step2 では，Step1 や Step5 で生成された新たな個体の適合度を評価する．これにより，集団に含まれるすべての個体の適合度が定まる．

Step3 では，あらかじめ決められた基準に基づいて終了判定を行う．本研究では世代数の最大値を決めておき，世代数がその値まで到達した時点で終了とした．

Step4 では，現世代の個体集団を親として用い，次世代の子個体の集団を生成する．Step4 の手続きを以下に示す．

Step4-1: エリート保存
Step4-2: 親 2 個体選択
Step4-3: 交叉
Step4-4: 突然変異
Step4-5: 終了判定
Step4-6: Step4-2 へ戻る

Step4-1 では，現世代の個体集団のうち評価値が優れている（ランク上位の）個体を一定個数，次世代の子個体集団へとコピーする．エリートとして保存する個体の数は事前に決めておく（L とする）．集団内の個体数（集団サイズ）を P とすると，次世代の P 個体のうち L 個を現世代からのコピーによって獲得し，残りの(P-L)個の個体を交叉と突然変異によって生成する．

Step4-2 では，現世代の P 個体から，交叉に用いる親個体を 2 個体選択する．その選択方法として本研究ではトーナメント選択を用いる．

Step4-3 では，Step4-2 で選択された 2 個体の親を交叉させて子個体を生成する．本研究では交叉手法としてブレンド交叉法を用いる．ブレンド交叉法では 1 回の交叉で出来る子個体は 1 つである．

Step4-4 では，Step4-3 で出来た子個体に突然変異を適用する．具体的には，genotype のベクトルに含まれる2MN + M + N個の要素のそれぞれを一定の確率でランダムな値に書き換える．このランダムな値の決め方は，その要素の値を Step1 において初期化した際の方法と同一である．

Step4-5 では，次世代の子個体が P 個すべて決定されたか判定し，Yes なら Step4 を終了する．

4. 実験

学習用データの集合$\boldsymbol{X}$として手書き数字のデータを用いて実験を行った．具体的には，UC Irvine Machine Learning Repository に掲載されている Optical Recognition of Handwritten Digits Data Set[1] を用いた．初期的な実験として，optdigits.tra ファイルに記録されたデータのなかから数字「0」～「9」の手書きデータをそれぞれ 1 つだけ抽出し，計 10 個のデータを用いた．各データは 8×8＝64 個の要素で構成されており，各要素の値は 0～16 の整数である．本実験ではこれらの要素の値をすべて 16 で割り，区間[0.0, 1.0]内の実数値に変換して用いた．用いた 10 個のデータを図 2 に示す．

データの次元数が 64 のため，AE の入力層および出力層のユニット数（図 1 の N の値）も 64 である．中間層ユニットの数は，4^2=16，5^2=25，6^2=36 を試行錯誤的に試した結果，36 個の場合に誤差関数の値が十分に小さくなるまで学習が進んだ．中間層ユニット数（図 1 の M の値）が 36 個のとき，AE に含まれる結合強度およびしきい値の総数は$2 * 36 * 64 + 32 + 64 = 4708$個である．したがって，GA における genotype は 4708 次元実数ベクトルとなる．

本実験における GA の設定を以下に示す．これらの設定は，著者のこれまでの経験と，試行錯誤的な予備実験に基づいて決定された．

[1] http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

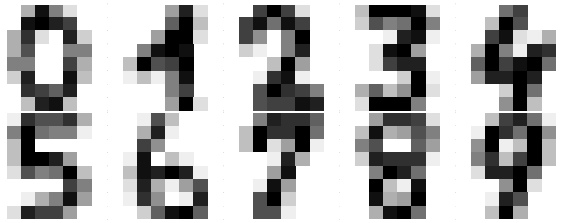

図 2: 実験で用いた手書き数字のデータ

・genotype の各要素の定義域： [-5.0, 5.0]
・個体数：100
・世代数：100,000
・エリート保存数：2
・トーナメントサイズ：10
・交叉方法：ブレンド交叉（α = 0.5）
・突然変異確率：1/4708

世代経過に伴う誤差 e の変化は図 3 の通りであった．このグラフは各世代において 100 個体のうち最良個体の誤差 e の値を示している．図 3 の 100 個体の genotype の値をすべて定義域[-5.0,5.0]内でランダムに初期化した時点では，誤差 e の値は 100 個体内の最良で 0.181 であったが，最終世代の時点ではその値は 0.00937 まで減少した．この値が 1 ピクセルあたりの平均誤差であり，その値が 0.0～1.0 であることから，初期化時点では 1 ピクセルあたり 18.1%の誤差があったのに対して，最終世代では 1 ピクセルあたり 0.973%の誤差まで減少したと言える．この誤差を得た最良個体の AE による出力値を図 2 と同様に 8×8 ピクセルで表示したものを図 4 に示す．図 4 を図 2 と比較すると，数字 0・1・8・9 の画像イメージにある程度の差異が見てとれるが，概ねよく入力を復元できていることがわかる．

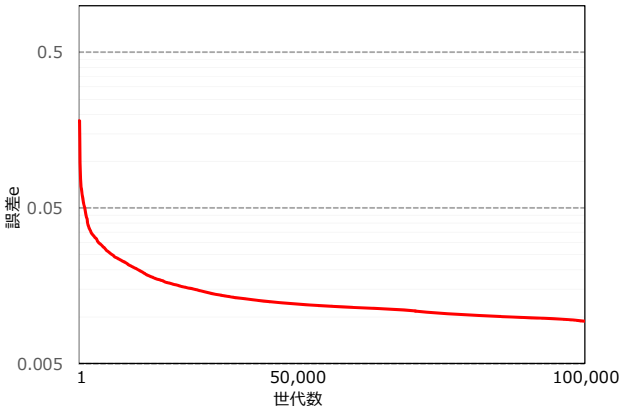

図 3: 世代経過に伴う誤差の変化

この誤差最良の AE が図 2 の入力に対して図 4 を出力した際，データ 10 個のそれぞれについて中間層ユニット 36 個が出力した実数値を視覚的に表現したものを図 5 に示す．一方，図 2 のデータを図 5 と同じ形式で表現したものを図 6 に示す．AE の出力が入力を良い精度で復元できた場合，その AE の中間層からの M 次元出力ベクトルは，N 次元入力ベクトルとして表現された学習用データの情報をなるべく損なわず別の形式へ変換したものと言える．また，N > M の場合にはこの変換は N 次元実数ベクトルから M 次元

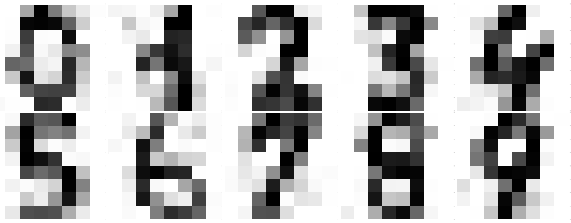

図 4: 誤差最良の AE による出力値

図 5: 誤差最良の AE における中間層からの 36 次元出力ベクトル

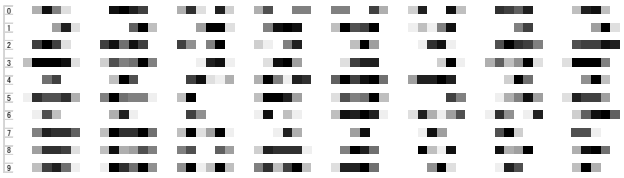

図 6: 図 5 と同じ形式で表した 10 個の学習用データ（64 次元入力ベクトル）

実数ベクトルへの圧縮にあたると言える．図 5 と図 6 を比べると，図 6 は 10 種類の数字の間で縦に共通して白い部分がよく見られるが，図 5 はそのような部分がより少ないことがわかる．つまり，学習用データ（64 次元実数入力ベクトル）には含まれていた無駄な情報を減らしつつ，手書き数字 0～9 の復元に必要な情報が抽出されるように 36 次元の実数ベクトルへと圧縮変換していることがわかる．

なお，前記の誤差最良の AE に対して，学習に用いていいない 0～9 のデータを入力したところ，そのデータは十分には復元されなかった．つまり，今回の実験では学習後の AE の汎化能力は低いことがわかった．この最大の理由は学習用データ数が少ないため（0～9 の 10 種類につきそれぞれ 1 個だけのため）と考えられる．

5. まとめ

遺伝的アルゴリズムを用いたオートエンコーダの進化的学習を試みた．手書き数字データを用いた実験により，学習に用いたデータは良好な精度で復元でき，中間層からの 36 次元出力ベクトルが元の 64 次元入力ベクトルの情報を適切に圧縮していることがわかった．しかし，今回の実験では学習後の AE は汎化能力が低いこともわかった．学習用データ数を増加させた実験が今後の課題の 1 つである．

参考文献

[1] G.E. Hinton and R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, 313(5786), 504-507, 2006.

[2] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, in *Neurocomputing: Foundations of Research*, J.A. Anderson and E. Rosenfeld (Eds.), MIT Press, 696-699, 1988.

[3] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Professional, 1989.

[4] L.J. Eshelman and J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in D.L. Whitley (ed), *Foundation of Genetic Algorithms 2*, 187-202, 1993.

[5] X. Yao, Evolving artificial neural networks, *Proc. of the IEEE*, 87(9), 1423-1447, 1999.

[6] D. Floreano, P. Durr and C. Mattiussi, Neuroevolution: from architectures to learning, *Evolutionary Intelligence*, 1(1), 47-62, 2008.