ISSN 2186-5647

2-38

結合重みを自己修正するニューラルネットワーク

日大生産工 (院)　○井坂 太一　　日大生産工　松田 聖

1　背景

ある目的関数を最小化 (又は最大化) する最適化問題において, 目的関数が線形の場合は単体法のように有効なアルゴリズムがあるのに対して, 目的関数が非線形又は不連続の場合においては, 決定的なアルゴリズムが存在しない. また, 問題を解くにあたり, どのような目的関数を設定すれば良いのかわからない場合もある. そこで本研究では, 人工知能の分野の一つであるニューラルネットワーク理論を活用して, 最適化問題の解を発見的に求める手法を提案する.

2　目的

本研究は, ニューラルネットワーク (以下 NN) から最適化問題の解を得ることを目的とする. そのためには, NN の出力が最適化問題の解となるように結合重みを修正することが要される. 一般的な NN の学習規則は, 入力に対しての実際の出力と正しい出力 (教師信号) との誤差を用いて結合重みを修正していく手法と. 最適化問題の目的関数が 2 次形式に出来る場合は, エネルギー関数なる量と対応させて最適化問題の解を得ることができる 2 つの手法がある. しかし, 実際の最適化問題に対しては, 正しい答えが分からないことや, 目的関数を 2 次形式にすることが出来ない場合が多い. 本来, NN はシステムの最適化に優れた手法なので, 従来ある学習規則を改良すれば最適化問題の解を得るようにすることが出来ると考えた. 本研究では, 目的関数が不明瞭な場合でも, 時間変化に対して状態の適合度は比較できると仮定し, その変化に応じて結合重みを修正することで, 最適化問題の解を得ることを目的とした. その学習規則を次章で定式化する.

3　提案手法

3.1　ニューラルネットワークの構造

3.1.1　ニューラルネットワークの概要

一般の NN の構造を簡単に説明する. NN は複数の素子から構成され, 各素子は, 出力値を持つ. 出力値は各素子の出力の重み付き総和をシグモイド関数なる関数の引数にして, 出力値を決定する. 各素子は重み (結合重み) を隔てて結合されていると考え, 素子同士の結合は片方のみ場合や双方向の場合がある. 結合重みの変化が素子の出力に影響を及ぼすため最適な結合重みを与えることで NN は目的の出力を得ることとなる. 一般に最適な結合重みはどのように設定すれば良いのか分からないことが多いので, 結合重みをアルゴリズムにより徐々に修正していく. そのアルゴリズムを学習規則と呼ぶ. すでに決まっている入出力関係を学習する場合は, 一方向のみの結合された素子を階層に分けて入出力関係を学習する. 双方向に結合された素子は主に TSP 問題やナップザック問題などの目的関数を 2 次形式にすることができる最適化問題を解く際に使用される. 本研究では, 各素子は双方向に結合されたものを想定し, 一部は一方向のみで結合された素子を考えた.

3.1.2　研究で用いるニューラルネットワ-ク

本研究で用いる NN の素子は内部層と出力層に分かれる. 内部層の素子同士は, 相互に結合され結合重みは対称とする. 内部層と出力層の素子は, 内部層から出力層への一方向的な結合のみ存在し, 出力層同士の結合は存在しない. つまり, 出力層の素子は内部層の素子の出力状態を取りだす役割をする. また, 素子の出力は確率的に 1 と-1 を出力する. i 番目の素子が 1 を出力する確率 $p(s_i = +1)$ と, -1 を出力する確率 $p(s_i = -1)$ はシグモイド関数 $g(u_i)$(fig. 1) を用いて次式で表す.

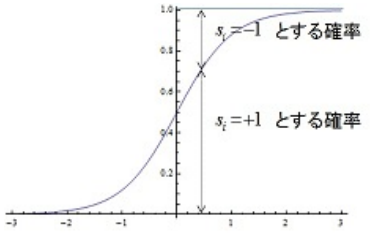

fig.1 シグモイド関数

$$
\begin{aligned}
p(s_i = +1) &= g(u_i) \\
p(s_i = -1) &= 1 - g(u_i) \\
u_i &= \sum_i^n s_j w_{i,j}
\end{aligned}
$$

ここで, s_i は, i 番目の素子の出力, $w_{i,j}$ は i 番目の素子と j 番目の素子の結合重み, u_i は, 各素子との重み付き総和を表す. また, u_i を内部状態と呼ぶ.

シグモイド関数は以下で示す連続関数で, 内部状態の値が大きくなると+1 を出力する確率が高くなり, 内部状態の値が小さいと-1 を出力する確率が高くなる.

3.1.3 提案する学習規則

本研究での結合重みを修正する方法を示す. 想定する問題は目的関数を最小化する最適化問題とし, 出力層の素子の出力は, 最適化問題の解とする. まず, NN の各素子と結合重みをランダムに初期化する. その後, すべての内部層の素子の値を一度更新した後, 出力層の素子を一つランダムに選択して. その素子の値を更新する. 出力層の素子の値が変化するので (変化しない場合もある), その状態間で目的関数の値を比較できる. ここでもし目的関数の値が減少したら近視的ではあるが状態がより良くなったといえる. また, 目的関数の値が増加したら, 状態が悪くなったといえる. ここで, 状態が良くなったときに+1 を, 悪くなったときに-1 をとるスカラー情報を考える. 内部層の素子を固定した後に, 1 回目に出力層の素子を更新した時刻を t, 次に出力層の素子を更新した時刻を $t+1$, そのときの目的関数の値をそれぞれ $F(t)$, $F(t+1)$ として以下にスカラー情報の値を定義する.

$$
r = \begin{cases} +1 & (F(t) - F(t+1) > 0 \text{ のとき}) \\ -1 & (F(t) - F(t+1) < 0 \text{ のとき}) \end{cases}
$$

この r を報酬と呼び, この報酬の値を頼りに結合重みを更新していく. この報酬値から現状の出力層の k 番目の素子の取るべき値が分かる. 時刻 $t+1$ に出力層の k 番目の素子の値 s_k を出力したときに, 報酬 $r = +1$ を得たとする. そのとき, k 番目の素子の値はそのまま保持することが望ましいといえるので, s_k を教師信号として良いと考える. 逆に, 時刻 $t+1$ に出力層の k 番目の素子の値 s_k を出力したときに, 報酬 $r = -1$ を得たとする. そのとき, k 番目の素子の値は s_k と逆の値すなわち $-s_k$ を教師信号として良いとする. しかし, この教師信号は時刻 t から $t+1$ 間の極めて近視的にみる教師信号であるので擬似的な教師信号と呼び以下に定式化する.

$$
T_k = \begin{cases} +1 & (r = +1 \text{ のとき}) \\ -1 & (r = -1 \text{ のとき}) \end{cases}
$$

この擬似的な教師信号 T_k の値と s_k との 2 乗誤差 E を次式で定式化する.

$$
E = \frac{1}{2}(T_k - \langle s_k \rangle)^2
$$

$\langle s_k \rangle$ は期待値 (平均値) を表し, 確率を用いて次式で計算できる.

$$
\begin{aligned}
\langle s_k \rangle &= +1p(s_k = +1) - 1p(s_k = -1) \\
&= +1g(u_k) - 1(1 - g(u_k)) \\
&= 2g(u_k) - 1
\end{aligned}
$$

結合重みの修正規則は, E を最小化するように最急降下法により結合重みを修正していく.

$$
\begin{aligned}
\Delta w^o_{i,k} &= -\varepsilon \frac{\partial E}{\partial w^o_{i,k}} \\
&= -\varepsilon \frac{\partial E}{\partial \langle s_k \rangle} \frac{\partial \langle s_k \rangle}{\partial g(u_k)} \frac{\partial g(u_k)}{\partial u_k} \frac{\partial u_k}{\partial w^o_{i,k}} \\
&= \varepsilon (T_k - \langle s_k \rangle) g'(u_k) s_i
\end{aligned}
$$

$w^o_{i,k}$ は内部層の i 番目の素子と出力層の k 番目の素子間の結合重みとする.

次に内部層のニューロン i と j 同士の結合重み $w^I_{i,j}$ を修正する規則を考える. 今, 出力層のニューロン k が出力変化したとして, k に接続されている内部層のニューロン i とそれに接続されているニューロン j の結合重みを $w^I_{i,j}$ とする. 先ほどの最急降下法と同様に,

$$
E = \frac{1}{2}(T_k - \langle s_k \rangle)^2
$$

として $w^I_{i,j}$ で勾配をとると,

$$
\begin{aligned}
\Delta w^I_{i,j} &= -\varepsilon \frac{\partial E}{\partial w^I_{i,j}} \\
&= -\varepsilon \frac{\partial E}{\partial \langle s_k \rangle} \frac{\partial \langle s_k \rangle}{\partial g(u_k)} \frac{\partial g(u_k)}{\partial u_k} \frac{\partial u_k}{\partial s_j} \frac{\partial s_j}{\partial w^I_{i,j}}
\end{aligned}
$$

しかし, $\frac{\partial u_k}{\partial s_j}$ は微分出来ないため, $s_j \simeq \langle s_j \rangle$ で近似する. 近似により, $\Delta w^I_{i,j}$ は次式で計算できる.

$$
\begin{aligned}
\Delta w^I_{i,j} &= -\varepsilon \frac{\partial E}{\partial w^I_{i,j}} \\
&= -\varepsilon \frac{\partial E}{\partial \langle s_k \rangle} \frac{\partial \langle s_k \rangle}{\partial g(u_k)} \frac{\partial g(u_k)}{\partial u_k} \frac{\partial u_k}{\partial \langle s_j \rangle} \\
&\quad \frac{\partial \langle s_j \rangle}{\partial g(u_j)} \frac{\partial g(u_j)}{\partial u_j} \frac{\partial u_j}{\partial w^I_{i,j}} \\
&= \varepsilon (T_k - \langle s_k \rangle) g'(u_k) g'(u_j) w^o_{i,k} s_i
\end{aligned}
$$

ただし, ε は, 学習率と呼ばれる任意の定数とする. 微分の結果から定数が計算されるため, 学習規則の最後の項の ε は, 定数をまとめて改めて, ε としている. 以上の学習規則を用いることで, 結合重みを修正し, 最適化問題の解を学習させる.

4 動作規則

前節で示した学習規則は, 最急降下法を用いている為に, 誤差 E が 0 に収束するように結合重みを修正しているが, 直近の目的関数の値を比較して作られた擬似的な教師信号 T_k を利用いているために確実に 0 に収束するとはいえない. そこで, 学習規則により, 素子の出力にどのような影響が起きるか確認してみる. 簡単のために内部層と出力層の素子はそれぞれ 1 つとして考える. 内部層の素子を i として, 出力層の素子は k とする. 今, 内部層の素子の出力 s_i を固定し, 時刻 t として, 出力層の素子 k の出力が $s_k = -1$ だったとする. その後, 内部層の素子の出力値を固定したままで, 時刻 $t+1$ として, 出力層の素子 k の出力が $s_k = +1$ と計算されたとする. 素子の出力は確率的なために, そのようなことが起こりうる. そのとき, 報酬 $r = +1$ だったとすると, 前節の更新規則 $\Delta w^o_{i,k}$ から重みの修正が行われる. そのときの内部状態 u_k に着目すると,

$$\begin{aligned} u_k(t) &= s_i w_{i,k} \\ u_k(t+1) &= s_i(w_{i,k} + \Delta w^o_{i,k}) \end{aligned}$$

$u_k(t+1)$ に $\Delta w^o_{i,k}$ と $T_k = +1$, を代入すると

$$\begin{aligned} u_k(t+1) &= s_i(w_{i,j} + \varepsilon(+1 - \langle s_k \rangle) g'(u_k) s_i) \\ &= s_i w_{i,j} + \varepsilon(+1 - \langle s_k \rangle) g'(u_k) s_i^2 \\ &= u_k(t) + \varepsilon(+1 - \langle s_k \rangle) g'(u_k) s_i^2 \end{aligned}$$

$\langle s_k \rangle < 1$ より, $+1 - \langle s_k \rangle > 0$, また, $g'(u_k) > 0$ から,

$$\varepsilon(+1 - \langle s_k \rangle) g'(u_k) s_i^2 > 0$$

よって, 時刻 t から $t+1$ 間で内部状態 u_k の値は増加して, $s_k = +1$ を出力する確率が増加していることが分かる. 同様に, $r = -1$ のときは $s_k = +1$ を出力する確率が減少することが確かめられる. また, 以上の結果と時刻 t から $t+1$ 間で s_k が 1 から -1 に出力変化したときの確率を計算した結果を以下に示す.

この結果より, 提案する学習規則は, $r = +1$ のときは現状の s_k を出力する確率を増加させ, $r = -1$ のときは, 現状の s_k と逆の値を出力する確率を増加させる.

5 報酬の収束性

次に報酬の収束性について考える. 最適化問題を解く上での目的は, 目的関数を最小化 (最大化) する状態を求めることだが, 今回の提案手法ではより多くの報酬を得ることが目的といえる (それが目的関数の最小化に繋がる). よって, 各時刻での報酬の値を加えていきそれが最大化されることを示したい. しかし, 和の最大化を求めるのは理論的に困難なため, 報酬の期待値 (平均値) の最大化を考える. 時刻 t での出力層のすべての素子の状態を $S(t)$ とし, 時刻 $t+1$ での出力層のすべての素子の状態を $S(t+1)$ とすると, 報酬 r は $S(t)$ と $S(t+1)$ により決まるので, 期待値の定義から,

	s_k の値の変化	
	-1 から $+1$	$+1$ から -1
$r = +1$	u_k が増加	u_k が減少
$r = -1$	u_k が減少	u_k が増加

表 1 学習規則

$$\langle r \rangle = \sum_t^{\infty} r(S_t, S_{t+1})\ p(S_t, S_{t+1})$$

ただし, $p(S_t, S_{t+1})$ は, 同時確率を示し, $S_t = S(t)$ とした. 今, 内部状態の素子を固定したとすると,

$$p(S_t, S_{t+1}) = p(S_t) p(S_{t+1})$$

とできる. この $\langle r \rangle$ に対して, $\mathrm{w}^o_{i,k}$での勾配を計算すると次式が得られる.
ここで, $\mathrm{r}=\mathrm{r}(\mathrm{S}_t, S_{t+1})$ としている.

$$\begin{aligned} \frac{\partial \langle r \rangle}{\partial w^o_{i,k}} &= \frac{\partial}{\partial w^o_{i,k}} \sum_t^{\infty} r\ p(S_t, S_{t+1}) \\ &= \frac{\partial}{\partial w^o_{i,k}} \sum_t^{\infty} r\ p(S_t) p(S_{t+1}) \\ &= \sum_t^{\infty} r \left(p(S_{t+1}) \frac{\partial p(S_t)}{\partial w^o_{i,k}} + p(S_t) \frac{\partial p(S_{t+1})}{\partial w^o_{i,k}} \right) \end{aligned}$$

また, $p(S_t)$ は確率の積で表すことができるので,

$$p(S_t) = \prod_{k=1}^{n} \begin{pmatrix} g(u_k) & \text{if } s_k = +1 \\ 1 - g(u_k) & \text{if } s_k = -1 \end{pmatrix}$$

n は素子の個数とする.

$$\begin{aligned} \frac{\partial p(S_t)}{\partial w^o_{i,k}} &= \frac{\partial p(S_t)}{\partial g(u_i)} \frac{\partial g(u_i)}{\partial u_i} \frac{\partial u_i}{\partial w^o_{i,k}} \\ &= \prod_{\substack{k=1 \\ k \neq i}}^{n} \begin{pmatrix} g(u_k) & \text{if } s_k = +1 \\ 1 - g(u_k) & \text{if } s_k = -1 \end{pmatrix} \cdot \begin{pmatrix} g'(u_i) s_i & \text{if } s_i = +1 \\ -g'(u_k) s_i & \text{if } s_i = -1 \end{pmatrix} \\ &= \prod_{k=1}^{n} \begin{pmatrix} g'(u_i) s_i & \text{if } s_i = +1 \\ -g'(u_k) s_i & \text{if } s_i = -1 \end{pmatrix} \cdot \begin{pmatrix} \frac{g'(u_i) s_i}{g(u_i)} & \text{if } s_i = +1 \\ \frac{g'(u_i) s_i}{1 - g(u_i)} & \text{if } s_i = -1 \end{pmatrix} \\ &= \prod_{k=1}^{n} \begin{pmatrix} g'(u_i) s_i & \text{if } s_i = +1 \\ -g'(u_k) s_i & \text{if } s_i = -1 \end{pmatrix} \cdot \frac{(s_i - \langle s_i \rangle) s_k}{2T} \\ &= p(S_t) \cdot \frac{(s_i - \langle s_i \rangle) s_k}{2T} \end{aligned}$$

同様に,

$$\frac{\partial p(S_{t+1})}{\partial w^o_{i,k}} = p(S_{t+1}) \cdot \frac{(s_i - \langle s_i \rangle) s_k}{2T}$$

以上の計算から,

$$\begin{aligned} \frac{\partial \langle r \rangle}{\partial w^o_{i,k}} &= \sum_t^{\infty} r \; p(S_{t+1}) p(S_t) \cdot \frac{(s_i - \langle s_i \rangle) s_k}{T} \\ &= \sum_t^{\infty} r \; \cdot \frac{(s_i - \langle s_i \rangle) s_k}{T} \end{aligned}$$

ただし, T はシグモイド関数の傾きを表す定数とする. この結果から, $r = +1$ のとき $T_k = s_k$ となるので, 提案する学習規則で, 重みを更新し続け, $(s_i - \langle s_i \rangle) s_k$ が 0 に向かうならば, 報酬は極大値に向かっていくと考える.

6 実験方法

実験方法について順を追って示す. なお, 一度に更新する素子は 1 つのみとする.

STEP1: 目的関数を設定し, 内部層の素子と出力層の素子の数を設定する.

STEP2: 各素子の出力値と結合重みをランダムに設定する.

STEP3: 内部層のすべての素子の出力を一度更新する.

STEP4: ランダムに選択した出力層の素子 1 つの出力を更新する.

STEP5: 内部層の素子と出力層の素子間の結合重みを更新する.

STEP6: 内部層の素子同士間の結合重みを更新する.

STEP3 から STEP6 を 1 サイクルとして, サイクルを繰り返すことで, 結合重みを更新していく.

初めに, 学習規則の有効性を確かめるために, 目的関数として下に凸の 2 次関数 $y = x^2$ の最小値を求める最適化問題を解いてみる. なお実験環境として, 内部層の素子の数を 8, 出力層の素子の数を 8 に設定した. 素子の出力は $+1$ か -1 のため, -1 を 0 として, 8 個の素子の出力を 8 桁の 2 進数として表現した. ただし, そのうちの一桁は出力がプラスかマイナスを表現しているとする.

7 実験結果

実験結果を示した. (fig. 2) 上記の実験方法で 1 万サイクル実験したところ, 2 次関数の最小値を得ることが出来た. (fig.2) から, 目的関数が 0 に収束し, そのとき $x = 0$ となっていることが確認できる. なお, グラフは, 縦軸が目的関数の値, 横軸がサイクル数を表している. また, 100 サイクルごとの平均を載せている. x の値は収束していないようにもみえるが, 素子の出力を確率的に動作している為に, サイクルを重ねても高い出力が計算されてしまう. 実際は, 出力変化の頻度がそれほど多くなく, ほとんど 0 に収束している.

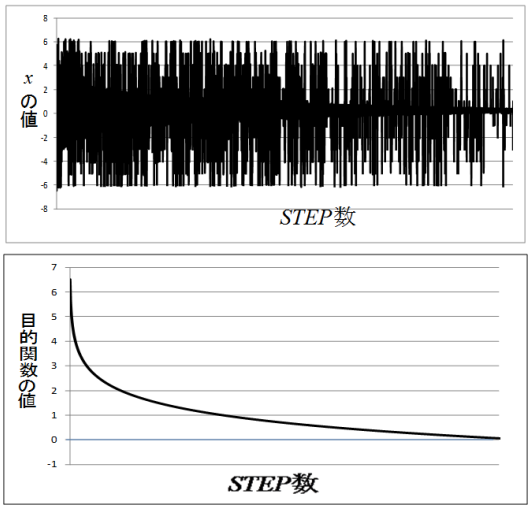

fig.2 2 次関数の収束

8 結論

提案する学習規則で最適化問題の解が得られることが分かった. また, 学習規則に目的関数の形状を設定していないことから, 不連続関数にも適応できることが期待できる. 今後は, 複数の極小値がある問題や最適化問題のベンチマーク関数において, アルゴリズムの有効性を確かめたい.

参考文献

[1] 熊沢　逸夫, **“学習とニューラルネットワーク”**, 森北出版, 1998.

[2] 小熊　崇, **“強化学習ととりいれたボルツマン・マシンによる非線形計画問題の大域的最適解の解法”**, 数理解析研究録 1349 巻, 2004.

[3] 新井　利幸, **“情報幾何学に基づく平均場ボルツマンマシン学習則の理論的検討’**, 電子情報通信学会論文誌, 1998.