ニューラルネットを用いた逐次的関数近似手法の提案

福井　健一　　（沼田　一道　助教授）

1．はじめに

関数形を予測し，新たな入力に対する未知の出力を予測する「関数近似」は様々な手法によって行われている．用いられるデータにも様々なものがあり，現実的なデータのなかには時間経過によって逐次的に与えられ，予測する関数の形が不鮮明であるものが考えられる．

解析的な関数近似手法を用いる場合，得られるデータが予め決められた関数の形をしているのであれば，精度の良い関数近似が行えることが知られているが，与えられるデータがどのような関数形をしているのかという情報が得られないデータであると解析は困難なものになってしまうため，機械学習を用いた関数近似手法が要求される．また，機械学習のなかでも逐次学習の概念を持たない関数近似手法では，与えられた全てのデータを用いて解析することで関数近似を行うため，逐次的にデータが与えられた場合には，随時新しいデータを含めた全てのデータに対して解析することが必要となってしまい，データのサイズが大きいと解析に必要な計算量は膨大なものになってしまう．そのため逐次的に学習することが効率的であると考えられる．

そこで本研究では，逐次的に与えられる不鮮明なデータを学習する「逐次学習手法」をニューラルネットの構造を用いて構築し，その効果を実験によって検討する．

2．ニューラルネット(NN)

人工ニューロン(artificial neuron)(図１)は，生物の神経系であるニューロンをモデル化したものである．このモデルは，他の人工ニューロンからの出力 x_j が人工ニューロン間の結合強度 w_{ij} により重み付けされ，人工ニューロンに達し，それがある閾値 θ (h[] は閾値関数を示す)を越えるとニューロンが発火するように，出力 y_i を生じさせることを示している．この人工ニューロンをネットワーク状に接続したものがニューラルネットであり，階層型ネットワーク/相互型ネットワークやフィードバックを持つ/持たないネットワークなど様々な型のニューラルネットが提唱されている．本研究では順方向のみのフィードバックを持たない階層型ニューラルネット（図２）の構造を用いて逐次学習手法の実装を目指す．ここで挙げる学習とは教師データと出力との差の総和 E (総誤差)を最小にするような重み w を求めることであり，具体的には，適当な重み w から出発して，微少量 w が変化したときに誤差 E が変化する量(勾配ベクトル) ∇E を求めて E の減少する方向に w を一定量変化させることを繰り返す．本研究では誤差の小さいシステムへと改善させていく(ステップ幅固定の最急降下法)という学習法を用いる．また，学習の過程は全てこの学習法によるものを改良することで実装した．以下に具体的な学習法を示す．

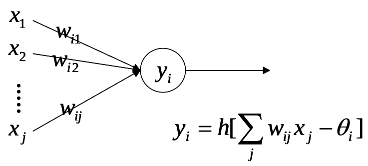

図１：人工ニューロン
(artificial neuron)

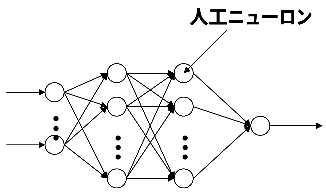

図２：階層型ニューラルネット

3．学習法

第 P 入力に対する第 $S-1$ 層の第 j ニューロンの出力を y_{jp}

第　S　層の第 i ニューロンの出力を x_{ip} としたとき，

ニューロン間の入出力関係を以下のように与える．

$$x_{ip} = h_i(z_{ip}) \qquad z_{ip} = \sum_j w_{ij} y_{jp} \tag{1}$$

(w_{ij} ：$S-1$ 層の第 j ニューロンと S 層の第 i ニューロン間の結合強度　，　$h[]$ ：閾値関数)

このとき第 P 入力に対する誤差関数 $E_p(w)$ は，第 P 入力の教師信号を d_{ip} としたとき，誤差関数を $E\ (w) = \sum_p E_p(w)$ であらわし，総誤差関数を $E_p(w) = \frac{1}{2}\sum_i (\overline{x_{ip}} - d_{ip})^2$ で示すものとする．この総誤差関数最小化問題を解くために， $E_p(w)$ を最小化することを考える．そのため $E_p(w)$ の勾配ベクトルを計算し，その逆方向に w_{ij} を変化させていくこととする．

式(１)の入出力関係より， w_{ij} が微小量変化することで z_{ip} が変化し， z_{ip} が微小量変化することで x_{ip} が変化することがわかるので，勾配ベクトルの偏微分式は以下のようになる．

$$\partial E_p(w)/\partial w_{ij} = \partial E_p(w)/\partial x_{ip} \times h'(z_{ip}) \times y_{jp} \tag{2}$$

ここで表記上の簡略化のため $\delta_{ip} = -\partial E_p(w)/\partial x_{ip} \times h'(z_{ip})$ とおく $\therefore \partial E_p(w)/\partial w_{ij} = -\delta_{ip} y_{jp}$

<u>STEP１</u>　第S層，第iニューロンが出力層に属しているとき

出力 x_{ip} は $E_p(w)$ の $\overline{x_{ip}}$ に等しいので $\partial E_p(w)/\partial x_{ip} = (x_{ip} - d_{ip})$ である．したがって勾配ベクトルは,

$$\partial E_p(w)/\partial w_{ij} = (x_{ip} - d_{ip}) \times h'(z_{ip}) \times y_{jp} \tag{3}$$

によって求められる．

<u>STEP２</u>　第S層，第iニューロンが出力層に属していないとき

第 S+1 層の第 k ニューロンの出力変化が $E_p(w)$ を変動させるので，勾配ベクトルは

$$\partial E_p(w)/\partial w_{ij} = h'(z_{ip})\sum_k \delta_{kp} w_{ki} \times y_{jp} \tag{4}$$

によって求められる．

STEP１, STEP２より出力層から前層へ δ_{ip} を δ_{kp} へと置き換えることで、全ての結合強度 w の勾配ベクトルを求めることが出来る．

ステップ幅＝0.01 として $w_{ij}^{(k+1)} = w_{ij}^{(k)} - 0.01 \times \partial E_p(w)/\partial w_{ij}$ の更新式によって， $E_p(w)$ が減少する方向に w を更新する．この作業が１つの教師データに対する１回の学習となる．この更新を続け，精度を上げていくまで学習を続ける．

４．逐次学習手法

逐次学習手法に求められることは，逐次的に与えられたデータに対してシステムを逐次改善し関数近似結果を出力することにある．従来研究[３]では，非線形関数を逐次的に学習するために，領域を区分的に線形近似するという手法を用いているが，本研究では非線形の関数を直接学習する方法を考える．このことによって，入力に対して予測するべき関数の値を直接出力として与えるシステムを構築することが出来る．一般的な学習手法では３層の階層型ネットワークが用いられるが，本研究では学習の精度を上げるために４層のネットワークで構成することを試み，３つの実装パターンを考え比較した．

Ⅰ：閾値関数を用いないネットワーク

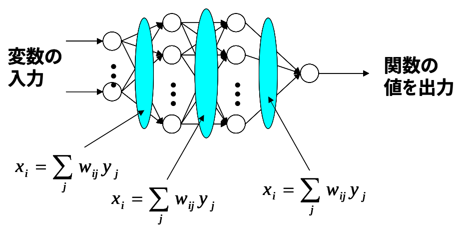

図３：閾値関数を用いないネットワーク

入力層→中間層→中間層→出力層の間が全て入力と重み w との線形結合のみによって表されるネットワーク．この構造は，ニューロンの発火という概念を無視して構成している．そのため全ての入出力の関係は重み w との積の和で構成されているので出力は入力の線形関数であるといえる．

全ての層からの出力 y_i は閾値関数を用いないため，勾配ベクトルは

中間層→出力層で $\frac{\partial E_p(w)}{\partial w_{ij}} = (x_{ip} - d_{ip}) \times y_{jp}$，出力層→中間層→中間層では $\frac{\partial E_p(w)}{\partial w_{ij}} = \sum_k \delta_{kp} w_{ki} \times y_{jp}$ となる．

Ⅱ：閾値関数を用いたネットワーク

入力層→中間層→中間層→出力層の間で中間層→中間層において閾値関数としてシグモイド関数を用い，他のニューロンからの出力は全て入力と結合強度との線形結合によってあらわす．

一般的な NN では出力層においても閾値関数を用いるので出力は０～１の値で出力されるが，関数近似を行う際には０～１以外の値を出力することが必要とされているため，出力層からの出力は入力と結合強度の線形結合によってあらわしている．

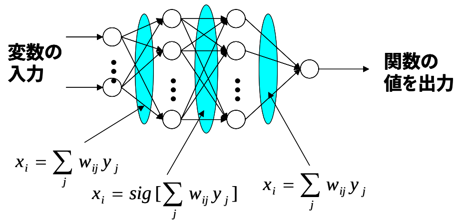

図４：閾値関数を用いたネットワーク

層によって出力の形が異なるので，勾配ベクトルは

中間→出力層で $\frac{\partial E_p(w)}{\partial w_{ij}} = (x_{ip} - d_{ip}) \times y_{jp}$　，　中間→中間層で $\frac{\partial E_p(w)}{\partial w_{ij}} = \sum_k \delta_{kp} w_{ki} \times y_{jp}$

入力→中間層で $\frac{\partial E_p(w)}{\partial w_{ij}} = h'(z_{ip}) \sum_k \delta_{kp} w_{ki} \times y_{jp}$ となる．

Ⅲ：閾値ありの区分学習

与えられたデータの定義域を分割し，それぞれの領域を区分する．区分された各領域でそれぞれ違う重み w の学習を行うことで，近似が難しい高次の関数の近似を可能にすることを目指す．この区分学習を用いると非線形関数の関数近似を行う際でも細かい分割を行うことで，その領域内で線形近似できることがわかるが，細かい分割を行うことは分割領域の数だけ学習データの数を増やさなければいけないことになり，線形近似することで満足できる誤差に留めるためには膨大な学習が必要となる．したがって本研究では従来研究[３]に対して分割した領域内で「Ⅱ：中間層にのみ閾値関数を用いたネットワーク」を用いて領域内を区分的に学習することを考える．

５．実験

提案する逐次学習手法の学習性能を次の関数について検証した．

１： $f(x_1, x_2) = x_1 + x_2$

２： $f(x_1, x_2) = x_1^2 + x_1 \times x_2 + x_2^2$

３： $f(x_1, x_2) = x_1^3 + x_1^2 x_2 + x_1 x_2^2 + x_2^3$

４： $f(x_1, x_2) = \frac{\tanh(9x_2 - 9x_1) + 1}{9}$

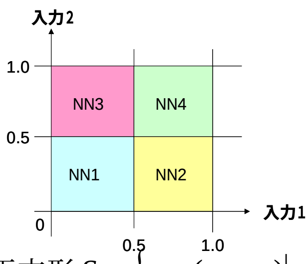

図５：領域分割

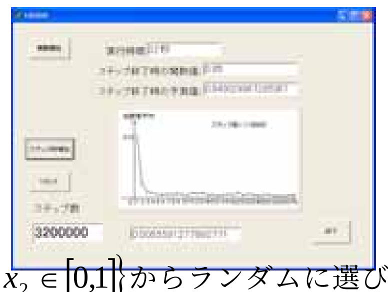

図６：実行画面

各関数の訓練データ $y = f_i(x_1, x_2)$ は単位正方形 $S = \{x = (x_1, x_2) | x_1, x_2 \in [0,1]\}$ からランダムに選び生成した．この訓練データの学習後，S 上の 10×10 の格子点上の関数値 $y = f_i(x_1, x_2)$ と実装したシステムによる推定値 $\hat{y}$ との差をとり，平均二乗誤差を算出しこれを用いて実装したシステムを評価した．なお全てのネットワークは入力層２個，中間層 20 個，出力層１個の人工ニューロンで構成した．ここ

で中間層のニューロンの数は可変であるが，計算時間を抑えるために 20 個で設定している．またⅢの区分学習においては領域を均等に４分割(図５)し，それぞれの区分された領域において学習を行うものとした．なお実験プログラムは Borland 社の Delphi6 で作成した(図６)．

６．実験結果および考察

Ⅰ：閾値関数を用いない４層階層型ネットワーク

線形関数の近似は 10×10^4 回の学習で誤差がなくなるまで収束したが，非線形の関数の近似では，どの関数でも 10×10^3 回以上の学習に対して精度を上げることは出来ず，非線形関数の学習は出来ないことがわかった．

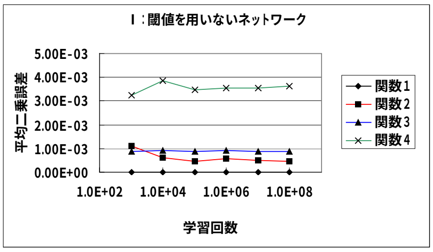

図７：閾値関数なし

Ⅱ：中間層にのみ閾値関数を用いた４層ネットワーク

線形関数の近似はⅠの閾値を用いない場合と比べ精度は悪くなっているが，学習ごとに精度が上がり，10×10^8 回の学習でおよそ 2.97×10^8 の平均二乗誤差に収束した．同様に非線形な関数は 10×10^7 回の学習でほぼ収束し 1.0×10^6 程度に収束した．また関数４は従来研究[１]において，10×10^8 回の学習終了時点で約 4.0×10^{-6} であるのに対し，同じ学習回数で 4.1×10^{-5} の精度となり先行研究と比べ近似性能が劣っていることがわかった．

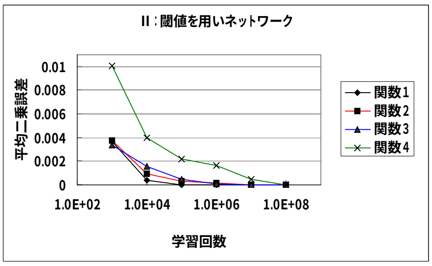

図８：閾値関数あり

Ⅲ：閾値ありの区分学習

いずれの関数においても，学習回数の少ないときは区分学習をしないⅡの方が精度良く近似できていることがわかるが，関数３および関数４の近似において 10×10^8 回の学習終了時点では区分学習を行った場合の方が精度良く近似することが出来た．これはいずれも高次の関数であり，学習回数を多くした場合には区分学習の方が精度の良い近似結果が得られることがわかる．

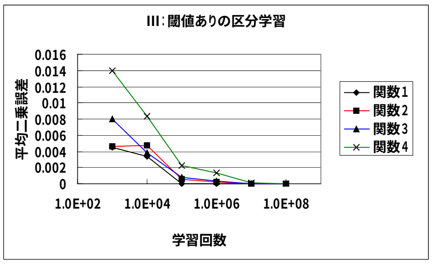

図９：区分学習

７．まとめ

本研究では，逐次的に関数近似を行う逐次学習手法を提案した．提案手法の関数近似の精度は 0.3%の限界があり，その誤差をゆるす時には，多くの関数において有効であることがわかった．この 0.3%は，不鮮明なデータに対する関数近似としては許容できるものと思われる．学習法においてステップ幅固定の最急降下法を用いているため局所最適解に陥って誤差が減少しなくなっている可能性も考えられるので，重み w を更新する際にランダム探索法などを用いることで，局所解に陥らないように学習法を強化することが必要であると思われる．

参考文献

［１］．馬場則夫，小島史夫，小澤誠一：ニューラルネットの基礎と応用；共立出版株式会社（1994）

［２］．茨木俊秀，福島雅夫：最適化の手法；共立出版株式会社（1993）

［３］．黒木修一：競合連想ネットの漸近最適性と非線形関数の逐次学習への応用；電子情報学会論文誌（pp.１～11）(2002)