バックプロパゲーションにおける高速学習アルゴリズム －補習学習－

5－8

A study on supplementary learning algorithm in back propagation

益岡 竜介 渡部 信雄 木本 隆 川村 旭 浅川 和雄
Ryusuke MASUOKA Nobuo WATANABE Takashi KIMOTO Akira KAWAMURA Kazuo ASAKAWA
（株）富士通研究所
Fujitsu Laboratories Ltd.

ABSTRACT

In this paper we propose a fast learning algorithm for neural networks called the Supplementary Learning Algorithm (SLA). Generally it takes a great deal of time for back propagation to converge. SLA has been developed by observing the learning processes of neural networks. In the last stages of learning, since networks produce correct outputs for almost all data, the calculations for propagating errors in the learned data can be reduced. SLA takes advantage of this fact.

SLA can also be used simultaneously with the Incremental Learning Algorithm (ILA) we have proposed. We explain the principles of SLA and ILA by applying these algorithms to font recognition. We show that when using these methods, a three-layer neural network learns four times faster than when using simple back propagation.

1. 序

近年，階層型ニューラルネットワークの学習方式として，バックプロパゲーション法（以下 BP と省略）が注目を集めている．しかし，BP を実行するためには膨大な計算時間を要し，その高速化が求められている．我々は，BP における柔軟性に富む高速な学習アルゴリズムを開発するため次のような検討を進めた．まず，アルゴリズムは，ニューラルネットワークの並列処理の特徴を生かしてなければならないことに着目し，高速化に対する１つの解答として，追記学習アルゴリズム[1,2]を既に開発した．今回は，学習中の無駄な計算を低減することに着目した補習学習法を報告する．この学習法は，追記学習アルゴリズムとも組み合わせることができる高速学習アルゴリズムである．更に，それらのアルゴリズムをアルファベットフォント認識の学習に適用した結果を示す．

2. いままでの BP 学習の問題点

まず BP 学習のどこで計算量を減らすことができるかを観察した．アルファベットフォントの認識を学習中のネットワークの状態を観察すると，第１図のように学習最後の方になると，ほとんどのフォントに関して学習は終わっている．これは，最後の方では，ほとんどのフォントに関して既に学習が終わっているにもかかわらず，最後まで全てのフォントに関して学習を繰り返しているということである．このことは，２つの意味で学習の進行を阻害している．

１つは，積極的な意味での阻害である．学習が済んでいるパターンが未学習のパターンの学習の進行を阻害するということである．すなわち次のようなことが起きる．主に未学習のパターンのエラーにより重みが変わり始めると，今まで小さかった既学習のパターンによるエラーが増加する．すると組織だ

っていない未学習のパターンによるエラーは増加した既学習のパターンによるエラーに負けてしまい，重みの変化を妨げてしまう．

もう1つは，阻害と言うよりは計算量に関する無駄である．既に学習が済んでいるパターンは，エラーが十分小さいので，それらのパターンに関するエラーのバックプロパゲーションは非常に小さくなる．

これらの BP における問題点を回避するには，エラーの大きなパターンに関してだけエラーをバックプロパゲートすればよい．これが補習学習の基本的な考え方である．

しかし補習学習では全てのパターンについてエラーをバックプロパゲートしないので，重みを最急降下ではない方向に変えていくことになる．それに関しては次のように考える．ニューラルネットワークで，エラーを小さくすることと，求める対応関係を実現することとは一般には異なっている．エラーを小さくするためには，最急降下の方向に重みを変えることが重要であろう．（しかし必ずしも global minimun に到達する保証はない．）しかし，我々はむしろバックプロパゲーションを，求める対応関係を実現するための Hebb の学習則のようにとらえ直すことにより，補習学習アルゴリズムを開発した．

3. 補習学習のアルゴリズム

補習学習のアルゴリズムを説明する．バックプロパゲーションでは，一度ネットワークを入力パターンに関して実行して，出力を出させる．そこでネットワークの出力と入力パターンに関する出力の教師信号を比較して，得られたエラーをバックプロパゲートする．補習学習のアルゴリズムでは，ネットワークを入力パターンに関して実行して，出力を出させた段階で，その出力が収束しているかを判定する．（ここで収束とは，何らかの意味で出力が教師信号に十分近いという意味である）もしその入力パターンに関する出力が収束していると判定されたら，エラーをバックプロパゲートしない．出力が収束していないと判定されたパターンに関してだけ，エラーをバックプロパゲートする．こうして，収束していないパターンに関してだけ，エラーをバックプロパゲートすることにより，無駄な計算を排し，効率的な学習を可能とする．補習学習のアルゴリズムを，フローチャートとして第2図にまとめた．

4. 追記学習アルゴリズムとの関係

補習学習のアルゴリズムは他の学習アルゴリズムとも簡単に組み合わせることができる．特に追記学習とのアルゴリズムとは，相性が良い．後で述べるように追記学習と組み合わせると一層の効果が出る．

まず簡単に追記学習のアルゴリズムを説明しよう．追記学習のアルゴリズムは，分割した学習パターンの組に対して，適用される．そのアルゴリズムは，第3図のとおりである．すなわち最初のパターンに関してだけ独立に学習し，後は新しいパターンごとに，その新しいパターンを学習する新規学習フェーズ，新しいパターンと今までのパターンを合わせて学習する復習フェーズを繰り返して学習を進める．

補習学習を追記学習に組み合わせるには，この追記学習の各段階の学習に対して，補習学習のアルゴリズムを適用すればよい．

特に追記学習の復習フェーズでは，ほとんどのパターンが一度学習していることなので，ほとんどのパターンが直ぐに再学習されてしまう．学習に手間のかかるのは，ほんの2，3のパターンについてだけのことが多い．補習学習では，その2，3のパターンに関してだけ，学習させることになり，効率的に学習を進めることができる．

5. 補習学習の実験

この補習学習アルゴリズムの効果を見るためにアルファベットフォント認識の学習を例に取ってみた．実験の設定は，いろいろな設定を試し，通常の BP がほぼ最適に学習するように選んだ．実験の設定の詳細は，以下のとおりである．

5.1. ネットワークの課題

同じ状態のネットワークからそれぞれ通常の学習，追記学習，補習学習，追記学習と補習学習のアルゴリズムでアルファベットフォント認識を学習させて，

学習の速さを比較した.

5.2. ネットワークの構造

使ったネットワークの構造は以下のようである.

入力層 64 ユニット
中間層 15 ユニット
出力層 26 ユニット

閾値は，中間層と出力層に入れた.

5.3. 重みと閾値の初期値と学習

重みと閾値の初期値は，$[-0.1, 0.1]$ からランダムに取った. ネットワークの学習は，学習させるパターンを変える以外は通常のバックプロパゲーションを用いた. 閾値も重みと同様に学習させた.

5.4. パターンの構造

入力パターンは各アルファベットについて 8 × 8 ドットのフォントのデータを 0,1 の2値としたものを与えた. 教師信号は各アルファベットにつきそれぞれ異なる一つのユニットだけが 1 でその他は全て 0 としたものを与えた. パターンの数はAからZまでの 26 個である.

入力信号	教師信号
00011000	1000000000000
00100100	0000000000000
01000010	
01111110	
01000010	
01000010	
01000010	
00000000	

第4図 入力信号と教師信号の例（A）

5.5. 学習定数

学習定数は以下のように取った.

(1) 学習レート

AからZの 26 文字全てを学習するときに 0.2 とした. 後は学習させるパターン数に反比例してとった. すなわち，p をパターン数としたときに学習レート ϵ を $\epsilon = 5.2/p$ とした.

(2) モーメンタム

一律にモーメンタム α を $\alpha = 0.4$ とした.

5.6. 学習の収束判定

学習の収束判定は，教師信号からの差が各ユニットで 0.4 未満になったときとした.

5.7. 学習パターンの分割の仕方

学習パターンの分割の仕方は，分割数を g，分割数で 26 を割り切れる場合は，アルファベットの最初から順に $26/g$ づつ組に分けた. 分割数で 26 を割り切れない場合には，1 番目から $(26 - g \times \lfloor 26/g \rfloor)$ 番目の各組に順に $\lfloor 26/g \rfloor + 1$ 個のアルファベットを割り当て，それ以外の組に順に $\lfloor 26/g \rfloor$ 個のアルファベットを割り当てた.

これらの学習アルゴリズムの効果を実験した結果が第5図のグラフである. 補習学習単独の効果では，約 1.2 倍の高速化である. 追記学習アルゴリズム（9分割）では通常のバックプロパゲーションに比べて約 1.4 倍の高速化であるが，追記学習（9分割）と補習学習を合わせたアルゴリズムでは通常のバックプロパゲーションに比べて約 4 倍の高速化という結果を得た. （このグラフにあるインストラクション数とは，実行，エラーバックプロパゲート，重み更新に要する全ての加減乗除の回数を合計したものである. ほぼ学習終了までの時間に比例する. ）

第6図は，同じ初期状態からのネットワークの学習についてグラフにしたものである. 学習中か否かに関わらず全パターンのエラーの平均を縦軸に，インストラクション数を横軸にプロットしたものである. この図では，分かり易いように4分割の場合で示している. 特徴的なのは，補習学習を入れたアルゴリズムでは，エラーの平均が高い所で終わっていることである. これは，学習が平均的に進んでいることを示している.

6. 結論と今後の課題

追記学習，補習学習の各アルゴリズムをアルファベットフォントの認識に適用して，両アルゴリズムを使った場合，通常の BP と比較して，約4倍の高速化と結果を得た.

追記学習，補習学習の各アルゴリズムではパターン数が動的に変化するので，学習定数の設定が難しい. 各学習段階，パターン数に応じて学習定数を最適に調整することが課題である.

また，この補習学習のアルゴリズムをアナログパターンの学習に適用する実験も進めている.

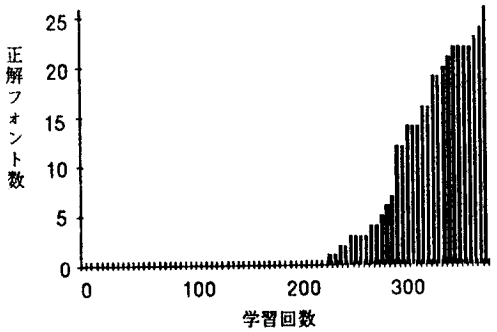
7. 謝辞

日頃から御指導頂く棚橋純一部門長，林弘部長ならびに人工知能研究部第二研究室の皆様に感謝します．

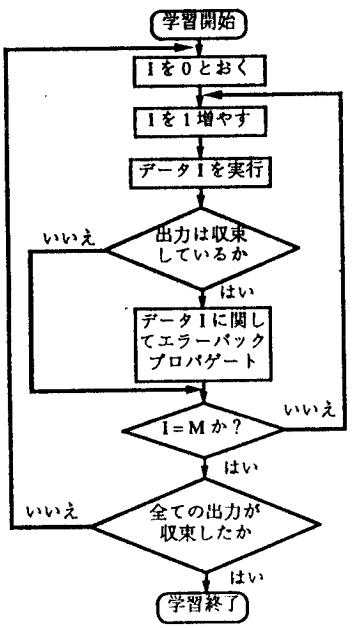
References

1. 益岡, 渡部, 木本, 川村, 浅川, "バックプロパゲーションの高速学習アルゴリズムー追記学習ー," 情報処理学会第３８回全国大会６Ｆ－５, pp. 492-493, 17 Mar. 1989.

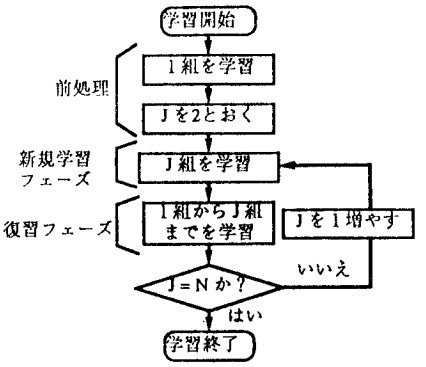
2. 益岡, "バックプロパゲーションの高速学習アルゴリズムー追記学習ー," 信学技報, vol. 88, no. 466, pp. 121-126, 15 Mar. 1989.



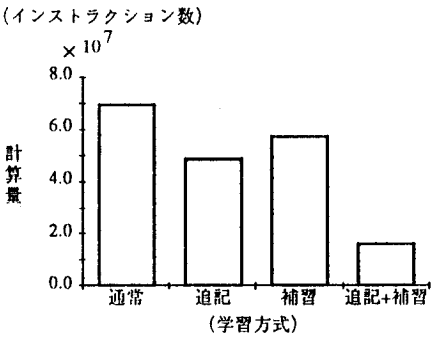
第１図　学習中の正解フォント数



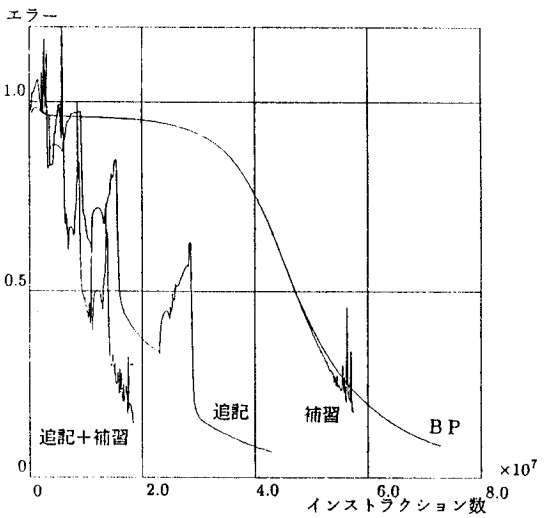
第２図　補習学習のアルゴリズム



第３図　追記学習のアルゴリズム



第５図　アルファベットフォントの認識学習実験結果



第６図　学習中のエラーの変化