

シグモイド関数の傾きを減少させたフィードフォワードニューラルネットワークにおけるバックプロパゲーション

橘高 新三郎[†] 上手 洋子[†] 西尾 芳文[†]

[†] 徳島大学工学部 〒770-8506 徳島県徳島市南常三島 2-1

E-mail: †{kittaka,uwate,nishio}@ee.tokushima-u.ac.jp

あらまし 私達の研究の目的はフィードフォワードニューラルネットワークの学習法であるバックプロパゲーションにおける学習の誤差の減少である。バックプロパゲーションでの学習において誤差を減らす方法として、学習係数の変更やモーメント項の追加などがあげられる。私達はシグモイド関数に注目して誤差の減少を行った。シグモイド関数はフィードフォワードニューラルネットワークのニューロンに入力された信号を出力信号に変換する時と、ニューロン間の重みの調整時に使われる関数である。私達はシグモイド関数の傾きを時間的に変化させ、その効果を調査した。キーワード ニューラルネットワーク、フィードフォワードニューラルネットワーク、バックプロパゲーション、シグモイド関数

Decreasing Gradient of Sigmoid Functions in Back Propagation of Feed-Forward Neural Network

Shinsaburo KITAKA[†], Yoko UWATE[†], and Yoshihumi NISHIO[†]

[†] Faculty of Engineering, Tokushima University

2-1 Minami-Josanjima, Tokushima 770-8506, Japan

E-mail: †{kittaka,uwate,nishio}@ee.tokushima-u.ac.jp

Abstract Our study is about the learning method of feed-forward neural network. Generally, the method of improving its learning is focused on learning rate and moment term. We focus on sigmoid functions. Sigmoid functions are used for converting input signal into output signal and adjusting connection weight of learning in feed-forward neural network. We change gradient of sigmoid functions and investigate the effect of our method.

Key words Neural Network, Feed-forward neural network, Back propagation, Sigmoid functions

1. はじめに

ニューラルネットワークは動物の脳をモデルにしたネットワークでニューロンと呼ばれる素子で構成されている。構成が動物の脳に似ているという点からニューラルネットワークは文字の認識や音声の認識に適していると言われている。また、ニューラルネットワークの特徴は並列処理をする回路であることと学習する回路であることがあげられる。この学習するという特徴からニューラルネットワークは、人間の脳波の認識にも適しているとされている。人間の脳波は個人差があり、また、体調によっても変化するが、学習によって信号を認識するニューラルネットワークは個人の脳波を学習でき、体調変化による脳波でさえも認識することができるためである。先行研究より、ニューラルネットワークにおける脳波の認識率は86%程度であるという記述を確認した [1]。この認識率を増加させるため

には学習によって脳波と出力の関係をより明らかにする必要がある。このようなニューラルネットワークを利用した回路を製作するにあたり、問題になるのが実際の出力と理想的な出力の差である。ニューラルネットワークはその学習法の性質上この差が0になることは起こりえない。この誤差をいかに0に近づけられるかがニューラルネットワークにおける大きな課題となっている。そこで私達は学習法に注目した研究を行う。今回、私達の研究ではフィードフォワードニューラルネットワークの学習法に変更を加える。フィードフォワードニューラルネットワークの学習方法はバックプロパゲーションと呼ばれる方法である。私達はニューロンの変換関数であるシグモイド関数に変化を加えることで学習精度の向上を測る。

2. フィードフォワードニューラルネットワーク

フィードフォワードニューラルネットワークはニューラルネットワークの代表的なネットワークのひとつである。ネットワークは入力層、中間層、出力層の3つの層で構成されている。図1にフィードフォワードニューラルネットワークの模式図を表す。

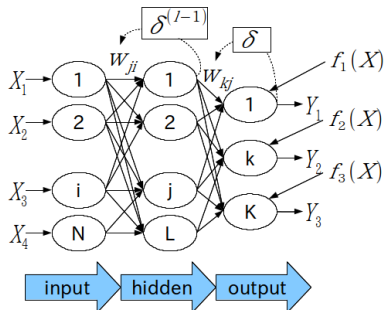


図1 フィードフォワードニューラルネットワーク模式図

入力された信号は一方にしか伝搬が行われず、ニューロンは入力された信号をシグモイド関数に従って、0から1の値に変換して出力する。式(1)にシグモイド関数を表し、図2にそのグラフを表す。

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (1)$$

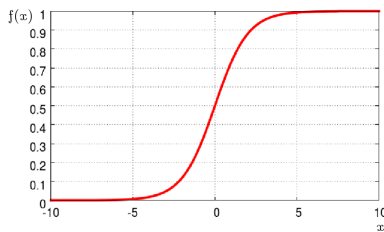


図2 シグモイド関数

3. バックプロパゲーション

フィードフォワードニューラルネットワークの学習法としてバックプロパゲーションを使用する。教師信号として、人為的に出力層に理想的な出力を与えることで出力層で実際の出力と理想的な出力の差を式(2)から計算する。その差を元にニューロン間の重みを式(3)から調節し、出力を理想的な出力に近づけるといった手法である。 $f_n(x)$ は実際の出力、 Y_n は理想的な出力を表す。 W は重み、 η は学習係数、 net_j は全てのつながっているニューロンからの入力を表す。

$$\delta = f_n(x) - Y_n \quad (2)$$

$$\Delta W = \eta \delta_n f'(net_j) \quad (3)$$

この計算を続けることで出力の信号を理想的な出力に近づけることができる。

4. 提案手法

今回私達は、シグモイド関数に含まれている“ k ”を学習回数ごとに減少させることで傾きを徐々に緩やかにしていく方法をとる。

$$f(x) = \frac{1}{1 + \exp(-kx)} \quad (4)$$

図3に“ k ”を変化させたときのシグモイド関数のグラフを表す。図4にその微分形のグラフを表す。

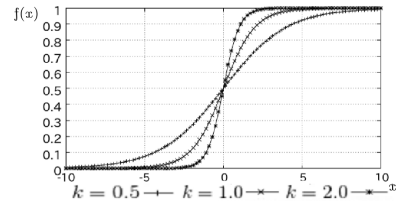


図3 “ k ”変化時のシグモイド関数

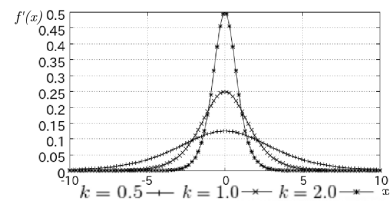


図4 シグモイド関数微分形

図3より“ k ”が減少するとシグモイド関数の傾きが緩やかになることがわかる。また、図4より“ k ”が減少すると重みの修正値が減少していくことがわかる。私達は、まず最初に最適な学習係数を決定し、次に“ k ”を変化させることで起こる学習の変化を調査する。その後に提案手法として、“ k ”を学習回数ごとに0.001ずつ減少させる。最後に“ k ”を減少させる値を変化させその学習の変化を調査する。

5. シミュレーション結果

私達はあやめという花の分類と、Sin関数変換という2つのシミュレーションを行い、提案手法の効果进行调查する。

初期設定として、学習回数は1000回、中間層のニューロン数は4つで固定してシミュレーションを行う。

5.1 あやめの分類

このシミュレーションではあやめを3種類に分類する。学習用のサンプルデータを150個用意し、4つの特徴を元に3種類のあやめの分類を行わせる。4つの特徴とはがく片の幅と長さ、花びらの幅と長さである。

初めに最適な学習係数を決定する。表1に学習係数が0.1から1.0までの誤差の大きさを表す。なお、誤差は全て50回のシミュレーションの平均値を表す。

表 1 学習係数 a 変更時の誤差

a	0.1	0.2	0.3	0.4	0.5
ave.	0.02896	0.02438	0.02974	0.02261	0.02144
a	0.6	0.7	0.8	0.9	1.0
ave.	0.02144	0.02079	0.02137	0.02112	0.02144

表 1 より、最も小さい誤差をもつ学習係数は 0.7 である。この学習係数を使用してシミュレーションを続行する。

次に、“ k ” を変化させると学習にどのような変化が起こるかを調査する。表 2 に “ k ” を 0.6 から 1.5 から変化させた時の誤差の大きさを表す。

表 2 “ k ” 変更による誤差

k	0.6	0.7	0.8	0.9	1.0
ave.	0.02295	0.02231	0.02177	0.02221	0.02079
k	1.1	1.2	1.3	1.4	1.5
ave.	0.02076	0.02237	0.02267	0.02364	0.02365

“ k ” を変更することで表 1 より小さな誤差を得ることができたがその差はごく僅かなものであった。そのため私達は “ k ” を変更することで学習に大きな変化は起こらないと考えた。

次に提案手法の検証を行う。“ k ” を学習回数ごとに 0.001 ずつ減少させる。また、減少を開始させる回数を変化させその効果を調査する。表 3 に学習後の誤差の大きさを表す。また、変更を加えない学習と提案手法の学習の誤差の変化を図 5、図 6 に表す。

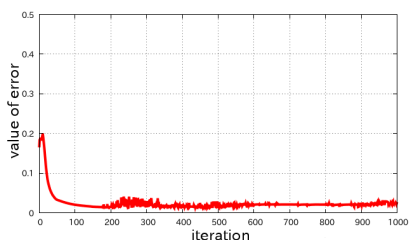


図 5 既存手法の学習誤差の変化 ($a=0.7, k=1.1$)

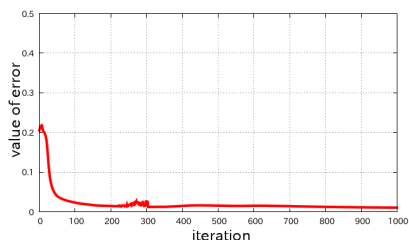


図 6 提案手法の学習誤差の変化 ($a=0.7, k=1.1$, 減少開始回数 = 300 回)

表 3 “ k ” の時間的変化による誤差 (あやめの分類)

		k										
		減少開始回数	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
学習回数	0		0.33406	0.19977	0.18239	0.01315	0.01150	0.01282	0.01576	0.01674	0.01780	0.01747
	100		0.22713	0.17337	0.21337	0.01244	0.01225	0.01301	0.01509	0.01622	0.01732	0.01743
	200		0.13594	0.16410	0.17844	0.01165	0.01143	0.01303	0.01492	0.01591	0.01598	0.01707
	300		0.10816	0.09496	0.08314	0.01304	0.01106	0.01465	0.01575	0.01514	0.01659	0.01690
	400		0.02578	0.03072	0.05036	0.01180	0.01395	0.01356	0.01587	0.01697	0.01644	0.01697
	500		0.01394	0.01325	0.01122	0.01071	0.01101	0.01301	0.01325	0.01556	0.01617	0.01584
	600		0.10579	0.02672	0.01150	0.01072	0.01148	0.01219	0.01589	0.01560	0.01574	0.01639
	700		0.02457	0.02290	0.03057	0.01090	0.01183	0.01378	0.01642	0.01610	0.01682	0.01624
	800		0.01371	0.01937	0.02613	0.01386	0.01336	0.01409	0.01471	0.01558	0.01644	0.01664
	900		0.01378	0.01228	0.01194	0.01336	0.01250	0.01498	0.01735	0.01528	0.01610	0.01648

表 3 より、多くのパラメータで誤差の減少が確認できる。このことから提案手法はバックプロパゲーションの性能の向上に効果的だと考えられる。また図 5 と図 6 を比較すると提案手法は誤差の増減がなくなり学習結果が安定した値が得られるようになっていることがわかる。また、最も小さい誤差を持つパラメータは “ k ” = 0.9, 減少開始回数は 500 回からの時であった。

最後にこのパラメータを使い、“ k ” の減少値を変化させた時の学習の変化を調査する。表 4 にその誤差を表す。

表 4 “ k ” 減少値変化時の誤差

value	0.0006	0.0007	0.0008	0.0009	0.001
ave.	0.01748	0.01547	0.01490	0.01271	0.01078
value	0.0011	0.0012	0.0013	0.0014	0.0015
ave.	0.01544	0.01711	0.01214	0.01693	0.01285

表 4 より、最も小さな誤差をもつパラメータは減少値が 0.001 の時であった。このシミュレーションでは提案手法を使うことで誤差が既存手法の 51.5 % となった。

5.2 Sin 関数変換

次に入力した角度を Sin 関数に変換するシミュレーションを行う。入力データは 180 個、データは 0 から 180 までの角度と角度に対応した Sin 関数を記している。先ほどのシミュレーション同様、初めに最適な学習係数を決定する。このシミュレーションでは学習係数が大きい値の時に誤差の最小値をとったため、表 5 には学習係数が 2.1 から 3.0 までの誤差の大きさを表す。

表 5 学習係数 a 変化時の誤差

a	2.1	2.2	2.3	2.4	2.5
ave.	0.01716	0.01667	0.01678	0.01592	0.01687
a	2.6	2.7	2.8	2.9	3.0
ave.	0.01651	0.01632	0.01527	0.01503	0.01541

表 5 より、最も小さい誤差をもつ学習係数は 2.9 であった。この値を使用してシミュレーションを続行する。

次に、“ k ” を変化させると学習にどのような変化が起こるかを調査する。表 6 に “ k ” を 1.1 から 2.0 から変化させた時の誤差の大きさを表す。

表 6 “ k ” 変更による誤差

k	1.1	1.2	1.3	1.4	1.5
ave.	0.01604	0.01415	0.01493	0.01349	0.01511
k	1.6	1.7	1.8	1.9	2.0
ave.	0.01383	0.01331	0.01422	0.01311	0.01335

表 7 “k” の時間的变化による誤差 (Sin 関数変換)

		k									
減少開始回数		1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
学習回数	0	0.02962	0.02434	0.02279	0.01834	0.01829	0.01684	0.01526	0.01552	0.01547	0.01291
	100	0.02528	0.02261	0.02145	0.02059	0.01926	0.01744	0.01485	0.01508	0.01467	0.01205
	200	0.02424	0.02661	0.02240	0.01832	0.01699	0.01657	0.01470	0.01329	0.01146	0.01097
	300	0.03176	0.02379	0.02006	0.01779	0.01650	0.01441	0.01495	0.01199	0.01204	0.01307
	400	0.02925	0.02362	0.02213	0.02002	0.02130	0.01562	0.01487	0.01322	0.01237	0.01241
	500	0.03032	0.02692	0.02504	0.02203	0.01798	0.01749	0.01547	0.01422	0.01321	0.01307
	600	0.03175	0.03012	0.02393	0.02003	0.01791	0.01752	0.01601	0.01521	0.01520	0.01392
	700	0.03351	0.02595	0.02590	0.02278	0.01905	0.01876	0.01899	0.01753	0.01568	0.01540
	800	0.03773	0.03035	0.02523	0.02360	0.01987	0.02429	0.02236	0.01889	0.01892	0.01642
	900	0.06038	0.04008	0.03369	0.02365	0.02586	0.02658	0.02741	0.02393	0.02168	0.01897

あやめの分類のシミュレーションとは違い, “k” を変更することにより誤差の減少が確認できた。

次に提案手法の検証を行う. “k” を学習回数ごとに 0.001 ずつ減少させる. また, 減少を開始させる回数を変化させその効果を調査する. 表 7 に学習後の誤差の大きさを表す. また, 変更を加えない学習と提案手法の学習の誤差の変化を図 5, 図 6 に表す.

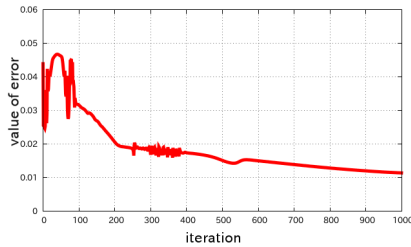


図 7 既存手法の学習誤差の変化

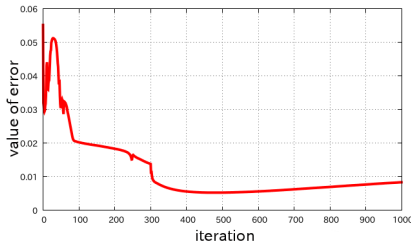


図 8 提案手法の学習誤差の変化

表 7 より提案手法による誤差の減少が確認できるパラメータが確認できたが, 前のシミュレーションと比較するとその数はごくわずかであった. 表 7 より, 最も小さな誤差を持つパラメータは “k”=2.0 で減少を 200 回から開始した時であった. また, 図 7, 図 8 を比較すると既存手法では誤差はしだいに減少をしていっているが, 提案手法は次第に誤差が増加していっている. この現象からあまり多くのパラメータで誤差の減少が確認できなかったと考える.

最後にこのパラメータを使い, “k” の減少値を変化させた時の学習の変化を調査する. 表 8 にその誤差を表す.

表 8 “k” 減少値変化時の誤差

value	0.0006	0.0007	0.0008	0.0009	0.001
ave.	0.01035	0.00906	0.01177	0.01044	0.01118
value	0.0011	0.0012	0.0013	0.0014	0.0015
ave.	0.01198	0.01273	0.01303	0.01672	0.01688

表 8 より, 最も小さい誤差を持つパラメータは “k” の減少値を 0.007 にした時であった. このシミュレーションでも提案手法を使うことで誤差が既存手法の 60.3 % となった.

6. ま と め

私達はフィードフォワードニューラルネットワークの学習法であるバックプロパゲーションに変更を加える提案を行った. 提案手法はシグモイド関数に含まれる “k” を学習回数ごとに 0.001 ずつ減少させるという手法をとった. この手法により 2 つのシミュレーションで変更前より小さな誤差を得ることを確認した. しかし, Sin 関数変換のシミュレーションでは誤差の減少が確認できたパラメータがあやめの分類と比較すると少量であったのでこの 2 つのシミュレーションを比較し, この差の原因を調べる必要がある. また, シグモイド関数の傾きを減少させることで学習の仕方に変化が確認できたため, 減少の方法を工夫しよりよい学習結果を得られるようにすることが今後の課題である.

文 献

- [1] A. Hiraiwa, N. Uchida, K. Shimohara, N. Sonehara “EMG Recognition with a Neural Network Model for Cyber Finger Control,” Transactions of the Society of Instrument and control Engineers, vol. 30, no.2, pp.216-224, 1994.
- [2] S. Kitayama, M. Sasaki, S. Ito, “Limb motion estimation by Bereitschafts Potential,” Journal of the Japan Society of Applied Electromagnetics and Mechanics, vol. 22, no. 2, pp. 318-323, 2014.
- [3] T. Kitamura, N. Tsuziuchi, T. Koizumi “Manipulation of Robot Hand Based on Motion Estimation Using EMG Signals,” Transactions of the Japan Society of Mechanical Engineers Series C, vol. 73, no. 735, pp3024-3030, 2007.
- [4] A. IBE, M. Gouko, K. Ito “Discrimination of Combined Motions for Prosthetic Hands Using Surface EMG Signals” Transactions of the Society of Instrument and Control Engineers, vol. 45, no. 12, pp. 717-723, 2009.
- [5] M. Kasuya, R. Kato, T. Takagi, S. Ito, S. Takayama, H. Yokoi “Prothetic hand and physical expansion : Toward the realization of the technology to redefine disability” Journal of Information Processing and Management, vol. 58, no. 12, pp. 887-889, 2015.