

6F-3

バック・プロパゲーションを利用したNNの学習異常とその対策に関する実験と考察

倉知 満 中川 徹 北川 一 (豊田工業大学)

1. はじめに

計算機に対するマン・マシン・インターフェースとしてのコマンドの変換機構を、ニューラル・ネットワーク(以下NN)を用いて構築しようと考えた。この予備実験として、入力コマンドによる利用者の特定実験をNNシミュレータを用いて行ったところ、予想外に学習時間が長く、このままアプリケーションを構築するのは難しいことが判明した。そこで、NNの動作を高速化するため学習加速アルゴリズムについて検討を行った。

2. NN学習の高速化

NN学習の高速化について、検討の段階で、アナログ値を用いていた従来のNNモデルとそのシミュレータを完全にデジタル化し、ニューロンの入出力において整数のみ(k of nの法則(1))を用いることが高速化に寄与すると考え、実験を行った。

本報告では、デジタル学習とはニューロンの入出力に整数値のみを使用し、アナログ学習とはそれ以外の中間値を使用するものとして分類する。

3. デジタル学習の確認実験

デジタル学習が存在するかどうかを調べる目的で、デジタル値を用いて記述できるはずの問題(exclusive_or, half_adder, full_adder)を、アナログ値(浮動小数)を用いているNNに学習させ、結果としてデジタル学習になっているかを調べた。

- <条件>
・ネットワーク: 3層構造・feed_forward型
・学習方法: back_propagation(以下BP)
・入出力関数: sigmoid
・結合度(W): 0~1で一樣な乱数
・中間ユニット数(hid): 2~50個

入力層 中間層 出力層

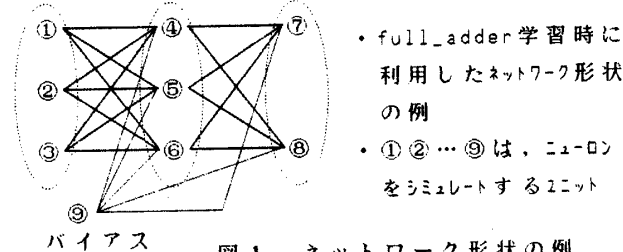


図1. ネットワーク形状の例

3.1 結合度の初期値についての実験結果

中間ユニット数が増加すると、ユニット1個当りの学習回数が増加した。これは結合度の初期設定値(以下W0)が原因であった。期待値0.5のW0と中間ユニットの出力値(入出力関数としてsigmoidを利用しているため、期待値0.5以上)の積の和(出力ユニットへの入力)が、外部からのNNへの入力に拘らず、大きな値となりNNの出力が1となり続けていた。

今回の場合、初期値W0を-1~+1の範囲で一樣な乱数にしてやることで解決することができる。(参: 図2)ただし、一般的なW0の与え方は、問題依存であり、より収束しやすい設定方法があると考えられる。

また、入出力関数のスロープをなだらかなものにすることで対処することもできる。そうすればNNは、入力パターンの差を出力ユニットへの入力の差として分離でき、学習を進めることかできる。ただしこの場合、初期値W0に対する適応力が向上する代わりに、学習が遅くなる。そこで、入出力関数自体を動的に変化させる機構が必要と考えられる。

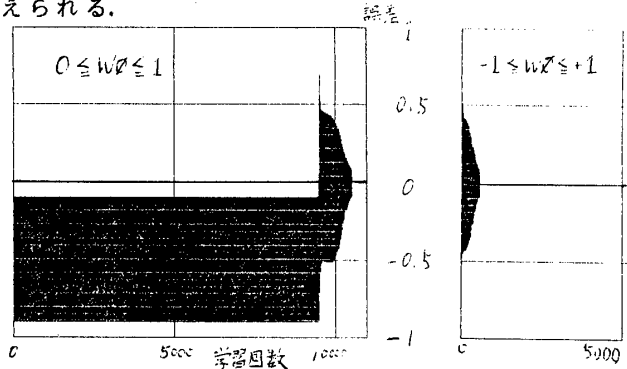


図2 学習過程における誤差の変化(hid=30)

3.2 学習の停滞についての実験結果

次のような場合、NNシミュレータはlocal_minimalに陥り学習が進みにくくなることがある。学習時(Wが変化してゆく過程)、またはWの初期設定時に同じ環境(結合関係とWが同一)のユニットが複数生じると、それらは実質上合成された1ユニット分の能力しか持たなくなる。従って、実質的な中間ユニットの数が不足すると、NN全体の能力が低下し、学習が停滞してしまう。

実験的に、同一環境に置かれたユニットが同じ動作をすることは、確認できている。

3.3 デジタル学習の実験結果

表1 full_adder学習後のW

	hid=2	hid=3		hid=2	hid=3
W_{14}	+1.4	-5.0	W_{36}	---	+7.1
W_{24}	+1.4	+6.2	W_{96}	---	-10.0
W_{34}	+1.4	+5.7	W_{47}	+7.2	-6.4
W_{94}	-2.2	-3.4	W_{57}	-10.3	+0.8
W_{15}	-11.3	+6.3	W_{67}	---	+1.4
W_{25}	-11.3	-0.6	W_{97}	+1.5	+10.7
W_{35}	-11.3	+1.0	W_{48}	+43.1	-5.2
W_{95}	+16.8	-3.4	W_{58}	+25.1	+10.7
W_{16}	---	+6.3	W_{68}	---	+10.3
W_{26}	---	+6.9	W_{98}	-33.9	-11.2

W_{ij} : ユニット*i*から*j*への結合度
ネットワーク形状は図1と同一

表2 各ユニットの出力値 (hid=2)

入力層 ①②③	中間層			出力層	
	④	⑤	⑥	⑦	⑧
0 0 0	0.1/0	1.0/1	---	0.0/0	0.0/0
0 0 1	0.3/0	1.0/1	---	0.0/0	1.0/0
0 1 0	0.3/0	1.0/1	---	0.0/0	1.0/0
0 1 1	0.7/1	0.0/0	---	1.0/1	0.0/1
1 0 0	0.3/0	1.0/1	---	0.0/0	1.0/0
1 0 1	0.7/1	0.0/0	---	1.0/1	0.0/1
1 1 0	0.7/1	0.0/0	---	1.0/1	0.1/1
1 1 1	0.9/1	0.0/0	---	1.0/1	1.0/1

注) sigmoid時/step関数時

表3 各ユニットの出力値 (hid=3)

入力層 ①②③	中間層			出力層	
	④	⑤	⑥	⑦	⑧
0 0 0	0.0/0	0.0/0	0.0/0	0.0/0	0.0/0
0 0 1	0.9/1	0.1/0	0.1/0	0.0/0	1.0/1
0 1 0	0.9/1	0.0/0	0.0/0	0.0/0	1.0/1
0 1 1	1.0/1	0.1/0	1.0/1	1.0/1	0.0/0
1 0 0	0.0/0	1.0/1	0.0/0	0.0/0	1.0/1
1 0 1	0.1/0	1.0/1	1.0/1	1.0/1	0.0/0
1 1 0	0.1/0	0.9/1	1.0/1	1.0/1	0.0/0
1 1 1	1.0/1	1.0/1	1.0/1	1.0/1	1.0/1

学習後のWを眺めると、表1ではアナログ的な中間値を持っているように見える。しかし、入出力関数として利用していたsigmoid関数をステップ関数 ($f(x)=0[x<0], 1[x>=0]$) に置き換えても認識時の動作は変化しない場合が多いことが実験により判明した。(参: 表1 表3)

つまりsigmoid関数のステップ関数化で動作の変わらなかったNNは、学習後のWについても実はアナログ値を必要としているわけではない。

この理由は、1個のユニットは線形分離を行っているに過ぎず、またこの場合の、分離すべき点(入力パターンの組合せ)は有限であり、Wが離散値を取っていたためと考えられる。

ただし、デジタル学習に必要な数だけの中間ユニットが存在しないと、あるユニットは入出力関数の中間値を利用して学習を行なう。この場合、sigmoid関数のステップ関数化はNN全体の入出力関係を崩すことになる。(参: 表1 表2)

結果として、今回のようにNNに与える問題がデジタル値で表現できるものならば、個々のユニットの入出力関数と学習後のWが、デジタル化可能である場合が多いことが解る。

3.4 バイアスについての実験結果

当初の実験においては、入力層を除く全てのユニットは、バイアス・ユニットからの出力値(定数: 1)を利用した。これはハードウェアで実現する際には、3層にまたがって配線するといった難がある。しかし今回の実験では、プログラム・シミュレートにおいて、BPの学習方法をバイアス決定にも用いるために考えられた便法⁽²⁾として利用した。

別の考え方として、入力層にバイアス発生用ユニットを1つ用意し、加えて、中間ユニット数を1つ増やすことで、全ニューロンにバイアスをかけた場合と同じ動作を期待できる。実験の結果、後者の方法は、僅かながら学習回数の減少が確認できた。

4. おわりに

本報告で述べた通り、学習結果がデジタルになるために必要なユニットの最低数が存在することが、実験的に解っている。つまり、これ以上のユニットを学習させてしまうことは、無駄であるし、一方ユニット数が少なければ、アナログ値を利用しなければならず、最悪の場合、学習が完了しないことになる。この対応策として、中間ユニット数を学習時に段階的に増加させる方法を検討している。

バック・プロパゲーションを用いたニューラル・ネットワークのデジタル化は、高速化のためだけでなく、動作の安定化にも寄与すると考えられ、今後この方面で積極的な研究を進める必要があると考える。

参考文献

- (1) Edward W. Page, Gene A. Tagliarini: "Algorithm development for neural networks," SPIE Symposium on Innovative Science and Technology, Los Angeles, CA (Jan. 1988)
- (2) 永野俊, 楽谷博司: 神経回路モデルの妥当性, 電子情報通信学会技術研究報告, CSPY87-38(1987)