

最適ニューラルネットワーク構造の探索方法について
～商品先物価格等分析法の一例～

辰 巳 憲 一
禱 道 守

1. はじめに
2. 先行研究の紹介と本研究のねらい
 - (1) 先行研究の内容と結論
 - (2) 先行研究の特徴・欠点
 - (3) われわれのこれまでの研究方法と本研究方法
 - (4) ARCHモデルの有用性
3. 最適ネットワーク構造の探索法
 - (1) 探索の考え方～安定的なバリデーション誤差最小化
 - (2) 探索の前提と手続き
4. ハル・ホワイトARCH (1) モデルの最適ネットワーク構造
 - (1) 探索結果とその分析
 - (2) 最適ネットワーク構造の選び方
 - (3) ネットワーク構造のその他の特徴
5. ハル・ホワイトARCH (2) モデルの最適ネットワーク構造
 - (1) 探索の前提
 - (2) 探索の結果
6. 全体の最適ネットワーク構造
7. 残された課題とANNモデル応用にあたっての示唆
 - (1) 時系列データのフィードバック特性
 - (2) ANN応用についての示唆～時系列フィードバックANN学習法
 - (3) 結論の導出法

1. はじめに

データを集めて計測してみるだけという分析スタイルは、ニューラルネットワーク（以下では時にANNと略）分析については既に昔のものになってしまっている。いまや、ANN分析は未熟な試行段階を大きく超えている。本研究では、与えられた時系列ARCHデータに対して、最適ANN構造をどのように探るべきか、を展開しよう。与えられた時系列データが時系列ARCHデータであるかどうかは事前には確実に判明しないので、データはシミ

ューレートして作るしかない。作られた時系列ARCHデータに対して様々なANNモデルを適用し、そのうちどれを最適とみなすべきかを分析することになる¹⁾。そうすることによって、ANNモデルの適用可能性を高められよう。

2 先行研究の紹介と本研究のねらい

(1) 先行研究の内容と結論

商品価格あるいは商品先物価格の予測によく用いられるのはARモデル (auto-regressive model、自己相関モデル) やARCHモデル (autoregressive conditional heteroscedasticity model、不均一分散自己相関モデル) である。そこで観察・発見されている事実は、商品価格等変化率はマーチンゲールであり、その誤差は商品価格等変化率の水準に単純比例しながら依存する条件付き不均一分散性を持っている、の2つである。ちなみに、ARCHモデルとはこの後者の事実を指す。

例えば、フランクとステンゴス (Frank, M. and Stengos, T. [1]) は1975年1月から1986年6月までの金と銀の日次、週次、隔週価格データを用いて、まずAR(1)モデルの係数値の有意性を棄却し、次に条件付き不均一分散性 (conditional heteroscedasticity) をテストしたが棄却できなかった、のである。

(2) 先行研究の特徴・欠点

これらのテストでは、線形性の検討がなされていない。実際、決定係数などの、当てはめの良さを示す指標は極めて低い。

われわれは、AR系モデルにおける係数間の線形性が実際になりたっているのかを大いに疑っている。そこで、非線形性を容易に処理できるニューラルネットワーク・モデル (neural network model) がこのような問題に対してどのように応用できるのか、どのような成果をあげるのか、特にARCHモデルへの適用可能性を探ったのである。

(3) われわれのこれまでの研究方法と本研究方法

われわれ (禱・辰巳[3]) がこれまで行ってきたのはARCHモデルのANNによる研究である。その分析の手順を簡単に示すと次のようになる。

まず、妥当な初期値と係数値を設定し、金利デリバティブ分野ではハル・ホワイト (Hull-White、以下では時にHWと略) ・モデル²⁾として知られているタイプのARCH(1)モデルから、商品価格変化率値をシミュレートし、サンプル・データとした (数値は後掲の表3を参照)。

このHW・ARCH(1)モデルの特徴は単純な変数変換を施せば線形OLS (ordinary least squares、最小2乗法) で良好な (最小線形不偏) 推計ができることである。このOLS推計結果がベンチマークになる。それゆえ、研究の焦点はニューラルネットワーク

・モデルがどこまでその成績に近づけるか、になる。ANNモデルとしては入力層と出力層が各1つ、中間層が2つのケースに限定した。結論的には、線形推計とかなり近い結果をニューラルネットワーク・モデルは導き出している事実を、袴・辰巳[3]では研究方法の詳細とともに展開し示した。

当てはめの良い計測を単に求めるだけでなく、本研究ではさらに、最新の分析技法であるバリデーション (validation) 法によって最適ネットワーク構造を探り、予測力の高いモデルを求めることに特徴がある。最適ネットワーク構造を探るとは、具体的に、2つの中間層のノードの数を何らかの基準で特定することである。そうすることによって、任意のデータでの高精度の予測が可能になる。

(4) ARCHモデルの有用性

金利や収益率などの変動をモデル化するにあたってARCHモデルが使われるのは、その特性が実態に合っているからである。まず第一に、金利や収益率の水準が高くなれば、それらのボラティリティが実際高くなっている。この良く観察される事実がARCHモデルにモデル化されているのである。第二に経路依存性 (path dependence) がある。経路依存性とは過去の経歴が現在の水準を決める現象である。例えば、正規誤差の場合累積誤差は時間とともにゼロに収束する。しかしながら、ARCHモデルが想定しているように金利や収益率の水準によって誤差が異なる場合この収束性が保証されず、累積誤差は一般にゼロに収束しない。その程度は金利や収益率の時間経路に依存するわけである。この現象は実際的であるとみなされている。

3 最適ネットワーク構造の探索法

(1) 探索の考え方～安定的なバリデーション誤差最小化

ネットワーク構造を決めるにあたって、訓練データの誤差最小化だけを基準にとれば、問題が生じることがよく知られている。ネットワークを大きくすればする程、また計算回数を増やせば増やす程、一般に学習結果が良くなる過学習 (overfitting) という現象があり、訓練データの誤差は原理的にはいくらでも小さくできる。それゆえ、訓練データの誤差最小化だけでネットワーク構造は選ぶようがないのである。同時に他方で、テスト・データでは誤差は益々大きくなっていく。

そこで、バリデーション・テスト (validation test) が考え出された。その手順は、まず訓練データの一部をバリデーション・データとして分離する。そして、残りの訓練データからネットワーク構造を学習し、その推定されたネットワークをバリデーション・データに当てはめ、バリデーション・データとその予測値との差の2乗和平方根 (root mean squared error) をバリデーション誤差 (validation error) とする。

バリデーション誤差は、学習の初期段階では減少していく。しかしながら、ある点から

上昇していく場合がある。これは、バリデーション・データには含まれない、訓練データの誤差（ノイズに相当する情報）までネットワーク構造が学習してしまったためである。したがって、バリデーション誤差が上昇し始める直前までが訓練データの構造を反映していると考えてよい。その点に該当するネットワーク構造を最適と見なせばよい³⁾。

バリデーション誤差がこのように上昇していくことがなく、訓練データの誤差と同じように下落し続けるだけであれば未学習（underfitting）の状態にあるという。

われわれが具体的にとったのは、バリデーション誤差の曲線が明瞭に極小値の存在を示し、しかもそれがすべての最小値にもなっているネットワーク構造を最適として選択する、方法である。ここで、明らかに最小値を示すとは一谷型曲線であることを指す。多峰多谷型の場合最小値がえられても、近辺にはノード間重み付け係数値が少し違うだけで、最小値に近い誤差をもたらすネットワーク構造がいくらかでも存在するからである。

(2) 探索の前提と手続き

とりえる最大のネットワーク構造として、中間層の数は2つまでに限ることにした。そして、HW・ARCH (1) モデルについてはそれぞれのノードの数は8つまで、またHW・ARCH (2) モデルについてはそれぞれのノードの数は10まで考えることにした。結果的にこれらの数は適切であることがわかる。当然ながら、いずれかのノードの数が0の場合中間層が1つのモデルに相当することになる。

原シミュレート・データは平均、標準偏差を計算した後規準正規化した（standard normalized）データに変数変換した。ANNのノード変換関数（node activation function）には、シグモイド関数より一般的な、ハイパータンジント関数 $\tanh(\cdot)$ を指定した。アルゴリズムには、最急降下法ではなく、より精緻ではあるが計算量の多い共役勾配法（conjugate gradient method）⁴⁾を選んだ。ノード間係数の初期値は一様乱数にして学習を始める。バリデーション・データは、慣例に従って、訓練データのなかからシーケンシャル（sequential、つまりデータ系列の期末）を選んだ10%にした⁵⁾。

4 . ハル・ホワイトARCH (1) モデルの最適ネットワーク構造

まず、ハル・ホワイトARCH (1) モデルの最適ネットワーク構造を探ろう。

(1) 探索結果とその分析

ノードの数の様々な組み合わせからなるANNモデルについて、それぞれの訓練テストとバリデーション・テストの結果のうち誤差（root mean squared error、以下では時にRMS誤差、rms errorとも記そう）だけを抜き出し表1に掲げた。それぞれの最終計算回（実は5800回目）の誤差とそれまでの誤差最小値（best error）である。

この表をみると、2つの中間層のうち、いずれかの層のノードの数が2つ以下の小さい

数になると、誤差は著しく大きい。これらのケースについて学習段階毎の誤差の動き（以下ではそれを誤差曲線と呼ぼう）を見ると、未学習（underfitting）状態になっている（例えば図1を参照）ことがわかる。つまり、バリデーション誤差も、訓練誤差と同じように、学習の度に減少していく。それゆえ、さらに、中間層が1つのANN構造はそれが2つのモデルに著しく劣っていることが明らかである。

中間第一層のノードの数と中間第二層のノードの数をその順にベクトル（ \cdot 、 \cdot ）の形で表して考えてみよう。選ぶべき候補となる最適ネットワーク構造としては3つある。バリデーション誤差最小値のもっとも低い（5、4）（7、4）とそれらに次いで低い（7、8）である。

表を見ればわかるように、それらの訓練誤差最小値については、全体の中で必ずしも低くないが、順番は（5、4）（7、8）（7、4）で、後者2者の順番はバリデーション誤差最小値の順番とは逆になる。さて最適ネットワーク構造としてどれを選ぶべきであろうか。テスト誤差や訓練誤差も考慮して考えてみよう。ここでテスト誤差とは、訓練データと同じように100のサンプルをまずシミュレートし、訓練データによって学習した係数推定値を用いてこのテスト・データを予測して得られるRMS誤差である。

(2) 最適ネットワーク構造の選び方

（5、4）の場合、テスト誤差最小値は1.7831（規準正規化データを原データと同じスケールに戻して計算したRMS誤差である。以下も同様。数値は表2に要約した）で特に低くない。訓練誤差最小値も1.1856で3候補の中で悪い方である。しかも、図2を見れば、未学習の嫌いがありそうである。3候補のなかで最も構造が簡単な（5、4）は最適ではないかもしれないわけである。

（7、4）の場合、テスト誤差最小値は1.8174（規準正規化データを原データと同じスケールに戻して計算したRMS誤差である。以下も同様）で少し高いが、訓練誤差最小値は1.1925でかなり悪い。そして、図3からわかるように、バリデーション誤差曲線の動きは不安定で複数の谷があり、ネットワーク構造は不安定である。

ところが、（7、8）の場合、テスト誤差最小値と訓練誤差最小値はそれぞれ1.7603と1.1429でどれよりも低い。テスト誤差最小値が低いこと自体には信頼を余り置けない（テスト・データには偶然性が入り込んでいるからである）。バリデーション誤差最小値が最低でなくても、しかしながら、図4から明瞭なようにバリデーション誤差曲線は明瞭に一つの谷を示している。

それゆえ、上記3候補のうち（7、8）を求める最適ネットワーク構造と考えるべきだろう。

(3) ネットワーク構造のその他の特徴

例えば（0、5）と（5、0）はモデルとして同じなので、表1では中間第一層のノー

ドの数がゼロのケースは計測していない。

例えば、表1においてノードの数の組み合わせが(3、8)と(8、3)などを比べてみればわかるように、中間第一層のノードの数と中間第二層のノードの数を入れ替えても、学習結果は一般に一致しない。

表1 ハル・ホワイト (1) に関する様々なニューラルネットワーク・モデルの訓練誤差とバリデーション誤差

中間第一層の ノードの数	中間第二層の ノードの数	訓 練		バリデーション	
		最終計算 時の誤差	誤 差 最小値	最終計算 時の誤差	誤 差 最小値
1	0	0.916	0.917	1.206	1.025
1	1	0.916	0.919	1.206	1.205
1	2	0.895	0.899	1.221	1.202
1	3	0.880	0.902	1.229	1.191
1	4	0.833	0.833	0.984	0.974
1	5	0.864	0.864	1.184	1.184
1	6	0.875	0.900	1.497	1.196
1	7	0.759	0.797	1.010	0.953
1	8	0.800	0.819	1.142	1.025
2	0	0.898	0.901	1.219	1.021
2	1	0.916	0.919	1.206	1.204
2	2	0.862	0.897	1.237	1.205
2	3	0.864	0.864	1.186	1.185
2	4	0.807	0.831	1.173	1.131
2	5	0.788	0.834	1.187	1.166
2	6	0.864	0.893	1.236	1.207
2	7	0.689	0.774	0.928	0.924
2	8	0.728	0.805	1.066	0.963
3	0	0.875	0.902	1.244	1.212
3	1	0.854	0.856	1.144	1.125
3	2	0.876	0.927	1.250	1.201
3	3	0.819	0.840	1.177	1.141
3	4	0.790	0.814	1.184	1.064
3	5	0.767	0.811	1.188	0.980
3	6	0.758	0.782	1.095	1.048
3	7	0.727	0.808	1.140	1.065
3	8	0.827	0.827	1.207	1.207
4	0	0.858	0.894	1.227	1.199
4	1	0.876	0.956	1.236	1.205
4	2	0.834	0.841	1.186	1.165
4	3	0.826	0.840	1.260	1.133
4	4	0.735	0.827	1.133	1.011
4	5	0.753	0.780	1.134	0.962
4	6	0.756	0.829	1.162	1.017

中間第一層の ノードの数	中間第二層の ノードの数	訓 練		バリデーション	
		最終計算 時の誤差	誤 差 最小値	最終計算 時の誤差	誤 差 最小値
4	7	0.694	0.768	1.106	0.933
4	8	0.730	0.809	1.150	0.988
5	0	0.849	0.849	1.205	1.204
5	1	0.916	0.917	1.206	1.204
5	2	0.786	0.786	0.887	0.885
5	3	0.779	0.808	1.120	1.079
5	4	0.775	0.777	0.870	0.832
5	5	0.735	0.782	1.404	0.967
5	6	0.718	0.795	1.131	1.003
5	7	0.727	0.811	1.116	0.995
5	8	0.685	0.688	0.889	0.859
6	0	0.830	0.835	1.157	1.140
6	1	0.831	0.840	1.168	1.149
6	2	0.782	0.795	1.038	0.961
6	3	0.792	0.807	0.924	0.890
6	4	0.742	0.800	1.123	0.923
6	5	0.723	0.770	1.117	1.025
6	6	0.745	0.780	1.050	0.931
6	7	0.689	0.808	1.029	0.940
6	8	0.690	0.766	1.101	0.897
7	0	0.823	0.827	1.156	1.131
7	1	0.831	0.833	1.172	1.144
7	2	0.798	0.799	1.017	1.003
7	3	0.803	0.817	1.163	1.153
7	4	0.675	0.782	1.103	0.848
7	5	0.696	0.699	1.034	1.022
7	6	0.683	0.818	1.112	0.002
7	7	0.648	0.814	1.085	0.993
7	8	0.646	0.749	1.024	0.585
8	0	0.770	0.778	1.104	1.075
8	1	0.916	0.917	1.206	1.205
8	2	0.861	0.899	1.226	1.196
8	3	0.817	0.822	1.165	1.149
8	4	0.633	0.637	0.952	0.933
8	5	0.761	0.792	1.028	0.967
8	6	0.708	0.797	1.519	1.006
8	7	0.703	0.735	1.119	0.990
8	8	0.732	0.741	1.028	0.995

表2 原データのスケールで表示したRMS誤差

中間第1層 のノード数	中間第2層 のノード数	訓練誤差 の最小値	バリデーション 誤差の最小値	テスト誤差 の最小値
5	4	1.1856	1.2685	1.7831
7	4	1.1925	1.2938	1.8174
7	8	1.1429	1.3086	1.7603

表3 ハル・ホワイトARCH (1) データ

訓練データ		バリデーションデータ	テストデータ	
7.694476	8.17711	4.677808	6.601054	6.43921
10.14464	6.747566	8.547547	6.872335	7.524949
8.597323	6.036705	5.156175	6.154656	7.586064
5.723438	6.074094	4.68475	6.712879	7.32286
6.894374	5.71204	6.905338	6.294471	9.00021
7.423394	7.756592	8.656974	7.270362	7.729644
7.837901	8.837458	7.33067	6.996072	4.259238
7.243143	7.800679	6.3418	6.822262	5.449369
6.48498	6.825215	5.751554	7.609289	5.976331
7.299494	5.748441	8.161506	9.882388	5.987954
4.525199	7.387043	11.31822	11.49573	6.796087
6.991984	5.274335		11.73539	7.572924
7.488942	8.033071		4.347132	4.301917
6.662049	8.10992		5.970709	7.042558
5.68243	11.2527		6.734452	5.401761
6.846771	9.718392		5.485822	8.110504
6.382795	9.121094		6.309261	7.384334
7.535423	7.98306		9.348229	4.050258
7.29476	6.025683		7.536512	5.087628
5.327557	8.028179		8.585148	7.153099
7.867913	7.022923		10.9876	9.01202
9.381422	2.189158		11.81693	6.743494
6.901576	6.051879		5.78913	6.974855
6.25436	8.172299		6.312176	7.571423
7.100467	7.095134		5.208845	8.702676
3.838581	9.179751		7.011498	5.296018
7.653447	10.17128		7.346944	5.350687
7.804873	6.903313		7.429275	7.728578
8.812929	7.952694		9.032601	8.41079
7.979598	11.74015		7.641887	9.068213
7.389384	6.35179		7.990738	8.746124
5.605486	6.894321		7.209559	5.799444
7.25479	7.813654		8.381585	7.877628
8.744672	6.677493		7.006461	4.470657
6.552232	5.336865		10.72326	5.987226
6.243192	6.245044		6.684115	6.986142
7.097473	7.907155		6.745969	6.899416
7.421993	7.120075		4.859864	4.681422
5.964437	6.617914		6.911953	4.724097
6.27902	5.059392		6.572185	9.198798
5.742781	5.735313		7.600098	7.693077
8.696596	7.345939		6.599598	7.745476
8.448842	8.031673		7.252157	8.09773
6.327295	5.454334		6.269543	8.195187
6.504321	8.35119		7.387533	5.2852
	4.677808		3.228015	7.110529
			5.770674	7.302909
			5.437743	7.635862
			7.708768	7.041434
			8.850343	8.015458
				4.547461

図1 ハル・ホワイトARCH (1) に関する中間層ノード数が (8、0) のニューラルネットワーク・モデルの訓練誤差曲線とバリデーション誤差曲線

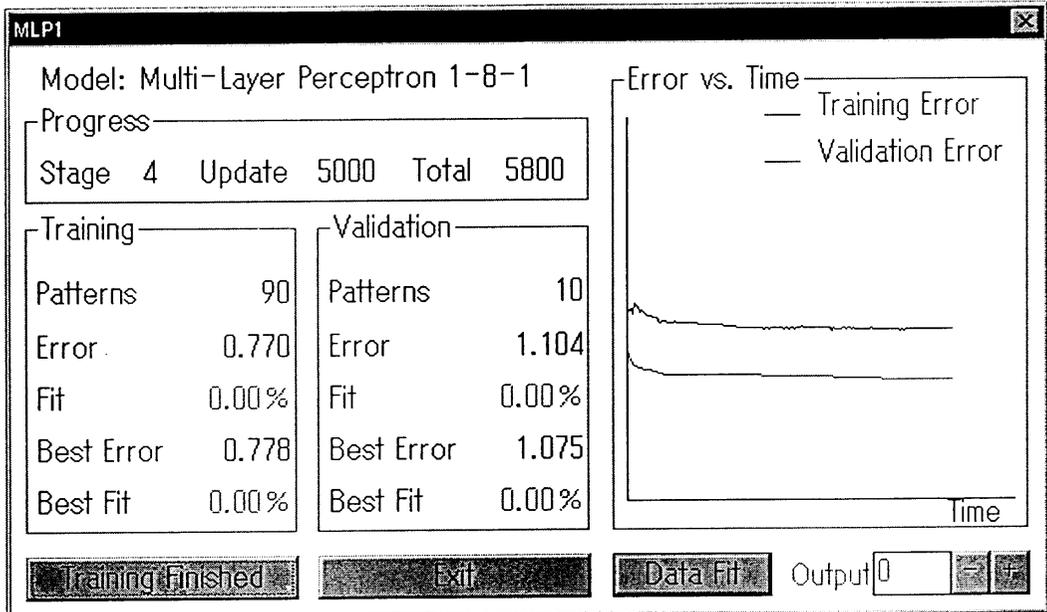


図2 ハル・ホワイトARCH (1) に関する中間層ノード数が (5、4) のニューラルネットワーク・モデルの訓練誤差曲線とバリデーション誤差曲線

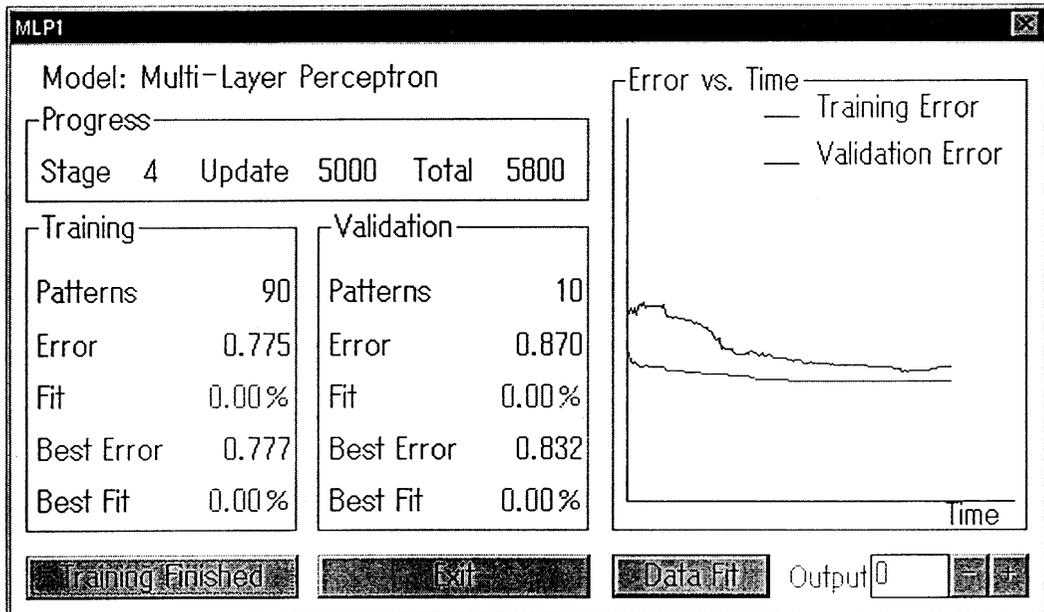


図3 ハル・ホワイトARCH (1) に関する中間層ノード数が (7、4) のニューラルネットワーク・モデルの訓練誤差曲線とバリデーション誤差曲線

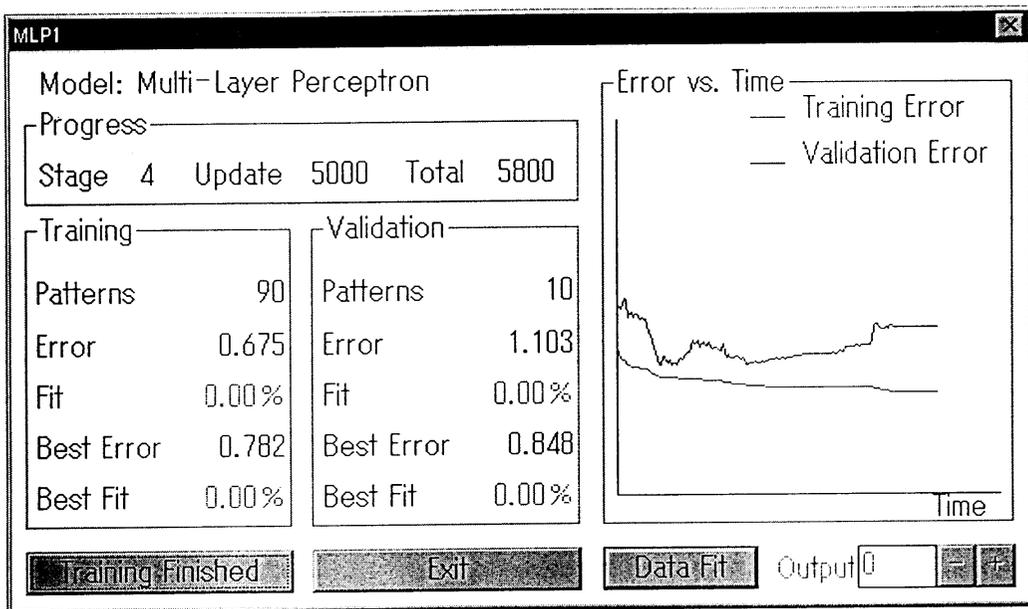
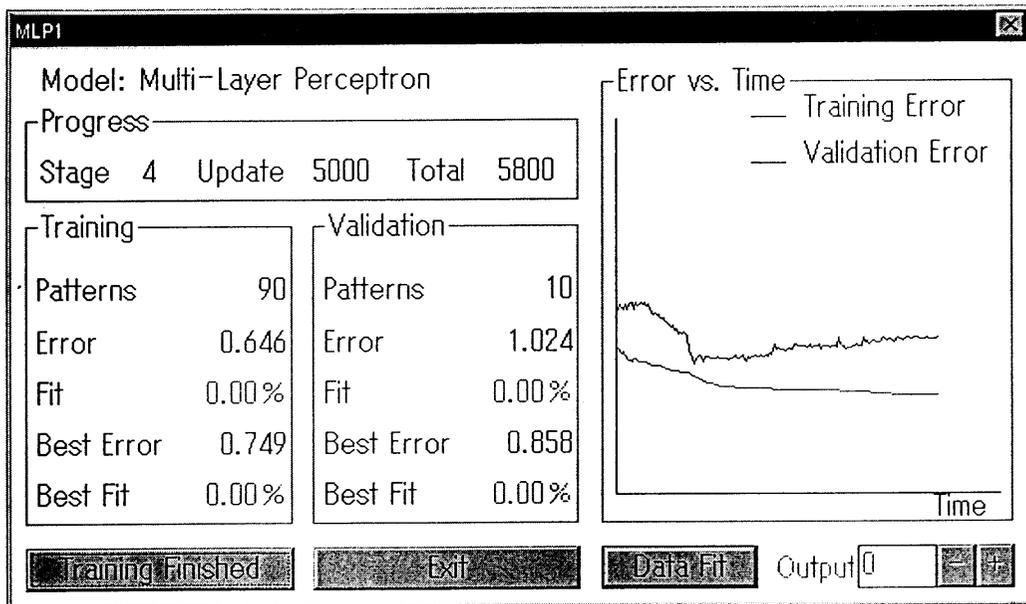


図4 ハル・ホワイトARCH (1) に関する中間層ノード数が (7、8) のニューラルネットワーク・モデルの訓練誤差曲線とバリデーション誤差曲線



5 . ハル・ホワイトARCH (2) モデルの最適ネットワーク構造

次に、ハル・ホワイトARCH (2) モデルの最適ネットワーク構造を探ろう。シミュレート・データはまだ作っていないので、その説明から始めよう。

(1) 探索の前提

データについては、HW・ARCH (2) モデル

$$r_{t+1} = m + b_0 r_t + b_1 r_{t-1} + s r_t e_{0t} + h s r_{t-1} e_{1,t-1}$$

から発生させた。ここで、 e_{0t} と $e_{1,t-1}$ は相互に独立な正規乱数である。係数値は $\mu = 5$ 、 $\sigma_0 = 0.3$ 、 $\sigma_1 = 0.25$ 、 $\rho = 0.2$ 、 $h = 0.5$ 、初期値は $r_0 = r_1 = 5$ と設定して、1、2期前に実現した r の値を代入していくことによって、HW・ARCH (2) データをシミュレートした(数値は表4を参照)。

HW・ARCH (2) モデルへの拡大にともなって、方程式の未定係数の数は3から5に増えるために、最適ネットワーク構造は大きくなることが予想される。しかしながら、上にみたHW・ARCH (1) モデルの結果との比較を容易にするため、換言すれば最適ネットワーク構造の比較可能性を維持するために、我々は中間層の数は増やさずに、ノードの数だけを増やす方法をとってみよう。その他の点はHW・ARCH (1) モデルのケースと同じANN計測法をとることになる。ただし、入力は r_t と r_{t-1} の2つ、出力は r_{t+1} の1つである。

(2) 探索の結果

図5には、主要なケースについてバリデーションの誤差最小値を3次元棒グラフで示した。棒グラフの高さがゼロの場合は計測されていないことを示している。いくつかのケースは既にHW・ARCH (1) モデルの場合で詳しく調べたので、計測はノードの数がより大きいケースに集中したからである。

バリデーションの誤差最小値は(9、7)で達成されるが、その近辺にも候補があり、(8、6)などもバリデーション誤差最小値が低い。現在、これらのうちどちらが最適かを定める根拠はなく、結論を下せない。

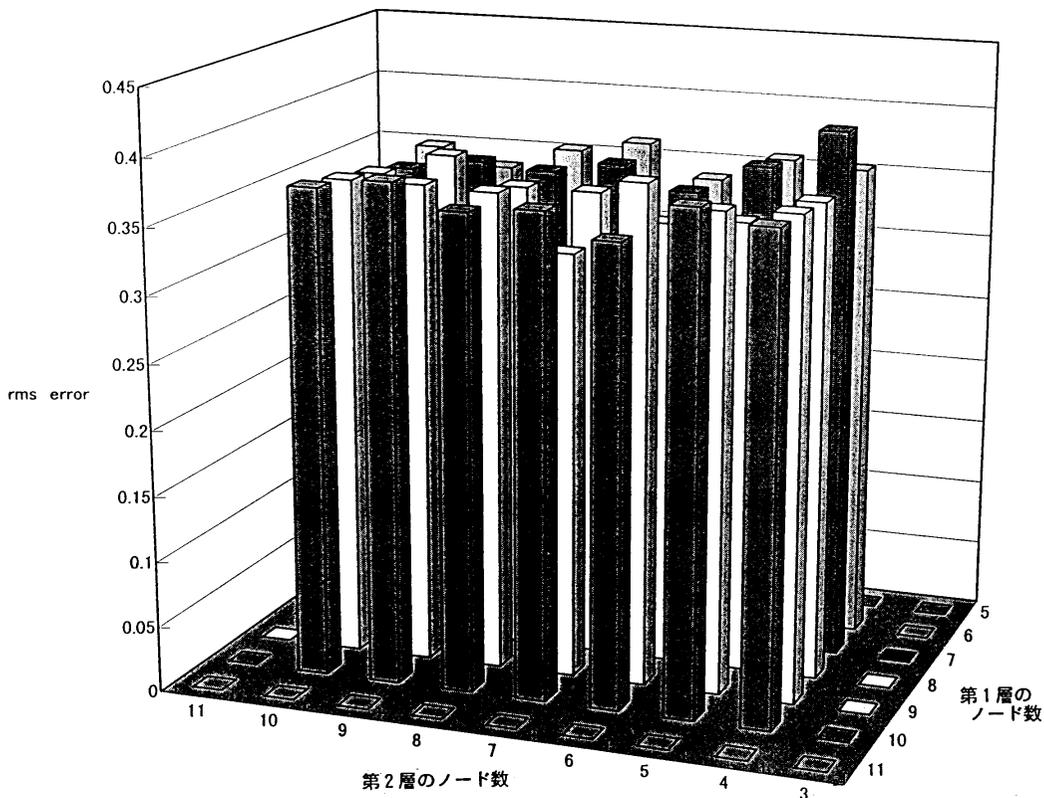
図6と図7を見ればわかるように、これらの最適ANN構造には共通の特徴がみられる。つまり、訓練誤差曲線もバリデーション誤差曲線も学習の初期段階で複雑な動きをする点である。特にバリデーション誤差曲線は最初のいくつかの点で激しい下降ジャンプをしている。4段階の学習段階⁶の変わり目でジャンプするのは、前学習段階での最小誤差(Best Error)に対応する係数に設定し直して計算を再出発するため誤差曲線は急降下するからである。ただし、それは次の学習段階に進む前に最小誤差を既に達成して誤差は反転してかなり上昇している場合に限られる。

また、4つの学習段階を進むに従って学習係数は低められ、モーメント係数は高めら

れる事実も影響していよう。次の学習段階に進み、係数が変われば、極めて急速に誤差を低める学習をする場合があるからである。しかしながら、その学習モードの下バリデーション誤差はいずれ増大していく。

最初の3段階の計算回数を増やせば激しいジャンプはなくなるかもしれない。しかしながら、原因はそれだけでもなさそうであり、アルゴリズムも関係していそうである⁷⁾。

図5 ハル・ホワイトARCH (2) に関する様々なニューラルネットワーク・モデルのバリデーション誤差最小値



第2層のノード数	第1層のノード数					
	10	9	8	7	6	5
10	0.371	0.366	0.360	0.353	0.360	0.364
9	0.379	0.366	0.377	0.362	0.347	0.364
8	0.361	0.364	0.357	0.358	0.364	0.373
7	0.366	0.323	0.357	0.367	0.373	0.349
6	0.348	0.379	0.338	0.350	0.368	0.364
5	0.377	0.363	0.343	0.373	0.368	0.364
4	0.367	0.365	0.362	0.403	0.364	0.364

図6 ハル・ホワイトARCH (2) に関する中間層ノード数が (9、7) のニューラルネットワーク・モデルの訓練誤差曲線とバリデーション誤差曲線

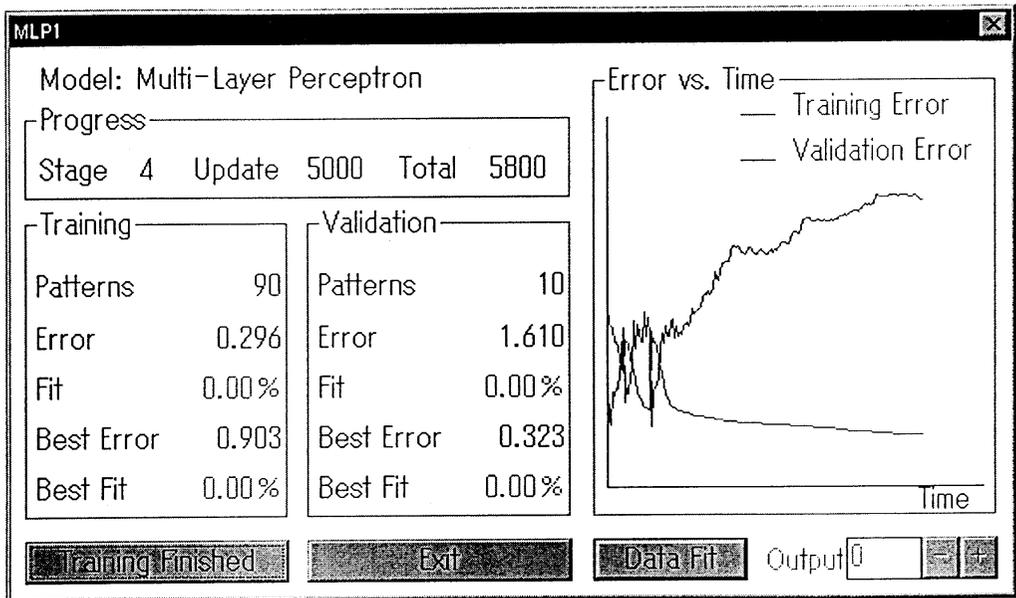


図7 ハル・ホワイトARCH (2) に関する中間層ノード数が (8、6) のニューラルネットワーク・モデルの訓練誤差曲線とバリデーション誤差曲線

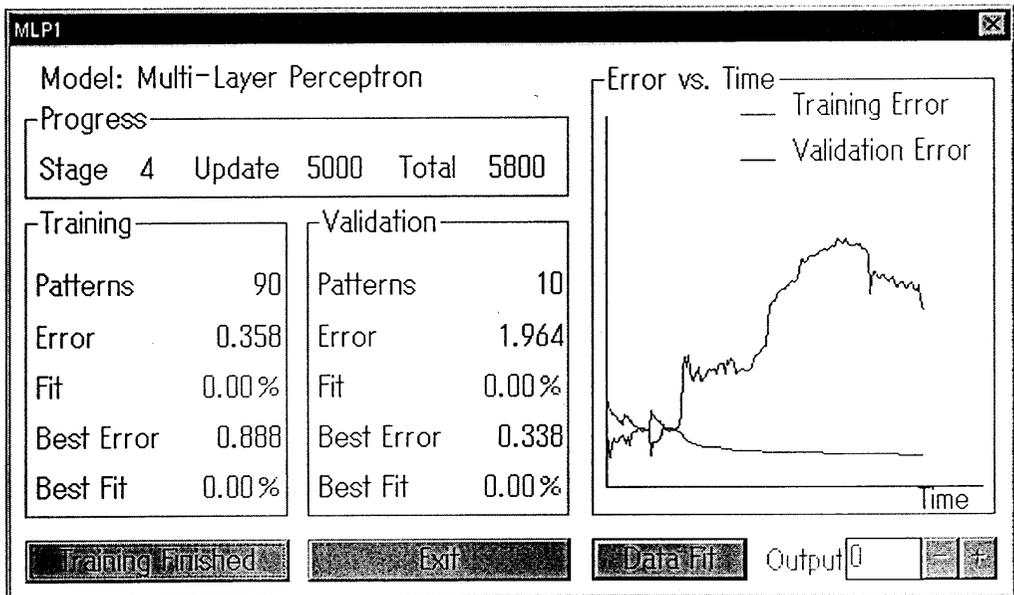


表4 シミュレートしたハル・ホワイトARCH (2) データ

訓練データ		バリ-ションデータ
5	7.266495	7.780819
5	9.760311	7.922398
6.310926	8.688562	10.10159
8.589813	10.91849	8.506415
11.67607	7.031253	6.895927
4.490991	7.648876	8.373016
7.813416	8.145317	7.885874
6.179931	8.96169	7.992165
4.979373	9.81493	7.972507
5.341619	10.2653	7.826126
6.541474	8.995577	9.643073
7.74488	9.121537	9.617218
7.555993	10.96137	
10.39148	12.65969	
8.476172	11.85723	
13.3445	14.42522	
16.06005	14.27177	
15.00464	14.48832	
14.97667	10.76325	
17.36467	9.923314	
11.64793	10.33407	
11.52632	7.322218	
5.888998	11.07994	
6.285278	9.441644	
7.744663	16.23577	
9.145631	15.4052	
5.147791	13.51568	
6.586398	8.809616	
5.725832	11.51159	
8.880813	7.441658	
10.41768	10.61938	
5.608261	9.677566	
9.359662	11.45357	
6.754695	9.670159	
9.130134	12.19651	
8.390215	5.95157	
12.49426	12.28569	
10.88358	10.2512	
11.22026	13.66815	
9.136295	9.772578	
5.479945	11.38302	
9.377299	12.29075	
13.56303	9.002022	
14.11841	3.62148	
11.89084	7.780819	
9.135302	7.922398	

6. 全体の最適ネットワーク構造

それでは、時系列ARCHデータが与えられた時HW・ARCH (1) とHW・ARCH (2) のどちらのモデルを選ぶべきだろうか。バリデーション誤差が小さいという観点では後者を選ぶべきことになるのは確実に予想できる。

しかしながら、モデルが複雑になる欠点は飛躍的に増している。複雑さをもペナルティにするモデル選択法が多く提案されている（辰巳[6]を参照）。もっとも、ペナルティ関数の形やペナルティとRMS誤差の間のウエイトなどについては恣意的であり、まだまだ研究途上・論争の途中であり、決着が着くとは思えない。

7. 残された課題とANNモデル応用にあたっての示唆

(1) 時系列データのフィードバック特性

一般の時系列データ・モデルには良く知られた独特の特性がある。例えばAR (1) モデルを想定してみればよくイメージできるように、モデルの1期前のアウトプット（出力）は当期のインプット（入力）になるのである。モデル構造の妥当性を検証するのが研究の目的ならば、このフィードバック特性を前提にして計測しなければならないだろう。これを考慮した計測方法を時系列フィードバックANN学習法と呼んでおこう。

ちなみに、本稿で用いたデータは上記のようにシミュレートしたものであるが、そのような作り方をしていない。

しかしながら、一般に、推計にあたってモデルのインプットとしては1期前のモデル・アウトプットではなく、1期前の実現値が用いられている。実現値をモデルで表現すると、被説明変数の実現値のことで、説明変数の値に推定された係数値を加重値にして総和し、誤差変数の実現値を加えたものになる。

それは、計量経済学においては、誤差の捉え方から正当化されている。つまり、誤差はモデルの一部であり、説明変数以外の多様な要因が被説明変数の変動をもたらしているという考え方である。

ところが、ANNモデルでは、誤差のモデル化は意図的になされていない。それゆえ、誤差はあくまで誤り（error）であり、できる限り小さい方がよいことになる。また例えば、誤差が非線形性から生じているのなら、そもそもそれは一切ない方がよいわけである。

(2) ANN応用についての示唆～時系列フィードバックANN学習法

何かの時系列データが与えられたとしよう。非線形性が特に強く、それをANN分析やANN予測する必要があるなら、時系列フィードバックANN学習しなければならない。

ちなみに、われわれも、いくつかのケースについて時系列フィードバックANN学習法による計測を行ったが、その結果はすべての点で前節までの結果と大差はなかった。その

原因は、当初設定したARCH (2) モデルの誤差項が説明変数の水準と比較して小さいからであると推察できる。

われわれのANN応用研究はほぼ完全にコントロールした環境で行われた。しかしながら、一般には、誤差項の大きさについて事前の情報は一切持てないと考えるべきであろう。そうすると、まず時系列フィードバックANN学習法を試みるべきだろう。ANNモデルの正しさを検討する作業においては特にそうすべきである。

(3) 結論の導出法

与えられた実際の時系列データからネットワーク構造を学習し最適構造が選べたとしよう。それは、HW・ARCH (1) モデルあるいはHW・ARCH (2) モデルのどちらに対応しているのであろう。そもそもデータは何らかのモデルに対応付けられるのであろうか。

2種類のデータが与えられ、それらの最適ネットワーク構造に大小の差がある場合、しかも上記2つのHW・ARCHモデルだけに対応付けなければならないという条件の下でのみ、対応可能になるに過ぎない。構造の大きい方は、あえて対応付ければHW・ARCH (2) モデルにより近い、と言えるに過ぎないのである。

【脚注】

- *) 本稿は辰巳憲一・袴 道守[7]の改訂版である。本研究には日本商品取引員協会から研究助成を受けたことを記し感謝に代えたい。野澤昌弘氏(東京理科大学)、古山徹氏(日経quick情報)もわれわれが開催した勉強会に積極的に参加されたが、データ作業・論文作成の最終段階で多忙などの理由で論文執筆には参加されなかった。著者連絡先: 学習院大学経済学部(教授)辰巳憲一 〒171-8588 東京都豊島区目白1-5-1、DI: 03-5992-4382、Fax: 03-5992-1007、E-mail: Kenichi.Tatsumi@gakushuin.ac.jp
- 1) ANN全般への入門については多数の啓蒙書・参考書があるが、読みやすいものとしてSmith[4]などを参照のこと。またANN構造の最適性に関する様々な考え方と実際例については現時点では辰巳[6]がもっとも纏まっており、参照のこと。ただし、辰巳[6]は格付け・信用リスクに係わるクロスセクション・データに関するものである。
 - 2) ANNへの応用を念頭に置いた金利モデルの展望は袴・辰巳[3]とその脚注を参照のこと。
 - 3) クロス・バリデーション(cross validation)とは、訓練データとバリデーション・データを入れ替える、モデル検証方法である。バリデーション・データは訓練データに戻し、戻す前の訓練データのなかから新たにバリデーション・データを選ぶ必要が生じる。
 - 4) 共役勾配法(conjugate gradient method)は、事前に設定した学習係数値(learning coefficient)とモーメント係数値(momentum coefficient)に基づき、ノード間係数推定値をつぎの公式で新しい係数推定値に変更していきながら求める。

$$\begin{aligned} \text{係数推定値の変化} = & - \text{学習係数} \cdot (\text{係数値の変化が誤差の大きさに影響する程度}) \\ & + \text{モーメントム係数} \cdot 1 \text{ 期前の係数推定値の変化} \end{aligned}$$

ノード間係数推定値は、一様乱数を初期値にする。計算の各段階においては、直前の係数推定値を元に都度モデルのRMS誤差を計算する。そして、係数値の変化がRMS誤差の大きさに影響する程度を計算して、上記公式に基づいて係数推定値を変更していく。計算を終了するのは、収束条件を満たした時つまりRMS誤差の変化率が0.001以下になった時あるいは当初設定した計算回数(5800回)に達した時である。

学習係数が高い程収束は速いが、最適値への収束は必ずしも保証されない。モーメントム係数は0.9がよく選ばれる(Hertz-Krogh-Palmer[2])という記述もみられるが、研究対象によって違ってこよう。

- 5) これらはいずれも、われわれが使用したソフト(S P S Sニューラル・コネクション)を動かす際のオプションの中から選択したものである。バリデーション・データとしてデータの後ろ端の一塊りを選ぶシークエンシャル(sequential)・バリデーション以外に、一定間隔数毎のデータを選ぶブロックング(blocking)と無作為に選ぶランダム(random)がある。また、アルゴリズムとしては共役勾配法以外に、誤差をもっとも下げる方向を選ぶ最急降下法(steepest descent method)が選べる。後述のように、これら以外のオプションもある。
- 6) 特に記しておくべきは、収束を速める、最初から局所的な極小化に陥らないようにする、などの目的で、計算を学習モードの違う4段階に分けて、それぞれの計算回数を設定できる機能がわれわれが使用したソフト(S P S Sニューラル・コネクション)にはあることであろう。つまり、各学習段階では、前段階の最小誤差のANNモデルを初期値にし、学習係数やモーメントム係数が次の表のように変えられていく。なお、これらの係数値は共役勾配法を実行する場合には変更できない。

段 階	1	2	3	4
学 習 係 数	0.9	0.7	0.5	0.4
モーメントム係数	0.1	0.4	0.5	0.6

各段階の計算回数については、われわれは狙いの最小点を早い段階で目指せるよう、最小点に近づいてからはそれに速く近づくことに配慮して、(100、200、500、5000)と設定した。合計計算回数は5800回である。

上のように設定された学習係数とモーメントム係数の下で、各段階の計算回数を適切に設定することは極めて重要である。われわれはかなりうまく設定できたと判断している。

- 7) 学習係数とモーメントム係数の関係について更に詳しく説明しておかねばならない。上記の係数値設定は、当初第一、二段階ではモーメントム係数を低く、学習係数を高くして、最小値を捜して八方に突き進んでいる。後ろの段階ではこれらの係数値の大小が逆転され、最小値が近いという情報を活用して最小値に効率良く軟着陸する、ことを狙っている。

また、有効学習係数値(effective learning coefficient)にも注意しなければならない

い。もし誤差曲線のほぼ平らな部分を最適化を目指して移動しているとするなら（最小値に近ければそうなる）ば、係数値の変化が誤差に及ぼす影響は小さく、上記共役勾配法公式の係数推定値の変化は毎回ほぼ等しくなり、公式は

$$\text{係数推定値の変化} = - [\text{学習係数} / (1 - \text{モーメントム係数})] (\text{係数値の変化が誤差の大きさに影響する程度})$$

と変形できる。ここで、[]内、つまり学習係数 / (1 - モーメントム係数) が有効学習係数と呼ばれる。4段階の上記数値は有効学習係数値を1に設定しているようにみえる。

ところが、第二段階においてのみ有効学習係数値が $0.7 / (1 - 0.4) = 0.7 / 0.6 = 1.1667$ に設定されている。これが計測結果に大きな影響を与えているように思われる。

【参考文献】

- [1] Frank, M. and Stengos, T., "Measuring the Strangeness of Gold and Silver Rates of Return", Review of Economic Studies, Vol.56 (1989), pp.553-567.
- [2] Hertz, J., Krogh, A. and Palmer, R. G., Introduction to The Theory of Neural Computation, Addison Wesley, 1991.
- [3] 禱 道守・辰巳憲一「金利ARCHモデルによる3つのニューラルネットワーク・ソフトの比較」学習院大学経済経営研究所ディスカッションペーパー 97-9、1998年2月。
- [4] Smith, M., Neural Networks for Statistical Modeling, Thomson Computer Press, 1996.
- [5] 辰巳憲一「信用リスクの推移、その評価法とデリバティブ・プライシング」学習院大学経済経営研究所ディスカッションペーパーNo.97-7、1998年1月。
- [6] 辰巳憲一「格付け・信用リスクの多変量非線形モデルによる計測方法」学習院大学経済経営研究所ディスカッションペーパー 98-4、1998年10月。
- [7] 辰巳憲一・禱 道守「最適ニューラルネットワーク構造の探索方法について～商品先物価格等分析法の一例～」学習院大学経済経営研究所ディスカッションペーパー 98-8、1999年2月。