

ニューラルネットワークによる株価の変動解析

Trend Analysis of Stock Prices by Neural Network

80918124 吉田聡 (Satoru Yoshida) Supervisor: 大森浩充 (Hiromitsu Ohmori)

1. 緒論

一般的な NN の誤差逆伝播法 (BP) は学習レートの設定が難しく、ネットワークが常に入出力関係を正しく学習するとは限らない。特に予測対象が時変の場合、その汎化性能は著しく低下する。そこで本研究では、これらの問題を改善するために、GJR Filter に基づくニューラルネットワークのオンライン学習アルゴリズムを提案する。これは、条件付き分散の非対称性を表すことができる GJR(Glosten/Jagannathan/Runkle) モデルを用いて、ネットワークの入出力の関連性のある程度事前情報として与え、入力に付加し、その信頼度をリアルタイムで補正して学習を行うオンライン学習アルゴリズムである。これにより、入出力の信頼度の高い入力に基づいたリアルタイム予測が可能となり、信頼度の低い入力から多くを学習することを防ぐことができる。また、4 つのパラメータによって特徴付けられる NIG(Normal Inverse Gaussian) 分布による予測値を入力に加えた場合も考察する。そして、提案したオンライン学習アルゴリズムを実際の株価の変動解析に適用することで有効性を検証した。

2. GJR モデル

GJR(p, q) モデルを次に示す。

$$A(z^{-1})y_t = B(z^{-1})\epsilon_t \quad (1)$$

$$\epsilon_t \sim N(0, h_t) \quad (2)$$

$$C(z^{-1})h_t = D(z^{-1})\epsilon_t^2 + E(z^{-1})S_t\epsilon_t^2 \quad (3)$$

ここで、 $A(z^{-1}) = 1 + \sum_{i=1}^r a_i z^{-i}$, $B(z^{-1}) = 1 + \sum_{i=1}^m b_i z^{-i}$, $C(z^{-1}) = 1 - \sum_{i=1}^p c_i z^{-i}$, $D(z^{-1}) = \sum_{j=1}^q d_j z^{-j}$, $E(z^{-1}) = \sum_{j=1}^q e_j z^{-j}$, $c_i, d_j, e_j \geq 0$, y_t は分析対象となる変数, ϵ_t は平均 0, 分散 h_t の独立した正規分布に従うと仮定する。また、式 (3) の S_t は次のダミー変数であり、

$$S_{t-j} = 1 (\epsilon_{t-j} < 0), S_{t-j} = 0 (\epsilon_{t-j} \geq 0)$$

これによりボラティリティの変動の非対称性を考慮できる。また、次の尤度関数を最大にするパラメータを同定する。

$$\ell_t = -\frac{1}{2} \sum_{t=1}^T \ln(h_t) - \frac{1}{2} \sum_{t=1}^T \left(\frac{\epsilon_t^2}{h_t} - 1 \right) \quad (4)$$

3. NIG 分布

NIG 分布は、 α が形状パラメータ、 β が歪度パラメータ、 δ が尺度パラメータ、 μ が位置パラメータという 4 つのパラメータによって特長付けられ、 $\delta > 0$, $\alpha > 0$, $0 \leq |\beta| \leq \alpha$, $\mu \in R$ を満たす。

$$f_{NIG(\alpha, \beta, \delta, \mu)}(x) = A \times \frac{K_1(\alpha \sqrt{\delta^2 + (x - \mu)^2})}{\sqrt{\delta^2 + (x - \mu)^2}} \quad (5)$$

$$A = \frac{\alpha \delta}{\pi} \exp \left(\delta \sqrt{\alpha^2 - \beta^2} + \beta(x - \mu) \right)$$

ここで、 K_1 は第三種の修正ベッセル関数である。また、NIG 分布のモーメント母関数は存在し次式で定義される。

$$M_x(t) = \exp \left(\mu t + \delta \left(\sqrt{\alpha^2 - \beta^2} - \sqrt{\alpha^2 - (\beta + t^2)} \right) \right) \quad (6)$$

$f_{NIG(\alpha, \beta, \delta, \mu)}(x)$ の裾での挙動は次式で定義される。

$$f_{NIG(\alpha, \beta, \delta, \mu)}(x) = O \left(|x|^{-\frac{3}{2}} \exp \{ \beta x - \alpha |x| \} \right) \quad (7)$$

ここで $\gamma = \sqrt{\alpha^2 - \beta^2}$ とおくと、NIG 分布の期待値は $\mu + \delta \frac{\beta}{\gamma}$, 分散は $\delta \frac{\alpha^2}{\gamma^3}$, 歪度は $3 \frac{\beta}{\alpha(\delta\gamma)^{\frac{1}{2}}}$, 尖度は $3 \left(1 + 4 \left(\frac{\beta}{\alpha} \right)^2 \right) \frac{1}{\delta\gamma}$ となる。また、式 (9) に従い、NIG 乱数を生成する。

$$x_{NIG}(t) = F_{NIG}(t) + x(t-1) \quad (8)$$

$$F_{NIG} = \mu + \beta z + \sqrt{z} N_2 \quad (9)$$

ここで、

$$z = \begin{cases} \frac{\delta^2}{V}, & \gamma = 0 \\ \left(\frac{\delta}{\gamma} + \frac{V}{2\gamma^2} \right) - \sqrt{\frac{V\delta}{\gamma^3} + \left(\frac{V}{2\gamma^2} \right)^2}, & U_3 < \frac{\delta}{\delta + \gamma z_1} \\ \frac{1}{z_1} \left(\frac{\delta}{\gamma} \right)^2, & otherwise \end{cases}$$

$N_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2)$, $N_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2)$, U_1, U_2, U_3 は互いに独立な $[0, 1]$ 上の一様分布である。

4. 最適学習レートを用いたオンライン学習アルゴリズム

一般に BPNN において、適切な固定学習レートを決定することは非常に困難である。そこで、新しいデータが利用可能になったとき、そのデータを逐次学習しつつ予測を行える、最適学習レートを用いたオンライン学習アルゴリズムを用いる。時刻 t を新たな訓練ペアが提示された瞬間としたとき、入力層に p ノード、中間層に q ノード、出力層に k ノードを持つ三層 BPNN の出力は次のように表せる。

$$\hat{Y}(t) = F_2 \left[V^T (t-1) F_1 (W(t-1) X(t)) \right] \quad (10)$$

ここで、 $X = (x_0, x_1, \dots, x_p)^T \in R^{(p+1) \times 1}$ は、BPNN の入力ベクトルであり、 $\hat{Y} = (\hat{y}_0, \hat{y}_1, \dots, \hat{y}_k)^T \in R^{k \times 1}$ は BPNN の出力ベクトルを表す。また、

$$W = \begin{bmatrix} w_{10} & \dots & w_{1q} \\ \dots & \dots & \dots \\ w_{q0} & \dots & w_{qp} \end{bmatrix} \in R^{q \times (p+1)}, V = \begin{bmatrix} v_{10} & \dots & v_{k0} \\ \dots & \dots & \dots \\ v_{1q} & \dots & v_{kq} \end{bmatrix} \in R^{(q+1) \times k}$$

$F_1(W(t-1)X(t)) = (F_1(\text{net}_0(t)) \dots F_1(\text{net}_q(t)))^T \in R^{(q+1) \times 1}$, $\text{net}_i(t) = \sum_{j=0}^q w_{ij}(t-1)x_j(t)$, $i = 0, 1, \dots, q$ は、 i 番目の中間ノードの出力である。また $F_1(\bullet)$, $F_2(\bullet)$ は、活性化関数であり、 F_1 は $\tanh$ 関数、 F_2 は線形伝達関数を用いる。評価関数は次のように与える。

$$E(t) = \frac{1}{2} \sum_{j=1}^t \sum_{i=1}^k e_{i,j}^2(t) + \lambda E(t-1) \quad (11)$$

$$e_j(t) = [e_{1,j}(t), e_{2,j}(t), \dots, e_{k,j}(t)]^T \in R^{k \times 1}, j = 1, \dots, t$$

ここで、 λ , $0 < \lambda \leq 1$ はモメンタムを表し、 t は訓練ペアの数を表す。また $e_j(t)$ は時刻 t で考慮される重みを用いて導出された j 番目の訓練ペアにおける誤差を表す。重みは式 (11) に対し、勾配法を適用することで、更新していく。また最適学習レートは下記の式で与えられる。

$$\eta(t) = \frac{[\xi(t)e(t) - \lambda \zeta(t-1)]^T e(t)}{[\xi(t)e(t) - \lambda \zeta(t-1)]^T [\xi(t)e(t) - \lambda \zeta(t-1)]} \quad (12)$$

ただし、

$$\xi(t) = F_2' \left[(F_1^T F_1) I_{k+2} + (X^T X) (\bar{V}^T F_1' F_1 \bar{V}) \right] F_2'$$

$$\zeta(t-1) = F_2' \left[\nabla_V^T E(t-1) F_1 + \bar{V}^T F_1' \nabla_W E(t-1) X \right] F_2'$$

$$\mathbf{F}_2' = \begin{bmatrix} F_{2(1)}' & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & F_{2(t)}' \end{bmatrix}, \mathbf{X}^T \mathbf{X} = \begin{bmatrix} x_1^T x & \cdots & x_1^T x_t \\ \cdots & \cdots & \cdots \\ x_t^T x_1 & \cdots & x_t^T x_t \end{bmatrix}$$

$$\mathbf{F}_1^T \mathbf{F}_1 = \begin{bmatrix} F_{1(1)}^T F_{1(1)} & \cdots & F_{1(1)}^T F_{1(t)} \\ \cdots & \cdots & \cdots \\ F_{1(t)}^T F_{1(1)} & \cdots & F_{1(t)}^T F_{1(t)} \end{bmatrix}$$

$$\mathbf{F}_1' \mathbf{F}_1' = \begin{bmatrix} F_{1(1)}' F_{1(1)}' & \cdots & F_{1(1)}' F_{1(t)}' \\ \cdots & \cdots & \cdots \\ F_{1(t)}' F_{1(1)}' & \cdots & F_{1(t)}' F_{1(t)}' \end{bmatrix}$$

$$F_1' = \text{diag} [f_{1(1)}' \cdots f_{1(q)}'] \in R^{q \times q}$$

$$F_2' = \text{diag} [f_{2(1)}' \cdots f_{2(k)}'] \in R^{k \times k}$$

$$f_{1(i)}' = f_1'(\text{net}_i) = \frac{\partial f_1(\text{net}_i)}{\partial \text{net}_i}, \quad i = 1, 2, \cdots, q$$

$$f_{2(i)}' = f_2' [v_i^T F_1(WX)] = \frac{\partial f_2[v_i^T F_1(WX)]}{\partial [v_i^T F_1(WX)]}, \quad i = 1, 2, \cdots, k$$

また $\bar{V}$ はバイアス成分を除いた純粋な重み行列を表す。モメンタムは下記の式にしたがって適応的に変化させる。

$$\lambda(t) = s(t-1)/s(t)$$

$$s(t) = [(\tau-1)/\tau]s(t-1) + (e(t)^T e(t)/\tau) \quad (13)$$

ここで、 τ はモメンタムの変化分を決める定数である。

5. GJR Filter に基づくオンライン学習法

BPNN を実際に予測器として用いるとき、訓練データにおいて、その入出力関係を正しく学習する必要がある。提案法では、まず入力変数と出力の分散を GJR モデルを用いてリアルタイムで推定し、得られた値に基づいて入力変数の値を修正する。図 1 に GJR Filter に基づく BPNN の構造を示す。ここで、 y は予測対象の時系列データを表し、 x は外部要因を表す。入力データ x, y は h_t の値に基づいて補正された後、前章で言及した学習アルゴリズムを用いて訓練される。GJR Filter では式 (14), (15) に基づいて入力データが補正される。

$$\bar{y}(t-1) = \begin{bmatrix} u_y(t-1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & u_y(t-l) \end{bmatrix} y(t-1) \quad (14)$$

$$\bar{x}_p(t-1) = \begin{bmatrix} u_{x_p}(t-1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & u_{x_p}(t-l) \end{bmatrix} x_p(t-1) \quad (15)$$

ここで、 p は外部変数の数、 l は入力変数のラグである。

$$y(t-1) = [y(t-1), \cdots, y(t-l)]^T$$

$$x_p^T(t-1) = [x_p(t-1), \cdots, x_p(t-l)]^T$$

$$x_p(t-1) = [x_1^T(t-1), \cdots, x_p^T(t-l)]^T$$

$$u_y(t-l) = \frac{u(h_y)}{\sum_{m=1}^p (u(h_y) + u(h_{x_p}))}$$

$$u_{x_p}(t-l) = \frac{u(h_{x_p})}{\sum_{m=1}^p (u(h_y) + u(h_{x_p}))}$$

$$u(h_y) = \exp\left(-\sqrt{h_y(t-l)}\right)$$

$$u(h_{x_p}) = \exp\left(-\sqrt{h_{x_p}(t-l)}\right)$$

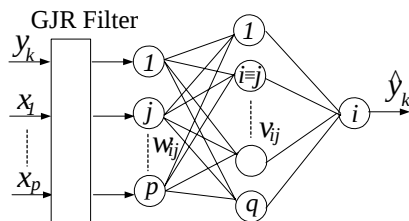


図 1: GJRzNIG Filter を導入した BPNN 構造

表 1: 条件の組み合わせ

入力データ	NN	GJR Filter & NN
y	ケース 1	ケース 4
y, x	ケース 2	ケース 5
$y, x, \text{NIG 乱数}$	ケース 3	ケース 6

6. 提案法によるバリデーション

条件の組み合わせを表 1, 結果である予測パフォーマンスの比較を表 2 に示す。また、表 2 は 20 回の試行の平均値と標準偏差を表示している。また図 2 は、ベストパフォーマンスをグラフ化したものである。

表 2: 予測パフォーマンスの比較

	中間層	RMSE	D_{stat}
ケース 1	5	242.2 ± 2.649	49.11 ± 2.462
	10	239.4 ± 1.891	52.88 ± 1.262
	15	239.9 ± 2.852	51.06 ± 0.434
ケース 2	40	271.5 ± 22.45	58.41 ± 1.489
	80	287.1 ± 47.25	59.76 ± 2.856
	120	265.4 ± 37.16	60.46 ± 1.802
ケース 3	40	269.2 ± 8.153	61.21 ± 1.981
	80	278.9 ± 20.58	61.96 ± 1.704
	120	256.9 ± 12.60	62.77 ± 1.661
ケース 4	5	234.7 ± 1.314	51.80 ± 0.914
	10	235.3 ± 1.211	53.02 ± 0.868
	15	235.0 ± 1.566	52.04 ± 1.070
ケース 5	40	212.9 ± 2.318	66.56 ± 0.930
	80	214.3 ± 3.001	66.88 ± 0.796
	120	216.5 ± 3.158	67.19 ± 0.685
ケース 6	40	209.9 ± 1.113	67.09 ± 0.683
	80	212.5 ± 2.228	67.13 ± 0.813
	120	214.9 ± 1.141	67.27 ± 0.806

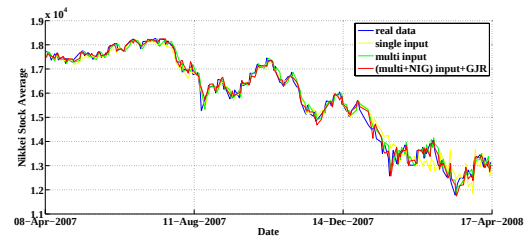


図 2: ベストパフォーマンスの比較

7. 結論

提案手法により、入出力関係を適切に学習することが可能となり、予測精度の向上と安定した予測性能が確認できた。