CENTER FOR ADVANCED RESEARCH IN FINANCE

# CARF ワーキングペーパー

CARF-J-107

# 深層学習を用いた投資手法

福井貴也
みずほ証券株式会社

高橋明彦
東京大学

平成 29 年 7 月 20 日

現在、CARF は第一生命保険株式会社、野村ホールディングス株式会社、株式会社三井住友銀行、株式会社三菱東京 UFJ 銀行、株式会社 FINATEXT、株式会社 GCI アセット・マネジメント、株式会社東京大学エッジキャピタルから財政的支援をいただいております。CARF ワーキングペーパーはこの資金によって発行されています。

# 深層学習を用いた投資手法

福井貴也*
高橋明彦†

平成 29 年 7 月 20 日

**概 要**

本稿では，ニューラルネットワークを用いた深層学習に基づく投資手法を考察する．特に，損失関数の特定化により様々な投資手法を構築できることに着目し，その例として，資産リターン (収益率) の時系列を入力し，教師付き深層学習 (Supervised Deep Learning, SL) によるアノマリー検出を投資判断に活用する手法と，収益最大化に基づく投資判断を深層強化学習 (Deep Reinforcement Learning, RL) により求める方法を示す．また，各資産のリターンに関する学習結果を，複数資産を対象とするポートフォリオ運用についても活用する．

さらに，現実の市場データを用いた検証を行い，対象資産が同じであっても，月次／日次リターン等の入力データの種類, 学習手法の選択，投資期間中の再学習の有無，ネットワークの層数，中間層のユニット数等の外生的に設定する項目により投資パフォーマンスが大きく異なることを例示する．なかでも，RL に基づく投資判断が，ポートフォリオ運用において相対的には安定して良好なパフォーマンスを示すことが明らかとなる．

## 1 序論

深層学習の応用は多くの分野において目覚しい成果を上げている．一般物体認識の分野では Krizhevsky 等 [1] が深層学習を応用して高精度で画像の被写体の判別を行っている．また，一時点のデータの判別だけでなく，パターンの時間方向の組み合わせが意味を持つような音声認識の分野でも，Hinton 等 [2] によって深層学習を応用して高精度で判別を行う方法論が提案されている．このことは深層学習の応用が時系列の予測において効果的であることを示唆している．さらにゲームの人工プレイヤーの意思決定に対しても深層学習は適用され，大きな成果が得られている．(Mnih 等 [3]) このことはある環境下においてエージェントが得られる利得（ゲームのスコアや勝敗）を最大化するような意思決定問題に対して深層学習が効果的であることを示唆している．

物体認識・音声認識で応用されている学習は教師付き深層学習 (Supervised Deep Learning, SL) と言われる．それは，画像データのピクセル値や音声データの周波数などの説明変数を入力し，被説明変数である，画像の被写体の種類や音声の意味等の観測値に対する予測値を出力するニューラルネットワークのパラメータ推定を行う．推定は観測値とその予測値を用いて定義される損失関数の最小化問題を数値的に解くことにより行われる．

*みずほ証券株式会社
†東京大学

一方，ゲームなどの人工プレイヤーの意思決定に応用されている学習は，深層強化学習 (Deep Reinforcement Learning, RL) と言われる．RL では入力を画像データ等のゲームの状態を表す変数，出力をエージェントの行動様式とし，学習期間で得られる総利得を最大化するようにニューラルネットワークのパラメータを推定する．

また，ファイナンスの分野においても時系列の予測や市場環境に基づく意思決定は重要な問題である．これまでは金融市場のモデル化を行い，そのモデル・パラメータの推定値に基づき意思決定が行われてきた．近年では，上述の他分野で得られた示唆に基づき，多くの研究者が金融市場の意思決定の分野で深層学習を適用した分析を行っている．

SL を応用した研究には，Yudong-Lenan[4], Araujo 等 [5], Patel 等 [6] などがある．分析対象となる時系列について [4] では多クラスへの分類, [5], [6] では 2 つのクラスへの分類を行い，入力にはテクニカル指標の値を，出力にはリターンの大小に基づく分類結果を設定したニューラルネットワークのパラメータを教師付き学習で求めている．

RL を応用した研究には，Zhang-Maringer[7] や Deng 等 [8] などがある．[7] ではテクニカル指標，[8] では分析対象の過去の時系列データを入力し各期の資産の投資量を出力するようなニューラルネットワークを設定し，投資により得られる収益の最大化を目標として強化学習が行なわれている．

しかしながら，これらの研究は，ある一つの資産に着目しその資産のみへの投資のパフォーマンス分析が主であり，さらに SL と RL のどちらかを選択し分析を行っている．これに対し，本研究では複数の資産を対象に SL，RL 両方の分析を行い，その結果を個別資産への投資のみならず，ポートフォリオを対象とする投資の意思決定に活用する方法を提案する．また，本研究ではテクニカル指標など複雑な計算により得られるものはニューラルネットワークの入力として採用せず，入力データの加工は観測データをその平均と分散によって標準化することにとどめる．一方，多層順伝播ニューラルネットワークを構成し，テクニカル指標に相当する特徴量が出力と入力の間のネットワークの中で内生的に生成され，分析者の変数選択の負担を軽減することを企図する．但し，観測データのノイズを除去するために指数移動平均の利用は許容する．

以上の設定の下で実データを用いた検証を SL, RL 各々について行い，投資のパフォーマンスを比較する．尚，資産価格の表示通貨が円建てとは異なる（US ドル建ての）場合は，為替（FX）リスクのヘッジの意思決定も学習に基づき同時に行う状況も想定する．

本稿の構成は以下のとおりである．次節では，ニューラルネットワークの出力を投資判断に活用する手法を例示する．第 3 節では本稿で用いるネットワーク構造や活性化関数を，第 4 節では多層ネットワークの学習手法を概観する．第 5 節，第 6 節では，各々月次データ，日次データをネットワークの入力データに用いて検証を行う．第 7 節で結論を述べる．

## 2　ネットワーク出力の投資判断への活用

ニューラルネットワークを投資手法に応用する場合，ニューラルネットワークの入力 $X$ としてマーケットで観測されるデータを用いる．特に本研究では，各資産のリターン (収益率) の時系列を入力 $X$ としてニューラルネットワークの学習を行い，その結果から 1 期先の投資判断を得る手法を構築する．また，ニューラルネットワークの初期値を効率的に決めるため，Deep Belief Network(DBN) に基づく事前学習を行う．

本稿における学習とは，ニューラルネットワークの出力 $\hat{Y}(\theta)$ から計算される損失関数 $E(\theta) = E(\hat{Y}(\theta))$ を最小化するニューラルネットワークのパラメータ $\theta$ を求めることである．以下では，投資への応用のためのニューラルネットワークの出力と損失関数を SL, RL 各々に関し定義する．

## 2.1 リターンの分類に基づく教師付き深層学習

本節では，学習によりアノマリーを検知し，その結果に従い投資判断を行う手法を構築する．特に，出力 $\hat{Y}(\theta)$ を，リターンの観測値を閾値により分類した指標に対する予測値と定義し，その指標の実現値 $Y$ を被説明変数とする．損失関数としては，$\hat{Y}(\theta)$ の $Y$ に対する適合度を表す関数を設定する．$(X, Y)$ を教師データとして SL を実行することにより，アノマリーの予測を行うことができる．

具体的には，まず学習データの生成時点を $T = \{T_s, \ldots, T_e\}$ とし，各時点で観測されたリターン $\{R_t\}_{t\in T}$ を $N_c$ 種類に分類する．$R_t$ が $c$ 番目の種類に分類された時，その分類結果を用いて，ニューラルネットワークの学習の被説明変数 $\{Y_{t,k}\}_{k=0,\ldots,N_c-1}$ を $k = c$ の場合，$Y_{t,k} = 1$，それ以外の場合は $Y_{t,k} = 0$ となるように生成する．また，$t$ 時点より以前の情報から説明変数 $X_t$ を生成し，教師データ $(X_t, Y_t)$ をすべての $t \in T$ で生成する．

上記で生成した教師データに基づきリターンの値の分類を学習するため，出力層の出力ユニットに (3.2 節で導入する) Classification Unit が設定された $L$ 層のニューラルネットワークを構築する．以上により，説明変数 $X_t$ から後述する式 (3), (4) で出力層のユニットの出力変数 $z^{(L+1)}$ を計算し，その値と後に示す式 (10), (11) から指標に対する予測値 $\hat{Y}_t$ を計算できる．(但し，3.2 節の $\{1, 2, \cdots, N^{(L)}_{\mathrm{OUT}}\}$ は本節の $\{0, 1, \cdots, N_c - 1\}$ に対応する.)

この予測値 $\hat{Y}_t$ を指標の実現値 (被説明変数)$Y_t$ に近づけることが目的であるため，損失関数としては，以下の式 (1) により定義される，推定すべきパラメータ $\theta$ を伴ったクロス・エントロピー $E(\theta)$ を設定する．

$$E(\theta) = \sum_{t\in T} E_t(\theta),\ E_t(\theta) = -\sum_{k\in C} Y_{t,k} \ln[p(\hat{Y}_{t,k} = 1|\theta)] \tag{1}$$

但し，$C = \{0, 1, \cdots, N_c - 1\}$ である．学習は $E(\theta)$ を最小化するようなパラメータ $\theta$ を得ることが目的であるため，$E(\theta)$ の $\theta$ に関する最適化問題を解く．具体的には $\partial E(\theta)/\partial\theta$ を計算し，数値的に最適化問題を解くことにより最適な $\theta$ を得る．(具体的な方法は後述する.)

学習により最適な $\theta$ が得られた後は，学習期間外 (out-of-sample) の $t'$ 時点において，($t'$ 時点以前の資産価格，リターン等) 時点 $t'$ より以前の情報に基づく入力 $X_{t'}$ から指標の予測値 $\hat{Y}_{t'} = \{\hat{Y}_{t',k}\}_{k\in C}$ を計算し，その値と外生的に与えたパラメタに従い投資の意思決定を行う．

例えば，第 5，6 節の実証分析では，リターンを 3 種類に分類 ($C = \{0, 1, 2\}$) し，学習により得た $\theta$ を用いて，入力データ $X_{t'}$ から指標の予測値 $\hat{Y}_{t',k}(k = 0, 1, 2)$ に関する確率，$P_0 = p(\hat{Y}_{t',0} = 1|\theta)$, $P_1 = p(\hat{Y}_{t',1} = 1|\theta)$, $P_2 = p(\hat{Y}_{t',2} = 1|\theta)$ を計算する．後述する式 (62), (63) の定義により，$1 - P_1$ は $R_{t'}$ がアノマリーである確率を表すと解釈できる．特に，検証に用いる投資ルールとしては，$P^{th}_1, P^{th}_2$ を外生的に与えるハイパー・パラメータとして，$P_1 < P^{th}_1$ かつ $P_2 - P_0 > P^{th}_2$ となる場合に買い（ロング）のシグナルを生成するルール

を採用する．即ち,「アノマリーが生ずる確率が $1-P_1^{th}$ 以上」かつ,「リターンの高いアノマリーが低いアノマリーよりも，$P_2^{th}$ より大きい確率で生じる」場合にロングする，という投資戦略である．尚，他の投資戦略等の検討は今後の課題としたい．

### 2.2 収益最大化に基づく投資判断の深層強化学習

特定の資産を投資対象とした場合に次の投資戦略を考える．学習期間中のあらかじめ決められた期間ごとにその資産のロングポジションを取るか否かの意思決定を行い，次の意思決定の直前にロングポジションを解消することを学習期間中に繰り返す．

具体的には，学習期間中のポジションが清算される時点を $T = \{T_s, \ldots, T_e\}$ として，ニューラルネットワークの出力を，時点 $t \in T$ においてロング (買い) ポジションをとるか否かを示す 2 値変数 $\hat{Y}_t$ とする．つまり，時点 $t$ でロングポジションを取るときは $\hat{Y}_t = 1$，とらないときは $\hat{Y}_t = 0$ である．

この投資戦略におけるロングポジションをとるか否かの意思決定を RL により得るために，$t$ 時点より以前に得られた情報 $X_t$ を入力して $\hat{Y}_t$ を出力するような $L$ 層のニューラルネットワークを構成する．$\hat{Y}_t$ は 2 値変数であるからニューラルネットワークの出力層の出力ユニットは 2 値のユニット (Binary Unit) を 1 つだけ設定する．

このようなニューラルネットワークに基づき，変数 $X_t$ を入力し，後述する式 (3), (4) を用いて出力層のユニットの出力変数 $z^{(L+1)}$ を計算する．その値と後述の式 (6) により確率変数 $\hat{Y}_t$ が得られる．(但し，$N_{\mathrm{OUT}}^{(L)} = 1$ で $\hat{Y}_t = \hat{Y}_{t,1}$ と置く．)

次に，この投資戦略により得られる，投資対象資産の $t$ 時点における収益 $\hat{R}_t$ は $\hat{R}_t = \hat{Y}_t R_t$ と表される．また，学習期間におけるリターンの観測値 $R_t (t \in T)$ を所与とすると，期待収益 $\tilde{R}_t(\theta) = E[\hat{R}_t | R_t, \theta]$ は $\tilde{R}_t(\theta) = p(\hat{Y}_t = 1|\theta) R_t$ と表現できる．本稿の学習においては，この $\tilde{R}_t(\theta)$ を当該戦略に基づく時点 $t$ の収益と見做して，総収益を $U(\tilde{R}_{T_s}, \ldots, \tilde{R}_{T_e}) = \sum_{t=T_s}^{T_e} \tilde{R}_t$ により定義する．以上の設定により，ニューラルネットワークのパラメータ $\theta$ は，4 節で概説する手法を用い，次の最適化問題を解くことで得られる．

$$\max_{\theta} \; U(\tilde{R}_{T_s}(\theta), \ldots, \tilde{R}_{T_e}(\theta)) = \max_{\theta} \; \sum_{t=T_s}^{T_e} \tilde{R}_t(\theta) = \max_{\theta} \; \sum_{t=T_s}^{T_e} p(\hat{Y}_{t,1} = 1|\theta) R_t. \qquad (2)$$

ここで損失関数を $E(\theta) = -U(\tilde{R}_{T_s}(\theta), \ldots, \tilde{R}_{T_e}(\theta)) = \sum_{t=T_s}^{T_e} E_t(\theta)$, $E_t(\theta) = -\tilde{R}_t(\theta)$ と定義すれば，$\theta$ の学習は $E(\theta)$ の最小化問題として扱うことができる．(最適化の具体的な方法については後述する．)

学習より得た $\theta$ を用い，学習期間外 (out-of-sample) の $t'$ 時点に関して，まず，($t'$ 時点以前の資産価格，リターン等) 時点 $t'$ より以前の情報に基づく入力 $X_{t'}$ から $P_1 = p(\hat{Y}_{t',1} = 1|\theta)$, $P_0 = p(\hat{Y}_{t',1} = 0|\theta) = 1 - p(\hat{Y}_{t',1} = 1|\theta)$ を計算する．次に，その値に基づき，(リターン $R_{t'}$ を未知として) $t'$ 時点で結果が明らかとなる投資の意思決定を行う．実データによる検証に用いる投資ルールとしては，$P_1 > P_0$ となるとき，即ち，$P_1 > 0.5$ となる場合にロングする (買う) 戦略を採用する．

尚，他の投資戦略，損失関数の検討等は次の論文に譲ることとしたい．例えば，以下の設定も考えられる．学習期間のリターンの観測値 $R_t (t \in T)$ を所与とすると，$t$ 時点の収益の分散 $V_t(\theta) = E[\hat{R}_t^2 | R_t, \theta] - (\tilde{R}_t(\theta))^2$ は $V_t(\theta) = R_t^2 [p(\hat{Y}_t = 1|\theta)\{1 - p(\hat{Y}_t = 1|\theta)\}]$

と表現できる．そこで，$\tilde{R}_t(\theta)$ の他，$V_t(\theta)$ も用いて，損失関数を $E(\theta) = \sum_{t=T_s}^{T_e} E_t(\theta)$, $E_t(\theta) = \gamma V_t(\theta) - \tilde{R}_t(\theta)$ ($\gamma$:正定数) と定義して $\theta$ を学習し，この $\theta$ に基づき投資ルールを決定する．

# 3　多層順伝播ネットワーク

## 3.1　ネットワーク構造

多次元の説明変数の観測値 $X$ が与えられた時に多次元の被説明変数 $Y$ の予測値 $\hat{Y}$ を与えるため，入力と出力に複数ユニットを持つ $L$ 層のニューラルネットワークを考える．ここで，第 $l \in \{1, \ldots, L\}$ 層のネットワークの入力のユニット数を $N_{\rm IN}^{(l)}$，出力を $N_{\rm OUT}^{(l)}$ としたときに $l = 1, \ldots, L$，$N_{\rm OUT}^{(l)} = N_{\rm IN}^{(l+1)}$ となる．ネットワークの構造 $\{N_{\rm IN}^{(l)}\}_{l=1,\ldots,L+1}$ とユニットの種類は外生的にあたえられる．さらに，ユニットの間の関係性を表すパラメータとして $N_{\rm OUT}^{(l)} \times N_{\rm IN}^{(l)}$ 行列で表される各層のユニット間の重み $W^{(l)} = [w_{ij}^{(l)}]$ と $N_{\rm OUT}^{(l)}$ 次元ベクトルで表される出力層のバイアス $b^{(l)} = [b_i^{(l)}]$ が $l = 1, \ldots, L$ で与えられる．このネットワークのパラメータを $\theta = \{\theta^{(l)}\}_{l=1,\ldots,L}$, $\theta^{(l)} = \{W^{(l)}, b^{(l)}\}$ と表す．

ニューラルネットワークの予測値の計算のために各層のネットワークの出力ユニットごとに入力側と出力側の潜在変数を定義する．いま，$l = 1, \ldots, L$ で第 $l$ 層の出力ユニットごとに定義された $N_{\rm OUT}^{(l)} = N_{\rm IN}^{(l+1)}$ 個の入力側の潜在変数を $h^{(l+1)} \in R^{N_{\rm IN}^{(l+1)}}$，出力側の潜在変数を $z^{(l+1)} \in R^{N_{\rm IN}^{(l+1)}}$ で表す．予測値の計算ではまず $N_{\rm IN}^{(1)}$ 次元の入力データ $X$ が与えられたとき，$z^{(1)} = X$ とすることでニューラルネットワークに $X$ を入力し，以下のように $h^{(2)}, z^{(2)}, h^{(3)}, \ldots, z^{(L)}, h^{(L+1)}, z^{(L+1)}$ を逐次求めていく．

$$h_i^{(l+1)} = \sum_{j=1}^{N_{\rm IN}^{(l)}} w_{ij}^{(l)} z_j^{(l)} + b_i^{(l)} \tag{3}$$

$$z_i^{(l+1)} = f(h_i^{(l+1)}),\ i = 1, \ldots, N_{\rm OUT}^{(l)} \tag{4}$$

ここで，$f$ は各 $l$ 層において異なる関数を用いることも可能だが，本稿では，出力層 (第 $L$ 層) を除き同一の関数を適用するため，簡略化のため同一の記号 $f$ を用いる．

最後に，$z^{(L+1)}$ を用いて出力として $N_{\rm OUT}^{(L)}$ 次元のベクトルの予測値 $\hat{Y}$ を得る．但し，$\hat{Y}$ は出力層の出力ユニットの種類により異なる．

学習は，$t$ 時点以前の情報から得られる説明変数 $X_t$ を入力し，被説明変数の $t$ 時点の予測値 $\hat{Y}_t$ を計算したのち，$\hat{Y}_t$ と観測実現値より計算される損失関数を最小化する $\theta$ を推定することである．なお，SL と RL の違いは，前節でみたように，損失関数 $E(\theta)$ の定義の違いで表現される．$E(\theta)$ を所与とすれば，$\partial E(\theta)/\partial\theta$ を計算しでパラメータ $\theta$ を逐次更新する．具体的な方法は第 4 節で説明する．

次に，学習により得られた $\theta$ に基づき，ニューラルネットワークを用いて学習期間外 (out-of-sample) の時点 $t'$ における（$t'$ 時点より以前の情報から得られる）説明変数 $X_{t'}$ から予測値 $\hat{Y}_{t'}$ を計算する．前節で概観したように，本稿ではこの予測値 $\hat{Y}_{t'}$ を投資の意思決定に活用する．

### 3.2 活性化関数とネットワークの出力

本稿に関連する活性化関数とネットワークの出力形態について概説する．

**Binary Unit** 式 (4) における活性化関数 $f$ は以下のロジスティック関数で与えられるとする．

$$z_i^{(l+1)} = f(h_i^{(l+1)}) = \sigma(h_i^{(l+1)}) = \frac{1}{1+\exp(-h_i^{(l+1)})},\ i = 1, \ldots, N_{\mathrm{OUT}}^{(l)} \tag{5}$$

さらに，ニューラルネットワークの出力が 0 または 1 を取る値のベクトルで表され，その要素はベルヌーイ分布から取り出された乱数であることを想定する．具体的には，ある入力サンプル $X_t$ から式 (3), (4) によって第 $L$ 層（出力層）の出力ユニットの入力潜在変数 $h_t^{(L+1)}$ が得られる．この値を用いてネットワークの出力 $\hat{Y}_t$ を以下のようにして得る．

$$\hat{Y}_{t,j} = \begin{cases} 1 & \text{with prob. } \sigma(h_{t,j}^{(L+1)})(= z_{t,j}^{(L+1)}) \\ 0 & \text{with prob. } 1-\sigma(h_{t,j}^{(L+1)})(= 1 - z_{t,j}^{(L+1)}) \end{cases},\ j = 1, \ldots, N_{\mathrm{OUT}}^{(L)} \tag{6}$$

**Rectified Linear Unit** 式 (4) における活性化関数 $f$ は以下の Rectified Linear Unit (ReLU) で与えられるとする．

$$f(h_i^{(l+1)}) = \max\{0, h_i^{(l+1)}\} \tag{7}$$

さらに，後述の 4.2.3 節で導入するネットワークの出力においては，次の Noisy Rectified Linear Unit (NReLU) (Nair-Hinton[9]) を用いる．即ち，まず，ある入力サンプル $X_t$ から式 (3), (4) により計算された出力層の各ユニットに対する入力の潜在変数を $h_t^{(L+1)}$ と表す．この潜在変数を用いて，以下によりネットワークの出力値 $\hat{Y}_t$ を得る．

$$\hat{Y}_{t,j} = \max\{0, h_{t,j}^{(L+1)} + u\},\ j = 1, \ldots, N_{\mathrm{OUT}}^{(L)} \tag{8}$$

$$u \sim N(0, \sigma(h_{t,j}^{(L+1)})) \tag{9}$$

但し，$N(0, \sigma(h_{t,j}^{(L+1)}))$ は平均 0, 分散 $\sigma(h_{t,j}^{(L+1)})$ の正規分布を表す．

**Classification Unit** ニューラルネットワークの出力が，時点 $t$ の分類を表すベクトルであることを想定する．この想定における第 $L$ 層の出力ユニット ($L+1$ 番目のユニット) として，Classification Unit を考える．

活性化関数は Soft-max 関数とし，出力層における出力ユニットの入力潜在変数 $h^{(L+1)} = (h_i^{(L+1)})_{i=1,\cdots,,N_{\mathrm{OUT}}^{(L)}}$ をこの関数に代入し，出力潜在変数 $z^{(L+1)} = (z_i^{(L+1)})_{i=1,\cdots,,N_{\mathrm{OUT}}^{(L)}}$ を以下により計算する．

$$z_i^{(L+1)} = f(h_i^{(L+1)}) = \left[\sum_{j=1}^{N_{\mathrm{OUT}}^{(L)}} \exp(h_j^{(L+1)})\right]^{-1} \exp(h_i^{(L+1)}),\ i = 1, \ldots, N_{\mathrm{OUT}}^{(L)} \tag{10}$$

従って，$z^{(L+1)}$ は，各ユニット $i = 1, \ldots, N_{\mathrm{OUT}}^{(L)}$ において，$0 \leq z_i^{(L+1)} \leq 1$ で，その合計は，$\sum_{i=1}^{N_{\mathrm{OUT}}^{(L)}} z_i^{(L+1)} = 1$ となる．

ここで，想定している分類の集合を $C = \{1, \ldots, N_{\mathrm{OUT}}^{(L)}\}$ とし，入力 $X_t$ からあるネットワークを通して得られる $h_t^{(L+1)} = (h_{t,i}^{(L+1)})_{i=1,\cdots,,N_{\mathrm{OUT}}^{(L)}}$ を所与とすれば，式 (10) で $h_i^{(L+1)} = h_{t,i}^{(L+1)}$ と置いて計算される $z_i^{(L+1)}$ は，入力 $X_t$ に対して分類 $i$ が実現する確率を表すと解釈できる．これを用いて，確率変数 $\hat{Y}_{t,i}$ $i = 1, \ldots, N_{\mathrm{OUT}}^{(L)}$ を，$P(\hat{Y}_{t,i} = 1) = z_i^{(L+1)}$, $P(\hat{Y}_{t,i} = 0) = 1 - z_i^{(L+1)}$ により定義する．

また，取り得る値の集合が $C$ で，その確率分布が $[z_{t,1}^{(L+1)}, \ldots, z_{t,N_{\mathrm{OUT}}^{(L)}}^{(L+1)}]$ である確率変数 $c$ (つまり，$P(c = i) = z_i^{(L+1)}$) を用いて，$\hat{Y}_{t,i}$ を以下で表すこともできる．

$$\hat{Y}_{t,i} = \begin{cases} 1 & \text{if } i = c \\ 0 & \text{otherwise} \end{cases}, \ i = 1, \ldots, N_{\mathrm{OUT}}^{(L)} \tag{11}$$

# 4 多層ネットワークの学習法

## 4.1 誤差逆伝播法

いま，観測されるデータセット $D$ に属するあるサンプル $d_t \in D$ について推定すべきパラメータ $\theta$ に対する損失関数 $E_t(\theta) = E_t$ の勾配を求める．ただし SL を実行する時は $d_t$ は教師データであり $d_t = (X_t, Y_t)$ となる．一方，RL を実行する時は $d_t$ はニューラルネットワークの入力データで $d_t = X_t$ である．$\frac{\partial E_t}{\partial \theta}$ は，$w_{ij}^{(l)}, b_j^{(l)}$ に関して各々以下で表される．

$$\frac{\partial E_t}{\partial w_{ij}^{(l)}} = \frac{\partial E_t}{\partial h_i^{(l+1)}} \frac{\partial h_i^{(l+1)}}{\partial w_{ij}^{(l)}} = \frac{\partial E_t}{\partial h_i^{(l+1)}} f(h_j^{(l)}) \tag{12}$$

$$\frac{\partial E_t}{\partial b_j^{(l)}} = \frac{\partial E_t}{\partial h_i^{(l+1)}} \frac{\partial h_i^{(l+1)}}{\partial b_j^{(l)}} = \frac{\partial E_t}{\partial h_i^{(l+1)}} \tag{13}$$

ここで $\delta_i^{(l)} = \partial E_t / \partial h_i^{(l)}$ とおくと，

$$\delta_i^{(l+1)} = \sum_k \delta_k^{(l+2)} \frac{\partial h_k^{(l+2)}}{\partial h_i^{(l+1)}} = \sum_k \delta_k^{(l+2)} w_{ki}^{(l+1)} f'(h_i^{(l+1)}) \tag{14}$$

と出力層に近い層の $\delta^{(l+2)}$ によって $\delta^{(l+1)}$ が計算できることがわかる．そのため，まず $\delta^{(L+1)}$ を計算し，順次 $\delta^{(L)}, \delta^{(L-1)}, \ldots, \delta^{(1)}$ を計算することで各層の $w_{ij}^{(l)}, b_j^{(l)}$ の勾配が計算できる．そして $D$ に属するサンプルについて総和をとることで，損失関数 $E(\theta)$ の勾配を計算できる．

- ミニバッチ勾配降下法

  上記で求めた $d_t \in D$ に関する勾配を $D$ の各要素でたし合わせて勾配降下法を実行すると，推定パラメータの更新量が小さくなってしまうことがある．この現象は観測数を増やした時に顕著である．そのため，$D$ を分割したミニバッチについて確率的勾配降下法を実行する．

まず，最適化の前に $D$ を $M$ 個の部分集合 $D^{(m)}(m = 1, \ldots, M)$ に分割する．勾配降下法の各ステップの実行前に確率的に部分集合を選択し，選択された部分集合 $D^{(m)}$ について，

$$\Delta w_{ij}^{(l)} = \frac{1}{N_{obs}^{(m)}} \sum_{d_t \in D^{(m)}} \frac{\partial E_t}{\partial w_{ij}^{(l)}} \tag{15}$$

$$\Delta b_{j}^{(l)} = \frac{1}{N_{obs}^{(m)}} \sum_{d_t \in D^{(m)}} \frac{\partial E_t}{\partial b_{j}^{(l)}} \tag{16}$$

を計算する．但し，$N_{obs}^{(m)}$ は $D^{(m)}$ の要素数である．そしてこの値に従って

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \rho \Delta w_{ij}^{(l)} \tag{17}$$

$$b_{j}^{(l)} = b_{j}^{(l)} - \rho \Delta b_{j}^{(l)} \tag{18}$$

を計算する．ただし $\rho$ は学習率と呼ばれるパラメータであり外生的にあたえられる．また，このパラメータ更新は事前に繰り返しの回数が定められている．

## 4.2 事前学習

先述の誤差逆伝播法は誤差の情報が逆伝播を繰り返すたびに失われていき，入力層に近い $\delta^{(l)}$ の値が小さくなってしまうことが度々報告されている．この問題に対処するために Hinton 等 [10] では $l = 1, \ldots, L-1$ 層 ($L \geq 2$) のネットワークを Deep Belief Network (DBN) として事前学習を行い，その結果を誤差逆伝播法の初期パラメータとして利用することを提案している．本研究でもこの方法を採用する．

### 4.2.1 準備

本節で導入する DBN においては，出力層の一つ手前の第 $L-1$ 層に制約ボルツマンマシン (Ristricted Boltzmann Machine-RBM-) を仮定し，$l = 1, \ldots, L-1$ 層全てのユニットに対して Binary Unit を仮定する．また，第 $L-1$ 層ネットワークである制約ボルツマンマシンでは入力 $v^{(L-1)}$ と出力 $v^{(L)}$ に相当する変数（可視変数）を導入し，入力と出力いずれの可視変数も確率変数と考える．さらに，それよりも入力に近い層，$l = 1, \ldots, L-2$ のネットワークにおいても，入力の可視変数 $v^{(l)}$ と出力の可視変数 $v^{(l+1)}$ を導入し，入力 $v^{(l)}$ が与えられた時の出力 $v^{(l+1)}$，出力 $v^{(l+1)}$ が与えられた時の入力 $v^{(l)}$ が，それぞれベルヌーイ分布に従う確率変数であると考える．

より具体的には，第 $l$ 層を一つの独立したネットワークと捉えるとき，入力側の可視変数 $v^{(l)}$ を所与とすれば，出力側の可視変数 $v^{(l+1)}$ は，式 (3) で $z^{(l)} = v^{(l)}$ として計算した $h^{(l+1)}$ を用い，式 (6) と同様に以下で求められる．

$$v_i^{(l+1)} = \begin{cases} 1 & \text{with prob. } \sigma(h_i^{(l+1)}) \\ 0 & \text{with prob. } 1 - \sigma(h_i^{(l+1)}) \end{cases}, \ i = 1, \ldots, N_{\mathrm{OUT}}^{(l)} \tag{19}$$

但し，$\sigma(\cdot)$ はロジスティック関数を表す．また，$h_i^{(l+1)}$ は，式 (6) において入力 $v^{(l)}$ と第 $l$ 層のニューラルネットワークのパラメータ $\theta^{(l)}$ を所与として計算されるので，$v_i^{(l+1)}=1$ となる条件付き確率を $p(v_i^{(l+1)}=1|v^{(l)},\theta^{(l)})$ と表記する．

一方，入力の可視変数 $v^{(l)}$ は，出力の可視変数 $v^{(l+1)}$ を所与とすれば，入力側のバイアス $a^{(l)}=\{a_j^{(l)}\}_{j=1,\ldots,N_{\mathrm{IN}}^{(l)}}$ と，式 (3) における重み $w_{ij}^{(l)}$ を用い，以下のように計算される．

$$g_j^{(l)}=\sum_{i=1}^{N_{\mathrm{OUT}}^{(l)}} w_{ij}^{(l)}v_i^{(l+1)}+a_j^{(l)} \tag{20}$$

$$q_j^{(l)}=\sigma(g_j^{(l)}) \tag{21}$$

$$v_j^{(l)}=\begin{cases}1 & \text{with prob. } q_j^{(l)}\\ 0 & \text{with prob. } 1-q_j^{(l)}\end{cases},\ j=1,\ldots,N_{\mathrm{IN}}^{(l)} \tag{22}$$

従って，第 $l$ 層を一つの独立したネットワークと考える場合，順伝播ネットワークのパラメータ $\theta^{(l)}$ に $a^{(l)}$ を加えた $\theta_b^{(l)}=\{\theta^{(l)},a^{(l)}\}$ が学習対象となる．

また，第 $L-1$ 層が制約ボルツマンマシンであれば，$(v^{(L-1)},v^{(L)})$ は以下の確率分布に従う．

$$p(v^{(L-1)},v^{(L)}|\theta_b^{(L-1)})=\frac{1}{Z(\theta_b^{(L-1)})}\exp[-\Phi(v^{(L-1)},v^{(L)},\theta_b^{(L-1)})] \tag{23}$$

ここで，$v^{(L-1)},v^{(L)}$ に関するボルツマン型のエネルギー関数は，

$$\Phi(v^{(L-1)},v^{(L)},\theta_b^{(L-1)})=-\sum_j a_j^{(L-1)}v_j^{(L-1)}-\sum_i b_i^{(L-1)}v_i^{(L)}-\sum_i\sum_j w_{ij}^{(L-1)}v_j^{(L-1)}v_i^{(L)} \tag{24}$$

で与えられ，規格化項 $Z(\theta_b^{(L-1)})$ は以下のように書ける．

$$Z(\theta_b^{(L-1)})=\sum_{v^{(L-1)}}\sum_{v^{(L)}}\exp[-\Phi(v^{(L-1)},v^{(L)},\theta_b^{(L-1)})] \tag{25}$$

ただし

$$\sum_{v^{(l)}}=\sum_{v_1^{(l)}=0,1}\cdots\sum_{v_{N_{\mathrm{IN}}^{(l)}}^{(l)}=0,1} \tag{26}$$

である．また，この設定の下で，$v^{(l)}$ が与えられたときの $v^{(l+1)}$ の条件付き確率と $v^{(l+1)}$ が与えられたときの $v^{(l)}$ の条件付き確率は，各々，式 (19)，(22) で与えられる．

一方，$l=1,\ldots,L-2$ 層においては，式 (22) で表されるように出力の可視変数 $v^{(l+1)}$ が与えられた条件の下では，$v^{(l)}$ はベルヌーイ分布に従うと考えることができる．この確率分布を $p(v^{(l)}|v^{(l+1)},\theta_b^{(l)})$ と表記すると，本節の DBN における $(v^{(1)},\ldots,v^{(L)})$ の同時分布は以下のように表される．

$$p(v^{(1)},\ldots,v^{(L)}|\theta_b^{(1)},\ldots,\theta_b^{(L-1)})=\left[\prod_{l=1}^{L-2}p(v^{(l)}|v^{(l+1)},\theta_b^{(l)})\right]p(v^{(L-1)},v^{(L)}|\theta_b^{(L-1)}) \tag{27}$$

#### 4.2.2 事前学習の方法 (1)

本稿の実証分析における事前学習は，前節で述べた DBN を用いた方法を簡略化し，$l = 1, \ldots, L-1$ 層を各々独立した制約ボルツマンマシン (RBM) として扱い，各層毎に，最尤法により RBM の学習を行う．この手法は [10] で提案され，勾配法に基づく学習の初期パラメータを与えるための方法として有効であると言われている．具体的には $t$ 時点における $l$ 層の入力の可視変数 $v_t^{(l)}$ が与えられたもとで第 $l$ 層の RBM の学習を行い，学習されたネットワークから出力の可視変数 $v_t^{(l+1)}$ を生成する．そして $l$ 層の出力の可視変数を $l+1$ 層の入力の可視変数として $l+1$ 層に与えることを繰り返す．なお，第 1 層の RBM の学習は入力の可視変数として DBN の入力をそのまま用いる．つまり $v_t^{(1)} = X_t$ として学習を行う．このことはネットワークの入力 $X_t$ の要素自体がベルヌーイ分布に従う確率変数であることを意味している．従って $X_t$ の要素は全て 0 もしくは 1 である必要がある．

以下では各層毎の RBM の学習方法を説明する．学習データの生成時点を $T = \{T_s, \ldots, T_e\}$，学習データの数を $N = (T_e - T_s)/\Delta t + 1$ とする．第 $l$ 層のネットワークの入力データ $D_{\mathrm{IN}}^{(l)} = \{v_t^{(l)}\}_{t \in T},\ v_t^{(l)} = \{v_{t,j}^{(l)}\}_{j=1,\cdots,N_{\mathrm{IN}}^{(l)}}$ が与えられた時の対数尤度関数は，式 (24) で定義されるエネルギー関数と式 (25) で定義される規格化項を用いて以下のように与えられる．

$$L(\theta_b^{(l)}) \propto \frac{1}{N} \sum_{t \in T} \left[ \ln \sum_{v^{(l+1)}} \exp\{-\Phi(v_t^{(l)}, v^{(l+1)}, \theta_b^{(l)})\} \right] - \ln Z(\theta_b^{(l)}) \tag{28}$$

この対数尤度関数を最大化するために勾配上昇法を適用する．推定すべきパラメータの微分係数は $w_{ij}^{(l)}$ については次のようにあたえられる．

$$\begin{aligned} \frac{1}{N} \frac{\partial L}{\partial w_{ij}^{(l)}} &= \frac{1}{N} \sum_{t \in T} \sum_{v_i^{(l+1)}=0,1} v_{t,j}^{(l)} v_i^{(l+1)} p(v_i^{(l+1)} | v_t^{(l)}, \theta_b^{(l)}) \\ &\quad - \sum_{v^{(l)}} \sum_{v^{(l+1)}} v_j^{(l)} v_i^{(l+1)} p(v^{(l)}, v^{(l+1)} | \theta_b^{(l)}) \qquad (29) \\ &= \frac{1}{N} \sum_{t \in T} v_{t,j}^{(l)} p(v_i^{(l+1)} = 1 | v_t^{(l)}, \theta_b^{(l)}) \\ &\quad - \sum_{v^{(l)}} \sum_{v^{(l+1)}} v_j^{(l)} v_i^{(l+1)} p(v^{(l)}, v^{(l+1)} | \theta_b^{(l)}) \qquad (30) \end{aligned}$$

同様に入力と出力のバイアス項については次のようにあたえられる，

$$\frac{1}{N} \frac{\partial L}{\partial a_j^{(l)}} = \frac{1}{N} \sum_{t \in T} v_{t,j}^{(l)} - \sum_{v^{(l)}} \sum_{v^{(l+1)}} v_j^{(l)} p(v^{(l)}, v^{(l+1)} | \theta_b^{(l)}) \tag{31}$$

$$\frac{1}{N} \frac{\partial L}{\partial b_i^{(l)}} = \frac{1}{N} \sum_{t \in T} p(v_i^{(l+1)} = 1 | v_t^{(l)}, \theta_b^{(l)}) - \sum_{v^{(l)}} \sum_{v^{(l+1)}} v_i^{(l+1)} p(v^{(l)}, v^{(l+1)} | \theta_b^{(l)}) \tag{32}$$

従って，最適化のパラメータ更新幅は以下のように計算する．

$$\Delta w_{ij}^{(l)} = \langle v_j^{(l)} v_i^{(l+1)} \rangle_{\mathrm{data}} - \langle v_j^{(l)} v_i^{(l+1)} \rangle_{\mathrm{model}} \tag{33}$$

$$\Delta a_j^{(l)} = \langle v_j^{(l)} \rangle_{\mathrm{data}} - \langle v_j^{(l)} \rangle_{\mathrm{model}} \tag{34}$$

$$\Delta b_i^{(l)} = \langle v_i^{(l+1)} \rangle_{\mathrm{data}} - \langle v_i^{(l+1)} \rangle_{\mathrm{model}} \tag{35}$$

但し，$\langle\cdot\rangle_{\rm data}$ は式 (30)–(32) の右辺第 1 項を表し，$v_t^{(l)}$ が与えられれば計算が容易である．一方，$\langle\cdot\rangle_{\rm model}$ は式 (30)–(32) の右辺第 2 項を表し，確率分布 $p(v^{(l)}, v^{(l+1)}|\theta_b^{(l)})$ に従う確率変数 $v^{(l)}, v^{(l+1)}$ のとり得る全ての組合せを考慮しなければならず計算負荷が大きい．そのため，この項は $v_t^{(l)}$ を初期値とするコンストラティブ・ダイバージェンス (CD) 法を実行することにより近似計算される．

CD 法では，$v^{(l,0)} \equiv v_t^{(l)}$ を所与として，式 (3) で $z^{(l)} = v^{(l,0)}$ として計算された $h^{(l+1)}$ を用いて式 (19) で表されるベルヌーイ分布から，$p_i^{(l,0)} := p(v_i^{(l+1)} = 1|v^{(l,0)}, \theta_b^{(l)})$, $i = 1, \ldots, N_{\rm OUT}^{(l)}$ を得る．その確率に基づき $v^{(l+1,0)}$ をサンプリングし，その実現値を用いて式 (22) で表されるベルヌーイ分布から $q_j^{(l,1)} := p(v_j^{(l)} = 1|v^{(l+1,0)}, \theta_b^{(l)})$, $j = 1, \ldots, N_{\rm IN}^{(l)}$ を得る．次に，その確率に従って $v^{(l,1)}$ をサンプリングし, その実現値を用い，$p_i^{(l,0)}$ を得たときと同様に，式 (19) から $p_i^{(l,1)}$ を計算する．

この操作を $T$ 回繰り返すことによって得られた $v^{(l,T)} = (v_j^{(l,T)})_{j=1}^{N_{\rm IN}^{(l)}}$, $p^{(l,T)} = (p_i^{(l,T)})_{i=1}^{N_{\rm OUT}^{(l)}}$ より，

$$\langle v_j^{(l)} v_i^{(l+1)}\rangle_{\rm model} \approx v_j^{(l,T)} p_i^{(l,T)},\ \langle v_j^{(l)}\rangle_{\rm model} \approx v_j^{(l,T)},\ \langle v_i^{(l+1)}\rangle_{\rm model} \approx p_i^{(l,T)}, \tag{36}$$

と近似計算する．

一般的には，$T$ は大きい必要があるが，$v_t^{(l)}$ を初期値とする場合は，$T = 1$, 即ち，$v^{(l,1)}, p^{(l,1)}$ を用いたとしても，多層の RBM によりデータの特徴を捉えるために各層の RBM を学習する状況では問題ない報告されている (Hinton[11] 14 節を参照)．故に本稿では，計算の効率化を考慮し，$T = 1$ とする．さらに [11] 3 節にならい，サンプリングノイズを減少させるために，サンプリングに基づく $v^{(l,1)}$ の代わりに確率 $q^{(l,1)}$ を用いる．以上の近似に基づき $\langle\cdot\rangle_{\rm model}$ を計算し，パラメータ更新を行う．

$$w_{ij}^{(l)} = w_{ij}^{(l)} + \rho\Delta w_{ij}^{(l)} \tag{37}$$

$$a_j^{(l)} = a_j^{(l)} + \rho\Delta a_j^{(l)} \tag{38}$$

$$b_i^{(l)} = b_i^{(l)} + \rho\Delta b_i^{(l)} \tag{39}$$

#### 4.2.3 事前学習の方法 ( 2 )

次に，前節の方法を拡張して，$v^{(l)}$, $v^{(l+1)}$ の取り得る値として非負の整数を許容し，入力データに関しても非負整数を許容することを試みる．

具体的には，重み $w_{i,j}^{(l)}$，バイアス $b_i^{(l)}$，$i = 1, \ldots, N_{\rm OUT}^{(l)}$, $j = 1, \ldots N_{\rm IN}^{(l)}$ 及び，$0, 1, \cdots, K$ $(K \geq 2)$ の値を取り得る $v_i^{(l)}$, $i = 1, \ldots, N_{\rm OUT}^{(l)}$ を所与として，$v^{(l+1)} = (v_i^{(l+1)})_{i=1,\ldots,N_{\rm OUT}^{(l)}}$

を以下のように計算する．

$$h_i^{(l+1)} = \sum_{j=1}^{N_{\mathrm{IN}}^{(l)}} w_{ij}^{(l)} v_i^{(l)} + b_i^{(l)} \tag{40}$$

$$v_i^{(l+1,k)} = \begin{cases} 1 & \text{with prob. } \sigma(h_i^{(l+1)} - k + 0.5) \\ 0 & \text{with prob. } 1 - \sigma(h_i^{(l+1)} - k + 0.5) \end{cases} \tag{41}$$

$$v_i^{(l+1)} = \sum_{k=1}^{K} v_i^{(l+1,k)} \tag{42}$$

但し，$\sigma(\cdot)$ はロジスティック関数である．

同様に，重み $w_{i,j}^{(l)}$，バイアス $a_j^{(l)}$，$i = 1, \ldots, N_{\mathrm{OUT}}^{(l)}$, $j = 1, \ldots N_{\mathrm{IN}}^{(l)}$ 及び，$0, 1, \cdots, K$ $(K \geq 2)$ の値を取り得る $v_i^{(l+1)}$, $i = 1, \ldots, N_{\mathrm{OUT}}^{(l)}$ を所与として，$v^{(l)} = (v_j^{(l+1)})_{j=1,\ldots,N_{\mathrm{IN}}^{(l)}}$ を以下のように計算する．

$$g_j^{(l)} = \sum_{i=1}^{N_{\mathrm{OUT}}^{(l)}} w_{ij}^{(l)} v_i^{(l+1)} + a_j^{(l)} \tag{43}$$

$$v_j^{(l,k)} = \begin{cases} 1 & \text{with prob. } \sigma(g_j^{(l+1)} - k + 0.5) \\ 0 & \text{with prob. } 1 - \sigma(g_j^{(l+1)} - k + 0.5) \end{cases} \tag{44}$$

$$v_j^{(l)} = \sum_{k=1}^{K} v_j^{(l,k)} \tag{45}$$

このようなネットワークのパラメータを推定するため，$V^{(l)}$, $V^{(l+1)}$ を各々，$V^{(l)} = (v_j^{(l,k')})_{k'=1,\ldots,K,\, j=1,\ldots,N_{\mathrm{IN}}^{(l)}}$, $V^{(l+1)} = (v_i^{(l+1,k)})_{k=1,\ldots,K,\, i=1,\ldots,N_{\mathrm{OUT}}^{(l)}}$ とおいて，ネットワークのエネルギー関数を以下で定義する．

$$\begin{aligned} \Phi(V^{(l)}, V^{(l+1)}, \theta_b^{(l)}) = &- \sum_j \sum_{k'=1}^{K} (a_j^{(l)} - k' + 0.5) v_j^{(l,k')} \\ &- \sum_i \sum_{k=1}^{K} (b_i^{(l)} - k + 0.5) v_i^{(l+1,k)} - \sum_i \sum_j \sum_{k'=1}^{K} \sum_{k=1}^{K} w_{ij}^{(l)} v_j^{(l,k')} v_i^{(l+1,k)} \end{aligned} \tag{46}$$

このとき，$V^{(l)}$, $V^{(l+1)}$ の同時分布は，

$$p(V^{(l)}, V^{(l+1)} | \theta_b^{(l)}) = \frac{1}{Z(\theta_b^{(l)})} \exp[-\Phi(V^{(l)}, V^{(l+1)}, \theta_b^{(l)})] \tag{47}$$

と定義される．ここで，$Z(\theta_b^{(l)})$ は規格化項で以下のように書ける．

$$Z(\theta_b^{(l)}) = \sum_{V^{(l)}} \sum_{V^{(l+1)}} \exp[-\Phi(V^{(l)}, V^{(l+1)}, \theta_b^{(l)})] \tag{48}$$

但し，

$$\sum_{V^{(l)}} = \sum_{v_1^{(l,1)}=0,1} \cdots \sum_{v_1^{(l,K)}=0,1} \cdots \sum_{\substack{v_{N_{\mathrm{IN}}^{(l)}}^{(l,1)}=0,1}} \cdots \sum_{\substack{v_{N_{\mathrm{IN}}^{(l)}}^{(l,K)}=0,1}} \tag{49}$$

である．また，この設定の下では，$V^{(l)}$ が与えられたときの $(v_i^{(l+1,k)})_{k=1,\cdots,K,i=1,\cdots,N_{\rm OUT}^{(l)}}(=V^{(l+1)})$ の条件付き確率と，$V^{(l+1)}$ が与えられたときの $(v_j^{(l,k)})_{k=1,\cdots,K,j=1,\cdots,N_{\rm IN}^{(l)}}(=V^{(l)})$ の条件付き確率は，各々，($v^{(l)}$ $v^{(l+1)}$ を通して) (41) 式と (44) 式により与えられる．

次に，尤度関数は，

$$L(\theta_b^{(l)}) \propto \frac{1}{N}\sum_{t\in T}\left[\ln\sum_{V^{(l+1)}}\exp\{-\Phi(V_t^{(l)},V^{(l+1)},\theta_b^{(l)})\}\right]-\ln Z(\theta_b^{(l)}) \tag{50}$$

と表されるため，尤度関数の重みパラメータに関する微分係数は下記のように書くことができる．

$$\begin{aligned}\frac{1}{N}\frac{\partial L}{\partial w_{ij}^{(l)}} &= \frac{1}{N}\sum_{t\in T}\sum_{k'=1}^{K}\sum_{k=1}^{K}\sum_{v_i^{(l+1,k)}=0,1} v_{t,j}^{(l,k')}v_i^{(l+1,k)}p(v_i^{(l+1,k)}|V_t^{(l)},\theta_b^{(l)}) \\ &\qquad -\sum_{V^{(l)}}\sum_{V^{(l+1)}}\sum_{k'=1}^{K}\sum_{k=1}^{K}v_j^{(l,k')}v_i^{(l+1,k)}p(V^{(l)},V^{(l+1)}|\theta_b^{(l)}) \qquad (51)\\ &= \frac{1}{N}\sum_{t\in T}v_{t,j}^{(l)}\sum_{k=1}^{K}p(v_i^{(l+1,k)}=1|v_t^{(l)},\theta_b^{(l)}) \\ &\qquad -\sum_{V^{(l)}}\sum_{V^{(l+1)}}\sum_{k'=1}^{K}v_j^{(l,k')}\sum_{k=1}^{K}v_i^{(l+1,k)}p(V^{(l)},V^{(l+1)}|\theta_b^{(l)}) \qquad (52)\end{aligned}$$

同様に，入力と出力のバイアスパラメータに関する微分係数は，次のように与えられる，

$$\frac{1}{N}\frac{\partial L}{\partial a_j^{(l)}} = \frac{1}{N}\sum_{t\in T}v_{t,j}^{(l)} - \sum_{V^{(l)}}\sum_{V^{(l+1)}}\sum_{k'=1}^{K}v_j^{(l,k')}p(V^{(l)},V^{(l+1)}|\theta_b^{(l)}) \tag{53}$$

$$\begin{aligned}\frac{1}{N}\frac{\partial L}{\partial b_i^{(l)}} &= \frac{1}{N}\sum_{t\in T}\sum_{k=1}^{K}p(v_i^{(l+1,k)}=1|v_t^{(l)},\theta_b^{(l)}) \\ &\qquad -\sum_{V^{(l)}}\sum_{V^{(l+1)}}\sum_{k=1}^{K}v_i^{(l+1,k)}p(V^{(l)},V^{(l+1)}|\theta_b^{(l)}) \qquad (54)\end{aligned}$$

なお，式 (52), (54) の右辺第 1 項の条件付き確率分布は $V_t^{(l)}$ ではなく，$v_{t,i}^{(l,k)}$, $k=1,\cdots,K$ の和，$v_t^{(l)}=(v_{t,i}^{(l)})_{i=1,\cdots,N_{\rm IN}^{(l)}}$ が与えられたときの条件付き分布で表現されている．これは，式 (41) で表されるベルヌーイ分布が，和 $v_i^{(l)}(=v_{t,i}^{(l)})$ の値により決まり，その $v_{t,i}^{(l)}$ の値を構成する $v_{t,i}^{(l,k)}$,$k=1,\cdots,K$ の値の組み合わせに依らないためである．

以上から，式 (52)–(54) の右辺第 1 項を $\langle\cdot\rangle_{\rm data}$，右辺第 2 項を $\langle\cdot\rangle_{\rm model}$ と表すと，これらが計算できれば前節と同様の手順でパラメータ更新を行うことができる．

さらに，[9] による以下の ReLU による近似，

$$\sum_{k=1}^{\infty}\sigma(x-k+0.5)\approx\ln(1+\exp(x)) \tag{55}$$

を大きな $K$ について $\sum_{k=1}^{K}\sigma(h_{t,j}^{(l+1)}-k+0.5)$ に対して適用し，さらに，$\ln(1+\exp(x))\approx\max\{0,x\}$ の近似を用いて，$\langle\cdot\rangle_{\rm data}$ の計算を以下のように行う．但し，$h_{t,i}^{(l+1)}$ は式 (40) で $v_i^{(l)}=v_{t,i}^{(l)}$ として計算された値である．

$$\begin{aligned}\langle v_j^{(l)}v_i^{(l+1)}\rangle_{\rm data}&=\frac{1}{N}\sum_{t\in T}v_{t,j}^{(l)}\sum_{k=1}^{K}p(v_i^{(l+1,k)}=1|v_t^{(l)},\theta^{(l)})\\&=\frac{1}{N}\sum_{t\in T}v_{t,j}^{(l)}\sum_{k=1}^{K}\sigma(h_{t,i}^{(l+1)}-k+0.5)\approx\frac{1}{N}\sum_{t\in T}v_{t,j}^{(l)}\max\{0,h_{t,i}^{(l+1)}\}\end{aligned}\tag{56}$$

$$\begin{aligned}\langle v_i^{(l+1)}\rangle_{\rm data}&=\frac{1}{N}\sum_{t\in T}\sum_{k=1}^{K}p(v_i^{(l+1,k)}=1|v_t^{(l)},\theta^{(l)})\\&=\frac{1}{N}\sum_{t\in T}\sum_{k=1}^{K}\sigma(h_{t,i}^{(l+1)}-k+0.5)\approx\frac{1}{N}\sum_{t\in T}\max\{0,h_{t,i}^{(l+1)}\}\end{aligned}\tag{57}$$

一方，重み，バイアスを所与とした場合の $\langle\cdot\rangle_{\rm model}$ の計算については，まず，式 (41), (44) のベルヌーイ分布が，それぞれ，$v^{(l)}$, $v^{(l+1)}$ により一意に定まる $h^{(l+1)}$ と $g^{(l)}$ から求まることに注意する．従って, 前述の CD 法に則れば，初期値を $v_t^{(l)}$ として，各 $k$ に関するベルヌーイ分布に基づくサンプリングで得られた $K$ 個の確率変数を足し合わせ，$v^{(l+1)},v^{(l)}$ をもとめる操作を繰り返すことになる．しかしながら，その計算負荷は大きいため，本研究では [9] に従い，第 3.2 節で導入した Noisy Rectified Linear Unit (NReLU) を活用し計算負荷の軽減を図る．

より具体的には，まず，$v_i^{(l+1,k)}$, $k=1,2,\cdots,$ は，ベルヌーイ分布に従う独立確率変数列で，平均は $\sigma(h_i^{(l+1)}-k+0.5)$, 分散は $\sigma(h_i^{(l+1)}-k+0.5)(1-\sigma(h_i^{(l+1)}-k+0.5))$ であることに注意する．次に，$v_i^{(l+1)}=\sum_{i=1}^{K}v_i^{(l+1,k)}$ の平均は，先述と同様,

$$\sum_{k=1}^{K}\sigma(h_i^{(l+1)}-k+0.5)\approx\frac{d}{dx}\ln(1+\exp(h_i^{(l+1)})\approx\max\{0,h_i^{l+1}\}$$

により近似する．分散は，$\frac{d}{dx}\sigma(x)=\sigma(x)(1-\sigma(x))$ に注意し，

$$\begin{aligned}&\sum_{k=1}^{K}\sigma(h_i^{(l+1)}-k+0.5)(1-\sigma(h_i^{(l+1)}-k+0.5))=\sum_{k=1}^{K}\frac{d}{dx}\sigma(h_i^{(l+1)}-k+0.5)\\&=\frac{d}{dx}\sum_{k=1}^{K}\sigma(h_i^{(l+1)}-k+0.5)\approx\frac{d}{dx}\ln(1+\exp(h_i^{(l+1)}))=\sigma(h_i^{(l+1)})\end{aligned}$$

により近似する．

さらには，これらの近似と，$v_i^{(l+1)}$ の非負性を考慮し，(8), (9) 式と同様の以下の式に基づき，$v_i^{(l+1)}$ のサンプリングを行う．

$$v_i^{(l+1)}=\sum_{i=1}^{K}v_i^{(l+1,k)}=\max\{0,h_i^{(l+1)}+u\},\ i=1,\dots,N_{\rm OUT}^{(l)}$$

$$u\sim N(0,\sigma(h_i^{(l+1)}))$$

### 4.3 学習アルゴリズム

以上をまとめると以下のようなアルゴリズムで学習は実行される．

1. 観測値からユニットの定義に沿うように入力データ $X$ を生成し，これをランダムに 4 等分することでミニバッチを 4 つ生成する．
2. 4.2.3 節で概観した事前学習の実行：$l = 1$ と初期設定し，$l = L - 1$ となるまで下記を繰り返す．
   (a) 式 (37)–(39) に従って $\theta^{(l)}$ を更新することをあらかじめ決められた回数繰り返す．
   (b) $l = l + 1$ とする．
3. 学習の実行：事前学習によって得られた $\theta^{(1)}, \ldots, \theta^{(L-1)}$ を初期値として，以下のプロセスをあらかじめ決められた回数繰り返す．ただし $\theta^{(L)}$ の初期値は Hinton 等 [2] にならい $N(0, 0.1)$ からサンプルしたものを用いる．
   (a) (15), (16) に従い $l = L, L-1, \ldots, 1$ で各層のパラメータ更新幅を逐次計算する．
   (b) 式 (17), (18) に従って $l = L, L-1, \ldots, 1$ でパラメータを更新する．

さらに，実データを用いた検証においては，以下に列挙する学習理論の標準的手法を用いる．

- 重み減衰 (岡谷 [12] 3.5.2 節): $\lambda = 0.001$ を採用
- ドロップアウト ([12] 3.5.3 節): $p = 0.5$ を採用
- モメンタム ([12] 3.6.5 節): $\mu = 0.5$ を採用
- 学習率の自動調整 (RMSprop)(Tieleman-Hinton[13])：ミニバッチ勾配降下法における学習パラメータの更新の際，学習率を繰り返し回数毎に更新する．

  実装では，http://sebastianruder.com/optimizing-gradient-descent/ の RMSprop を参考にし，式 (17), (18) で表されるパラメータ $\eta \in \theta$ の更新において，$k$ 回目の演算の際，学習率 $\rho$ を以下の $\rho_k$ に変えて計算する．

$$\rho_k = \frac{\rho}{\sqrt{E[\Delta\eta^2]_k + \epsilon}},\ k = 1, 2, \cdots,$$
$$E[\Delta\eta^2]_k = 0.9E[\Delta\eta^2]_{k-1} + 0.1\Delta\eta^2,\ E[\Delta\eta^2]_0 = 0$$

  ただし，$\Delta\eta$ は式 (15), (16) で計算される．また，$\epsilon = 10^{-8}$ を採用する．

  尚，これらのテクニックは事前学習のパラメータ更新（式 (37)–(39)）においても使用する．また，上式の基準となる学習率 $\rho$ と学習の繰り返し回数は末尾の付録に掲載する．

## 5　月次データを用いた検証

本節では $L = 3$ として構成したニューラルネットワークを用いて実データを学習し，その学習結果を活用した投資戦略の検証を行う．また，このネットワークを単純化した $L = 1, 2$ の場合においても同様の分析を行う．なお，$L = 1$ の場合は DBN を構成できないため，事前学習は行われない．入力層のユニット数は，各資産リターンの時系列のデータ数 $N_{\mathrm{IN}}^{(1)} = 30$ とし，$L = 2, 3$ の場合の中間 (隠れ) 層のユニット数は，リターンとそのボラティリティに対応する変数を想定し，$N_{\mathrm{IN}}^{(l)} = 60,\ l = 2, \ldots, L$ とする．但し，第 6 節の日次データを用いた検証においては，層数やユニット数の変化の影響も分析する．

さらに，SL の場合はリターンの分類を 3 種類と設定するため $N_{\mathrm{OUT}}^{(L)} = 3$，RL の場合は $N_{\mathrm{OUT}}^{(L)} = 1$ である．また，出力層の出力ユニットには第 3 節で概説した，Binary Unit または Classification Unit を用い，それ以外のユニットの活性化関数は ReLU を設定する．これらをまとめると表 1 のようになる．

表 1: ネットワーク構造

| | $L$ | ユニットの種類 | ユニット数 |
|---|---|---|---|
| SL | 3 | ReLU-ReLU-ReLU-Classification | 30-60-60-3 |
| | 2 | ReLU-ReLU-Classification | 30-60-3 |
| | 1 | ReLU-Classification | 30-3 |
| RL | 3 | ReLU-ReLU-ReLU-Binary | 30-60-60-1 |
| | 2 | ReLU-ReLU-Binary | 30-60-1 |
| | 1 | ReLU-Binary | 30-1 |

学習に用いる実データは表 2 の時系列データである．各資産の時系列データの学習によってニューラルネットワークのパラメータが得られたのちに，各資産の学習結果を用いて予測を行い，各資産クラスへの投資を行う．但し，第 2 節で想定した投資戦略と整合的となるために，各期とも買い（ロング）のみとし空売りはできないものとする．また，各期ごとにポジションは解消されるとする．さらに，円建て以外の資産（本稿では全て USD 建ての資産）については，USD 建てリターンに基づき学習及び運用シミュレーションを行う，(FX 完全ヘッジ) のケースと，USDJPY の FX レートに対しても同様に学習データ生成と学習を行い，その結果に基づき USD 建ての資産の FX リスクをヘッジするか否かの判断を行う (FX 動的ヘッジ) のケースの両方を実施する．(5.2 節の FX ヘッジを参照のこと．)

表 2: 使用データ

| | 表示通貨 | 説明 |
|---|---|---|
| Japan Equity | JPY | Topix 株式インデックス |
| U. S. Equity | USD | S&P 株式インデックス |
| Emerging Equity | USD | FTSE 新興国株式インデックス |
| Emerging Bond | USD | J. P. Morgan 新興国債券インデックス |

### 5.1 データ生成

まず，入力に対応するデータの生成方法を説明する．いま資産の (月次) リターン $R_t$ の時系列の観測間隔を $\Delta t(=1$ヶ月) とする．学習データの生成時点 $T = \{T_s, \ldots, T_e\}$ でニューラルネットワークの入力データの生成のために，まず時点 $t \in T$ よりも前に得られた $M$ 個のリターンの観測値からなるデータ $[R_{t-M\Delta t}, \ldots, R_{t-\Delta t}]$ を用意する．ここで入力データの次元はニューラルネットワークの入力層の入力ユニットの数に等しいため $M = N_{\mathrm{IN}}^{(1)}(= 30)$ である．また，学習データの生成時点の間隔はデータの観測間隔と等しく $\Delta t(=1$ヶ月$)$, $T_e = T_s + (N-1)\Delta t$ として $N$ 個の時点のデータを生成する．ここで，各時点のデータは $M$ 次元のベクトルである．

しかしながら，第 4.2.3 節で説明した事前学習を行う場合，全ての層のネットワークの入力ユニットの可視変数はユニットの出力の定義に基づいた確率変数と仮定するため，上記の $N$ 時点のデータをニューラルネットワークの入力データとしてそのまま用いることはできない．即ち，入力データの各要素は第 1 層のネットワークにおける出力の定義に従う必要がある．具体的には，$[R_{t-M\Delta t}, \ldots, R_{t-\Delta t}]$ を次のように非負の整数に変換したものをニューラルネットワークの入力データとする．

最初に学習データ生成のために観測するすべての $R_t$ を，その平均と分散で標準化する．学習データの説明変数は $[R_{t-M\Delta t}, \ldots, R_{t-\Delta t}]$ を $t = T_s, \ldots, T_e$ で生成したものに基づくため，リターンの時系列の観測開始時点は $t_s = T_s - M\Delta t$ であり観測終了時点は $t_e = T_e - \Delta t$ である．また，学習データの非説明変数は $R_t$ を $t = T_s, \ldots, T_e$ で観測したものに基づく．従って学習データ生成のためにリターンの観測は $[t_s, T_e]$ で行われることになる．このとき，リターンの時系列をその平均と分散で標準化した $Z^{(t_s)} = \{Z_t^{(t_s)}\}_{t=t_s,\ldots,T_e}$ を以下のように生成する．

$$Z_t^{(t_s)} = \frac{R_t - \mu^{(t_s)}}{\sigma^{(t_s)}} \tag{58}$$

$$\mu^{(t_s)} = (N+M)^{-1} \sum_{t=t_s}^{T_e} R_t \tag{59}$$

$$[\sigma^{(t_s)}]^2 = (N+M)^{-1} \sum_{t=t_s}^{T_e} (R_t - \mu^{(t_s)})^2 \tag{60}$$

そして $Z_t^{(t_s)}$ の大きさに基づき，変数 $x^{(t_s)} = \{x_t^{(t_s)}\}_{t=t_s,\ldots,t_e}$ を以下のように生成する．

$$x_t^{(t_s)} = \begin{cases} 3 & Z_t^{(t_s)} \geq 1/2 \\ 2 & 0 \leq Z_t^{(t_s)} < 1/2 \\ 1 & -1/2 \leq Z_t^{(t_s)} < 0 \\ 0 & Z_t^{(t_s)} < -1/2 \end{cases} \tag{61}$$

こうして得られた $x^{(t_s)}$ から $M$ 次元のベクトル $X_t^{(t_s)} = [x_{t-M\Delta t}^{(t_s)}, \ldots, x_{t-\Delta t}^{(t_s)}]$ を $t = T_s, \ldots, T_e$ で生成することでニューラルネットワークの入力データ $X^{(t_s)} = \{X_t^{(t_s)}\}_{t\in T}$ を得る．

次に出力において用いるデータの生成について説明する．RL の場合は $\{R_t\}_{t\in T}$ をそのまま利用できるが，SL の場合は $\{R_t\}_{t\in T}$ からリターンの大きさに従い以下のように分類することで，教師データの被説明変数 $Y^{(t_s)} = \{Y_t^{(t_s)}\}_{t\in T}$, $Y_t^{(t_s)} = \{Y_{t,i}^{(t_s)}\}_{i=0,1,2}$ を生成する．

$$Y_{t,i}^{(t_s)} = \begin{cases} 1 & \text{if } i = K_t^{(t_s)} \\ 0 & \text{otherwise} \end{cases}, \tag{62}$$

$$K_t^{(t_s)} = \begin{cases} 2 & Z_t^{(t_s)} \geq 1/2 \\ 1 & -1/2 \leq Z_t^{(t_s)} < 1/2 \\ 0 & Z_t^{(t_s)} < -1/2 \end{cases} \tag{63}$$

但し，$K_t^{(t_s)}$ は 3.2 節における (11) 式の $c$ に対応し，分類 $\{0, 1, 2\}$ は $C = \{1, \cdots, N_{\mathrm{OUT}}^{(L)}\}$ に対応する．

また，$K_t^{(t_s)} = 2$ である時点は観測期間の中で相対的に大きいリターンが得られた時点，$K_t^{(t_s)} = 1$ である時点は変動があまりなかった時点，$K_t^{(t_s)} = 0$ は異常に小さいリターンが得られた時点と解釈することができる．

## 5.2 運用シミュレーション

運用期間を $T^I = [T_s^I, T_e^I]$ と設定し，以下で説明するように，学習スケジュール・学習の種類・FX ヘッジの有無により異なる複数のパターンで運用シミュレーションを行う．

**学習スケジュール**　運用期間中に学習を再実行するか否かにより以下の 2 通りのスケジュールを設定する．

$S_1$ : $t = T_s^I$ において $X^{(t_s)}, Y^{(t_s)}$ について学習する．ただし学習期間を $T_e = T_s^I - \Delta t$ となるように設定するため $t_s = T_s^I - (N+M)\Delta t$ である．そして推定されたパラメータに基づき $T^I = [T_s^I, T_e^I]$ の期間で運用する．運用期間中にパラメータ更新は行わない．

$S_2$ : まず，同一のパラメータ $\theta$ を用いて運用する期間 $\Delta T$ を新たに設定し，$T_{s,0}^I = T_s^I$, $T_{s,i}^I = T_{s,i-1}^I + i\Delta T$, $T_{e,i}^I = \min\{T_{s,i+1}^I, T_e^I\}$ として運用期間を分割する．そして各 $i$ に対し $t = T_{s,i}^I$ において $X^{(t_s)}, Y^{(t_s)}$ に関して学習する．但し，学習期間を $T_e = T_{s,i}^I - \Delta t$ となるように設定するため $t_s = T_{s,i}^I - (N+M)\Delta t$ である．そして推定されたパラメータで $T_i^I = [T_{s,i}^I, T_{e,i}^I]$ の期間で運用する．これを $T_{s,i}^I > T_e^I$ となるまで繰り返す．

**FX ヘッジ**　（FX 完全ヘッジ）のケースでは，USD 建てのリターンに基づく学習・運用シミュレーションを行い，そのベンチマークも対象資産の常時ロングに対する，USD 建てのリターンとする.（FX 動的ヘッジ）のケースでは，FX の学習も行い，USD 買いシグナルが出ない場合にドル売り円買いのヘッジを実施する．そのベンチマークは，為替ヘッジを全くしない場合の対象資産の常時ロングに対する円建てリターンとする.

### 5.3　シミュレーション結果

運用期間を $T_s^I =$ Dec., 2009, $T_e^I =$ Jun., 2016，リターンの時系列の観測間隔を $\Delta t =$ 1 カ月 として，表 2 の 4 つの資産クラスのリターンの時系列から，各時点における $M$ 次元 ($M = 30$) のニューラルネットワークへ入力データを $N = 160$ 時点で生成し（これに加えて SL の場合は，ニューラルネットワークの出力に対応するデータも $N = 160$ 時点で生成し），学習を行う．さらに $S_2$ の場合は $\Delta T = 20$ カ月 として $T^I$ を 4 分割した．また，SL の結果の活用に必要なパラメータを $P_1^{th} = 0.4$, $P_2^{th} = 0.0$ と設定した．そして特定の資産クラスのみに投資を行った場合と，下記の Port 1,2,3 のように複数資産を投資対象を設定した場合の検証を行った．さらに，資産価格の表示通貨が円建てではなく US ドル建ての場合は，対ドルの為替変動リスクをフルヘッジするケース（FX 完全ヘッジ）と対ドル為替レートに関する学習に基づき動的にヘッジするケース（FX 動的ヘッジ）の両方を検証した.

尚，複数資産を投資対象とする場合は，投資対象の中で買いシグナルが出た資産に等配分でロングする戦略を採用する.

また，単一資産を投資対象とする場合のベンチマークはその対象資産を常にロングする戦略であり，複数資産を投資対象とする場合のベンチマークは対象資産全てを等しい割合でロングする戦略である．さらに,（FX 完全ヘッジ）の場合のベンチマークは対ドルの為替変動リスクをフルヘッジする場合,（FX 動的ヘッジ）の場合のベンチマークは対ドルの為替変動リスクをヘッジしない場合のリターンとする.

尚，この分析の実行のためには Jan., 1994 からの資産のリターンの時系列の観測値が必要である.

**Port1**　Japan Equity, U. S. Equity, Emerging Equity, Emerging Bond

**Port2**　Japan Equity, U. S. Equity, Emerging Bond

**Port3**　Japan Equity, U. S. Equity

次に，(月次再投資運用の) 年率換算利回り $R^*$ とシャープレシオ SR を表 3–6 に示す.

表 3: SL による運用結果（FX 完全ヘッジ）

(a) $L = 3$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 12.32% | 0.79 | 6.20% | 0.41 | 8.65% | 0.61 |
| Port2 | 15.62% | 1.10 | 6.36% | 0.44 | 10.14% | 0.82 |
| Port3 | 12.79% | 0.89 | 5.62% | 0.38 | 10.11% | 0.71 |
| Japan Equity | 9.44% | 0.73 | 3.99% | 0.29 | 8.11% | 0.44 |
| U. S. Equity | 9.93% | 0.95 | 4.23% | 0.57 | 11.58% | 0.89 |
| Emerging Equity | -0.14% | -0.01 | 2.61% | 0.22 | 3.60% | 0.17 |
| Emerging Bond | 2.34% | 0.45 | 0.88% | 0.36 | 9.89% | 0.90 |

(b) $L = 2$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 8.36% | 0.62 | 1.63% | 0.16 | 8.65% | 0.61 |
| Port2 | 9.75% | 0.79 | 3.06% | 0.32 | 10.14% | 0.82 |
| Port3 | 10.02% | 0.82 | 3.84% | 0.40 | 10.11% | 0.71 |
| Japan Equity | 3.24% | 0.62 | 2.46% | 0.34 | 8.11% | 0.44 |
| U. S. Equity | 7.50% | 0.64 | 1.08% | 0.16 | 11.58% | 0.89 |
| Emerging Equity | -3.19% | -0.52 | -2.14% | -0.57 | 3.60% | 0.17 |
| Emerging Bond | 3.06% | 0.63 | -0.36% | -0.26 | 9.89% | 0.90 |

(c) $L = 1$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 11.00% | 0.92 | 4.99% | 0.48 | 8.65% | 0.61 |
| Port2 | 9.75% | 0.92 | 5.27% | 0.56 | 10.14% | 0.82 |
| Port3 | 3.37% | 0.36 | 1.68% | 0.19 | 10.11% | 0.71 |
| Japan Equity | 2.57% | 0.29 | 0.47% | 0.05 | 8.11% | 0.44 |
| U. S. Equity | 1.85% | 0.22 | 1.43% | 0.22 | 11.58% | 0.89 |
| Emerging Equity | 1.68% | 0.25 | 0.21% | 0.03 | 3.60% | 0.17 |
| Emerging Bond | 5.63% | 0.79 | 3.96% | 0.71 | 9.89% | 0.90 |

表 4: SL による運用結果（FX 動的ヘッジ）

(a) $L = 3$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 12.49% | 0.71 | 7.54% | 0.46 | 10.64% | 0.53 |
| Port2 | 16.37% | 1.04 | 7.03% | 0.46 | 11.88% | 0.67 |
| Port3 | 13.58% | 0.87 | 6.29% | 0.41 | 11.35% | 0.64 |
| Japan Equity | 9.44% | 0.73 | 3.99% | 0.29 | 8.11% | 0.44 |
| U. S. Equity | 11.64% | 0.92 | 5.05% | 0.58 | 14.22% | 0.76 |
| Emerging Equity | 0.04% | 0.00 | 3.26% | 0.26 | 6.22% | 0.22 |
| Emerging Bond | 2.63% | 0.33 | 0.88% | 0.36 | 12.53% | 0.62 |
| USDJPY FX | 0.75% | 0.13 | 1.71% | 0.40 | 2.62% | 0.26 |

(b) $L = 2$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 11.15% | 0.63 | 1.16% | 0.11 | 10.64% | 0.53 |
| Port2 | 12.86% | 0.78 | 2.59% | 0.25 | 11.88% | 0.67 |
| Port3 | 14.64% | 0.93 | 3.74% | 0.37 | 11.35% | 0.64 |
| Japan Equity | 3.24% | 0.62 | 2.46% | 0.34 | 8.11% | 0.44 |
| U. S. Equity | 12.08% | 0.78 | 0.96% | 0.13 | 14.22% | 0.76 |
| Emerging Equity | -4.09% | -0.59 | -2.72% | -0.67 | 6.22% | 0.22 |
| Emerging Bond | 2.51% | 0.30 | -1.50% | -0.61 | 12.53% | 0.62 |
| USDJPY FX | 3.63% | 0.40 | 1.30% | 0.25 | 2.62% | 0.26 |

(c) $L = 1$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 12.76% | 1.00 | 4.76% | 0.43 | 10.64% | 0.53 |
| Port2 | 11.51% | 1.00 | 5.04% | 0.51 | 11.88% | 0.67 |
| Port3 | 5.00% | 0.48 | 1.64% | 0.19 | 11.35% | 0.64 |
| Japan Equity | 2.57% | 0.29 | 0.47% | 0.05 | 8.11% | 0.44 |
| U. S. Equity | 3.88% | 0.40 | 1.36% | 0.21 | 14.22% | 0.76 |
| Emerging Equity | 1.68% | 0.25 | -0.41% | -0.06 | 6.22% | 0.22 |
| Emerging Bond | 5.75% | 0.81 | 3.76% | 0.59 | 12.53% | 0.62 |
| USDJPY FX | 1.27% | 0.35 | 1.39% | 0.33 | 2.62% | 0.26 |

表 5: RL による運用結果（FX 完全ヘッジ）

(a) $L = 3$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 7.84% | 0.53 | 10.59% | 0.72 | 8.65% | 0.61 |
| Port2 | 10.80% | 0.83 | 11.14% | 0.85 | 10.14% | 0.82 |
| Port3 | 9.65% | 0.69 | 11.77% | 0.82 | 10.11% | 0.71 |
| Japan Equity | 4.31% | 0.29 | 6.75% | 0.44 | 8.11% | 0.44 |
| U. S. Equity | 13.32% | 1.14 | 10.98% | 0.90 | 11.58% | 0.89 |
| Emerging Equity | 0.86% | 0.05 | 6.98% | 0.40 | 3.60% | 0.17 |
| Emerging Bond | 5.26% | 0.61 | 6.70% | 0.80 | 9.89% | 0.90 |

(b) $L = 2$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 9.96% | 0.72 | 11.13% | 0.75 | 8.65% | 0.61 |
| Port2 | 10.92% | 0.86 | 11.87% | 0.84 | 10.14% | 0.82 |
| Port3 | 11.73% | 0.85 | 11.87% | 0.81 | 10.11% | 0.71 |
| Japan Equity | 6.44% | 0.41 | 7.98% | 0.45 | 8.11% | 0.44 |
| U. S. Equity | 11.58% | 0.89 | 11.44% | 0.88 | 11.58% | 0.89 |
| Emerging Equity | 5.84% | 0.39 | 1.87% | 0.15 | 3.60% | 0.17 |
| Emerging Bond | 5.71% | 0.60 | 5.48% | 0.74 | 9.89% | 0.90 |

(c) $L = 1$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 12.03% | 0.90 | 10.75% | 0.73 | 8.65% | 0.61 |
| Port2 | 13.05% | 1.10 | 12.27% | 0.90 | 10.14% | 0.82 |
| Port3 | 12.04% | 0.96 | 11.38% | 0.75 | 10.11% | 0.71 |
| Japan Equity | 3.02% | 0.26 | 2.57% | 0.17 | 8.11% | 0.44 |
| U. S. Equity | 9.03% | 0.84 | 5.64% | 0.54 | 11.58% | 0.89 |
| Emerging Equity | 2.67% | 0.20 | 3.61% | 0.27 | 3.60% | 0.17 |
| Emerging Bond | 4.73% | 0.55 | 5.67% | 0.76 | 9.89% | 0.90 |

表 6: RL による運用結果（FX 動的ヘッジ）

(a) $L = 3$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 9.91% | 0.52 | 12.78% | 0.66 | 10.64% | 0.53 |
| Port2 | 12.75% | 0.77 | 12.57% | 0.73 | 11.88% | 0.67 |
| Port3 | 10.78% | 0.65 | 13.10% | 0.75 | 11.35% | 0.64 |
| Japan Equity | 4.31% | 0.29 | 6.75% | 0.44 | 8.11% | 0.44 |
| U. S. Equity | 16.41% | 1.05 | 14.09% | 0.84 | 14.22% | 0.76 |
| Emerging Equity | 4.72% | 0.21 | 11.76% | 0.52 | 6.22% | 0.22 |
| Emerging Bond | 6.30% | 0.43 | 8.20% | 0.53 | 12.53% | 0.62 |
| USDJPY FX | 3.02% | 0.35 | 3.17% | 0.35 | 2.62% | 0.26 |

(b) $L = 2$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 12.21% | 0.69 | 13.53% | 0.74 | 10.64% | 0.53 |
| Port2 | 12.66% | 0.78 | 14.29% | 0.83 | 11.88% | 0.67 |
| Port3 | 12.72% | 0.77 | 13.96% | 0.81 | 11.35% | 0.64 |
| Japan Equity | 6.44% | 0.41 | 7.98% | 0.45 | 8.11% | 0.44 |
| U. S. Equity | 14.81% | 0.89 | 15.24% | 0.90 | 14.22% | 0.76 |
| Emerging Equity | 10.35% | 0.53 | 2.31% | 0.14 | 6.22% | 0.22 |
| Emerging Bond | 9.54% | 0.63 | 8.44% | 0.68 | 12.53% | 0.62 |
| USDJPY FX | 3.20% | 0.43 | 4.53% | 0.54 | 2.62% | 0.26 |

(c) $L = 1$

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 14.54% | 0.93 | 13.57% | 0.82 | 10.64% | 0.53 |
| Port2 | 15.43% | 1.12 | 14.33% | 0.95 | 11.88% | 0.67 |
| Port3 | 13.46% | 0.99 | 12.68% | 0.80 | 11.35% | 0.64 |
| Japan Equity | 3.02% | 0.26 | 2.57% | 0.17 | 8.11% | 0.44 |
| U. S. Equity | 9.68% | 0.81 | 7.20% | 0.60 | 14.22% | 0.76 |
| Emerging Equity | 4.53% | 0.27 | 7.70% | 0.47 | 6.22% | 0.22 |
| Emerging Bond | 7.26% | 0.61 | 8.22% | 0.72 | 12.53% | 0.62 |
| USDJPY FX | 2.59% | 0.43 | 3.65% | 0.58 | 2.62% | 0.26 |

計算結果に見られる特徴を以下に列挙する．

- SL(FX 完全ヘッジ): ベンチマークと比較すると，ポートフォリオ (Port1,2,3) にお

いて，リターン (利回り)，シャープレシオ共に $L = 3$ の $S_1$(パラメータ更新なし) が良好なパフォーマンスを示している．それらのポートフォリオにおいて，全般的に，パラメータ更新によりパフォーマンスは悪化し，多層化による著しい改善効果は見られない．

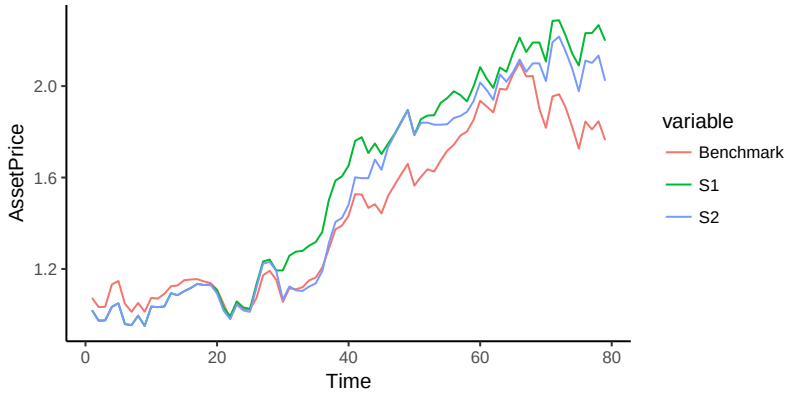
- SL(FX 動的ヘッジ): ベンチマークと比較すると，ポートフォリオ (Port1,2,3) において，Port3 の $L = 1$ を除き，$S_1$(パラメータ更新なし) がリターン (利回り)，シャープレシオ共に良好なパフォーマンスを示している．それらのポートフォリオにおいて，全般的に，パラメータ更新によりパフォーマンスは悪化し，多層化による著しい改善効果は見られない．

- RL(FX 完全ヘッジ): ベンチマークと比較すると，ポートフォリオ (Port1,2,3) において，$L = 3$ の $S_1$(パラメータ更新なし) を除き，リターン (利回り)，シャープレシオ共に良好なパフォーマンスを示している．それらのポートフォリオにおいて，全般的に多層化による改善効果は見られず，むしろ $S_1$ では悪化している．

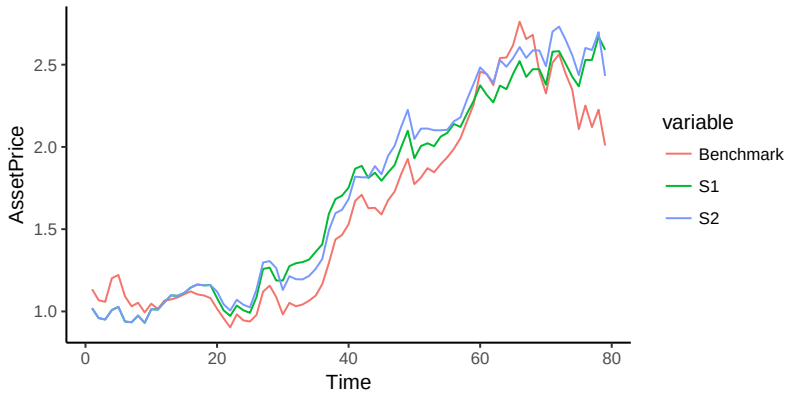
- RL(FX 動的ヘッジ): ベンチマークと比較すると，ポートフォリオ (Port1,2,3) において，Port 1 の $L = 3$，$S_1$(パラメータ更新なし) を除き，リターン (利回り)，シャープレシオ共に良好なパフォーマンスを示している．それらのポートフォリオにおいて，多層化によりパフォーマンスが悪化する傾向があり，$L = 1$ のケースが最良である．また，表 6 に見られる (USD JPY FX) の結果より，学習効果に基づく動的ヘッジがある程度は有効であろうと思われる．

全体として，本稿で採用した SL, RL に基づく運用は，相対的にはポートフォリオ (Port1,2,3) に対して有効である結果を示している．また，それらのポートフォリオに対して，パラメータ更新によるパフォーマンスの明確な改善効果は見られず，むしろ SL においては悪化する．さらに，多層化による改善効果も見られず，RL のパフォーマンスは $S_1$ で悪化する傾向が観測される．総じて，$RL$ の $L = 1, 2$ がポートフォリオにおいてベンチマーク対比で良好なパフォーマンスを示す結果となっている．

本小節の最後に，$L = 1$ におけるベンチーマークと RL の Port1 の資産価値の時系列を図 1 に示す.（但し時点 $t = 1$ からの図示となっている.）FX 完全ヘッジ，FX 動的ヘッジ何れの場合もベンチーマークに対する一定の改善効果が観測される．




(a) FX 完全ヘッジ




(b) FX 動的ヘッジ

図 1: Port1 の価格の時系列

### 5.4 SL の結果活用におけるハイパー・パラメータの影響

本研究では SL の結果から買いシグナルを導くためにハイパー・パラメータ $P_1^{th}$, $P_2^{th}$ を導入した．以下では $L = 3$ の場合でこれらパラメータの影響を分析する．学習スケジュール $S_2$ のケースで運用期間 $T^I$ を分割することにより得られた運用期間 $T_i^I$, $i = 0, 1, 2, 3$ について，それぞれの運用期間 $T_i^I$ におけるシャープレシオが最大となる時の $P_1^{th}, P_2^{th}$ をグリッド・サーチによって求めた．但し，パラメータの探索の範囲は $P_1^{th}, P_2^{th}$ ともに $\{0.0, 0.1, \ldots, 1.0\}$ である．学習スケジュール $S_1, S_2$ で（FX 完全ヘッジ）の場合結果を表 7 に示す．

得られたパラメータを見ると，当初設定した $P_1^{th} = 0.4$, $P_2^{th} = 0.0$ と乖離している場合があることがわかる．また，同じ学習スケジュールであっても対象の運用期間によって最適なパラメータが異なるケースがある．このような結果が得られたケースでは，買いシグナルの性質が運用期間ごとに異なっていたと解釈することができる．

表 7: 分割した各運用期間における最適な $P_1^{th},\ P_2^{th}$ $(L=3)$

| | | $S_1$ | | $S_2$ | |
|---|---|---|---|---|---|
| | Term | $P_1^{th}$ | $P_2^{th}$ | $P_1^{th}$ | $P_2^{th}$ |
| Japan Equity | $T_0^I$ | 0.2 | 0.1 | 0.4 | 0.4 |
| | $T_1^I$ | 0.4 | 0.0 | 0.3 | 0.3 |
| | $T_2^I$ | 0.5 | 0.0 | 0.7 | 0.0 |
| | $T_3^I$ | 0.3 | 0.2 | 0.5 | 0.3 |
| U. S. Equity | $T_0^I$ | 0.2 | 0.0 | 0.1 | 0 |
| | $T_1^I$ | 0.4 | 0.0 | 0.7 | 0.1 |
| | $T_2^I$ | 0.6 | 0.0 | 0.7 | 0.0 |
| | $T_3^I$ | 0.2 | 0.0 | 0.6 | 0.2 |
| Emerging Equity | $T_0^I$ | 0.3 | 0.3 | 0.3 | 0.2 |
| | $T_1^I$ | 0.3 | 0.2 | 0.3 | 0.2 |
| | $T_2^I$ | 0.5 | 0.1 | 0.5 | 0 |
| | $T_3^I$ | 0.2 | 0.1 | 0.4 | 0.1 |
| Emerging Bond | $T_0^I$ | 0.5 | 0.1 | 0.5 | 0.1 |
| | $T_1^I$ | 0.4 | 0.0 | 0.6 | 0.0 |
| | $T_2^I$ | 0.4 | 0.0 | 0.4 | 0.0 |
| | $T_3^I$ | 0.4 | 0.0 | 0.5 | 0.0 |
| USDJPY FX | $T_0^I$ | 0.4 | 0.0 | 0.7 | 0.0 |
| | $T_1^I$ | 0.3 | 0.1 | 0.4 | 0.0 |
| | $T_2^I$ | 0.3 | 0.1 | 0.7 | 0.0 |
| | $T_3^I$ | 0.3 | 0.1 | 0.6 | 0.0 |

ここで $i=1,2,3$ で $T_{i-1}^I$ に得られたデータは，$T_i^I$ の運用実行時には検証データとして扱うことが可能である．そこで，$T_{i-1}^I$ で得られた検証データから推定された表 7 のハイパー・パラメータで $T_i^I$ で運用することを考える．このように運用期間によって $P_1^{th},\ P_2^{th}$ を更新することで，買いシグナルの性質の変化に対処できることが期待される．表 8 に上記のように $P_1^{th},\ P_2^{th}$ を更新したときの $T^I$ での運用結果を示す．ただし $T_0^I$ では検証データを定義できないため第 5.3 節と同様に $P_1^{th}=0.4,\ P_2^{th}=0.0$ として運用する．

しかしながら，殆どの場合で運用パフォーマンスは悪化してしまい，推定パラメータは検証データに基づいた買いシグナルの変化を捉えるが，その変化が実際の運用期間で適合していないことが明らかとなった．今回設定した検証データは 20ヶ月で得られたものであり，学習期間である 160ヶ月に対して短い．そのため，$P_1^{th},\ P_2^{th}$ の最適化によって買いシグナルの性質の局所的な変化まで取り入れてしまい，パフォーマンスが劣化してしまったとも考えれる．

表 8: 最適化された $P_1^{th}, P_2^{th}$ のもとでの SL による運用結果（$L = 3$, FX 完全ヘッジ）

| | $S_1$ | | $S_2$ | | Benchmark | |
|---|---|---|---|---|---|---|
| | $R^*$ | SR | $R^*$ | SR | $R^*$ | SR |
| Port1 | 10.49% | 0.75 | 3.91% | 0.32 | 8.65% | 0.61 |
| Port2 | 8.78% | 0.75 | 4.13% | 0.34 | 10.14% | 0.82 |
| Port3 | 6.82% | 0.59 | 2.88% | 0.24 | 10.11% | 0.71 |
| Japan Equity | 4.64% | 0.46 | -1.70% | -0.16 | 8.11% | 0.44 |
| U. S. Equity | 7.49% | 0.81 | 5.52% | 0.67 | 11.58% | 0.89 |
| Emerging Equity | -0.45% | -0.04 | -1.99% | -0.23 | 3.60% | 0.17 |
| Emerging Bond | 0.50% | 0.12 | 1.65% | 0.39 | 9.89% | 0.90 |

# 6 日次データを用いた検証

月次データを利用する場合には異なる時点に対応した入力データのベクトル間で大部分の要素が共有されてしまい，入力データの実質的な次元数が小さくなる．これにより推定パラメータの過適合が発生し，パフォーマンスに悪影響を及ぼしている可能性がある．従って，日次データを用いてより多数の入力データを生成し，そのデータを学習することでパフォーマンスが改善することが期待される．

## 6.1 データ生成

月次データを利用する場合と同様に以下の方法で非負の整数のベクトルを生成する．最初に入力データ生成のための観測期間に観測した全ての (日次) リターン $R_t^{\mathrm{IN}}$ を，その平均と分散で標準化する．入力データは $[R_{t-M\Delta t^{\mathrm{IN}}}^{\mathrm{IN}}, \ldots, R_{t-\Delta t^{\mathrm{IN}}}^{\mathrm{IN}}]$ ($\Delta t^{\mathrm{IN}}$=1 日) を $t = T_s, \ldots, T_e$ (時点 $t$ の間隔は 1ヶ月) で生成したものに基づくため，リターンの時系列の観測開始時点は $t_s = T_s - M\Delta t^{\mathrm{IN}}$ であり観測終了時点は $t_e = T_e - \Delta t^{\mathrm{IN}}$ である．ここで観測期間 $[t_s, t_e]$ に得られた全ての日次データの個数を $N_{obs}$ とすると，リターンの時系列をその平均と分散で標準化した $Z^{(t_s),\mathrm{IN}} = \{Z_t^{(t_s),\mathrm{IN}}\}_{t=t_s,\ldots,t_e}$ は以下のように生成される．

$$Z_t^{(t_s),\mathrm{IN}} = \frac{R_t^{\mathrm{IN}} - \mu^{(t_s),\mathrm{IN}}}{\sigma^{(t_s),\mathrm{IN}}} \tag{64}$$

$$\mu^{(t_s),\mathrm{IN}} = N_{obs}^{-1} \sum_{t=t_s}^{t_e} R_t^{\mathrm{IN}} \tag{65}$$

$$[\sigma^{(t_s),\mathrm{IN}}]^2 = N_{obs}^{-1} \sum_{t=t_s}^{t_e} (R_t^{\mathrm{IN}} - \mu^{(t_s),\mathrm{IN}})^2 \tag{66}$$

そして $Z_t^{(t_s),\mathrm{IN}}$ をその大きさに基づき分類し，変数 $x^{(t_s)} = \{x_t^{(t_s)}\}_{t=t_s,\ldots,t_e}$ を以下のように生成する．

$$x_t^{(t_s)} = \begin{cases} 3 & Z_t^{(t_s),\mathrm{IN}} \geq 1/2 \\ 2 & 0 \leq Z_t^{(t_s),\mathrm{IN}} < 1/2 \\ 1 & -1/2 \leq Z_t^{(t_s),\mathrm{IN}} < 0 \\ 0 & Z_t^{(t_s),\mathrm{IN}} < -1/2 \end{cases} \tag{67}$$

こうして得られた $x^{(t_s)}$ から $M$ 次元のベクトル $X_t^{(t_s)} = [x_{t-M\Delta t^{\mathrm{IN}}}^{(t_s)}, \ldots, x_{t-\Delta t^{\mathrm{IN}}}^{(t_s)}]$ を $t = T_s, \ldots, T_e$ で生成することでニューラルネットワークの入力データ $X^{(t_s)} = \{X_t^{(t_s)}\}_{t\in T}$ を得る．

次に出力に対応するデータの生成について説明する．月次データを利用する場合と同様に，RL の場合は $\{R_t\}_{t\in T}$ をそのまま利用でき，SL の場合は $\{R_t\}_{t\in T}$ からリターンの大きさに従って分類を行う必要がある．分類を行うときは，$\{R_t\}$ が月次リターンで観測間隔が 1ヶ月であることに注意し，以下のように教師データの被説明変数 $Y^{(T_s)} = \{Y_t^{(T_s)}\}_{t\in T}$, $Y_t^{(T_s)} = \{Y_{t,k}^{(T_s)}\}_{k=0,1,2}$ を生成する．

$$Y_{t,k}^{(T_s)} = \begin{cases} 1 & \text{if } k = K_t^{(T_s)} \\ 0 & \text{otherwise} \end{cases}, \tag{68}$$

$$K_t^{(T_s)} = \begin{cases} 2 & R_t \geq \sigma^{(T_s)}/2 + \mu^{(T_s)} \\ 1 & -\sigma^{(T_s)}/2 + \mu^{(T_s)} \leq R_t < \sigma^{(T_s)}/2 + \mu^{(T_s)} \\ 0 & R_t < -\sigma^{(T_s)}/2 + \mu^{(T_s)} \end{cases}, \tag{69}$$

$$\mu^{(T_s)} = N^{-1} \sum_{t=T_s}^{T_e} R_t \tag{70}$$

$$[\sigma^{(T_s)}]^2 = N^{-1} \sum_{t=T_s}^{T_e} (R_t - \mu^{(T_s)})^2 \tag{71}$$

なお，$Y^{(T_s)}$ の解釈は月次データを利用した場合と同様である．

## 6.2 シミュレーション結果

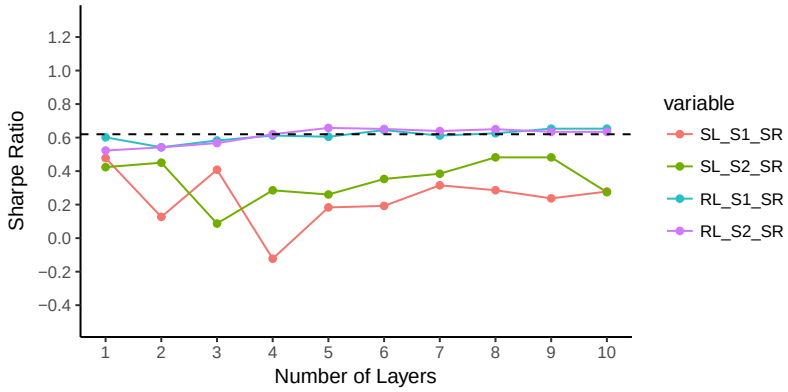
月次データを用いる場合と同様の運用期間 $T^I = [T_s^I, T_e^I]$ と設定し，月次データと同様に学習スケジュール $S_1$, $S_2$ に従ってそれぞれ学習を行う．尚，パラメータは，$\Delta t^{\mathrm{IN}} = 1$ 日以外は 5.3 節と同様のパラメータを設定した．さらに，$L = 1, 2, 3, \cdots, 10$ の場合の学習とその結果に基づく検証を行った．なお，ネットワークの構造は月次データの場合に倣い，入力ユニットは ReLU を $N_{\mathrm{IN}}^{(1)} = 30$，中間層のユニットは ReLU を $N_{\mathrm{IN}}^{(l)} = 60$ $(l = 2, \cdots, L)$，出力ユニットは SL の場合は Classification Unit を $N_{\mathrm{IN}}^{(L+1)} = 3$，RL の場合は Binary Unit を $N_{\mathrm{IN}}^{(L+1)} = 1$ と設定した．このとき得られた Port1–3 のシャープレシオ (SR) をプロットしたものを図 2–3 に示す．

結果を見ると，RL の場合は $L$ の上昇に伴い安定的にベンチマーク以上のパフォーマンス得られる傾向が見られた．一方，SL の場合は，RL と比べて $L$ の変化によるシャープレ

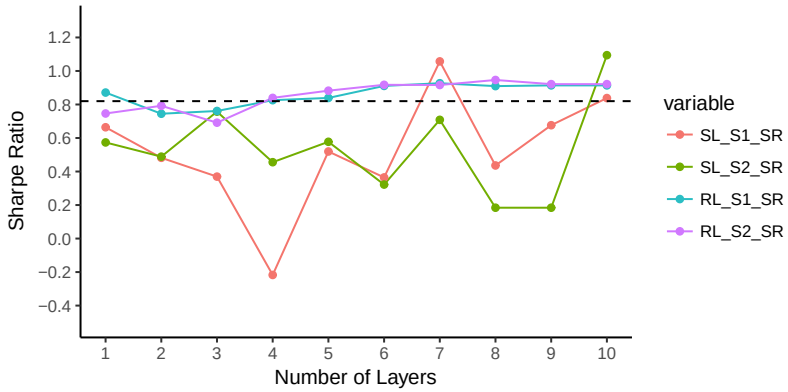
シオの変動が激しく，$L$ を増加させても，安定的にベンチマークを上回るパフォーマンスは得られなかった．

図 2: 日次データ入力時のネットワークの層数変化によるシャープレシオの変化（FX 完全ヘッジ，黒点線はベンチマークのシャープレシオを表す）

(a) Port1, Benchmark: 0.61




(b) Port2, Benchmark: 0.82




(c) Port3, Benchmark: 0.71

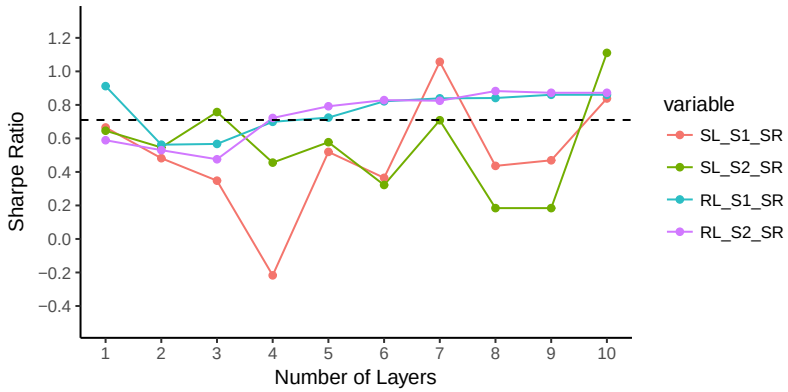
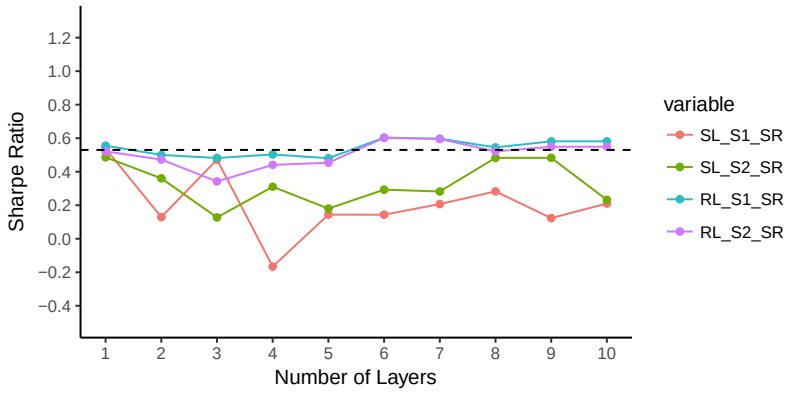
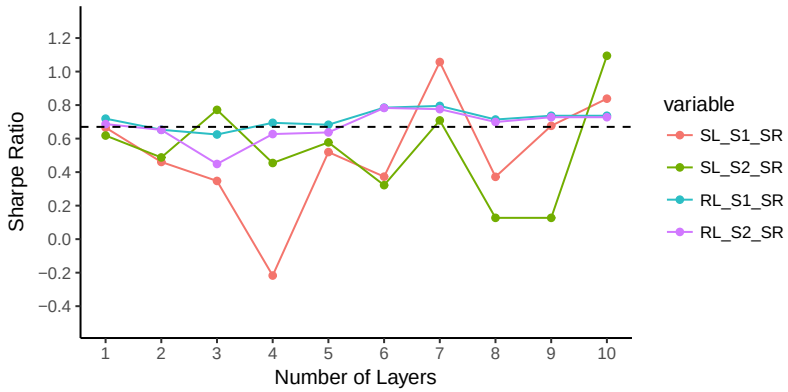


図 3: 日次データ入力時のネットワークの層数変化によるシャープレシオの変化（FX 動的ヘッジ，黒点線はベンチマークのシャープレシオを表す）

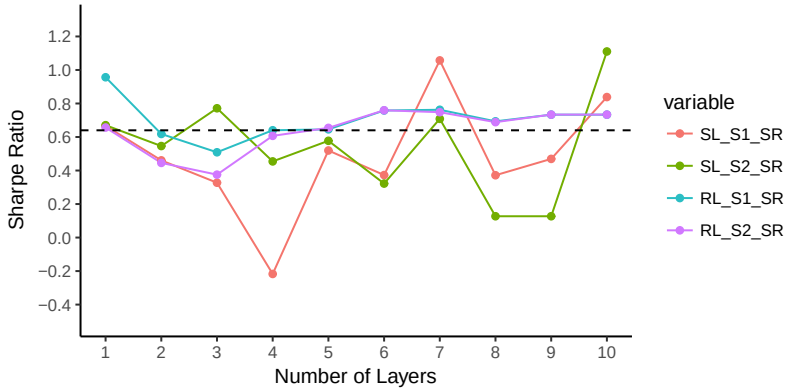
(a) Port1, Benchmark: 0.53




(b) Port2, Benchmark: 0.67




(c) Port3, Benchmark: 0.64




資産別では，RL の Japan Equity に関してのみベンチマークと比して明確なパフォーマンスの改善が観察された．その例として，図 4 は $L = 10$ として RL を実行したときの Japan Equity のみに投資した場合のポートフォリオ価値の推移を示している．(但し時

点 $t = 1$ からの図示となっている.)

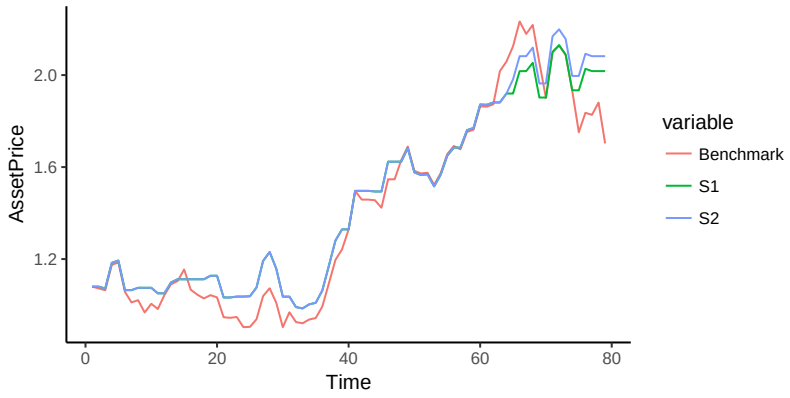
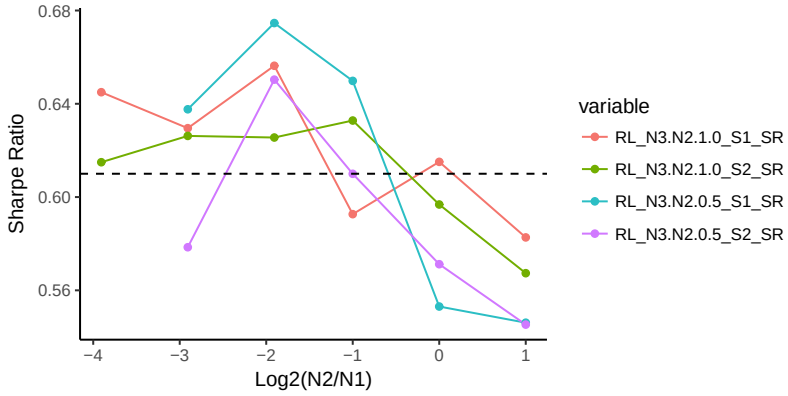



図 4: Japan Equity のみに投資を行った場合のポートフォリオ価格の時系列

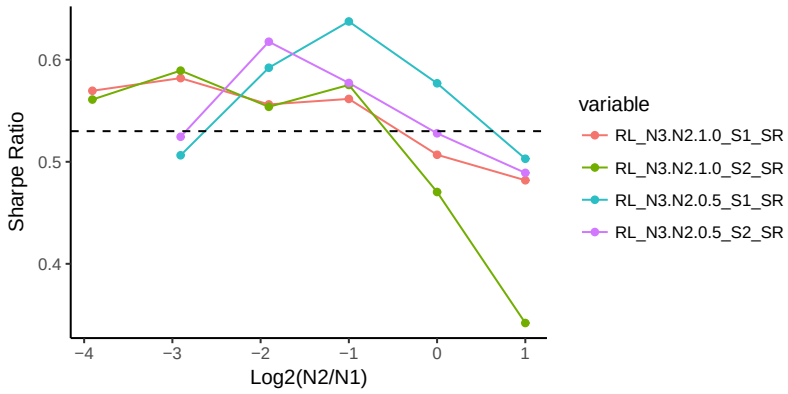
最後に，中間層のユニット数を変化させた場合の影響を検証する．$N_{\mathrm{IN}}^{(1)} = 30$，$L = 3$ として, $N_{\mathrm{IN}}^{(2)}$ を変化させた時の Port1 のシャープレシオの推移を図 5 に示す．但し，点線はベンチマークのシャープレシオを示しており、横軸は　$\log_2(N_{\mathrm{IN}}^{(2)}/N_{\mathrm{IN}}^{(1)})$ である．さらに, $N_{\mathrm{IN}}^{(3)}/N_{\mathrm{IN}}^{(2)} = 1.0, 0.5$ の 2 通りの検証も行った.

まず，全体的な傾向として，$N_{\mathrm{IN}}^{(2)} = 30, 60$ と中間層のユニット数を入力層のユニット数 $N_{\mathrm{IN}}^{(1)} = 30$ 以上にするとシャープレシオが悪化する一方，$N_{\mathrm{IN}}^{(2)} = 15, 8$ 等，中間層のユニット数を入力層のユニット数より少なくすると，相対的に高いシャープレシオが得られることが観察される．さらに，$S_1$ では $(N_{\mathrm{IN}}^{(2)} = 8, N_{\mathrm{IN}}^{(3)} = 4)$ や $(N_{\mathrm{IN}}^{(2)} = 15, N_{\mathrm{IN}}^{(3)} = 8)$ のように, また $S_2$ でも $(N_{\mathrm{IN}}^{(2)} = 8, N_{\mathrm{IN}}^{(3)} = 4)$ のように $L = 2$ から $L = 3$ でもユニット数を減少させると，$N_{\mathrm{IN}}^{(2)} = N_{\mathrm{IN}}^{(3)}$ の場合に比べてシャープレシオが改善することが見て取れる.

以上により，これまでの分析で用いた $N_{\mathrm{IN}}^{(1)} = 30$, $N_{\mathrm{IN}}^{(2)} = N_{\mathrm{IN}}^{(3)} = 60$ 等のようにユニット数を入力層から中間層で増加させるのではなく，減少させる方が運用パフォーマンス改善に繋がることが示唆される.




(a) FX 完全ヘッジ




(b) FX 動的ヘッジ

図 5: $N_{\mathrm{IN}}^{(2)}, N_{\mathrm{IN}}^{(3)}$ を変化させた時の Port1 のシャープレシオ

### 6.3 指数移動平均 (EMA) の利用

前節で SL の場合に問題となった，入力データのノイズへの過適合を抑制するために，入力データからノイズを取り除くことを考える．例えば，Nakano 等 [14] は，入力データに観測値をそのまま用いるよりも EMA を用いた場合の方が良い予測結果をもたらすと報告している．本稿でも，EMA の生成過程によってデータ観測のノイズを除去し，パフォーマンス向上や層数の変化に対するパフォーマンスの安定性が向上するか否かを検証する．特に，EMA の計算は以下のようにして行う．

$$\mathrm{EMA}(\beta)_t = \beta R_t^{\mathrm{IN}} + (1-\beta)\mathrm{EMA}_{t-\Delta t^{\mathrm{IN}}},\ t > t_0, \tag{72}$$

$$\mathrm{EMA}(\beta)_{t_0} = R_{t_0}^{\mathrm{IN}}. \tag{73}$$

ここで $\beta \in (0,1]$ はハイパーパラメータであり，$\beta = 1.0$ のときは入力データをそのまま用いた時に対応する．また，$t_0$ はデータが観測可能となった時点であり，$t_0 \leq t_s$ である．この値は対象となるアセットクラスによって異なり，表 9 のようになる．

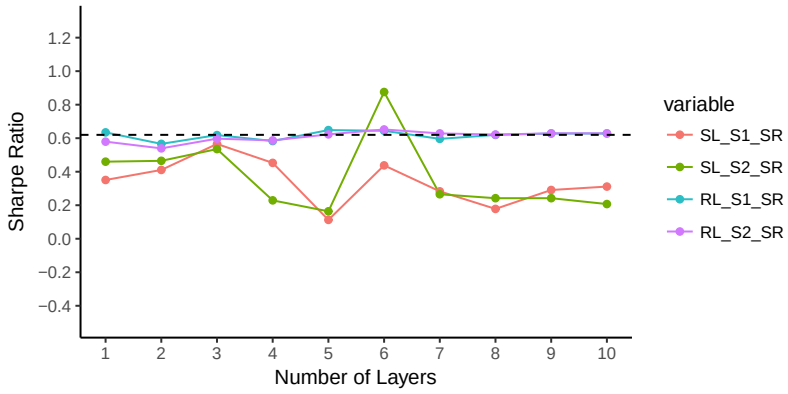
表 9: 各アセットクラスの日次データ観測開始時点

| | |
|---|---|
| Japan Equity | 4 Jan., 1990 |
| U. S. Equity | 2 Jan., 1990 |
| Emerging Equity | 2 Jan., 1994 |
| Emerging Equity | 2 Jan., 1994 |
| USDJPY FX | 1 Jan., 1990 |

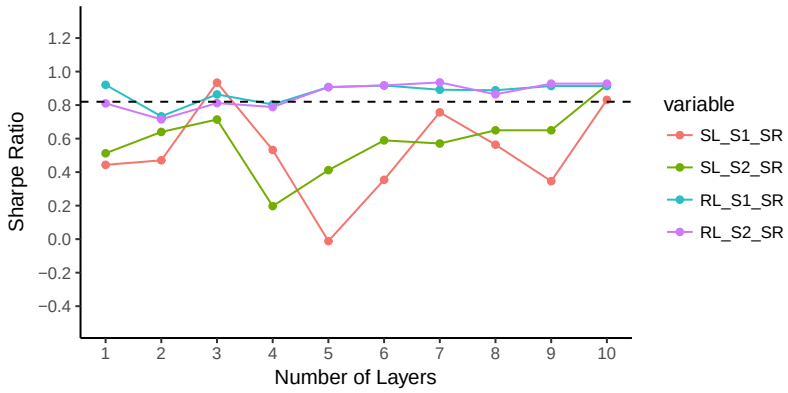
本稿では，$\beta = 0.9,\ 0.5,\ 0.1$ に対する $\{\mathrm{EMA}(\beta)_t\}$ を 式 (64) – (66) において $R_t^{\mathrm{IN}}$ の替わりに用いることにより，ノイズの除去を企図する．しかしながら，結果的には，RL では $\beta$ の値に依らず，層数の変化に対するパフォーマンスの変化の傾向に大きな違いはみられなかった．SL においても，ベンチマークと比べてパフォーマンスは依然低いケースが多く，パフォーマンスの向上には結びつかなかった．その例として，以下の図 6, 7 において，$\beta = 0.9$ の場合の，$L = 1, \cdots, 10$ に対する Port1–3 のシャープレシオ (SR) をプロットしたものを示す.

図 6: $\beta = 0.9$ の日次 EMA を入力時のネットワークの層数変化によるシャープレシオの変化（FX 完全ヘッジ，黒点線はベンチマークのシャープレシオを表す）

(a) Port1, Benchmark: 0.61




(b) Port2, Benchmark: 0.82




(c) Port3, Benchmark: 0.71

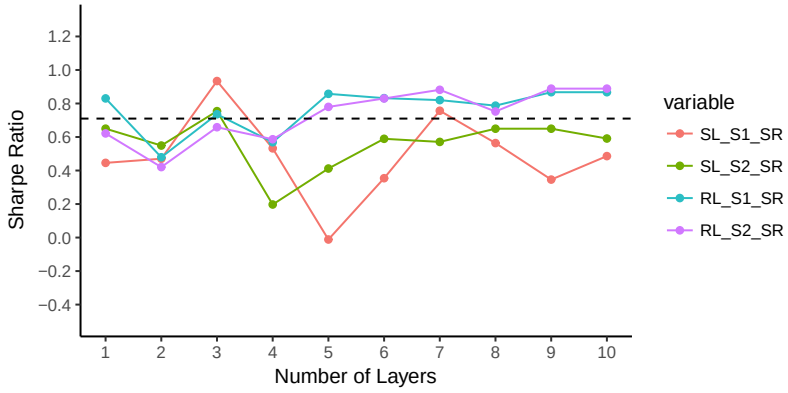
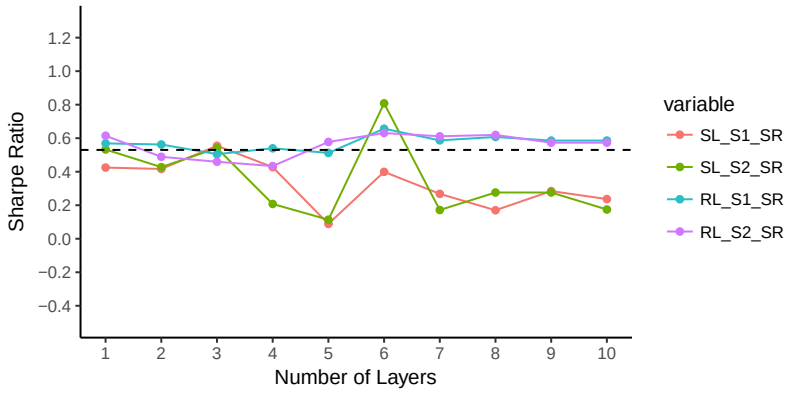
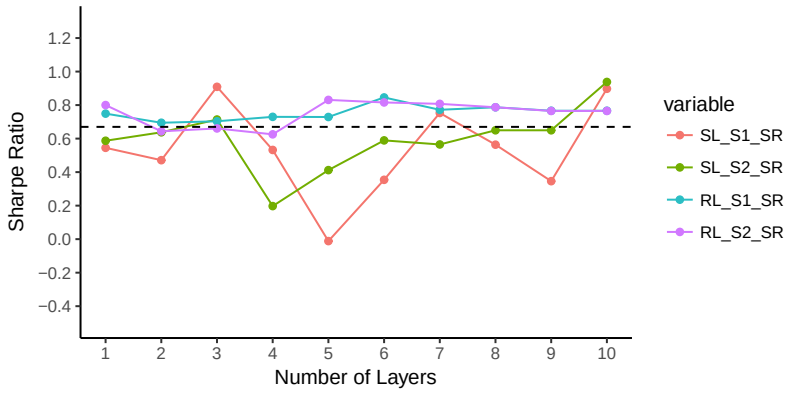


図 7: $\beta = 0.9$ の日次 EMA を入力時のネットワークの層数変化によるシャープレシオの変化（FX 動的ヘッジ，黒点線はベンチマークのシャープレシオを表す）

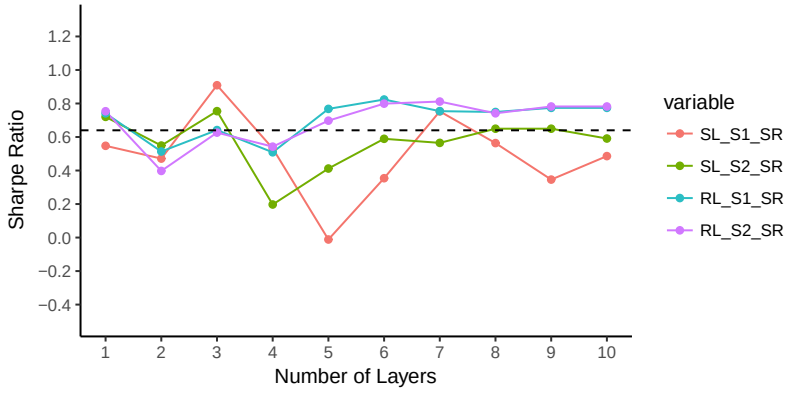
(a) Port1, Benchmark: 0.53




(b) Port2, Benchmark: 0.67




(c) Port3, Benchmark: 0.64



# 7 結論

## 7.1 本稿の要約

本稿は，異なる損失関数を定めることにより様々な投資戦略を表現できることに注目し，ニューラルネットワークを活用した深層学習に基づく投資手法を検討した．特に，その例として，多層のニューラルネットワークを構成することにより，リターンの観測値のアノマリーを検知し，それを投資に活用する手法を教師付き深層学習（SL）を用いて開発した．また，多層のニューラルネットワークに基づく深層強化学習（RL）を用いて，リターンの観測値が与えられたときに学習期間で得られる収益を最大にする投資手法も開発した．さらに多層のニューラルネットワークの学習でしばしば問題となる勾配消失問題克服のために DBN(Deep Belief Network) を用いた事前学習を行い，効率のよい学習を可能にした．

次に，学習結果に基づく投資戦略により月次データと日次データを用いた検証を行い，SL, RL の選択によるパフォーマンスの違いを分析した．その結果，SL, RL の選択はもちろん，同一の学習方法においても，ネットワークに入力するデータ，検証期間中のパラメータ再学習の有無，ネットワークの層数，各中間層のユニット数等，分析者が外生的に設定する項目により投資パフォーマンスが大きく異なることが明らかとなり，それらを注意深く選択する必要があることが分かった．以下では，月次，日次データ各々の検証結果を要約する．

月次のリターン・データに基づく検証結果の特徴:

- 全体として，本稿で採用した SL, RL に基づく運用は，相対的にはポートフォリオ (Port1,2,3) に対して有効となる結果を示した．また，それらのポートフォリオに対して，パラメータ更新によるパフォーマンスの明確な改善効果は見られず，むしろ SL においては悪化した．さらに，$L = 1, 2, 3$(層数=1,2,3) の範囲における多層化による改善効果も見られず，むしろ RL の $S_1$(パラメータ更新なしの場合) においてはパフォーマンスは悪化する傾向が観測された．総じて，RL の $L = 1, 2$ がポートフォリオにおいてベンチマーク対比で良好なパフォーマンスを示す結果となった．

- SL において，前期の検証データから推定された最適なハイパー・パラメータを用いて当期に運用することを試みたが，殆どの場合で運用パフォーマンスは悪化した．

日次のリターン・データに基づく検証結果の特徴：

- ポートフォリオ運用に関し層数 $L = 1, \cdots, 10$ に対して，RL の場合は $L$ の上昇に伴い安定的にベンチマーク以上のパフォーマンスが得られる傾向が見られた．一方，SL の場合は，RL と比べて $L$ の変化によるシャープレシオの変動が激しく，$L$ が増加してもパフォーマンスは安定的にベンチマーク以上にはならなかった．資産別では，RL の Japan Equity に関してのみベンチマーク対比で明確なパフォーマンスの改善が観察された．

- RL によるポートフォリオ (Port1) 運用において中間層のユニット数を変化させたときの影響：$L = 3$ の場合, $N_{\mathrm{IN}}^{(2)} = 30, 60$ のように中間層のユニット数を入力層のユニット数 $N_{\mathrm{IN}}^{(1)} = 30$ 以上にすると相対的にシャープレシオが悪化する一方，$N_{\mathrm{IN}}^{(2)} = 15, 8$ 等，中間層のユニット数を入力層のユニット数より少なくすると, ベンチマークを上

回る相対的に高いシャープレシオが得られた．さらに，$N_{\mathrm{IN}}^{(2)} = 8, N_{\mathrm{IN}}^{(3)} = 4$ のように $L = 2$ から $L = 3$ でもユニット数を減少させると，減少させない $N_{\mathrm{IN}}^{(2)} = N_{\mathrm{IN}}^{(3)} = 8$ の場合に比べてシャープレシオが改善することも観察された．

従って，当初設定した $N_{\mathrm{IN}}^{(1)} = 30$, $N_{\mathrm{IN}}^{(2)} = N_{\mathrm{IN}}^{(3)} = 60$ 等のようにユニット数を入力層から中間層で増加させるのではなく，逐次減少させる方が運用パフォーマンス改善に繋がる可能性が示唆された．より詳細な分析は今後に譲ることとしたい．

- リターンの観測値の替りにその指数移動平均 (EMA) を用いることにより，観測データのノイズを除去しパフォーマンス向上や，層数の変化に対するパフォーマンスの安定性が向上することを企図したが，RL のポートフォリオ運用では EMA のパラメータ $\beta$ の値に依らず，層数の変化に対するパフォーマンスの変化の傾向に大きな違いはみられなかった．SL のポートフォリオ運用においても，ベンチマークと比べてパフォーマンスは依然低いケースが多く，パフォーマンスの向上には結びつかなかった．

## 7.2 今後の課題

資産運用においては，各投資主体の目的に適合したパフォーマンスの達成が目標であり，そのための投資戦略の設定が肝要となる．特に，ニューラルネットワークを用いた深層学習に基づく場合，様々なパフォーマンス尺度を反映した損失関数等を用いた戦略の表現及び，ネットワークの入出力形態，層数，各層のユニット数とその推移構造や活性化関数の特定等ネットワークのデザインが重要であることが，本稿の分析においても明らかとなった．従って，深層学習の投資への活用のためこの問題意識に沿った様々な研究課題が考えられる．推定に関する技術的な点も含めたより具体的な課題としては，本稿で例示した手法以外の投資戦略の検討の他，例えば以下の事項が挙げられる．

- 推定すべきパラメータが大量に存在するため，次元確保のために学習期間を多くとる必要がある．日次データを用いることでデータの次元を増やすことはできるものの，推定すべきパラメータはネットワークの層数を増やすことにより爆発的に増加する．この点に対処する方法として [8] と同様に確率的に入力データの個数を増幅する手法を適用することが考えられる．

- ニューラルネットワークの構造をハイパー・パラメータとして扱い，損失関数の最適化によって推定することが考えられる．なお，深層学習におけるハイパー・パラメータの推定をランダムサーチによって行う方法は Bergstra-Bengio[15] で既に提案されている．本稿で例示した戦略を含む様々な投資手法において，ネットワーク構造を推定対象とするアプローチの有効性を検証する必要がある．その一方，ランダムサーチでは，設定された検証データに基づいてハイパー・パラメータが決定されるため，データへの過剰適合が問題となる可能性があり，ファイナンスの理論的，実務的視点を考慮しネットワーク構造を検討する必要もある．

- 市場データの非定常性への対処が必要である．入力した観測データに適合するようにニューラルネットワークのパラメータが決定されてしまうため，将来の変動に対

してどの程度柔軟に対処できるか否かはわからない．一方，例えば，将来の予測に対して有意義なリターンモデルを作成しそのパラメータをフィルタリング等で推定できれば，その結果を活用することにより入力市場データに対する投資パフォーマンスの頑健性が向上する可能性もある．

## 参考文献


[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 25:1097–1105, 2012.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, nov 2012.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei a Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[4] Zhang Yudong and Wu Lenan. Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications*, 36(5):8849–8854, 2009.

[5] Ricardo de A. Araújo, Adriano L.I. Oliveira, and Silvio Meira. A hybrid model for high-frequency stock market forecasting. *Expert Systems with Applications*, 42(8):4081–4096, 2015.

[6] Jigar Patel, Sahil Shah, Priyank Thakkar, and K. Kotecha. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268, 2015.

[7] Jin Zhang and Dietmar Maringer. Using a Genetic Algorithm to Improve Recurrent Reinforcement Learning for Equity Trading. *Computational Economics*, 47(4):551–567, 2016.

[8] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–12, 2016.

[9] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning*, number 3, pages 807–814, 2010.

[10] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–54, 2006.

[11] Geoffrey E. Hinton. A Practical Guide to Training Restricted Boltzmann Machines. Technical report, Department of Computer Science, University of Toronto, 2010.

[12] 岡谷貴之. **深層学習**. 講談社, 2015.

[13] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[14] Masafumi Nakano, Akihiko Takahashi, and Soichiro Takahashi. Generalized exponential moving average (EMA) model with particle filtering and anomaly detection. *Expert Systems with Applications*, 73:187–200, 2017.

[15] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

## 付録

以下の表は，本文の数値例で適用したミニバッチ勾配降下法と事前学習の層数 ($L$ の数) 毎の学習率と繰り返し回数を掲載する.（尚，表の「学習率」,「繰り返し回数」はそれぞれ，ミニバッチ勾配降下法の学習率，繰り返し回数である.）

表 10: 学習率と学習の繰り返し回数

(a) $SL$

| 層数 | 学習率 | 繰り返し回数 | 学習率 (事前学習) | 繰り返し回数 (事前学習) |
|---|---|---|---|---|
| 1 | 0.001 | 20000 | 0.00005 | 3000 |
| 2 | 0.001 | 20000 | 0.00005 | 3000 |
| 3 | 0.001 | 20000 | 0.00005 | 3000 |
| 4 | 0.001 | 20000 | 0.00005 | 3000 |
| 5 | 0.001 | 20000 | 0.00005 | 3000 |
| 6 | 0.0005 | 20000 | 0.00005 | 3000 |
| 7 | 0.0005 | 20000 | 0.00005 | 3000 |
| 8 | 0.00025 | 20000 | 0.00005 | 3000 |
| 9 | 0.00025 | 20000 | 0.00005 | 3000 |
| 10 | 0.00025 | 20000 | 0.00005 | 3000 |

(b) $RL$

| 層数 | 学習率 | 繰り返し回数 | 学習率（事前学習） | 繰り返し回数（事前学習） |
|---|---|---|---|---|
| 1 | 0.001 | 20000 | 0.00005 | 3000 |
| 2 | 0.001 | 20000 | 0.00005 | 3000 |
| 3 | 0.0005 | 20000 | 0.00005 | 3000 |
| 4 | 0.0005 | 20000 | 0.00005 | 3000 |
| 5 | 0.0005 | 20000 | 0.00005 | 3000 |
| 6 | 0.0001 | 20000 | 0.00005 | 3000 |
| 7 | 0.00005 | 20000 | 0.00005 | 3000 |
| 8 | 0.00005 | 20000 | 0.00005 | 3000 |
| 9 | 0.00005 | 20000 | 0.00005 | 3000 |
| 10 | 0.00005 | 20000 | 0.00005 | 3000 |