タイトル未定

はじめに

従来型の時系列予測の研究に関しては主に統計分析手法と計量経済学モデルを用いて行っているが、1980年代以降、金融市場の分析に機械学習の手法が徐々に適用されるようになった。しかし、複雑な関数近似をしなければ分類や回帰ができない場合があり、従来型の機械学習手法ではうまくいかないケースも多く、このような問題に対して、ディープラーニング（深層学習）手法を使用するケースが増えている。ディープラーニングの領域の中でベースの手法はニューラルネットワークと呼ばれる機械学習手法である。しかし、通常のニューラルネットワークでは、ある層の出力は、次の層の入力に利用されるのみであり、時系列データを自然に扱うことができない。時系列データを予測するとき、過去の状態を中間層として定義したリカレント（再帰型）ニューラルネットワーク（以下：RNN）が用いられる場合が多い。RNNには時間を深く遡るほど勾配爆発あるいは勾配消失の問題があり、長・短期記憶（Long short-term memory、略称:LSTM）により解決された。金融市場の分析において、多くの研究はLSTMがいい予測結果が出られ、期待すべき研究方法であることを証明した。本文はLSTMを用いて　　　の予測を行う。

先行研究

White（1988）[1]は、ニューラルネットワークを用いて、IBMの株収益率を予測した。Atiyaなど(1997)[2]は上場企業の株収益率、売上高と利益率などの基本的な企業情報に基づいて、ニューラルネットワークを用いて、予測を行った。結果として、ニューラルネットワークの予測結果は投資者の参考になる。金融市場において、RNNの研究に関しては、Xieなど(2016)[3]はRNNを用いて、中国の株市場に関する実証分析を行った。RNNは投資者の参考になるよい予測結果を出した。また、LSTMの発展に関しては、HochreiterとSchmidhuber(1997)[4]は再帰型ニューラルネットワークを基づいて、LSTMを提出した。Gers&Schmidhuber（2000）[5][6]は忘却ゲートと覗き穴結合（peephole connection）をLSTMに導入し、LSTMを改良した。MaknickieneとMaknickas(2012)[7]はLSTMを用いて、為替市場のユーロ／ドルを予測し、取引戦略を立った。結果としてLSTMの予測に基づいた戦略は実行可能である。Murtaza　Roondiwalaなど(2015)[8]はLSTMを用いて、NIFTY50の収益率を予測した。LSTMはいい予測能力を持っていることを証明した。松井藤五郎と汐月智也(2017)LSTMを用いて1分ごとの1分後利益率をモデル化する手法を提案した。結果は、LSTMは1分後利益率の時系列をモデル化できることが確認された。南正太郎(2017)はLSTM-RNNを用いたイベント考慮後の株価時系列予測した。結果は、LSTMは金融の世界では幅広い可能性を持っていることである。

研究方法

1、RNNの勾配爆発と消失の問題

RNNでは、層自体は入力層、中間層、出力層と一般的なニューラルネットワークと変わらないが、時刻tにおけるに加え、時刻t-1における中間層の値を保持しておき、それも時刻tにおける中間層に伝える点が異なる。また、各層の入力として、前の層の入力のみではなく、時系列のデータポイントも入力とする。だから、RNNは時系列データの予測問題を解決することができる。

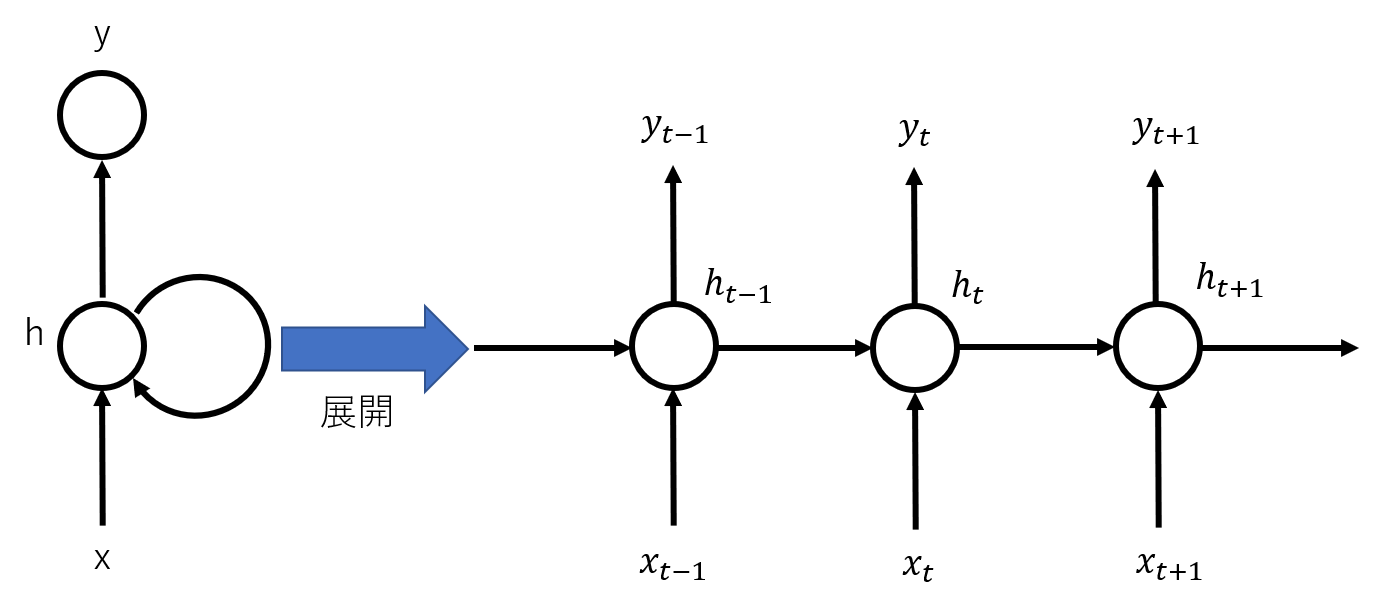


図１（RNNの概要図）

　　図１で示したように、時刻ｔの中間層の計算をする際に入力層からの入力と前の時刻の中間層の出力に重み付けを行った結果を加算したものを中間層への入力として扱う。この際にポイントが2つある。

　　一つ目は、一般的なRNNでは時刻のｔ＝0のときには時刻t-1の中間層が存在しないため、時刻t-1の中間層からの入力が0として設定する。

　　二つ目は、入力層から中間層への重み、中間層から中間層への重み、中間層からその上の層への重みは別の時間軸関でそれぞれ同じものを用いる。すなわち、一つのニューラルネットワークを構築して、その中間層の計算にとある特定の重みを用いて時間展開をしていくということになる。その際の活性化関数としてはよくtanhが用いられる。

　　そのため、ある時刻tにおける中間層からの出力を、時刻tにおける入力、時刻入力層から中間層への重みとバイアス、中間層から中間層への重みを用いて書くと以下のようになる。

これから、RNNの勾配を計算する

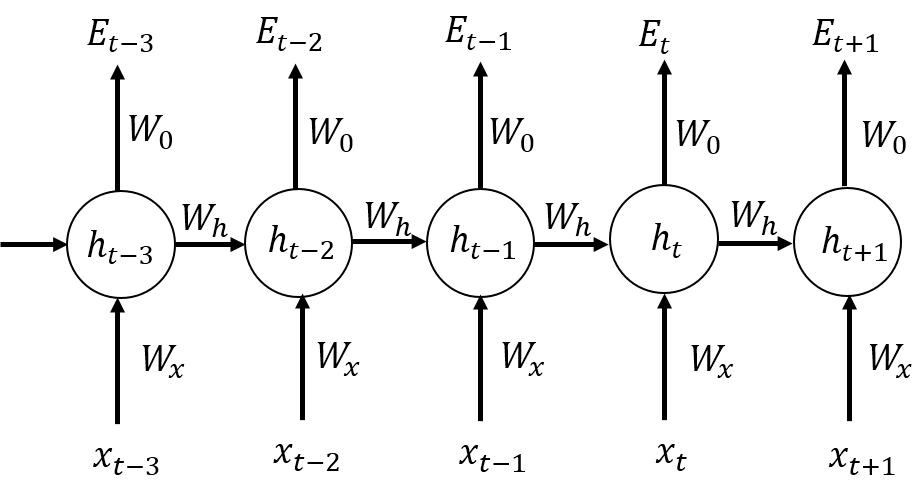


図２（RNNの展開図）

はｔ時刻の入力

はｔ時刻の中間層の出力

は誤差

は入力層から中間層の重み

は中間層同士の重み

は中間層から出力層への重み

RNNの勾配計算の手法としてメジャーなものはとしてBack Propagation Through Time(BPTT)である。

RNNでは各時刻が重みを共有して、中間層から出力層まで

t时刻における誤差関数はクロスエントロピーと定義される。

だから、時刻tにおける誤差をとすると、合計誤差は次のように表すことができる。

これから、誤差関数の値Eを最小になるために、すべての重み、、、についてそれぞれの勾配を計算する。

まずは、連鎖律を用いると、中間層から出力層への重みに関する誤差関数の偏微分は以下のようになる。

次は、中間層同士の重みについての勾配を計算すると以下のようになる。

はt時刻の中間層の出力を表すため、式（１）によると、はの関数であり、に関する計算はとに依存する。また前の時刻t-1のときに、であり、 に関する計算はとに依存する。だからを展開する必要がある。

を展開すると

そして、を展開すると、

式（８）を式（7）に代入すると

これにより、式(6)のという部分が展開された。

式（9）を式（6）に代入すると

最後に、入力層から中間層への重みに関する誤差関数の偏微分を計算する。の計算と同様に、連鎖律を用いると、重みに関する誤差関数の偏微分は以下のようになる

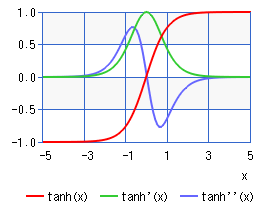
を展開すると、

先ほど計算した勾配の特性上、普通のRNNは長時間のデータを意識した学習をすることができないという問題が起こる。

重みに関する誤差関数の偏微分との計算に関しては、時刻tだけではなくて、時刻kからtまで、すべての時刻を一緒に計算する必要があるから、また、総乗の部分が時間に依存するため、時刻ｊと時刻ｋの間が大きく離れた場合は勾配爆発と勾配消失の問題が発生する。

なぜなら、の計算結果は

tanh関数の導関数の値の範囲がはであるため



Tanh関数のグラフ

勾配爆発が発生するの原因は

>4のとき、 、もしJとKの間の距離が大きすぎると、連乗も多すぎて，勾配爆発が発生し，無限に近づいてしまう。

勾配消失が発生する原因は

<4のとき、であるため、距离过大，会导致连乘项过多，

勾配消失が発生する。

勾配消失が起こると、ある程度先の時刻以降の誤差は全く考慮しないで重みを更新するという現象が起こっている。実際には4－５時刻程先の誤差を考慮した勾配を計算するとほぼ0になる。

2. 改良したRNN―LSTM

LSTMは、Hochreiter&Schmidhuber(1997)により提出され、RNNの中間層出力に対して、記憶期間の長さの考え方を導入することにより、遠い過去の出力の影響を保持することを可能にしている。

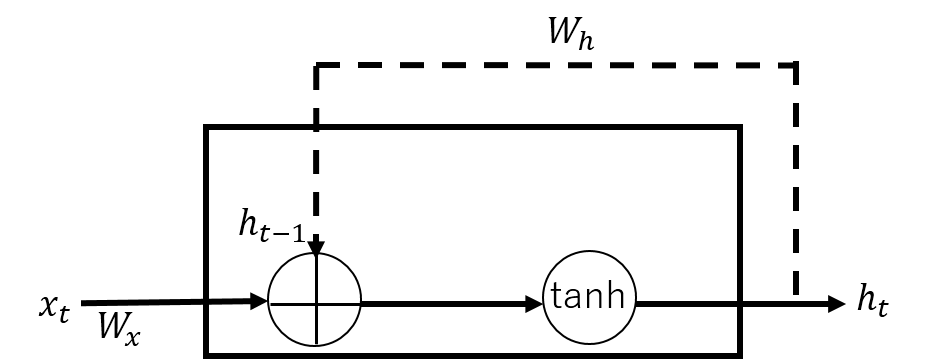


図　従来のRNN

LSTMは勾配爆発問題と勾配消失問題をCEC（Constant　Error　Carousel）という概念を導入するにより解決することができる。図の中心にあるCECは、受け取った値をそのまま過去の値として保持し、次の時刻に伝える。すなわち、破線の矢印が時間を遡った伝播を、+印のノードが値の掛け合わせを表す。例えば、前の時刻t-1における出力と今の入力の重み付き線形和をそのまま通すではなく、前の時刻t-1におけるCECの値を加算して出力される。

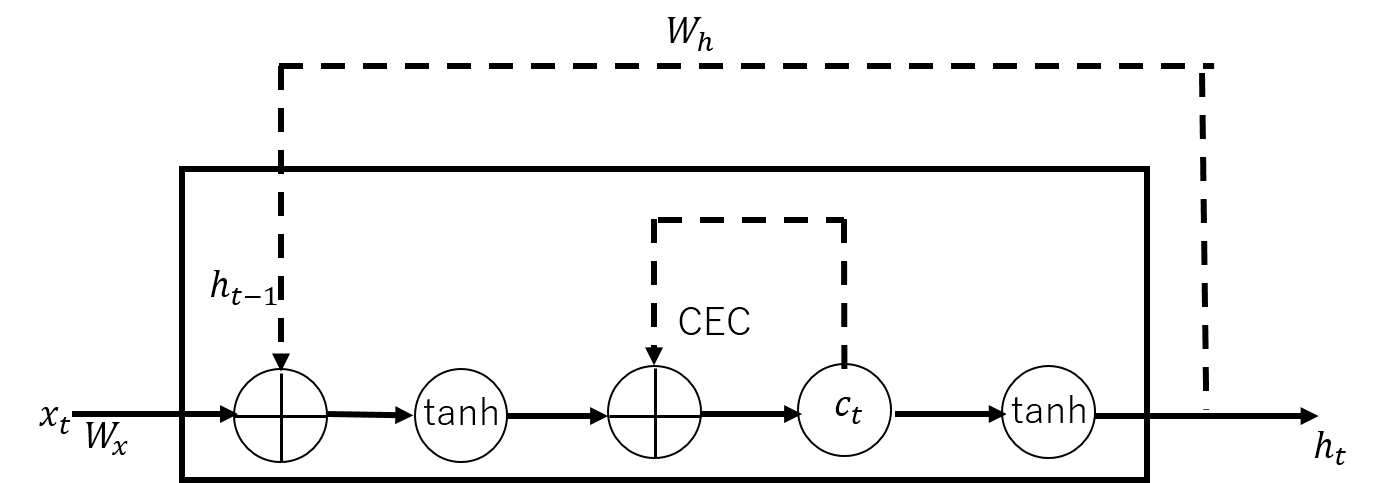


図　CECの導入

CECを導入することにより、過去の入力情報をすべて記憶し、過去を遡っても誤差を逆伝播できるようになったが、時系列データを学習するうえで、もう1つ大きな問題がある。時系列データを入力で受け取る場合は、時間依存性がある信号を受け取ったは重みを大きくし活性化すべき一方、依存性がない信号を受け取ったときは重みを小さくし非活性のままであるべきである。しかし実際は、ニューロンが同じ重みでつながっている限り、両者はお互いに打ち消し合う重みの更新となってしまうので、特に長期依存性の学習がうまくできないことに繋がる。

この問題を解決するためには、依存性のある信号を受け取ったときのみ活性化し、それ以外では依存性のありそうな情報を内部で保持しておく機構が必要になる。後者はCECにより実現できるが、前者の問題は、必要になったタイミングでのみ信号を伝播し、それ以外では信号を遮断するゲートのような存在を取り入れるべきである。ゲートの導入することによって各時刻ごとに入出力を制御することが可能になる。

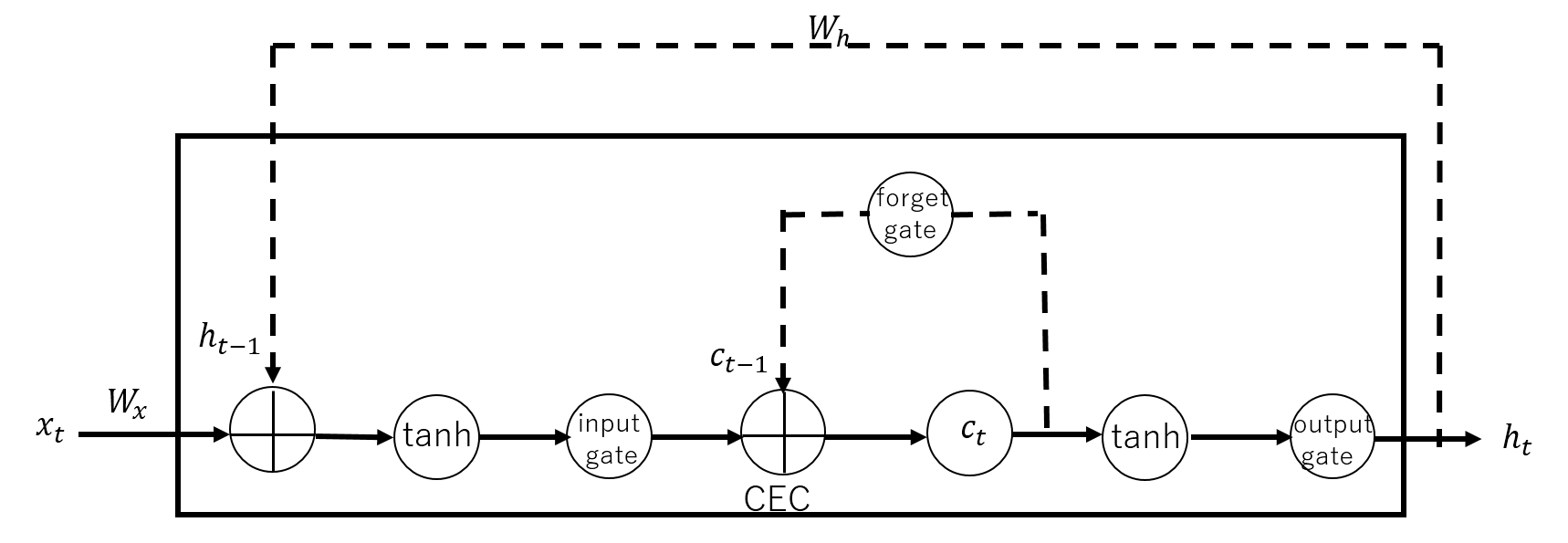


図　ゲートの導入

ゲートはLSTMが発表された1997年は入力ゲートと出力ゲートの二つだけであったが、1999年にはGers & Schmidhuberは入力系列のパターンが変わった時にCECの状態を一気に変更できるように忘却ゲートを導入した。図のようにCECの入力部分に入力ゲート、出力部分に出力ゲートを導入することで、入出力ともに過去の情報が必要になったタイミングでのみゲートを開け信号を伝播し、それ以外はゲートを閉じておくことで過去の情報を保持しておくことが可能になる。また、忘却ゲートに関してはCECからの誤差を受け取ることで、必要なタイミングでCECに記憶された値を忘れ去る機能を果す。

しかし、ゲートの制御に用いるのは時刻ｔにおける入力層の値がおよび時刻t-1における中間層の値であり、制御すべきCEC自身が保持している値は用いられていない。一見するとを制御に用いていることによりCECの状態が反映されているように思われるが、あくまでもLSTMブロックの出力は出力ゲートに依存しているので、仮に出力ゲートがずっと閉じている場合、どのゲートもCECにアクセスすることができず、CECの状態を見ることができないという問題が発生する。

この問題を解決するためにGers & Schmidhuber, 00は覗き穴結合（peephole connection）を導入した。これはCECから各ゲートをつなぐもので、これによりCECの状態を各ゲートに伝えることができるようになる。

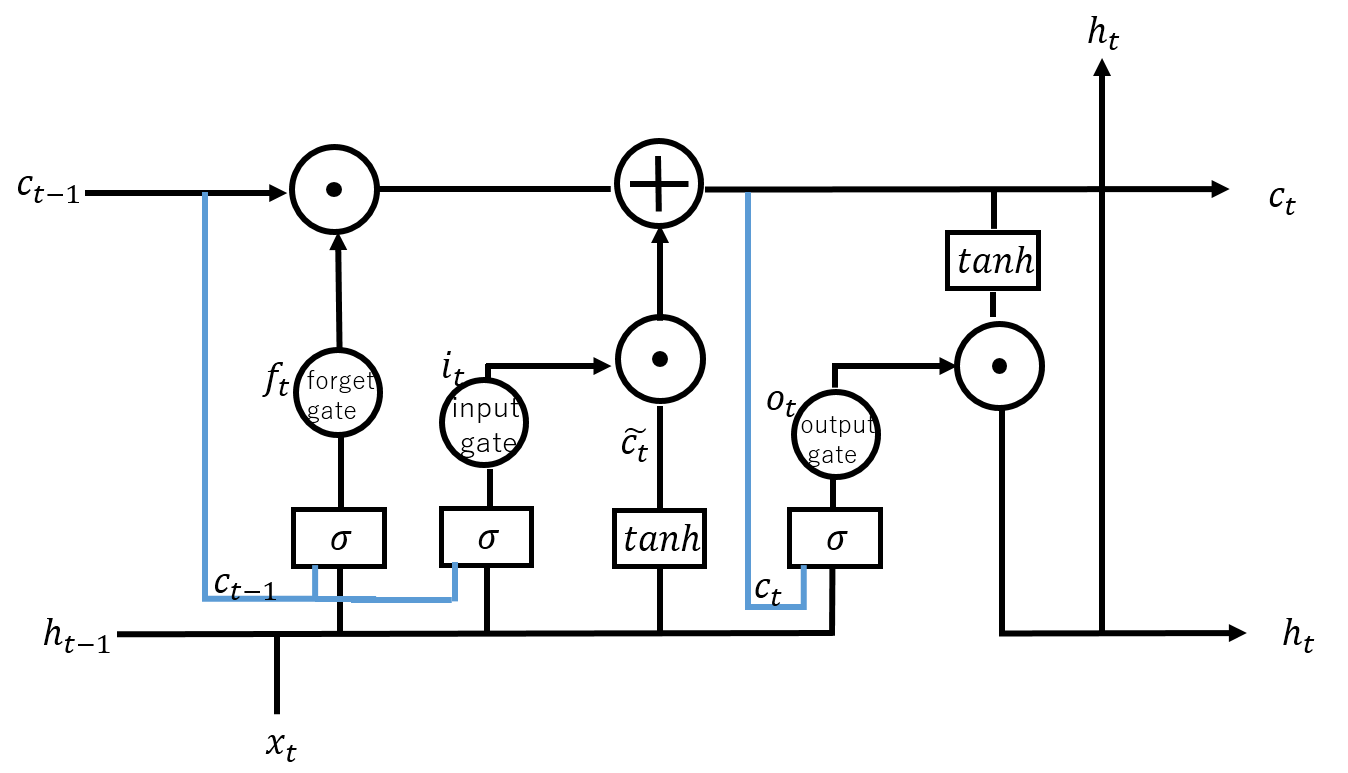


図　peephole connectionを導入したLSTM

具体的にそれぞれのゲートの式は以下のようになる。

入力ゲート

入力データ

出力ゲート

忘却ゲート

⊙はアダマール積であり、

、、の計算に関してはそれぞれt時刻における入力と前の時刻における中間層の出力の重み付き線形和にシグモイド関数を作用され、0.0～1.0の間の値に調整された。ゲートの値が1.0に近い場合はそれぞれの入力を通過させ、0.0.に近い場合はシャットダウンする。

また、はtanh関数を作用されることにより－1.0～1.0の間の値になるように調整された。はただの入力データであり、ゲートを制御するわけではない。

これから、忘却ゲートを作用させた前の時刻のCECの値Cｔ－１と入力ゲートを作用させた入力を時刻ｔの入力を加算してCECを更新する

それに活性化関数TANHを作用させた上で、出力ゲートを作用させて中間層出力とする。

⊙tanh()

最後の結果の出力は

=

これがLSTMの仕組みである。

、、、を定義する。すると、これらを

という形でまと　。こうしておくことで、を求めればよいことになるので、

よって：

これらの式にある、、、は、

一方、により

3データの前処理

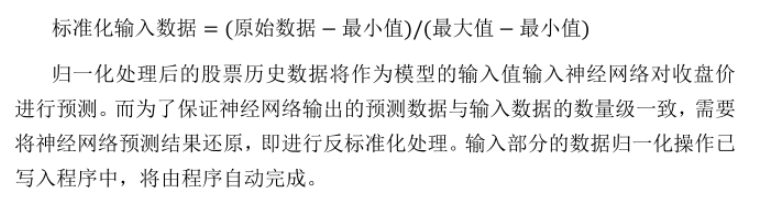
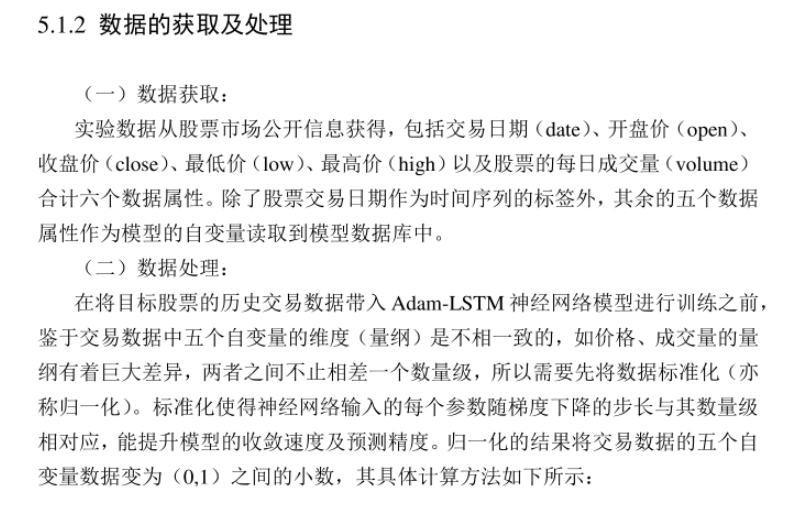
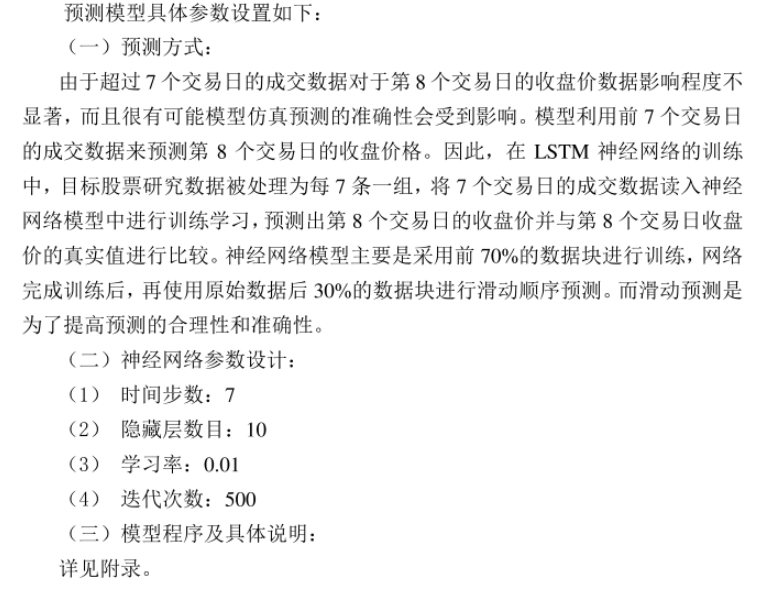
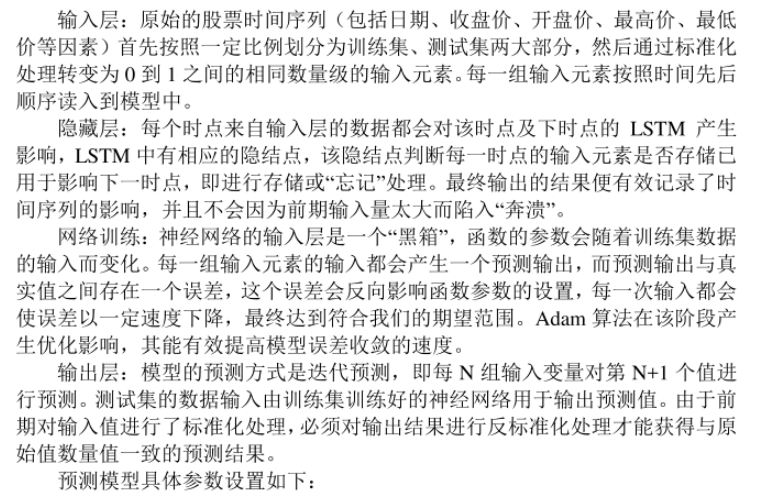
データの中における最大値と最小値を使って正規化する方法です。

この処理をすることで、データは最大値が1,最小値が0のデータとなります。あらかじめ最大値と最小値の範囲が限られている場合には有効な手法です。(逆に、外れ値が存在すると他の値の差がほとんど無くなってしまう可能性があります)

では、これもNumPyで実装していきます。

1999 年に Hochreiter らによって提唱された LSTM-RNNに基づいている。計算するにあたって、学習時には、Epoch 数を 50、100、500 の３つのパターンで計算。隠れ層のユニット数は 50、LSTM 層は 1 とした。学習のフレームワークとしては、フロントエンドに Keras、バックエンドには Tensorflowを使用して実装している。確率的勾配降下法の最適化を行う上で、Adam(Adaptive Moment Estimation) により学習率の設定を行っている。

研究結果



本文で使われるコード

参考文献

[1]H.White.Economic Prediction Using Neural Networks:the Case of IBM Daily Stock Returns.Neural Networks[J].IEEE International Conference on.1988.2(6):451-458

[2]Atiya A,Talaat N,Shaheen S.An Efficient Stock Market Forecasting Model Using Neural Network[C].International Conference on Neural Networks.1997.4:2112-2115

[3]Xie X K,Wang H.Recurrent Neural Network for Forecastin Stock Market Trend[J].International Conference on Computer Science,2016

[4]HOCHREITER S,SCHMIDHUBER J.Long short-term memory[J].Neural Computarion,1997,9(8):1735-1780

[5]GERS F A,SCHMIDHUBER J,CUMMINS F.Learning to forget:Continual prediction with LSTM[J].Neural Computation,2000,12(10):2451-2471

[6]GERS F A,SCHMIDHUBER J,CUMMINS F.Recurrent nets that time and count.Neural Networks,2000.IJCNN 2000,Proceedings of the IEEE-INNS-ENNS International Joint Conference on,volume 3,pp.189-194,IEEE,2000

[7]MAKNICKIENE N,MAKNICKAS A.Application of neural networks for forecastin of exchange rates and forex trading[C].The 7th International Scientific Conference"Bussiness and Management 2012".Vilnius,Lithuania:Vilnius Gediminas Technical University,2012:10-11

[8]Murtaza Roondiwala, Harshal Patel, Shraddha Varma.Predicting Stock Prices Using LSTM[J].International Journal of Science and Research(IJSR),2015,ISSN(Online):2319-7064

[9]

[10]

还有两个日文文献