APIR Discussion Paper Series No.48
2021/02

テキストデータを利用した新しい景況感指標の開発と応用(下)
― 応用編：深層学習を利用したテキスト分析―

生田祐介　関和広　松林洋一

大阪産業大学経営学部商学科
甲南大学知能情報学部
神戸大学大学院経済学研究科

APIR 一般財団法人 アジア太平洋研究所
ASIA PACIFIC INSTITUTE OF RESEARCH

テキストデータを利用した新しい景況感指標の開発と応用(下)

― 応用編：深層学習を利用したテキスト分析 ―

生田　祐介[1]，関　和広[2]，松林　洋一[3]

【要旨】

本稿では，テキストマイニングに基づいて，新聞や雑誌の文字情報から景況感指数の計測を行う際の実践的な手続きについて解説していきます。

テキストマイニングの基本的概念については，入門編において丁寧に説明を行い，その簡単な実例についても紹介しました。文字情報から景気の動向を抽出するという試みは，極めて魅力的ですが，基本的なテキストマイニングの手法には限界もあります。最も重要な問題点は，新聞記事等の文章に含まれている単語を単体として取り出して，その単語のみで景気の良し悪しを判定することは適当ではないということです。表現を変えれば景気の良し悪しの基本となる単語を，文章全体の中で理解していくという手続きが不可欠です。

こうした手続きは，昨今，機械学習，深層学習と呼ばれる人間の脳神経回路を模した情報処理方法によって考察することが可能です。そこで本稿では，深層学習の代表的な手法であるニューラルネットワークと呼ばれるモデル（およびその修正版）について丁寧に解説を行います。あわせて上巻で紹介した内閣府「景気ウォッチャー調査」をもとに，こうした新たな手法を用いた景況感指数の計測を紹介し，その特性を見ていくことにします。

[1] 大阪産業大学経営学部商学科, ikuta@dis.osaka-sandai.ac.jp（照会先）
[2] 甲南大学知能情報学部, seki@konan-u.ac.jp
[3] 神戸大学大学院経済学研究科, myoichi@econ.kobe-u.ac.jp

本稿は，APIR における研究プロジェクト「テキストデータを利用した新しい景況感指標と開発と応用」の一環であり，その成果の一部を初心者向けに平易に解説したものです（本稿と入門編である上巻の２分冊から構成されており，いずれも専門外の一般読者を想定しています）。

本稿の作成に当たっては，宮原秀夫 APIR 所長，猪木武徳先生，本多佑三先生，稲田義久先生，青山秀明先生，池田裕一先生，岩野宏 APIR 代表理事，中山明 APIR 総括調査役から有益なコメントをいただきました。ここに記して感謝申し上げます。なお本稿における誤謬はすべて筆者の責任です。

下巻の方針

上巻では，単語のみに注目して，情報をとらえようとしました。つまり，単語を利用してセンチメントインデックスを作成していたことになります。作成されたインデックスは，既存の景気動向指数と比べて，期間を通じて動きの方向性や進度に違いが生じることを確認しました。そうした差異が生じる原因として，単語を含んだ文脈の情報が考慮されていないことが挙げられます。すなわち，単語自体が持つ印象（極性値）は，全体に占める一部の情報しかとらえることができず，それゆえ，まとまった情報に対して適切な極性値(その情報が有する良しあしの程度)を与えることができなくなるのです。このため，文全体に対して評点を与える方法を用いる必要があります。

下巻では，単語の順序を考慮した複雑な解析をすることで，人間のように景況感を推定するという研究を報告します。ビジネスエコノミストは，日々新聞等の記事を読み，景況感への重みづけを意識しています。この知的活動を人工知能（AI）に置き換え，曖昧性を排除し，客観的で効率的な判断を実現できないでしょうか。この課題に対するアプローチとして，機械学習を利用したテキストマイニングを適用します。

機械学習とは，「人間が機械（コンピュータ）に入力と出力の正解パターンを教えることで，新しい入力データに対して機械が自動的に正解を出力する」というものです。機械学習は，音声，画像，そして言語のように，様々な対象を扱うことができます。

機械学習には様々な手法がありますが，本稿では深層学習（ニューラルネットワークという人間の脳神経回路を模したモデルによる機械学習方法）を用います。具体的には，内閣府「景気ウォッチャー調査」における景気判断理由文（テキスト）と景気判断（数値）のペアを学習データに利用することで，モデルのパラメータ値を推定します。そして，推定されたパラメータに対して新聞記事を入力することで，指数を出力させて，それを景気動向指数とします。推定の基本モデルには，リカレントニューラルネットワーク(Recurrent Neural Network，以下 RNN)を採用します。推定誤差を小さくするよう RNN を修正することで，モデルからの出力値の精度を高めるよう工夫します。既存モデルを改善することにより，地域・国の景況感判断に適用を試みます。

下巻の構成は以下の通りです。第 5 章では，上巻の第 4 章で明らかとなった単語を分析単位とするテキストマイニングの課題を振り返り，文を評価する方法について検討します。さらに，深層学習という，人間の脳の構造を模した分析モデルの利用可能性について述べます。第 6 章では，テキストデータを念頭に置いて，深層学習の仕組みについて基礎的な解説を行います。第 7 章では，文脈を考慮して文を評価する方法を解説します。具体的に，リカレントニューラルネットワークという深層学習の応用モデルを紹介します。最後の第 8 章では，深層学習を利用して，テキストデータから景況感指数を推定した結果を紹介します。

全体構成のイメージ

【上巻】

テキストデータの位置づけ

- 第1章 データとは

テキストマイニングの基本的な手法の解説

- 第2章 言葉をデータ化することの意義
- 第3章 テキストを分析単位にする方法とは

テキストマイニングの経済分野への応用と課題

- 第4章 テキストマイニングを実践する

【下巻】

課題解決に向けた改良指針

- 第5章 単語の評価から文の評価へ

テキストマイニングの発展的手法の解説

- 第6章 深層学習の基礎
- 第7章 文脈を考慮した深層学習

発展的手法を用いた新たな景況感指数の開発

- 第8章 景況感指数の足元予測

第 5 章　単語の評価から文の評価へ

5-1. 伝統的なテキストマイニングの課題

上巻で説明した伝統的なテキストマイニングは，単語を分析単位としていました。この方法は，分析対象とするテキストの中に，どの単語が何回登場したのかを利用します。これは，例えば「単語コレクター」の行動を再現したものと言えます。単語を分析単位として文章評価を行う場合，単語に備わったセンチメント（ポジティブまたはネガティブの数値表現）が，結果を左右することになります。そうしたテキストマイニングは，文全体の構造を把握していないため，結果的に，得られた景気判断は実感に沿わないものでした。

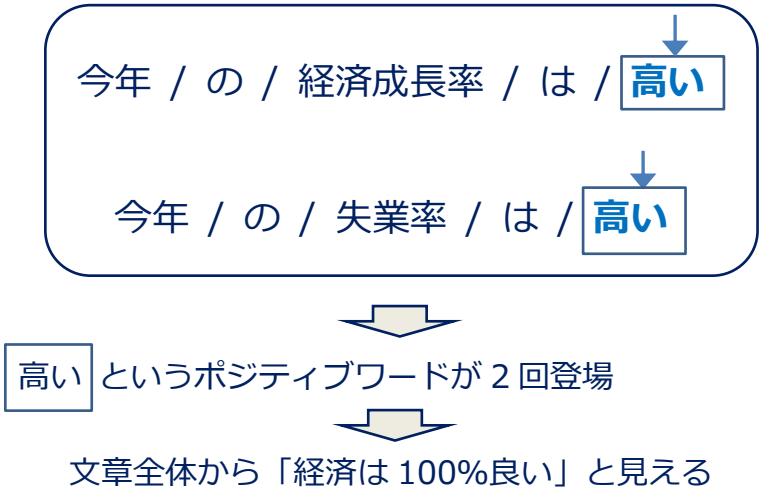
図表 5-1 は，単語を分析単位として景気を判断する場合，その結果に違和感が発生する例を表しています。分析対象は，「今年の経済成長率は高い」と「今年の失業率は高い」という二つの文で構成される文章であるとします。分析の前に，二つの仮定を置きます。まず，文章全体の評価は形容詞のみで決まるものとします。次に，“高い”という形容詞のセンチメントは，ポジティブであるとします。この下で，この文章に対して伝統的なテキストマイニングを行うと，「経済は 100%良い」と評価することができるでしょう。なぜなら，“高い”と言うポジティブな形容詞が，２回登場するからです。しかし，この結果は，人の目には不自然に映ります。

実際のビジネスエコノミストであれば，**図表 5-2** のように，文章全体を見渡して評価するでしょう。どちらの文にも，同じように“高い”というポジティブな単語がありますが，文脈から判断すると，文ごとの印象は異なります。きっと多くの人は，図の上の文をポジティブ，下の文をネガティブであると判断するのではないでしょうか。このため，文章全体では，「経済は 100%良い」と評価することはできません。こうした判断は，人にとっては当たり前に見えますが，単語を分析単位とする伝統的なテキストマイニングにおいては当たり前ではありません。

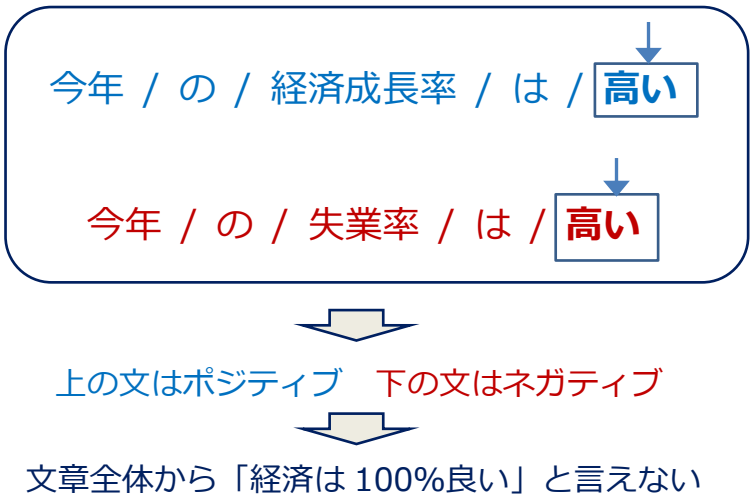
5-2. 解決のアイデア

伝統的なテキストマイニングをどのように発展させれば，自然な分析結果を得ることができるでしょうか。そのためのヒントは，上巻でも取り上げたとおり，文脈をとらえて評価できるような分析手法を用いることです。私たちは，文に登場するそれぞれの単語の意味や印象を理解していますが，同時に，文のどこに，どのような単語があるかという，単語同士の位置関係，すなわち文の構造も理解しています。文中に“高い”という述語があるとき，主語が“経済成長率”の場合はポジティブな意味になりますが，他方で“失業率”の場合はネガティブな意味になります。これは，単語の系列が一つの意味を有する，ということを意味します。

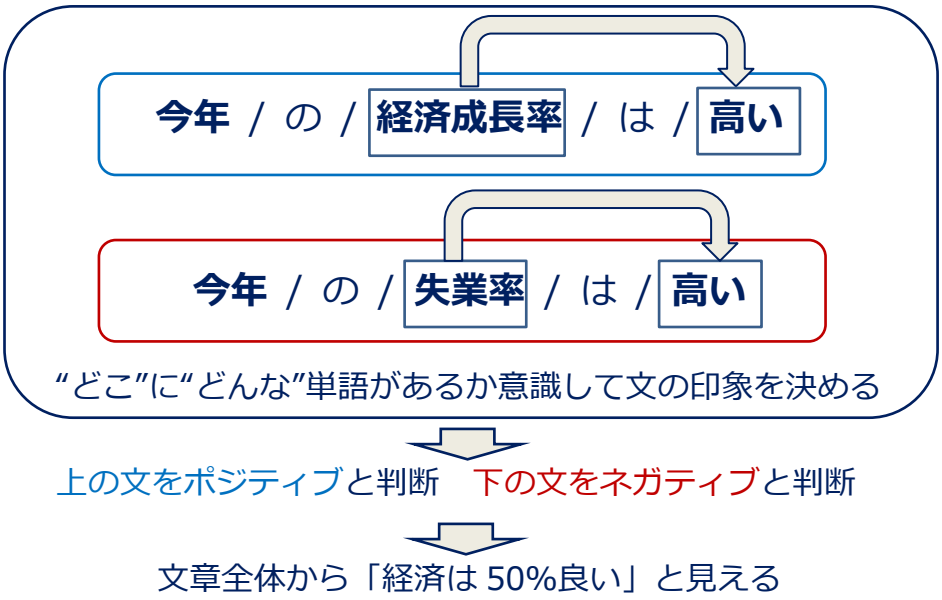
図表 5-1：単語を分析単位とする景気判断の結果



図表 5-2：単語を分析単位とする景気判断の課題



図表 5-3：単語の系列を考慮した景気判断



図表 5-3 は，単語の系列を分析単位と見なして，景気判断する例を表しています。二つの文のどちらも，“高い”という単語の前方に，どのような単語があるかによって，文全体の印象が決まります。上の文をポジティブ，下の文をネガティブと判断できるため，文章全体では「経済は 50%で良い」と評価できると考えます。これは，単語を分析単位とした**図表 5-1** と比べて，明らかに異なる結果です。

上記で述べた単語の系列を評価する方法は，単語ごとにセンチメントが与えられることを前提としています[4]。しかし，全ての単語についてセンチメントを準備することは，現実的ではないでしょう。また，たとえ全ての単語についてセンチメントを準備できたとしても，単語同士のセンチメントをどのように組み合わせて，文としてのセンチメントを導き出せばよいのでしょうか[5]。そうした疑問を解決する糸口は，人間の思考に近いフレームワークを使うことにあると考えます。

実際に，人が文章読解するとき，全ての単語の印象を覚えて，文の意味や印象を決めてはいないでしょう。むしろ，様々な文を読むという経験を通じて，適切な評価や感想を出せるようになると考えられます。その結果，伝えたい評価を言葉で記述することができるようにもなるでしょう。そうした人間に近い思考で文章評価を行うために，機械学習の一つであるニューラルネットワークを利用した深層学習に注目します。

---

[4] 上巻では，高村(2006)による感情極性辞書から，単語のセンチメントを参照している。

[5] 付録で述べている通り，n-gram という方法により，文中の任意の区間で単語の系列をとらえることができる。しかし，特定できた系列全体に対してセンチメントを与えることは難しい。

第６章　深層学習の基礎

6-1. ニューラルネットワーク

私たちは日常生活において，単語の位置関係から文脈を読み取り，文の持つ意味を理解しています。単語の系列として文を理解するという認知活動を行う人を「読書家」であるとしましょう。テキストマイニングを用いて景気判断を行うためには，「単語コレクター」ではなく，「読書家」の認知活動をモデル化して，コンピュータで解析する方が望ましいのではないでしょうか。そのためには，ニューラルネットワークが適していると考えます。**ニューラルネットワーク**とは，「人間の脳の構造を模した分析の枠組みであり，数学的には関数の一種で，入力値と出力値を関係付ける情報処理モデル」です。技術的なことを説明する前に，まずは，経済関連の文を，ニューラルネットワークで解析することのイメージを持っていただきたいと思います。

以下の**図表 6-1** は，「今年の経済成長率は高い」（以下，文①と表記）という文と，「今年の失業率は高い」（以下，文②と表記）という文に対して，それぞれ適切な評価を，ニューラルネットワークで解析する過程を表しています。ここでの解析とは，スタート（文）からゴール（評価）まで，経路を構築する作業であると考えてください。評価の集合は，{良い，悪い} で構成されています。文①にとっての評価は {良い} であり，文②にとっての評価は {悪い} であると想定して，それぞれ文と評価がペアになっています。以下では，文と評価を結びつけるように，人の頭の中でどのように情報を伝達させればよいかを考えます。

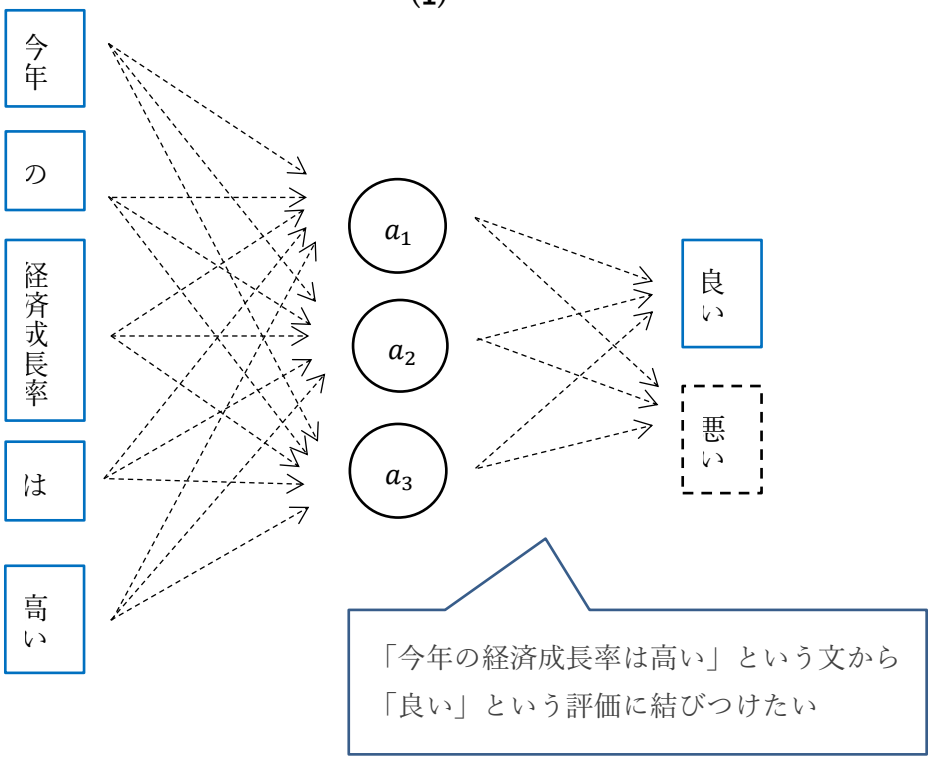
文と評価を結びつける経路を構築するためには，**ユニット**という「情報の交通整理をおこなう関所」が重要な役割を果たします。ユニットは，**図表 6-1** の中で $a_i\ (i = 1, 2, 3)$ と表されており，３つあります。各ユニットは，５つの単語から情報を受け取り，２つの景気判断へ反応信号を出力します。ユニットの数が１つではなく，２つ以上ある理由は，ユニットの情報処理能力に限度があるからです。たとえば，会議で５人の出席者から意見を聞き取って，経済の判断を下す場面を考えてみてください。聖徳太子のように，５人の意見を同時に一人で聞き分けて判断することは難しいでしょう。しかし，３人の審査員が異なる観点で５人の意見を聞き取り，その結果を照らし合わせた方が，情報量を減らすことができるため，判断は容易になります。

6-1-1. 会議の例

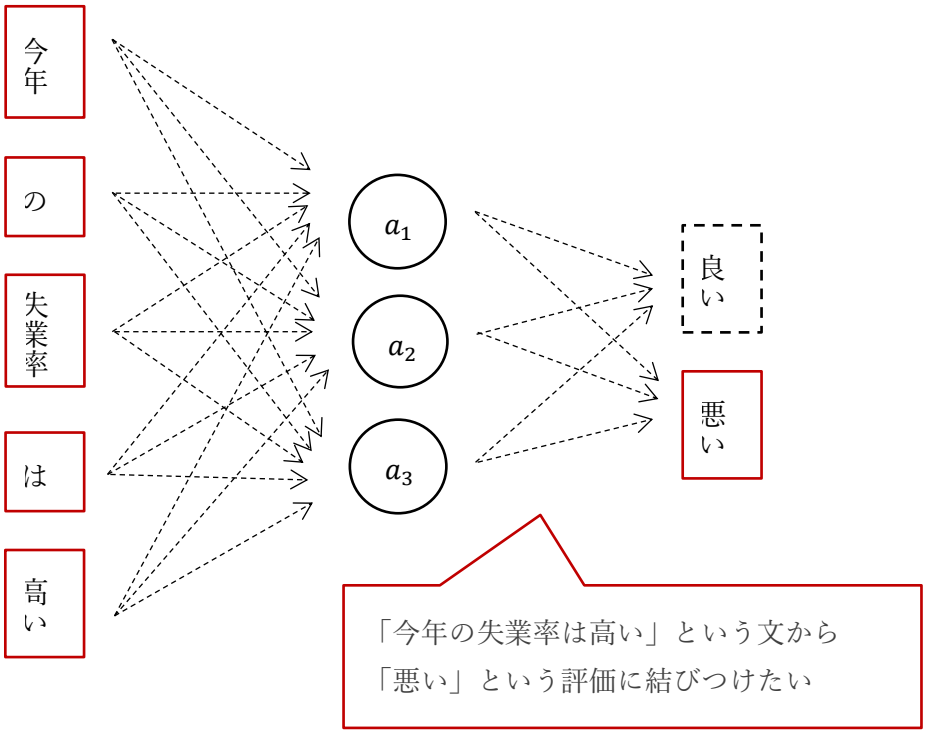
会議の例を使って，ニューラルネットワーク解析を，具体的に考えてみましょう。解析を進めるにあたって，３つのルールを設けます。第１に，会議の出席者の意見に対する総合評価の正解を事前に決めておきます。ここでは，会議の出席者からポジティブ（ネガティブ）な言葉が多く出てくれば，総合評価の正解はポジティブ（ネガティブ）であると仮定します。

図表 6-1：ニューラルネットワークによる文の評価

(1)



(2)



第２に，各審査員に聞き取りの役割を与えます。ここでは，審査員 A はポジティブな言葉に反応し，審査員 B はネガティブな言葉に反応し，審査員 C はポジティブな言葉とネガティブな言葉の両方に反応するように，役割を与えるとします。第３は，審査員の反応を通じた総合評価の決め方です。ここでは，審査員の反応の多数決によって総合評価が決まるとします。

上記のルールの下で，もしも会議で５人の出席者の意見からネガティブな言葉が出ず，ポジティブな言葉のみ出てきた場合，３人の審査員のうち審査員 A と審査員 C という２人が強く反応するため，総合評価はポジティブとなります。つまり，審査員は，会議の出席者の意見から正解を導きだすことができました。

会議の例において重要なことは，各審査員に適切な役割を与えることです。先述の設定と異なり，３人の審査員全てに対して，ネガティブな言葉に強く反応し，ポジティブな言葉を過小評価するような役割を与えたとします。この場合，５人の出席者の意見にポジティブな言葉が多く含まれていたとしても，どの審査員もポジティブな言葉を聞き流してしまうため，総合評価をネガティブと判定してしまうでしょう。結果的に，各審査員に誤った役割を与えていたことになります。一般的な言葉で説明すると，入力された情報を集約して正解を導くためには，大量の情報を濃縮して変換する役割を，各ユニットへ上手く調整して割り振ることが重要なのです。そして，ユニットによる情報変換から得られた複数の中間結果をブレンドすることで，最終評価を導きます。

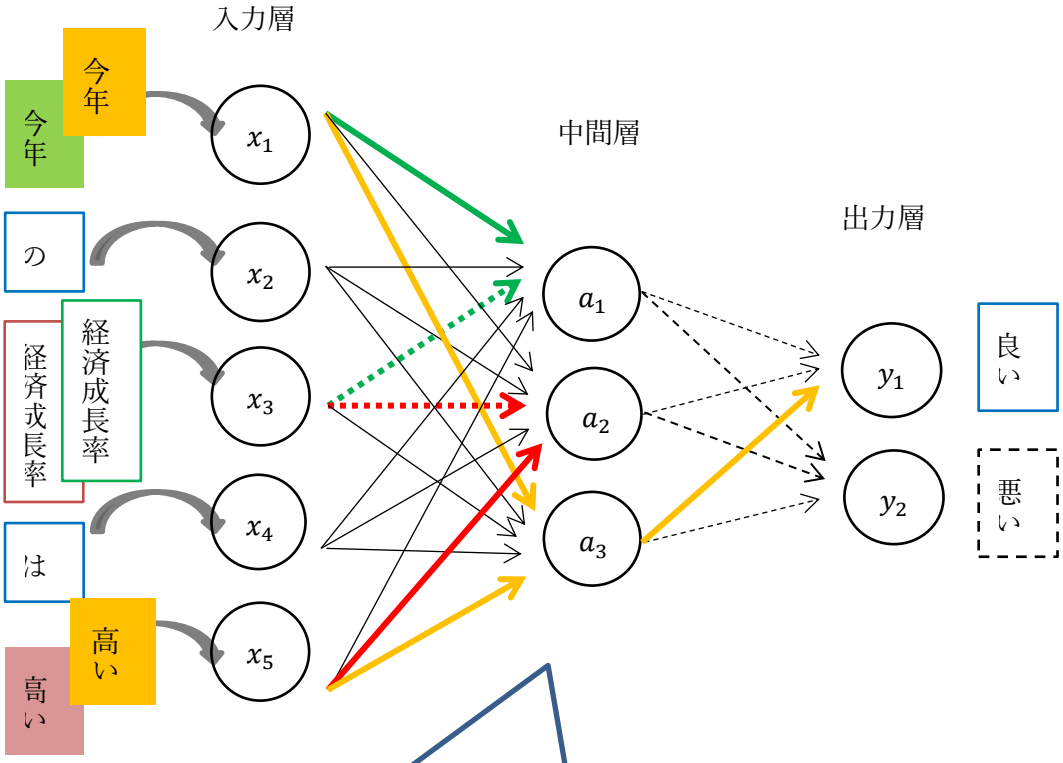
6-1-2. ニューラルネットワークの部品

ここまでの説明で用いた**図表 6-1** は，文と評価がユニットを通じて繋がっていることを表していました。次に，ユニットが情報を変換する仕組みについて，順を追いながらより詳細に説明します。そのために，**図表 6-1** で省かれていたニューラルネットワークを構成する部品を，明示的に扱います。

図表 6-2 は，先述の文①と文②を，ニューラルネットワークの一般的なモデル表現に当てはめて表現したものであり，**図表 6-1** から４つの修正があります。一点目は，**入力ユニット**という，入力される単語の受け皿を明示的に扱います。この例では，５つの単語を入力するので，入力ユニットは丸で囲まれた５つの $x_i\ (i = 1, \dots, 5)$ で表されています。「各入力ユニットのまとまり」を，**入力層**と呼びます。二点目は，会議の例で「審査員」として登場している通常のユニットを，**中間ユニット**と言い換えて，先ほどと同様に $a_i\ (i = 1, \dots, 3)$ で表しています。「各中間ユニットのまとまり」を，**中間層**と呼びます。三点目は，景気の評価を表す候補を，**出力ユニット**に置き換えて，$y_i\ (i = 1, 2)$ で表しています。評価の「良い」を y_1 に，「悪い」を y_2 に，それぞれ対応させます。「各出力ユニットのまとまり」を，**出力層**と呼びます。四点目は，各中間ユニットの役割を明示的に表していることです。各中間ユニットは，５つの入力ユニットから情報を受け取るのですが，反応の仕方が異なります。

図表 6-2：ニューラルネットワークの解析

(1)

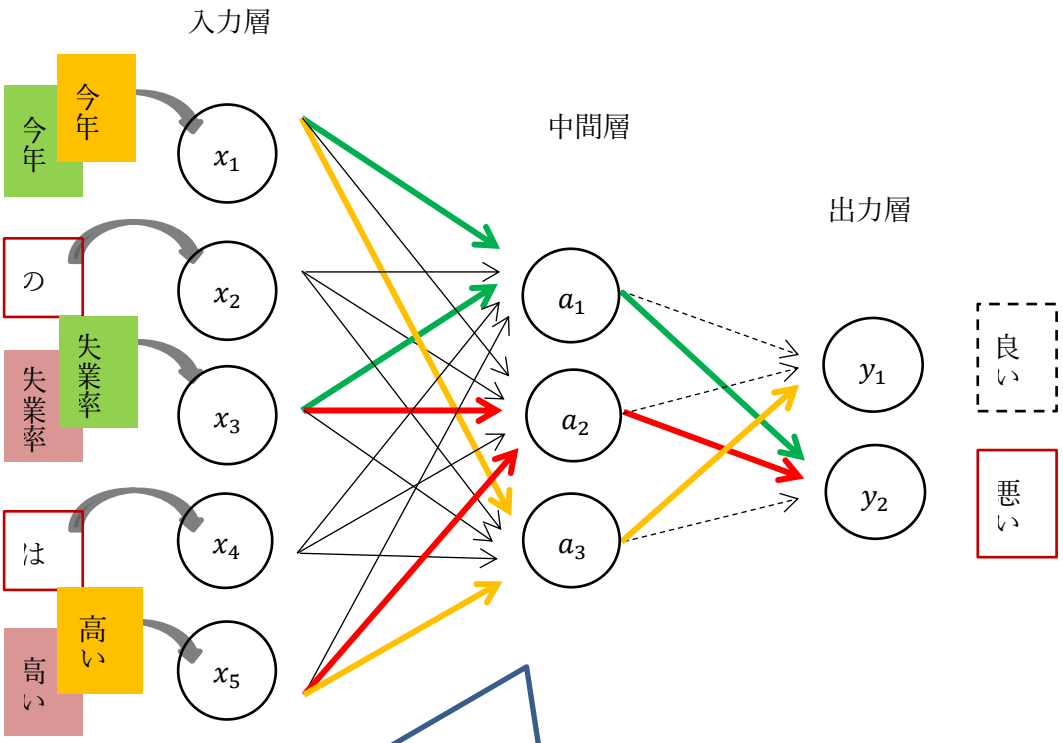


- 実線：（入）出力あり
- 破線：（入）出力なし
- 太線：重視

この例では，ポジティブでもネガティブでもない単語は受け入れて，ポジティブな名詞は受け入れず，ネガティブな名詞は受け入れるようにネットワークを構築する

1. 中間層の a_1 は，入力層の x_1 と x_3 を重視している。しかし，x_3 にポジティブな名詞“経済成長率”があるため，a_1 は x_3 からの出力を受け入れない。
2. 中間層の a_2 は，入力層の x_3 と x_5 を重視しているが，同様の理由により x_3 から出力を受け入れない。
3. 中間層の a_3 は，入力層の x_1 と x_5 を重視しており，どちらからも出力を受け入れている。

(2)



- 実線：（入）出力あり
- 破線：（入）出力なし
- 太線：重視

この例では，ポジティブでもネガティブでもない単語は受け入れて，ポジティブな名詞は受け入れず，ネガティブな名詞は受け入れるようにネットワークを構築する。

1. 中間層の a_1 は，入力層の x_1 と x_3 を重視しており，x_3 にネガティブな名詞“失業率”があるため，x_3 からの出力を受け入れる。

2. 中間層の a_2 は，入力層の x_3 と x_5 を重視しており，同様の理由により，x_3 からの出力を受け入れる。

3. 中間層の a_3 は，入力層の x_1 と x_5 を重視しており、どちらからも出力を受け入れている。

6-1-3. 中間ユニットの反応

以下では，中間ユニットの反応について，２つに分けて詳しく解説します。

ステップ１：中間ユニットの特徴

中間ユニットは，情報変換機能に関して個性があり，それは２つの側面によって特徴づけられます。一つ目は，情報を受ける側についてであり，中間ユニットは，全ての入力ユニットと繋がっていますが，繋がりの強さに違いがあります。二つ目は，情報を出す側についてであり，中間ユニットは，受け取る情報に対して反応の感度が異なり，また，出力の方向も異なります。そのような中間ユニットの個性は，入力される文から正しい判断にたどり着くように，データから学習されます。特に，どの入力ユニットに，どのような単語が入力されるかを考慮して，中間ユニットの個性は調整されます。

図表 6-2(1)は文①に対して正解が「良い」となるように，そして，**図表 6-2(2)**は文②に対して正解が「悪い」となるように，それぞれニューラルネットワークの構築例を表しています。入力層から中間層へ受け渡される単語の中には，条件付きで受け入れたり，そもそも受け入れる必要がなかったりするものが含まれています。これらの例では，受け入れる単語の選別について，①助詞（“の”，“は”）を受け入れない，②入力される単語がポジティブワード（“経済成長率”）なら受け入れず，ネガティブワード（“失業率”）なら受け入れる，と仮定します。さらに，入力層と中間層との繋がり方，そして，中間層と出力層との繋がり方は，中間ユニットごとに異なります。

ステップ２：中間ユニットの繋がり

中間層を通じたネットワークの繋がり方について，**図表 6-2(1)**を参照して説明します。まず，中間ユニット a_1 は入力ユニットの x_1 と x_3 から受け取る単語を重視する，と考えます。この場合，x_3 から受け取る単語が“経済成長率”というポジティブワードであるため，仮定より x_3 と a_1 の結びつきは弱いことになります。したがって，中間ユニット a_1 は，入力ユニット x_1を通じて，“今年”という情報のみ受け取ります。このとき，中間ユニット a_1 は，一つの情報しか受け取っていないため，何も反応しないように設定します。このことは，中間ユニット a_1から出力層への点線として表されています。

次に，中間ユニット a_2 は入力ユニットの x_3 と x_5 から受け取る単語を重視する，と考えます。すると，中間ユニット a_1 と同様の理由により，中間ユニット a_2 は入力ユニット x_5を通じて，“高い”という情報のみ受け取ります。やはり，この場合でも，中間ユニット a_2 は，一つの情報しか受け取っておらず，何も反応しないように設定します。このことは，中間ユニット a_2 から出力層への点線として表されています。

最後に，中間ユニット a_3 は入力ユニットの x_1 と x_5 から受け取る単語を重視する，と考えます。すると，先ほど説明した２つの中間ユニットとは異なり，中間ユニット a_3

は，“今年”と“高い”という二つの情報を受け取ります。このとき，中間ユニット a_3を通じて，評価が“良い”と判断できるため，出力ユニット y_1 へ反応するように設定します。このことは，中間ユニット a_3から出力層への黄色の実線で表されています。

6-1-4. 他の文への応用

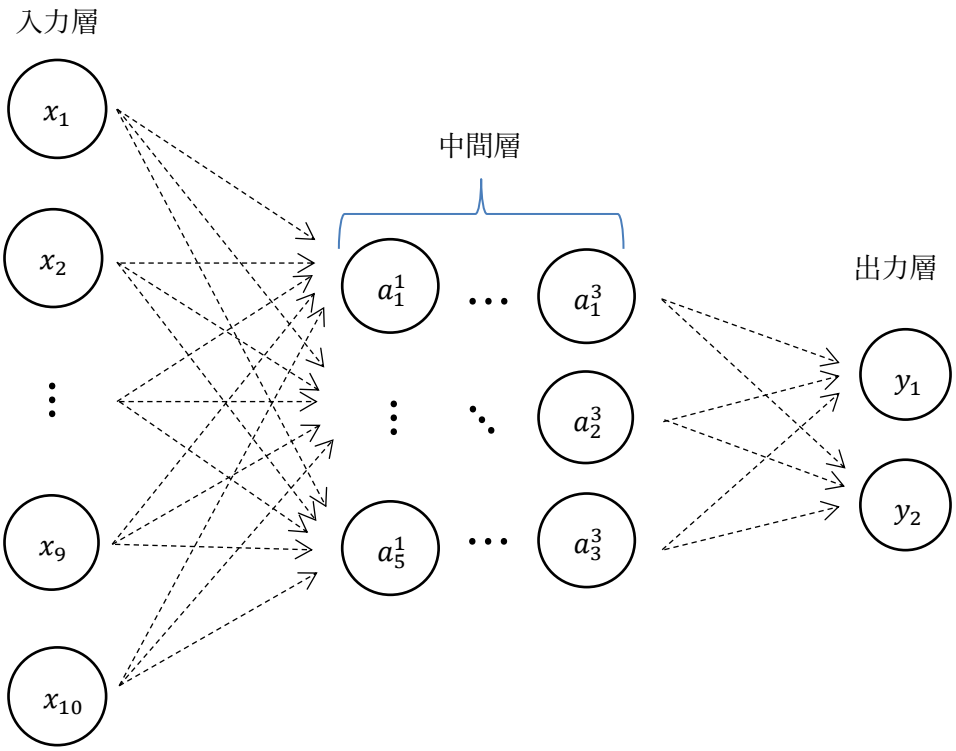
同様の考え方で，**図表 6-2(2)**についても，中間層を調整してニューラルネットワークを構築します。この例では，入力ユニット x_3 に，「失業率」というネガティブワードが入力されていることを考慮します。これにより，中間層の働きが**図表 6-2(1)**から変化します。具体的に，中間ユニットの a_1 は，“今年”と“失業率”とい二つの情報を受け取り，中間ユニットの a_2 は“失業率”と“高い”という二つの情報を受け取ります。このとき，中間ユニットの a_1 と a_2 は，評価を“悪い”と判断できるため，それぞれ出力ユニット y_2 へ反応するように設定します。それらの反応は，出力層への緑色の実線と赤色の実線で表されています。ところで，中間ユニットの a_3 は，**図表 6-2(1)**と同様の反応を見せます。つまり，“今年”と“高い”という二つの情報を受け取るため，評価は“良い”と判断して，出力ユニット y_1 へ反応するように設定します。

図表 6-2(2)での中間層の反応をまとめて，文②に対して正しい評価が導かれることを確認します。評価が“悪い”に向けて反応している中間ユニットは 2 つあり，評価が“良い”に向けて反応している中間ユニットは 1 つあります。中間ユニットごとに結果のバラつきはあるものの，多数決で考えると，中間層全体で評価は“悪い”へと導かれることが分かります。

6-1-5. ニューラルネットワークによる学習

図表 6-2(1)と**図表 6-2(2)**の例題より，ニューラルネットワーク解析において重要なことは，文と正解が結びつくように中間層の各ユニットに適切な役割を割り当てることであるということを理解できたと思います。先ほどは，中間層を会議における審査員にたとえましたが，そうした審査員の働きは，人間の頭の働きに置き換えることができます。つまり，中間層は，入力される文に対して正しく評価できるような，思考パターンを形成する場です。思考パターンが適切に形成されれば，様々な文に対して正しく評価できるようになります。以上を踏まえ，**学習**とは，「ニューラルネットワークを構成するユニットの働きを調整することで，入力データから正しい出力を得ること」をいいます。なお，ユニットの働きは，モデルにおけるパラメータの値で表されます。この説明には，若干の数式を必要とするため，詳細に関心のある読者は巻末の付録 1 をご覧ください。

図表 6-3：深層学習へ拡張したニューラルネットワーク



6-2. ディープニューラルネットワーク（DNN）

ここまでは，単純な文の入力を考えていたため，ニューラルネットワークの中間層は１層だけで済みました。しかし，長文で単語の数が多い複雑な文の入力を考えると，１層の中間層では情報を適切に変換できず，正解を導くようなネットワークの構築が困難になります。会議に例えると，出席者が多くて発言数が多いとき，１度限りの中間審査では出席者の発言を上手く集約できず，総合評価を正しく導けないかもしれません。そうした場合，中間審査の回数を増やすことで，正しく判断しようとするでしょう。このような処理をニューラルネットワークで考えると，中間層の数を増やすことに相当します。

図表 6-3 は，中間層を３つに増やしたニューラルネットワークを表しています。この例では，景気判断文が１０個の単語で構成されると想定するため，入力ユニットも１０個あります。中間層の第１層目には５つのユニット $a_i^1\ (i=1,\cdots,5)$ ，第２層目には４つのユニット $a_j^2\ (i=1,\cdots,4)$，そして第３層目には３つのユニット $a_k^3\ (k=1,\cdots,3)$があります。中間層というフィルタを３層に重ねることで，多くの情報を濾過して少量の質の高い情報を得て，正しい判断を導けるようにするのです。このように，「中間層に厚みを持たせてニューラルネットワークを解析すること」を，**深層学習**と言います。

第 7 章　文脈を考慮した深層学習

人間が書籍や新聞記事を読むとき，文脈を踏まえながら意味を理解しています。文が長くなるほど，文脈を理解することがより重要になります。第 6 章で述べたように，深層学習を利用して文章を評価する仕組みは DNN で作られていますが，そこでは文脈を考慮せずに解析されます。これは，文を構成する全ての単語が，その順番を無視して同時に分析に使われることを意味します。そうした課題への対策として，本章では，モデルをリカレントニューラルネットワーク（RNN）へと修正することによって，単語の順番を考慮できるように学習させる方法を紹介します。最後に，単語が多く構造な複雑を持つ長文の学習に関して，単語の情報処理を上手く行うために，中間層のユニットを後に説明する LSTM ユニットで置き換えた RNN について解説します。

7-1.　DNN の課題

DNN では，単語の順序を考慮した文の評価方法を，正しく学習できないことを紹介します。例えば「日本より中国の経済成長率は高い」という文による，日本経済の評価を考えます。人がこの文を読むと，「日本の経済成長率よりも，中国の経済成長率の方が高いというのは，普通のことだよね」という評価を頭に思い浮かべるとします。しかし，そのような評価方法を通常の DNN で学習させようとすると，上手くいかないかもしれません。なぜなら，DNN は単語の順番を考えず，「“日本”，“中国”，“経済成長率”，“高い”」という単語の集合に対して，「普通」という評価を学習しようとするからです[6]。

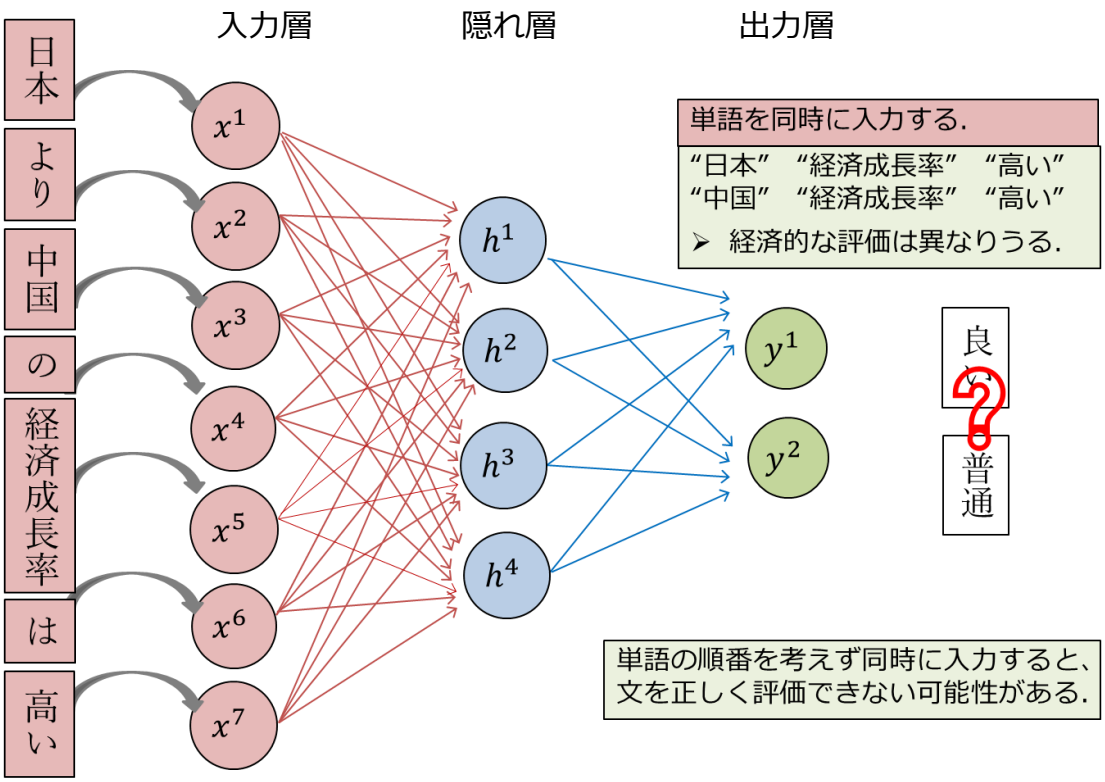
この場合，ニューラルネットワークの中のユニットは，文の構造を無視して，どの単語があるかに注目することで反応します。つまり，「“日本”，“中国”，“経済成長率”，“高い”」という集合と，“日本”と“中国”の位置を入れ替えた「“中国”，“日本”，“経済成長率”，“高い”」という集合を，同一の入力物とみなして分析します。こうした仮定の下，学習したモデルに「中国より日本の経済成長率は高い」という文を入力するとしましょう。そして，この文からは，日本経済は「良い」と評価されることが適切だと想定します。しかし，このモデルは文の構造を無視して学習しているため，予想に反して「普通」と評価しまうことになります。

図表 7-1 は，例文の評価方法を，DNN では正しく学習できないことを表したものです。**図表 6-3** で描かれた DNN の図と異なる点は，中間層ではなく**隠れ層**という言葉に置き換えて，そこに含まれる 4 つのユニットを $h^i\,(i=1,\cdots,4)$ という記号で表していることです。中間層は学習モデルの中に隠れており，分析者から見えない事と次節の RNN の説明（**図表 7-2**）と整合させるために，以下では，隠れ層と表現します。

---

[6] このように文中に登場する順番を無視した単語の集合は，専門用語で Bag of Words (BoW)といい，ベクトルで表現される。詳細に関心のある読者は，巻末付録 1 を参照すること。

図表 7-1：通常の DNN の課題



DNN は，単語の順序を考慮せずに学習するため，主語が何であるか区別していません。しかし，実際には，主語が何であるかに応じて，異なる評価を出すことが普通でしょう。主語の違いを強調するために，文を構成する単語の集合を単純化して，「“日本”，“経済成長率”，“高い”」と「“中国”，“経済成長率”，“高い”」に分けて考えます。日本経済について，前者の集合は「良い」と評価して，後者の集合は「普通」と評価できます。これとは対照的に，「“日本”，“中国”，“経済成長率”，“高い”」で構成される文については，その主語が“日本”か“中国”のどちらであろうが，「普通」と評価するように学習してしまいます。その理由は，繰り返しになりますが，DNN が単語の順番を考えずに学習してしまうからなのです。

7-2. リカレントニューラルネットワーク(RNN)

以上のように DNN の課題が分かったところで，ここからは，単語の順番を考慮して，文の評価方法を学習する仕組みについて説明します。単語の並び順を踏まえて分析するためには，DNN の背後で動くモデルの構造を，**リカレントニューラルネットワーク（Recurrent Neural Network, RNN）**へと拡張することが，有力な方法の一つです。

RNN は，通常のニューラルネットワークと異なり，**時系列データ**を扱うことができ

ます。時系列データとは，「発生した順番に並べたデータ」のことです[7]。テキストデータの分析単位を単語でなく文とする場合，単語の順序を考慮するので，そのようなテキストデータは時系列データとして分析されます。リカレント(recurrent)は，日本語で“回帰的”と訳されます[8]。RNN でテキストデータを分析することは，常に一つ手前の単語に立ち戻り，その情報を保持しながら，現在の単語の意味を考えることになります。したがって，文を読み終わるころには，それまでに登場した全ての単語の順番，すなわち文脈を考慮していることになるのです。

図表 7-2 は，RNN による学習のイメージを，５つに分けて表したものです。例文を構成する７つの単語を順番にネットワークへ入力することによって，文に対する評価方法を学習させます。RNN による学習では，各単語が入力されるたびに，隠れ層の全てのユニットがどのように反応するか調整します。**図表 7-2(1)**では，隠れ層の全てのユニット($h^1_{t=1},\cdots,h^4_{t=1}$) が，入力層のユニット ($x_{t=1}$) より，１番目の単語である“日本”のみを受け取る状況を表しています[9]。

次に，**図表 7-2(2)**では，隠れ層の全てのユニットが，１番目の単語に対する反応の仕方を記憶したまま，２番目の単語である“より”のみを受け取る状況を表しています。隠れ層のユニット自体は何も変化していませんが，２番目の単語に対する反応 ($h^1_{t=2},\cdots,h^4_{t=2}$) は，１番目の単語に対する反応 ($h^1_{t=1},\cdots,h^4_{t=1}$) から区別しています。隠れ層のユニットは，１番目の単語に対する反応の仕方を，図中の矢印の通りに受け継いだ下で，２番目の単語に対する反応の仕方を決めるのです。以上のロジックは，**図表 7-2(3)**と**図表 7-2(4)**で表すように，３番目と４番目の単語についても同様に適用されます。こうした作業は，それ以降の単語についても繰り返されます。

最後に，**図表 7-2(5)**では，７番目の単語に対して，隠れ層のユニットが反応の仕方を学習する様子を描いています。結果的に，単語の系列に対して「普通」という評価を学習するに至ります。RNN による学習では，単語を一つ入力するたびに，それ以前に入

---

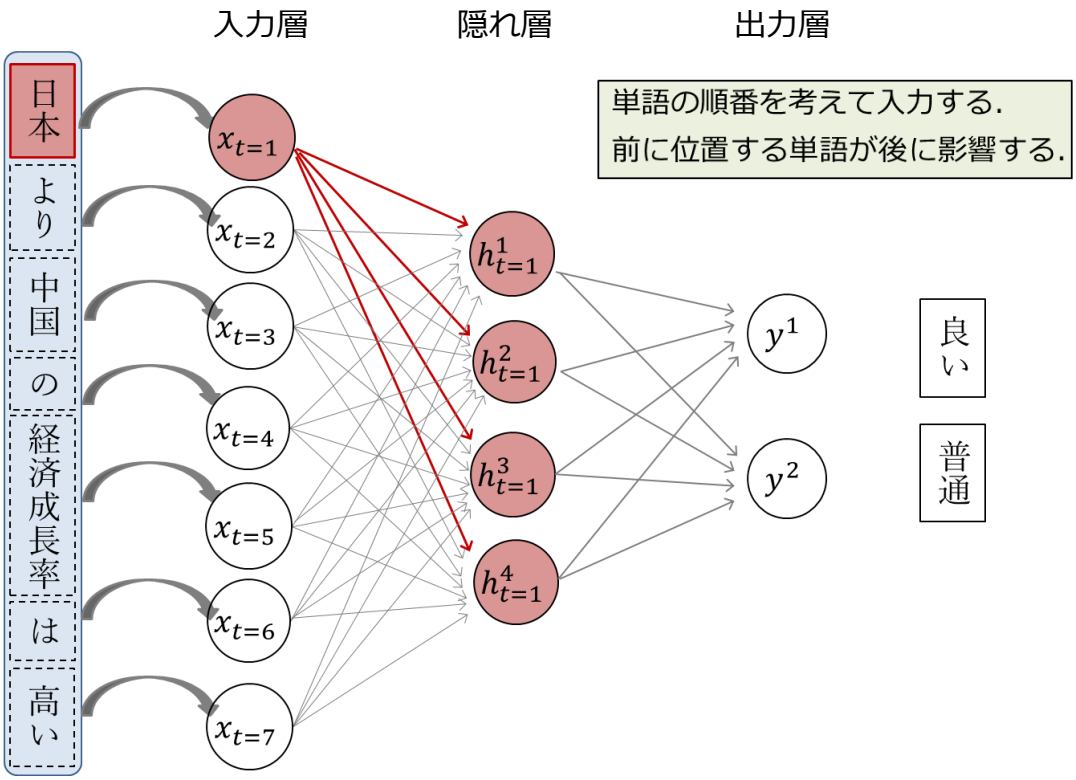
[7] 時系列データの例には，鉱工業生産，物価，GDP のように，月次，四半期，年次といった期間を通じて変化するものがある。GDP の数値は，2019 年と 2020 年の間で入れ替えることはできないため，並び順に意味がある。

[8]リカレント＝回帰的とは，“手前に戻って”という意味である。つまり該当する単語の意味合いを，文中においてそれより手前にある単語に戻って理解していくという意味である。ところで，教科書によってはリカレントを再帰的と訳すこともある。しかし，recursive neural network というモデルがあり，recursive を再帰的と訳すことがある。このため，本稿では，recurrent を recursive と区別するため，回帰的と訳すことにする。

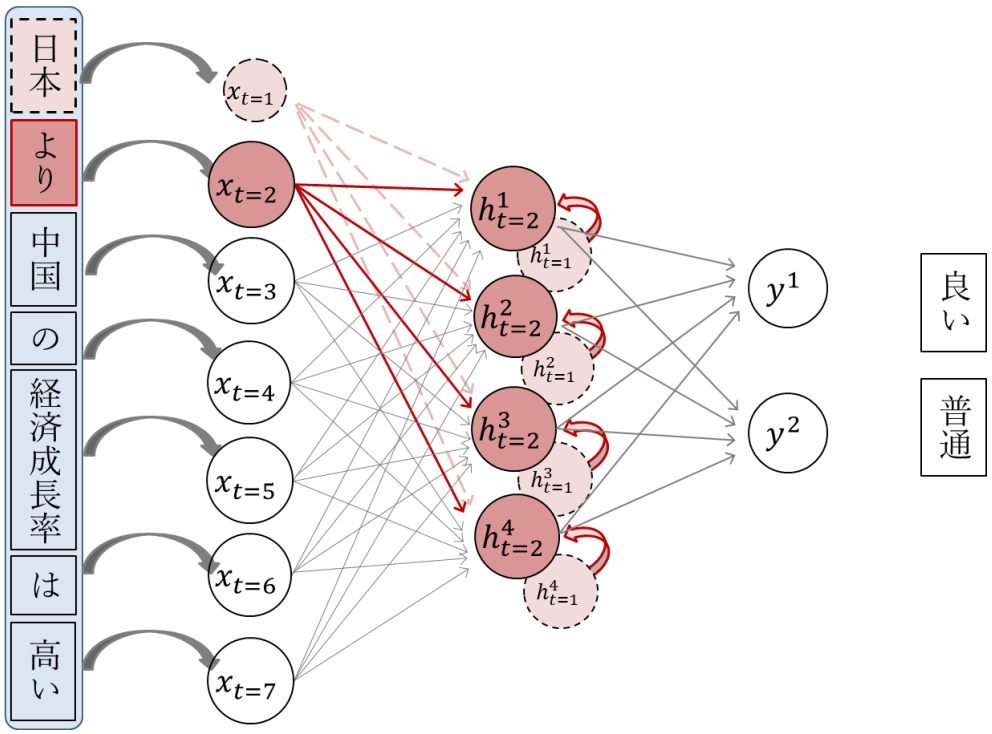
[9] ユニットを表す記号に含まれた下付き添え字 t は，timing の頭文字であり，入力される単語の順番を表している。つまり，$t=1$ は一つ目の単語，$t=2$ は二つ目の単語についてのユニットの反応を表す。

図表 7-2：RNN による学習

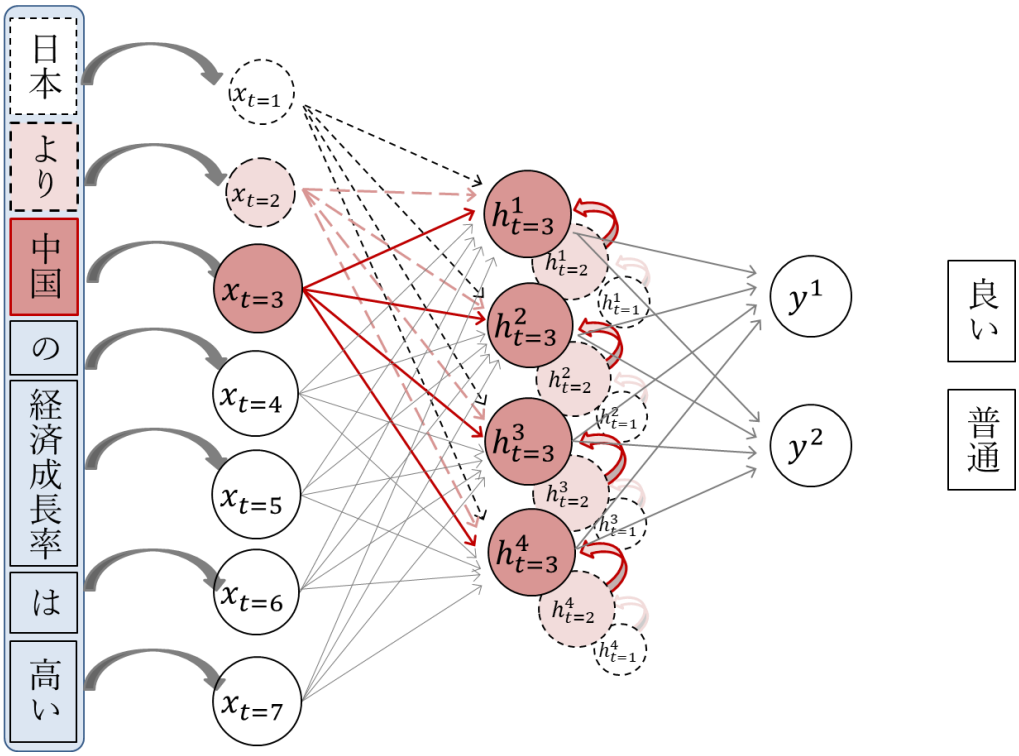
(1)



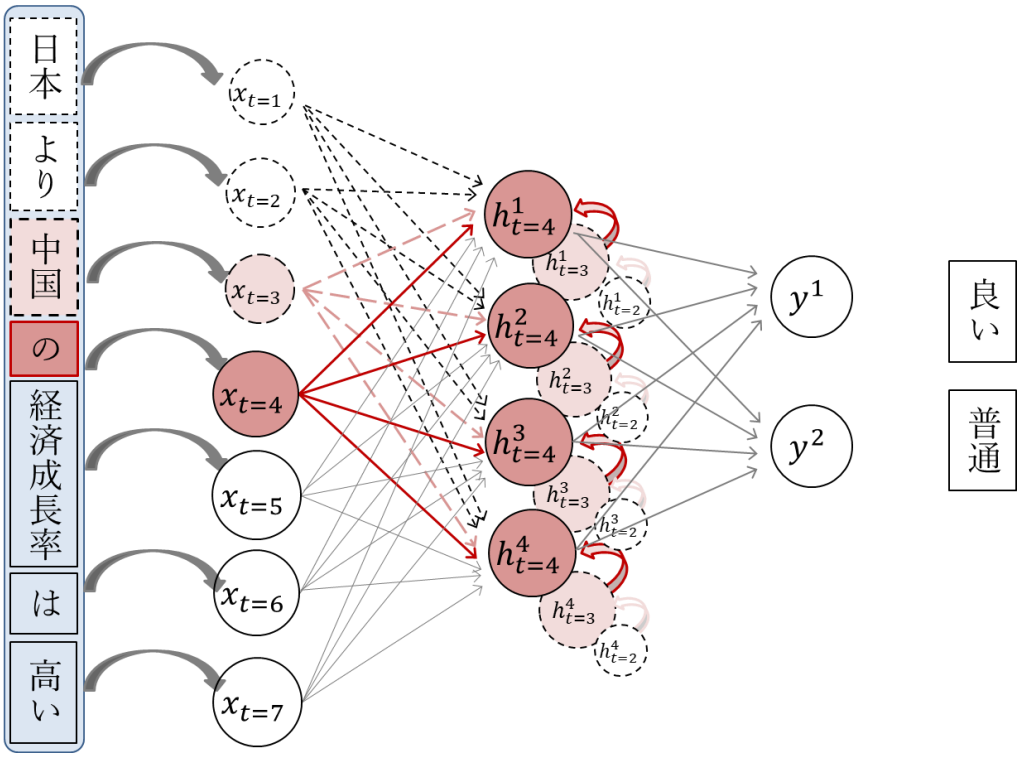
(2)



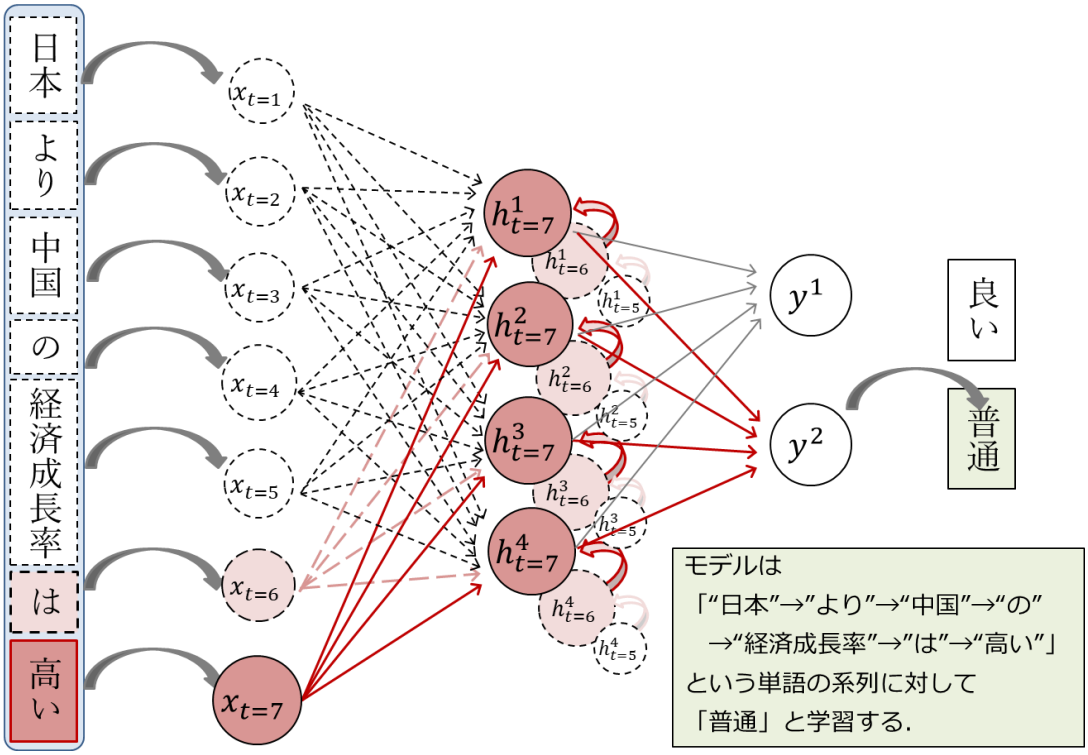
(3)



(4)



(5)



力された全ての単語に対する反応の仕方が，現時点へ引き継がれます[10]。この仕組みは，人が文を読むときに，登場した全ての単語を知っている，という認知活動に相当します。

ここまでは，RNN の長所に注目することで，文脈を考慮した文章評価の学習について解説しました。RNN による学習が上手くいくかどうかは，DNN による学習と同様に，情報を変換する役割を隠れ層のユニットへ適切に配分できるかに依存します。ところで，単語は数字に置き換えてコンピュータで処理することができます。図では単語を入力していますが，実際の処理では数字のベクトルとして入力されます（巻末付録 1 を参照）。したがって，モデルの出力値も数字で表されます。入力と出力が数字で繋がっているので，その関係の度合いをパラメータとよび，このパラメータの値を得ることを推定とよびます。つまり，「単語の入出力の関係を司るパラメータを推定する」ということです[11]。パラメータの推定は学習において非常に重要なものですが，実は普通の

[10] 隠れ層のユニットの記号は，ユニットのどのような番号 $j \in \{1,\cdots,4\}$ についても，一般的に $h_t^j = h_{t-1}^j$ と書くことができ，これは単語のあらゆる順番 $t \in \{1,\cdots,7\}$ に対して成立する。

[11] 本稿では，具体的な推定方法等について説明を省略する。関心のある方は，神嶌編(2015)，巣籠(2017)や坪井ほか(2017)を参照していただきたい。RNN では，長さ T の単語の系列 $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_T})$ が与えられた場合，t 番目の単語を受けた隠れ層の第 ℓ 層目のユニットの状態

RNN では, パラメータを上手く推定できない可能性があります。この原因は, RNN が, 文脈を考慮できるのですが, 文の始めの方に登場した単語については, その影響が徐々に減衰するという性質に基づきます。こうした課題を解決するために, 次の節では, 隠れ層のユニットを LSTM という機能によって補うことを解説します。

7-3. LSTM

RNN の課題を技術的に説明すると,「長期の依存関係を上手く学習できない」ということです。言い換えると, 学習する文が長い場合, 文を読み進めるほど, 始めの方で何が言われていたか忘れてしまい, 文全体としての意味を正しく理解できないということです。RNN は, 長期記憶が苦手で, 短期記憶だけを学習してしまう傾向があるのです。これは, 例えば試験の長文読解を苦手とするような, 受験生の思考にたとえることができるでしょう。こうした課題を解決するために, RNN のモデルに**ゲート**という要素を導入します。ゲートは, 隠れ層のユニットを通過する信号の動きを司り, 関所のように機能します。ゲートにより, 長期と短期の記憶をバランスよく学習させるように, RNN を改良できます。

RNN の改良は, ゲートを追加するだけでありません。RNN に限らず, 学習を完了させるためには, モデルの出力結果と予想される正解との誤差を, できる限り小さくすることを目標とします。学習過程において, 誤差の大きさを適切にとらえることが必要です。RNN の場合, 学習する文が長いほど長期記憶が苦手になるという特徴がありますが, これは技術的に, 文の前方に登場した単語の学習に関して, 残すべき誤差が消滅してしまうということです。この課題を解決するためには, **CEC** (Constant Error Carousel)という要素を追加します。CEC は, 文中の過去, すなわち文の前方に登場した単語に関して, 誤差を残す役割を果たします。

普通の RNN では学習のパラメータを上手く推定できないという問題に対して, 上記で述べたゲートと CEC を追加することで対処します。これら二つの要素をまとめて, **LSTM ブロック**と言います。LSTM とは, Long Short-Term Memory の頭文字であり, 長短期記憶と訳されます。ところで, ゲートは役割別に種類が分かれており, LSTM の場合, **入力ゲート, 忘却ゲート, 出力ゲート**の 3 種類を使います。入力ゲートは, ユニットへと信号を入力させます。忘却ゲートは, 入力した信号を消去させます。そして,

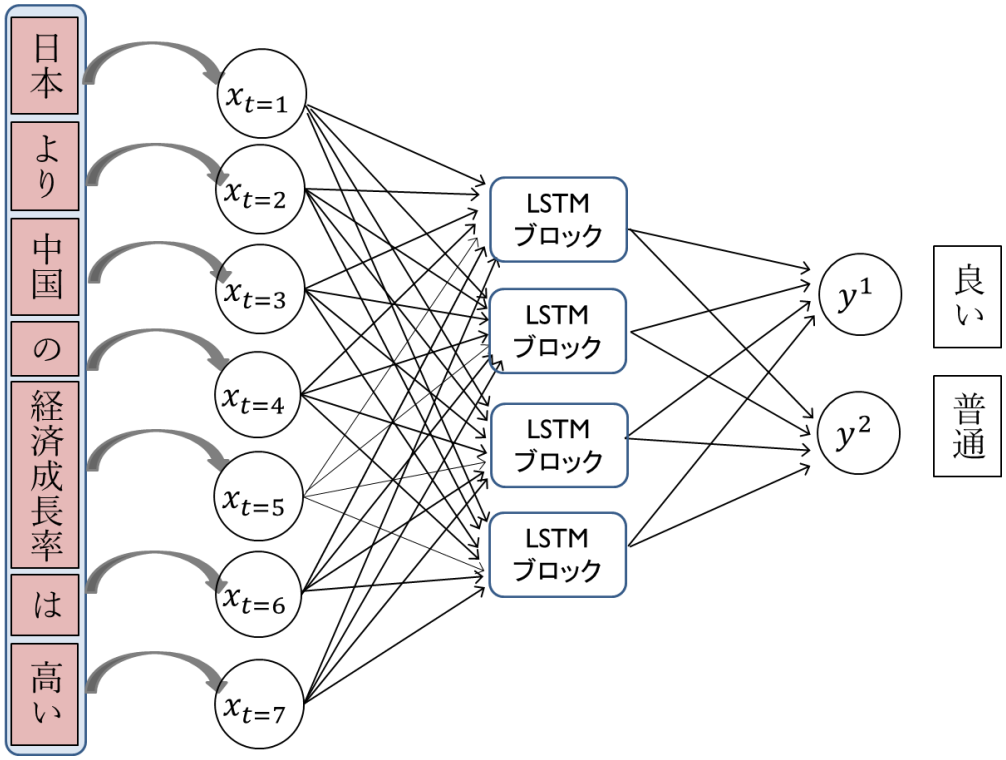
---

ベクトル $\boldsymbol{h}_t^{(\ell)}$ を, 以下のように再帰的に更新する。

$$\boldsymbol{h}_t^{(\ell)} = \boldsymbol{a}^{(\ell)}\left(\boldsymbol{W}^{(\ell)}\begin{bmatrix}\boldsymbol{h}_t^{(\ell-1)}\\ \boldsymbol{h}_{t-1}^{\ell}\end{bmatrix} + \boldsymbol{b}^{(\ell)}\right)$$

なお, 巻末付録 2 で説明している通り $\boldsymbol{a}^{(\ell)}$ は活性化関数である。そして $\boldsymbol{W}^{(\ell)}$ は重み行列で, $\boldsymbol{b}^{(\ell)}$ はバイアス・ベクトルという, それぞれ推定すべきパラメータである。

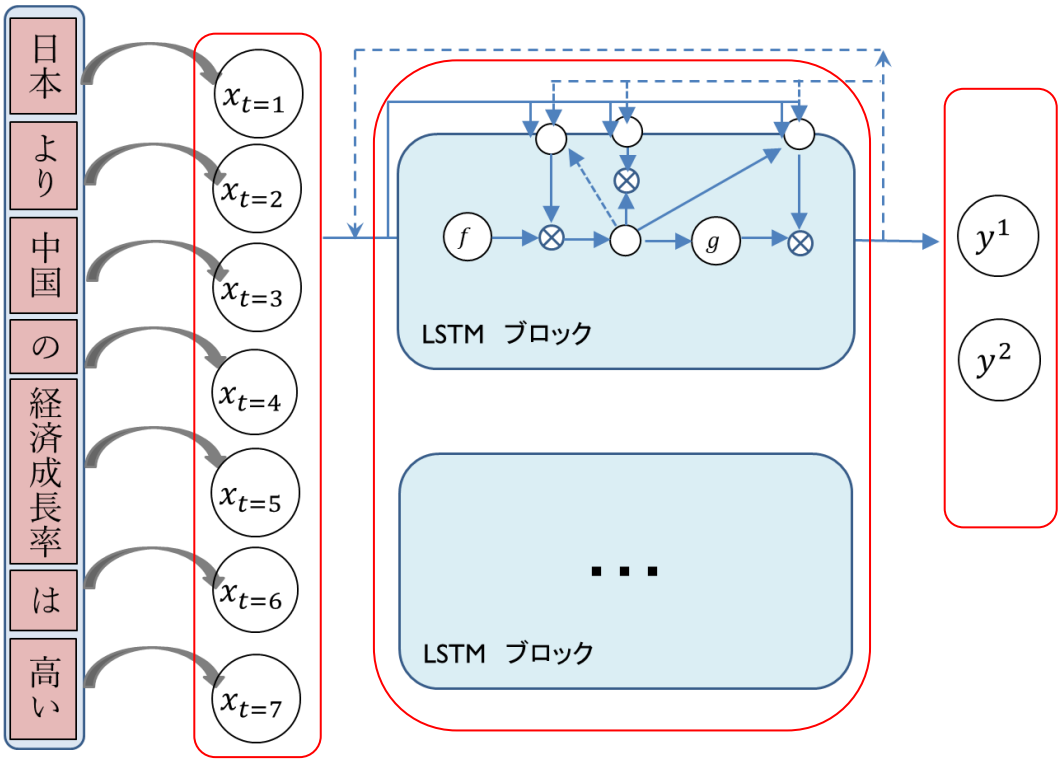
図表 7-3：LSTM-RNN のイメージ



出力ゲートは，ユニットから信号を出力させます。最終的に，隠れ層のユニットが LSTM ブロックに置き換えられて，RNN の改良は完了します。この改良版を，**LSTM-RNN** と呼ぶことにします。

LSTM-RNN のイメージは，**図表 7-3** のように，隠れ層のユニットが LSTM ブロックに置き換えられたもので表されます。そして，**図表 7-4** は，LSTM ブロックを拡大して，その要素であるゲートと CEC の働きを見えるようにしたものです。すでに述べたように，LSTM ブロックは，長文に対する学習のために，ユニットを経由する信号を取捨選択する役割を持ちます。実線は，入力層から出力層へ前向きに流れる信号を表します。点線は，後方へ戻される信号を表します。このような信号の動きは，アルゴリズムを通じて，分析者によって調整されます。その結果，どれだけの長さがあり，どのような単語の系列を持つ文に対して，どのような評価をするかということを，学習できるようになるのです。

図表 7-4：LSTM ブロックの構造



第 8 章 景況感指数の足元予測

本章では，第 7 章で紹介したリカレントニューラルネットワーク（RNN）を応用し，新聞記事を利用して景況感指数の足元予測を行う試みについて，予測結果も交えて紹介します。これは，従来ビジネスエコノミストが新聞や経済レポートから得た情報を自らの経験や知識を基に分析し，独自の景況感を形成してきた過程を機械学習によって自動化する試みとも言えるでしょう。

本章は 3 節から構成されています。第 1 節では，任意のテキストを入力として景況感を予測するために考慮すべき点や，詳細な分析手順について順番に説明していきます。そして，第 2 節で実際に 6 年分の日本経済新聞を入力として景況感指数を予測した結果を，既存の景況感指数と比較しながら紹介します。第 3 節でまとめと今後の展望について説明します。

8-1. 予測の方法

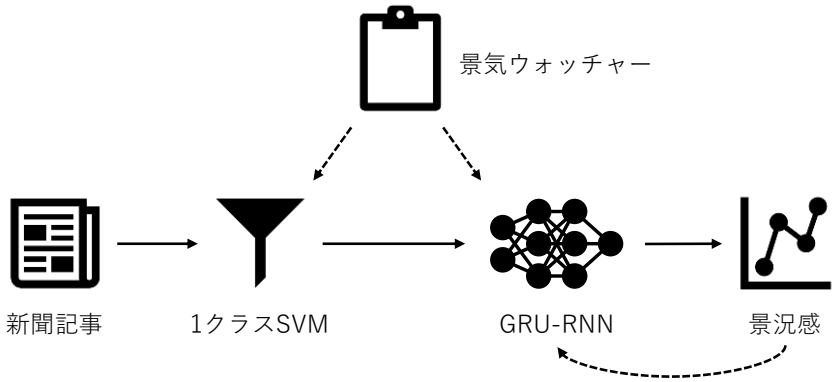
本節では，センチメント分析などに使われる RNN を応用して，新聞記事など任意のテキストから景況感を予測する方法について説明します。この予測の枠組みは**図表 8-1** の通りです。それでは，枠組みの中心となる（1）テキストから景気判断を予測する方法（GRU-RNN）から，(2) 経済・景気に関連しない文を除外する方法（1 クラス SVM），(3) 予測モデルを分析対象データに適合させるファインチューニングまで，順番に詳細を見ていきましょう。

(1) テキストから景気判断を予測する

テキストから景況感を予測する問題は，目的変数を景況感，説明変数をテキストが持つ情報とした回帰問題だと考えることができます。従来，このような問題では各単語あるいは n 語の連続する単語の組み合わせ（***n* グラム**）を独立した一つの説明変数としてモデルを学習することが一般的に行われてきました。つまり，本来順序に文法的な意味を持つテキストが，順序を無視した単語の集合（Bag of Words; BoW）として表現されてきたのです。しかしながら，本稿の上巻（生田ほか, 2020）で例示したように，独立した単語に注目していては，例えば「購買意欲が高いとは思えない」のような否定表現を含む文を適切に処理することはできません。また，5 章で述べたように，「GDP 成長率が高い」と「失業率が高い」のように，同じ「高い」という語でもその主語が何であるかによってテキストの景気に対する意味合いは全く変わってきます。テキストの意味をより適切に表現するためには，単語間の係り受け関係や文脈を考慮して文全体として意味を把握する必要があります。

近年，このようなテキストの文脈を考慮したモデル化の方法として，RNN がよく用いられています。ただし，第 7 章で説明したように，RNN は文中での距離が遠い単語

図表 8-1：新聞記事に基づく景況感指数予測の枠組み



（出所）筆者作成

図中の破線の矢印はモデルの学習時のデータあるいは処理の流れを示しており，実線の矢印は景況感指数を予測する際のデータあるいは処理の流れを示している。

間の依存関係をうまく学習できないという欠点があるため，RNN の隠れ層を構成するユニットとして，入力信号（ここでは単語）の依存関係を扱う特別な機構を持つ LSTM 等を用いるのが一般的です。本稿でも，LSTM と類似の機構を持つ **GRU**（Gated Recurrent Unit）（Cho et al., 2014）を用いて RNN を構成します。GRU と LSTM のどちらが効果的かはタスクやデータによるため一概には言えませんが，GRU はユニットの内部状態と出力を区別しないため，LSTM よりもモデルの大きさが小さくなるというメリットがあります。

モデルの学習データとしては，山本ほか（2016）にならって景気ウォッチャー調査のデータを利用します。具体的には，景気判断理由集を入力テキスト（説明変数），景気判断の結果を予測すべき目的変数とします。本稿の上巻（生田ほか, 2020）で紹介したように，景気ウォッチャー調査の景気判断は 5 つの記号（◎：良，○：やや良，□：不変，▲：やや悪，×：悪）で表現されているため，これらをそれぞれ 2, 1, 0, −1, −2 に置き換えて間隔尺度として扱います。

ところで，RNN には入力データを時系列の前方からモデルに入力する方法，後方から入力する方法，両方を組み合わせた方法があります。ここでは，話を簡単にするため前方から入力する方法を採用します。つまり，入力テキストを先頭の単語から順番に一単語ずつモデルに入力します。なお，本稿で扱う内容の範囲を越えますが，最近では **BERT**（Bidirectional Encoder Representations from Transformers）（Delvin et al., 2019）という言語表現モデルやその派生モデルを利用した方法がさかんに研究されており，センチメント分析を含め様々な自然言語処理タスクで RNN よりも高い性能が得られることが知られています。

(2) 経済・景気に関連しない文を除外する

前項のように景気ウォッチャー調査から回帰モデルを学習すれば, 形式的には任意の文章を入力として景気判断を予測することができます。しかし，景気に無関係な文章を基に景気判断を行っても実質的に意味がありませんし, むしろ景気に無関係な文章から不正確な景気判断が行われれば, 最終的な景況感の予測に悪影響を及ぼす可能性があります。実際に，本章では新聞記事を入力として景況感指数を予測することを想定しているため，入力される文章は経済に関係するとは限りません。たとえば，スポーツや芸能関係の記事は経済に無関係なことが多く, それらの文章から景気判断を行うことは避けるべきでしょう。経済や景気に関係する文章だけを予測に用いれば良さそうですが，どのようにすればそのような文章を見つけることができるでしょうか。

一つの方法は，経済や景気に関係する文章（以降「**景気関連文**」と呼びます）の特徴を学習し，そのような特徴を持たない文章を除外することです。これは，通常のデータとは異なるデータを検出する**異常値検出**の問題と考えることができます。異常値検出には様々なモデルが存在しますが，ここでは **1 クラス SVM** (One-Class Support Vector Machine) (Manevitz and Yousef, 2002)を利用することにします。通常の SVM (付録 1.2 参照) では 2 クラス分類問題を扱うのに対して，1 クラス SVM では 1 クラスのデータだけを学習データとして分類器を学習し, 学習データと類似していないデータを異常値として検出することができます。学習データとしては景気関連文を与えれば良いので, 景気ウォッチャー調査の景気判断理由集を利用することにしましょう。このように 1 クラス SVM を学習すれば，景気判断理由集と類似していない文を異常値として検出・除外することができるようになります。

なお，景気判断理由集を 1 クラス SVM に入力するためには，文章を特徴量に変換する必要があります。ここでは，形態素解析によって文章を単語に分割して BoW で表現します。また，**tf-idf** (term frequency-inverted document frequency) (Manning et al., 2008)と呼ばれる方法で，各特徴量（各単語）にそれぞれの重要度に応じた重み付けを行います。前述の通り BoW は単語の順序や依存関係を無視してしまうという欠点がありますが，景気関連文か否かを判定するという比較的単純なタスクにおいては，どのような単語がどの程度使われているかという頻度情報で十分だと考えられます。

(3) 予測モデルを分析対象データに適合させる

RNN の学習に用いる景気ウォッチャー調査の景気判断理由集と，景況感指数の予測に用いる新聞記事とでは, 文章中で使われる語彙や表現, 文体が異なると考えられます。このような語彙や文体を総称して**ドメイン**と呼ぶことにします。学習に用いるデータのドメインと予測に用いるデータのドメインの違いは, 一般的に機械学習モデルの予測性能に悪影響を与えます。よって，景気判断理由集で学習したモデルを予測に用いる新聞記事のドメインに適応させることができれば, 景況感指数のより正確な予測が期待でき

そうです。これをモデルの**ドメイン適応**と言います。

ドメイン適応の簡単な方法は，学習済みのモデルを実際に予測に用いるデータで学習し直すことです。この再学習によってモデルのパラメータが予測に用いるデータによりふさわしい値になることが期待できます。ただし，モデルの再学習には，景気ウォッチャー調査の景気判断理由集と景気判断のように，ラベル付きの学習データが必要になります。本稿では景況感の予測に新聞記事を用いることを考えていますので，理想的には各記事に－2～2 の景気判断が付与されているデータがあれば良いのですが，残念ながらそのようなデータは存在しません。そこで，予測に用いる新聞記事から新たな学習データを自動的に生成することを試みます。この手順を以下に示します。

1. 1 クラス SVM を用いて異常値の文を検出・除去する。
2. 景気ウォッチャー調査から学習した初期的なモデルを用いて，手順 1 で得られた新聞記事の各文の景況判断を予測する。
3. 手順 2 で得られた予測値の絶対値が大きい文は景気動向をよく表していると仮定して，高スコアの文と低スコアの（予測値が負で絶対が大きい）文を再学習用の学習データとする。

このように，初期的なモデルでデータを分析し，得られた結果（の一部）を学習データとしてより良いモデルを学習する方法は，**ブートストラップ法**と呼ばれます。

8-2. 予測の結果

(1) モデルの学習

まず，RNN モデルを学習するための学習データが必要になります。本稿では，内閣府のホームページ[12]から 2000 年～2018 年 9 月分の景気ウォッチャー調査の景気判断（現状）計 216,741 件をダウンロードして利用しました。このうち，無作為に選んだ 90%を学習データ，10%をテストデータとしました。景気判断理由集は形態素解析器 MeCab で単語（正確には形態素）に分割し，モデルへの入力とします。なお，RNN の学習の際にはモデルの大きさなどのパラメータをあらかじめ設定する必要があります。ここでは，学習データを使っていくつかのパラメータ値の組み合わせを試行した結果から，RNN の隠れ層の数を 2，各隠れ層ごとの GRU ユニットの数を 512，語彙数を 40,000 単語としました。また，入力の単語は Wikipedia の記事であらかじめ学習された 300 次元の**単語埋め込み**（word embedding）(Bojanowski et al., 2017）を用いて表現しました。

景況感指数の予測には，日経新聞 2013～2018 年度版の本紙朝刊と本紙夕刊に掲載された記事の見出しと本文を用いました。各記事は句点「。」に基づいて文分割し，記事単位ではなく文単位で景気判断の予測を行いました。そして，最終的な景況感指数の

[12] 内閣府景気ウォッチャー調査（https://www5.cao.go.jp/keizai3/watcher/watcher_menu.html）

図表 8-2：景気ウォッチャー調査の景気判断を予測した結果

モデル	平均 2 乗誤差（MSE）
リッジ回帰	0.509
RNN	0.351

（出所）関ほか（2020）から筆者作成

予測値は，文単位で予測した景気判断の予測値の平均値としました。本稿で利用した新聞記事は日刊のため一日ごとの集計（景況感指数の予測）も可能ですが，後述の比較対象が月次の指数のため，景況感指数の予測も月次で集計することにします。なお，モデルのドメイン適応に用いる学習データの生成には，景況感指数の予測に使うデータと重複しないように，日経新聞 2010 年度版を用いました。

(2) 景況感予測モデルの検証

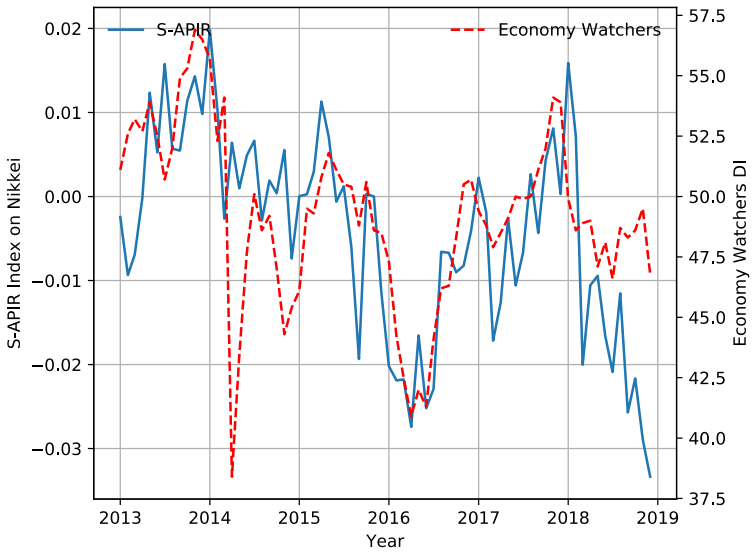
前項のように学習したモデルで，どの程度正確に景気判断の予測ができるのでしょうか。この疑問に答えるため，テストデータとして用意しておいた景気ウォッチャー調査の景気判断理由集を入力として，それぞれに対応する景気判断を予測しました。そして，予測された景気判断と実際の景気判断（−2〜2）との**平均二乗誤差**（mean squared error; MSE）を算出しました。この値が小さいほど，より正確に景気判断を予測できていることになります。結果を**図表 8-2** に示します。比較のため，入力テキストを tf-idf で重み付けした BoW で表現し，リッジ回帰で指数の予測を行った結果も表に示しています。前述のように，BoW は本稿で用いた RNN と異なり，語順が考慮されないという欠点があります。リッジ回帰と比較すると RNN では誤差が大きく減少し，指数の予測がより正確に行えていることが確認できます。なお，この結果は類似のモデル（両方向の LSTM-RNN）を利用した山本ほか（2016）の報告とほぼ一致しています。

(3) 既存の景況感指数との比較

景気判断予測モデルの性能が確認できたところで，つづいて月次の景況感指数を算出し，既存の景況感指数と比較してみましょう。ただし，景況感に正解は無いため，ここでの比較の目的は，(a) 新聞記事を基に算出した景況感指数が既存の景況感指数とどの程度類似したトレンドを持っているのか観察すること，また(b)どのような局面で既存の指数とは異なる動きをするのかといった特徴を探ることにあります。なお，以降ではテキスト情報を基に算出した景況感指数を「S-APIR 指数」と呼ぶことにします。

前述の日経新聞 2013〜2018 年度版の記事を基に，S-APIR 指数を算出した結果を**図表 8-3** に示します。比較対象として，内閣府発表の景気ウォッチャー調査から，現況判断 DI（季節調整値）も示します（以降では単に「景気ウォッチャーDI」と呼びます）。

図表 8-3：S-APIR 指数と景気ウォッチャーDI との比較



（出所）関ほか（2020）

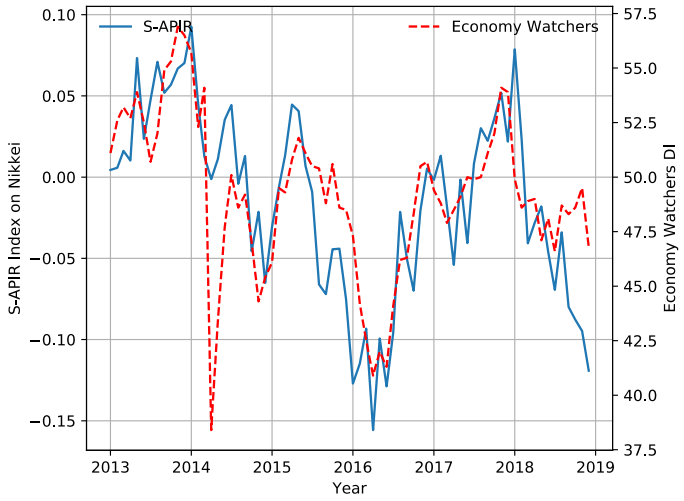
なお，ここでは景気に関連しない文を除外するための 1 クラス SVM の利用，およびモデルのドメイン適応はまだ行っていません。それでも，おおまかには S-APIR 指数は景気ウォッチャーDI のトレンドに近い動きを示していることが観察できます。実際に両者の相関係数は 0.546 であり，正の相関が見られました。

続いて，1 クラス SVM で異常値検出を行い，景気ウォッチャー調査の景気判断理由に似ていない文を除外した上で，S-APIR 指数をあらためて算出しました。グラフは省略しますが，全体的に S-APIR と景気ウォッチャーDI がさらに類似の動向を示すようになり，両者の相関係数は 0.546 から 0.686 まで上昇しました。なお，この結果からは景気に関連しない文がどれだけ正確に除去できたかは明らかではありませんが，景況感の推定を主目的とする景気ウォッチャー調査により近い結果が得られたことから，より適切な景況感の予測に効果があったと言えそうです。

(4) ドメイン適合の効果の検証

前項で，S-APIR 指数と景気ウォッチャーDI が正の相関を持つことが分かりました。しかし，景気判断理由集と新聞記事では文章の特徴（ドメイン）が異なると考えられるため，前者で学習したモデルが後者にどれほど適切なモデルであったのかは分かりません。そこで，8-1 節で述べた手順にしたがって学習済モデルのドメイン適合を行いました。ドメイン適合で用いる再学習用のデータを生成するための新聞記事としては，前述のように日経新聞 2010 年版を用いました。また，景気判断予測値の閾値については，予測値の分布を目視で確認し，スコアが－1.0 以下の文に教師信号－2，0.8 以上に教師

図表 8-4：ドメイン適応後の S-APIR 指数と景気ウォッチャーDI との比較



（出所）関ほか（2020）

信号 2 を与えて学習データとしました。この結果，スコアが高い文（教師信号=2）が 18,947 文，スコアが低い文（教師信号=−2）が 10,868 文，学習データとして獲得できました。続いて，このデータを学習データとして景気判断モデルの再学習を行いました。そして，再学習したモデルを用いて再度 S-APIR 指数の算出を行ったところ，景気ウォッチャーDI との相関係数が 0.686 から 0.706 に微増しました。この結果を**図表 8-4** に示します。

続いて，ドメイン適合によって景気判断モデルがどのように変化したのかをモデルの出力から見ていきましょう。そのため，再学習前のモデルと再学習後のモデルそれぞれに対して，単語のみをモデルの入力として与え，その単語について景気判断予測を行います。これによって，単語レベルでの景気判断モデルの変化を観察することができます。**図表 8-5** に，それぞれのモデルで予測値が大きかった 10 単語を上位から順に示します。上向きの矢印「↑」が付してある語はドメイン適合の前後で順位が上昇したもの，下向きの矢印「↓」は順位が下降したものを示します。「好調」，「享受」，「復調」などの語の順位が上昇する一方，「最高」，「絶好調」などの順位は降下していることが分かります。同様に，**図表 8-6** で予測値が小さかった 10 単語を比較します。「悪化」，「激減」などの順位が上昇する一方，「舗装」，「壊滅」の順位が下降しています。モデルの再学習の結果，両者とも新聞記事で使われるような比較的固い語彙が表の中でより上位に，アンケート回答で使われるような比較的カジュアルな語彙がより下位に移動している傾向が見てとれます。

なお，ここではモデルのドメイン適合の効果を調べるために単語ごとの指数を予測しましたが，このようにして得られた語と景気判断予測値の組は，単語とその極性値（感

図表 8-5：単語のみで予測した景況感指数（上位 10 件）

再学習前		再学習後	
快調	0.738	好調（↑）	1.591
絶好調	0.688	享受（↑）	1.503
好調	0.671	復調（↑）	1.502
最高	0.659	好転（↑）	1.488
上々	0.654	向上（↑）	1.485
好転	0.654	最高（↓）	1.484
伸	0.632	刺激（↑）	1.469
売れ れ	0.630	絶好調（↓）	1.456
上向い	0.624	伸（↓）	1.449
着実	0.617	図り（↑）	1.437

（出所）関ほか（2020）から筆者作成

図表 8-6：単語のみで予測した景況感指数（下位 10 件）

再学習前		再学習後	
舗装	−1.543	悪化（↑）	−2.029
不通	−1.441	激減（↑）	−2.028
最悪	−1.355	最悪（−）	−2.026
壊滅	−1.334	舗装（↓）	−2.026
急変	−1.303	急変（−）	−2.019
全滅	−1.285	壊滅（↓）	−2.018
追い打ち	−1.282	低落（↑）	−2.017
激減	−1.278	減益（↑）	−2.017
エスカレート	−1.257	損失（↑）	−2.013
漁船	−1.232	急落（↑）	−2.011

（出所）関ほか（2020）から筆者作成

情値）とも見なせます。つまり，この結果は経済ドメインの極性辞書としても利用することができます。一般ドメインにおいては Takamura ほか（2005）の極性辞書など利用可能な資源がいくつか存在しますが，経済ドメインかつ日本語で自由に利用できる大規模な極性辞書は著者の知る限り存在しません。よって，このようにして得られた結果は，経済分野のテキストの感情分析に用いる言語資源としても有用であると言えるでしょう。

図表 8-7：S-APIR 指数と業種別景気ウォッチャーDI の相関係数

業種	相関係数
家計動向関連業	0.600
企業動向関連業	0.816
雇用関連業	0.832
全体	0.706

（出所）筆者作成

(5) S-APIR 指数の特徴

図表 8-4 を見てみると，2014 年の前半に景気ウォッチャーDI は大きく落ち込んでおり，S-APIR 指数と大きな乖離があることが分かります。2014 年 4 月は消費税率が 8%に引き上げられた時期であり，景気ウォッチャーDI と比較して S-APIR 指数では増税の影響が小さいことは，S-APIR 指数の一つの特徴を示していると考えられます。この違いの原因はどこにあるのでしょうか。一つ考えられるのは，景気ウォッチャー調査の対象の 68.5%が家計動向関連業（小売や飲食など）の従事者であることです。消費税増税のように家計に与える影響が大きい事象は，より直接的に家計動向関連業に影響を与えると考えられますので，景気ウォッチャーDI は家計の動向のより強く反映した指標であると言えそうです。

ところで，景気ウォッチャー調査の回答者の残り約 3 割は，企業動向関連業が 21.4%，雇用関連業が 10.1%で構成されています。そして，それぞれの業種ごとの景況感指数も公表されています。そこで，S-APIR 指数がどの業種の景況感をより反映した指標であるのか，業種ごとの景気の現況判断 DI と S-APIR 指数で相関係数を計算してみましょう。結果を**図表 8-7** に示します。S-APIR 指数と雇用関連業の指数との相関係数は 0.832，企業動向関連業の指数との相関係数は 0.816 となり，家計動向関連業の指数との相関係数（0.600）よりも強い相関を示しました。この結果は，日経新聞から算出した S-APIR 指数は，より雇用の状況や企業動向を強く反映している指標であることを示しています。

8-3. まとめと今後の展望

本章では，テキストデータの経済分析への利用可能性を示す具体例として，深層学習モデルを用いて，日々発行される新聞記事を基に景況感の足元予測を行う方法を紹介しました。合わせて，景気に関係しない記事の存在や，学習と予測時のドメインの違いなど，新聞記事を景況感の予測に用いる場合に注意しなければいけない点についても解説しました。本章で紹介した方法で実際に日経新聞から景況感の予測を行った結果，既存

の景況感指標である景気ウォッチャーの現状判断 DI（季節調整値），特に雇用関連業および企業動向関連業の現状判断 DI で高い正の相関を示しました。景気ウォッチャー調査は，景気動向を判断することを目的に内閣府がコストと時間をかけて行っている全国的なアンケート調査です。一方，新聞は国内外の社会情勢一般のニュースを読者に伝えることを目的に発行されています。本稿で用いた日経新聞は経済紙ではありますが，そもそもの目的が異なる新聞記事から景気ウォッチャー調査の現状判断 DI と強い相関を持つ指標を自動的に生成できたことは，経済分野におけるテキストデータの利用可能性を強く示唆するものです。また，すでに存在する新聞記事を活用したことで，景気ウォッチャー調査のような大規模なアンケートを必要とする従来の景況感指数よりも低コスト，かつ即時性の高い指標が作成できたことにも注目すべきでしょう。

なお，今回は比較的仕組みを理解しやすい GRU-RNN を景気判断モデルに利用しましたが，前述のように，近年の自然言語処理の研究では，**自己注意機構**（self-attention mechanism）（Vaswani et al, 2017）に基づく BERT などの言語表現モデルの研究が進んでいます。APIR における研究プロジェクト「テキストデータを利用した新しい景況感指標の開発と応用」では，そのような先進的なモデルを使った景況感指数の検証も進めているところですが，その説明はまた別の機会に譲ります。また，検証のため本稿では 2013〜2018 年に発行された過去の新聞記事データベースを利用して S-APIR 指数を予測しましたが，新たに発刊された新聞記事をモデルに入力すれば，まさにその日の記事から読者が得られる景況感が，記事に目を通すことなく瞬時に得られることになります。新聞記事やニュース速報などのテキストデータを代替的に利用した経済指標は，その即時性やコスト面でのメリットから今後ますます利用が進んでいくものと予想されます。

【付録】

付録１　テキストを機械学習で扱うために

1-1. 文や単語を数学的に表現する方法

機械学習は，数値信号を処理するコンピュータによって実行されます。よって，言語処理に機械学習を応用する場合，文字情報を数値化して演算できるように加工する必要があります。どのようにすればよいのでしょうか。そのヒントは，このディスカッションペーパーの上巻で説明した，データの尺度にあります。テキストデータは定性的であり，本来は，名目尺度と順序尺度で扱うものです。つまり，言葉によって，物事は分類され，関係付けられます。しかし，このままでは，言葉を演算することはできません。そこで，テキストデータを定量的なものへ移行し，間隔尺度と比例尺度で扱えるように，数学的に表すことが必要です。

文章や文を数学的に表現するためには，文章や文を，**ベクトル**，すなわち数値の列に変換して，空間上の座標で表現する必要があります。文章をベクトルで表すとは，どういうことでしょうか。以下の例は，文章 d をベクトル $\boldsymbol{x}^{(d)}$ で表したものです。

文章 d =「東京都は東の京都ではありません。東京都は京都の東にあります。」

$$\boldsymbol{x}^{(d)} = (2, 2, 2, 2, 2, 1, 1, 1, 1)$$

この例の場合，ベクトルの要素は，各単語の頻度を表しています。文章や文をベクトルとして表現する場合，ベクトルの各要素は単語に関連する数字であり，単語の存在の有無や，単語の出現頻度を表します。そうしたベクトルの要素のことを，自然言語処理の分野では，**素性**（そせい）といいます。他方，機械学習の分野では，素性ではなく，**特徴**とよびます。本稿では，分かりやすさを優先して，ベクトルの中身を，素性や特徴ではなく，要素とよぶことにします。

文章や文をベクトルで表現することは，家の住所に例えることができます。家の住所は，郵便番号，町名，番地，部屋番号など，複数の種類による数字で表されます。家同士が近いか遠いかは，住所が近いかどうかで判断されます。これと同じ考え方を自然言語処理にも適用すると，文章や文の関係が近いかどうかは，ベクトルの向きがどの程度近いかで判断できそうです。

ベクトルの要素を得る，つまり，文に含まれる各単語の個数を知るためには，上巻で説明した形態素解析によって，まずは文を単語に区切る作業が必要です。特に，日本語は英語と異なり，文中に空白が無いため，単語が事前に区切られておらず，形態素解析が重要な処理となります。文が長ければ長いほど，多くの単語が含まれます。

形態素解析のあと，文を構成する単語のうち，その文の特徴を判断するために不要なものを削除することによって，効率的な分析が行えるようになります。不要な単語を削除することを，**前処理**といいます。例えば，“一”のような数字や，“は”のような助詞は，文の話題（トピック）を表す情報としてあまり有益でありません。また，滅多に表れない単語や，頻発する単語も，文の特徴を表すほど重要でなく，削除される場合があります。このように，分析したい話題と関連がない単語のことを**ストップワード**とよびます。前処理によりストップワードを削除することで，ベクトルの要素を減らし，計算の負荷を小さくすることができます。

二つ目の前処理は，**ステミング**と言い，語形変化を取り除いて語幹を取り出す作業です。たとえば，「eating」と「eats」は表層的には異なる語ですが，語幹はどちらも「eat」であり，一つの要素にまとめることができます。同様に，表現は異なるが意味が類似する単語（類似語または同義語）を一つの要素にまとめることがあります。例えば，“自由民主党”と“自民党“は，同じ要素と判断されるでしょう。他方で，形態素解析の結果，似たような単語であっても品詞が異なれば，それらは異なる要素と判断される場合があります。例えば，“走り［動詞］”と“走り［名詞］”は，異なる要素になりえます。

形態素解析の結果，文を 100 個の単語に区切ることが出来たとしましょう。全ての単語について頻度を求めると，ベクトルを構成する要素は，100 次元で表されます。これは高次元であり，計算に大きな負荷がかかります。そこで，余分な情報や，重複する情報を排除することによって，結果的にベクトルの次元を減らすことが求められます。前処理により，例えば，

「第一に，11 月の経済は，経済学的に堅調に推移していると判断した。」

という文は，どのように情報量を減らすことができそうでしょうか。あり得る結果は，

「11 月　経済　堅調　推移　判断　する」

です。こうすることで，元の文が持つ意味を残しつつ，要素の数を減らすことができました。本稿では，形態素解析と前処理によって，テキストデータは分析可能な状態になっていることを仮定します。文は一般的に使用される単語に区切られ，句読点のように不必要な情報が排除された状態であることを前提とします。したがって，ベクトルとして表された文の各要素には，文として判断できる必要な単語に関する情報のみが反映されていると考えます。

ここまでの話では，文の構成要素として，単語に注目していました。しかし，文には，隣り合う単語同士で意味を成す**語句（フレーズ）**も含まれています。語句を捉えたい場合，***n*-gram** といい「文の中に隣り合って出現した n 個の単語」の範囲で，文を区切っ

て見る方法があります。単語を区切る n には，関心に応じて任意の数字を入れることができます。例えば，$n=1$ はユニグラム（unigram），$n=2$ はバイグラム（bigram），$n=3$ はトライグラム（trigram）と呼ばれます。文をベクトルで表現する場合，n-gram の規模に応じて，ベクトルの要素の数は変わります。

文章は，ベクトルによってどのように表現できるでしょうか。文章 d が与えられたとき，それをベクトル $\boldsymbol{x}^{(d)}$ によって表すとします。このベクトルを構成する各次元は，文章に含まれる単語 w に対応して，その値は単語の頻度 $n(w,d)$ とします。例えば，文章 d を，「東京都は東の京都ではありません。東京都は京都の東にあります。」と考えてみましょう。この文章についてのベクトル表現は，以下の通りです。

$\boldsymbol{x}^{(d)}$	$=\Big(n\big(\text{"東京都"},d\big),n\big(\text{"東"},d\big),n\big(\text{"京都"},d\big),n\big(\text{"は"},d\big),n\big(\text{"の"},d\big),$ $n\big(\text{"では"},d\big),n\big(\text{"ありません"},d\big),n\big(\text{"に"},d\big),n\big(\text{"あります"},d\big)\Big)$
	$=(2,2,2,2,2,1,1,1,1)$

こうしたベクトル表現は，どの単語が何回出現したか知ることができるため，**頻度ベクトル**と呼ばれます。ただし，単語が文章中のどこに出現したのか知ることが出来ません。このような頻度ベクトルによる表現を，文章における単語の順番や文法的構造を無視しているため，**bag-of-words** といいます。この名称は，袋の中に単語が混ぜ合わせに入っている状態に由来します。

ベクトル表現は，文章を構成する文についても，同様の方法で表現できます。例えば，文 s_1 を「今年の景気は去年の景気よりも良い。」とし，文 s_2 を「去年の景気は今年の景気よりも良い。」と考えてみましょう。各文についての bag-of-words によるベクトル表現は，以下の通りです。

$\boldsymbol{x}_{\text{words}}^{(s_1)}$	$=\Big(n\big(\text{"今年"},s_1\big),n\big(\text{"の"},s_1\big),n\big(\text{"景気"},s_1\big),n\big(\text{"は"},s_1\big),$ $n\big(\text{"去年"},s_1\big),n\big(\text{"よりも"},s_1\big),n\big(\text{"良い"},s_1\big)\Big)$
	$=(1,2,2,1,1,1,1)$

$\boldsymbol{x}_{\text{words}}^{(s_2)}$	$=\Big(n\big(\text{"今年"},s_2\big),n\big(\text{"の"},s_2\big),n\big(\text{"景気"},s_2\big),n\big(\text{"は"},s_2\big),$ $n\big(\text{"去年"},s_2\big),n\big(\text{"よりも"},s_2\big),n\big(\text{"良い"},s_2\big)\Big)$
	$=(1,2,2,1,1,1,1)$

このとき，文 s_1 と文 s_1 がどれだけ似ているでしょうか。そのために，ベクトル同士がどの程度重なっているのか調べるための，**コサイン類似度（余弦類似度）**を使います。例えば，n 個の要素を持つ二つのベクトル $\vec{a}=(a_1,a_2,\cdots,a_n)$ と $\vec{b}=(b_1,b_2,\cdots,b_n)$ に対するコサイン類似度 $sim_{\cos}(\vec{a},\vec{b})$ は，$\vec{a}$ と $\vec{b}$ で挟まれた角度を θ としたとき，

$$sim_{\cos}(\vec{a},\vec{b}) = \cos\theta = \frac{a_1b_1 + a_2b_2 + \cdots + a_nb_n}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}\sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2}\sqrt{\sum b_i^2}}$$

と表します。コサイン類似度は，－1 以上 1 以下の値をとります。コサイン類似度が 1 に近いほど，二つのベクトルは同じ向きに近く，それが－1 に近いほど，二つのベクトルは逆向きに近くなります。コサイン類似度に $\boldsymbol{x}^{(s_1)}$ と $\boldsymbol{x}^{(s_2)}$ を当てはめると，

$$sim_{\cos}\left(\boldsymbol{x}_{\text{words}}^{(s_1)}, \boldsymbol{x}_{\text{words}}^{(s_2)}\right)$$
$$= \frac{1+4+4+1+1+1+1}{\sqrt{1+4+4+1+1+1+1}\sqrt{1+4+4+1+1+1+1}} = \frac{13}{\sqrt{13}\sqrt{13}} = 1$$

となります。文 s_1 である「今年の景気は去年の景気よりも良い。」と，文 s_2 である「去年の景気は今年の景気よりも良い。」という二つは，$sim_{\cos}\left(\boldsymbol{x}_{\text{words}}^{(s_1)}, \boldsymbol{x}_{\text{words}}^{(s_2)}\right) = 1$ より，ベクトルとして同じ方向に完全に重なっていると判断されます。しかし，二つの文に含まれる単語の順序は異なるため，文の意味は同一のものでありません。このように，ベクトルとしての同一性と文の構造としての同一性が一致しない原因は，コサイン類似度を求めるためのベクトルの要素が bag-of-words 表現であるからです。つまり，bag-of-words 表現は，文を構成するパーツが同じであるかどうかは知らせてくれますが，パーツの位置が等しいかどうかという情報までは知らせてくれないのです。

異なる文同士を比べて，文を構成する単語の位置がどれだけ等しいか，つまり，文の構造が互いにどれだけ似通っているのか知るためには，例えば，**bag-of-bigrams** によるベクトル表現を使うことが便利でしょう。これは，先述した bigram を，ベクトルの要素に取り入れたものであり，隣り合う 2 個の単語の順序が考慮されています。単語 bigram を使うと，bag-of-bigram によるベクトル表現は，以下の通りとなります。

$\boldsymbol{x}_{\text{bigrams}}^{(s_1)}$	$= \left(n\left(\text{"今年の"}, s_1\right), n\left(\text{"の景気"}, s_1\right), n\left(\text{"景気は"}, s_1\right), n\left(\text{"は去年"}, s_1\right), n\left(\text{"去年よりも"}, s_1\right), n\left(\text{"よりも良い"}, s_1\right), n\left(\text{"去年の"}, s_1\right), n\left(\text{"は今年"}, s_1\right), n\left(\text{"今年よりも"}, s_1\right)\right)$
	$= (1,1,1,1,1,1,0,0,0)$
$\boldsymbol{x}_{\text{bigrams}}^{(s_2)}$	$= \left(n\left(\text{"今年の"}, s_2\right), n\left(\text{"の景気"}, s_2\right), n\left(\text{"景気は"}, s_2\right), n\left(\text{"は去年"}, s_2\right), n\left(\text{"去年よりも"}, s_2\right), n\left(\text{"よりも良い"}, s_2\right), n\left(\text{"去年の"}, s_2\right), n\left(\text{"は今年"}, s_2\right), n\left(\text{"今年よりも"}, s_2\right)\right)$
	$= (0,1,1,0,0,1,1,1,1)$

そして，$\boldsymbol{x}_{\text{bigrams}}^{(s_1)}$ と $\boldsymbol{x}_{\text{bigrams}}^{(s_2)}$ をコサイン類似度に当てはめると，

$$sim_{\cos}\left(\boldsymbol{x}_{\text{bigrams}}^{(s_1)}, \boldsymbol{x}_{\text{bigrams}}^{(s_2)}\right) = \frac{0+1+1+0+0+1+0+0+0}{\sqrt{1+1+1+1+1+1+0+0+0}\sqrt{0+1+1+0+0+1+1+1+1}} = \frac{3}{\sqrt{6}\sqrt{6}} = \frac{1}{2}$$

となります。文 s_1 である「今年の景気は去年の景気よりも良い。」と，文 s_2 である「去年の景気は今年の景気よりも良い。」という二つは，$sim_{\cos}\left(\boldsymbol{x}_{\text{bigrams}}^{(s_1)}, \boldsymbol{x}_{\text{bigrams}}^{(s_2)}\right) = 1/2$ より，異なるベクトルであることが分かります（正確には，$\cos\theta = 1/2$ よりベクトル間の角度 θ が 60 度)。この結果は，先ほどの bag-of-words 表現に基づくベクトルと比べて異なります。この bag-of-bigram 表現に基づくベクトルは，単語の順序についての情報が幾分か残されているため，二つの文に対する照合をより適切に行うことができるのです。

もちろん，隣り合う単語の数は 2 個に限定する必要はなく，3 個や 4 個でも構いません。隣り合う単語の数を一般的に n 個とするならば，**bag-of-n-gram** によるベクトル表現を使うことになります。理屈としては，この n に当てはまる数字が大きくなるほど，語順の情報量が増えるため，より正確に文と文を照合できるようになりますが，部分的な一致は無視されてしまいます。

1-2. 文字や文章をパターン別に分類する方法

文章や文を，グループ別に分けるためにはどうすればよいでしょうか。分類するための規則を作る方法は 2 種類あり，人間が正解を教える方法と，そうでない方法です。人間が正解を教えるとは，ラベル付きデータをコンピュータに教えることです。これを**教師付き学習**とよびます。ここでの**学習**とは，データを用いることで，入力に対してどのような出力をすればよいかという処理方法を導くことです。

サポートベクトルマシン（SVM）は，データの集合に一本の線を引くことで，二つに分類する方法です。これを専門的な用語で，線形二値分類器といいます。分類器とは，定規のような物理的に存在するものではなく，単なる分析方法のことです。SVM で大切なことは，データの集合のどこに境界線を引けば，分析者にとって望ましい分類ができるかということです。集合の分類先を，**クラス**とよぶことにします。これは，学校に例えると，全校生徒という集合を，1 年 1 組や 1 年 2 組というクラスに分けることと同じ意味を持ちます。

データの集合を，2 つのクラス（好景気，不景気）に分けたいとします。全てのデータは，平面座標上に散らばっているとしましょう。2 つのクラスに分けることができる

境界線には，いくつか候補があります。その中で最も納得できる境界線の引き方は，両方のクラスから同じ距離に境界線を引くことです[13]。こうした方法を，**マージン最大化**といいます。境界線の近傍のデータのことを，サポートベクトルいい，これが SVM という名前の由来となっています。

付録２　深層学習

2-1. 文脈の解析方法と機械学習

ディスカッションペーパーの上巻では，テキストデータの分析単位が単語であったため，文脈情報を取り込むことができず，センチメント分析が不完全になってしまったことを解説しました。文脈情報を取り入れる試みは，いくつかの分析方法で見られます。

まずは，**言語モデル**です。これは，複数の単語で構成される文が，生成される確率をモデル化したものであり，違和感のない文は高い確率で，違和感のある文は低い確率で，それぞれ生成されると考えます。言語モデルには 2 種類あります。第一は，n-gram モデルといい，分析対象とする単語の前後に配置された n 個の単語の登場回数を計算します。第二は，**ニューラル言語モデル**です。これは，後述するニューラルネットワークを通じて，入出力処理方法の解を計算します。いずれの言語モデルあっても，文の生成確率を直接モデル化することには，困難が伴います。そこで，代替案として，文中の各単語の生成確率を，その単語の前に出現した単語（列）が与えられたという条件下で予測します。文脈とは，ある単語の出現確率を計算する際に用いる周囲の単語のことであると解釈します。

次の，文脈情報を取り入れた方法は，**分散表現**です。これは，単語同士の関係を座標（ベクトル）で表現する方法です。意味の近い単語同士は，座標の位置が近くなります。他方で，意味の異なる単語同士は，座標の位置が逆になります。分散表現を得るためには，ニューラル言語モデルを学習します。これは，ある単語の埋め込み層の変換行列に含まれたベクトルが似ていれば，最終的な出力も似通ったものとなります。意味的に似通った単語は，似通ったベクトルに変換されます。その結果，分散表現も学習できたこととなります。

分散表現を求める方法は，他にもあります。それは，ニューラル言語モデルの派生として登場した，**対数双線形モデル**です。このモデルによって獲得された分散表現は，**Word2vec** といいます。Word2vec の背後では，**skip-gram**，または **CBoW** という，モデルが利用されています。

以上で説明した言語モデルと分散表現は，ニューラルネットモデルを通じて結びつきます。テキストマイニングにおいて，単語ではなく文を分析単位とするためには，ニュ

[13] データ全体をクラス別に分ける境界線のことを，**分離平面**と言います。

ーラルネットワークの分析が有効です。

センチメント分析とは，言葉がもつ感情の良し悪しを，数値で表すということです。上巻で解説したセンチメント分析は，既に存在する単語の感情極性辞書を利用することで，単語に対して決められた数値を割り振るという作業でした。これに対して，この下巻で紹介するセンチメント分析は，文に対してふさわしい数値を推定して割り振るという作業を行います。この手法で重要なことは，事前に単語の感情極性辞書を必要とせず，コンピュータに，文と数値のペアを学習させるということです。学習といっても，コンピュータが自律的に学習するのではなく，人間が教え込ませる「教師付き学習」という方法を使います。これは，人にたとえると，教師が学生に宿題を出し，模範解答も教えることにより，学生は期末試験で出題された問題に対して正解できるということになります。

私たちは，どのようにして，教師付き学習を行うのでしょうか。目標は，機械であるコンピュータに文を入力したら，評価する値を自動的に出力してもらうことです。しかし，コンピュータは，初めから文を自律的に判断できません。まず，「例題と模範解答のペア」のことである**学習データ**を学ばせることで，コンピュータの判断能力を発達させることから始めます。例題と模範解答のどちらも，データとして分析者の手元にあるとします。例題を入力すると，模範解答を出力するような関数（数学的な入出力の変換装置）を設定します。この時点で，入力と出力の関係をある程度は決めておきます。例えば，入力する値が大きければ，出力される値も大きくなるという程度で構いません。

ところで，関数の内部では，入力と出力の関係の強さを表すパラメータ（見えない数値）が働いていますが，その大きさは分かりません。そこで，入力に対して所望の出力を得ることのできるように，パラメータを推定します。パラメータを推定し終えたら，例題とは異なりますが，類似する新しい問題を関数に入力することで，コンピュータは自動的に解答を出力できるようになります。この一連の作業が，機械学習における教師付き学習です。

2-2. ニューラルネットワーク

ニューラルネットワークとは，人間の脳内で情報伝達を行う神経細胞であるニューロンの働きを再現するような，数学モデルのことです。数学モデルと言うと，難しそうに聞こえるかもしれませんが，そのようなことはありません。ニューラルネットワークは，関数の一種であり，例えば，中学１年で学んだ一次関数のようなものです。一次関数では，入力する値に対して，直線のような形（線形）で出力される値が決まります。これに対してニューラルネットワークは，入力に対して曲がった形（非線形）で出力される値が決まります。このため，二次関数を使うこともあります。

なぜニューラルネットワークという関数は，一次関数とは異なり，複雑な反応をするのでしょうか。それは，ニューロンの働きに由来します。一つのニューロンは，他の複

数のニューロンから信号を受けて，その合計が一定の大きさを超えたときのみ反応します。ニューロンは生物学の用語であるため，これ以降では，ニューロンのように入力と出力の機能を持つ関数を，**ユニット**と呼び，ニューロンとは区別します。したがって，ニューラルネットワークとは，ユニットとユニットを繋げて合わせたネットワークのことを指します。

ここからは，ニューラルネットワークの数学モデルについて，詳しく説明していきます。まずは，ユニット単体の働きを考えます。ユニットは関数であり，その出力変数を y と表すことにします。ユニットは，複数の変数を持ちます。例えば，3 つの変数がある場合，それらを x_1, x_2, x_3 と表すことにしましょう。また，各変数は，重み付けされてユニットへ入力されると考えます。そうした重みを w_1, w_2, w_3 と表すことにします。すると，重み付けされた変数は，

$$w_1x_1 + w_2x_2 + w_3x_3$$

と表されます。これは，他の三つのユニットから受け取った入力信号です。ユニットが，出力反応を示すとき $y = 1$ ，出力反応を示さないとき $y = 0$ と表します。ユニットが，出力反応を示すかどうかは，受け取った入力信号の大きさに依存します。そこで，ユニットが反応する入力信号の閾値を θ と表すと，ユニットの反応条件は，

$$\begin{cases} \text{出力反応なし } (y = 0): \ w_1x_1 + w_2x_2 + w_3x_3 < \theta \\ \text{出力反応あり } (y = 1): \ w_1x_1 + w_2x_2 + w_3x_3 \geq \theta \end{cases}$$

と表現できます。こうしたユニットの出力反応を単純に表したものが，**付図 1** です。

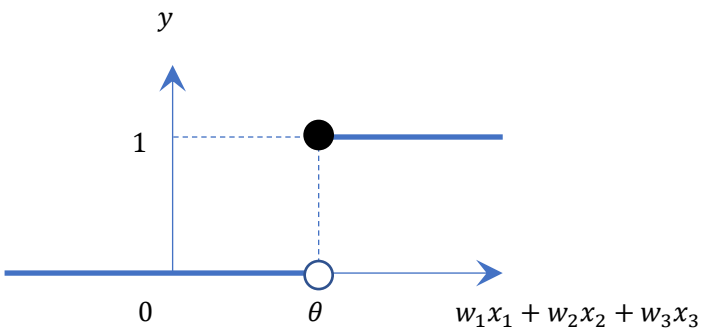
しかし，実際には，出力反応が図 1 のように極端であることは稀です。そこで，入力に対する出力は，0 から 1 の区間における任意の値であると考えることにします。具体的に，**付図 2** のように S 字型の曲線でユニットの反応を表します。**付図 2** は，2 つの点で**付図 1** との違いがあります。第一は，入力値 $(w_1x_1 + w_2x_2 + w_3x_3)$ と出力の閾値 (θ) の差分を， z という**潜在変数**で置き換えた点です。すなわち，

$$z = w_1x_1 + w_2x_2 + w_3x_3 - \theta$$

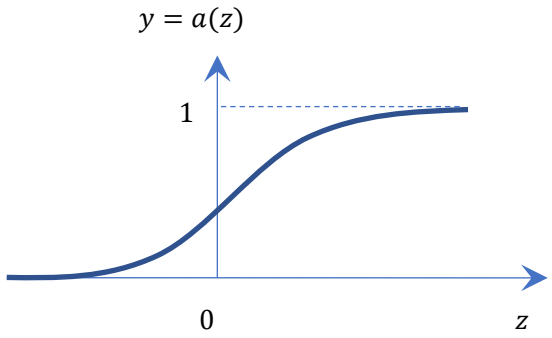
です[14]。潜在変数 z は，手元にある複数の入力データに，出力への影響という情報を加

[14] このような z は，深層学習の分野では「重み付き入力」とよばれるが，モデルの出力に直接作用するが分析者には見えない変数であるため，本稿では潜在変数とよぶことにする。

【付図 1】



【付図 2】



味して，濃縮変換されたものです。潜在変数 z が負値であれば，情報を十分に伝えることができず，出力反応が弱いことを表します。逆に，潜在変数 z が正値であれば，情報を十分に伝えることができるため，出力反応が強いことを表します。第二の違いは，潜在変数 z に対するユニットの反応 y を表す S 字型曲線は，**活性化関数**（activation function）という変換装置によって描写されている点です。活性化関数は，具体的な計算式で表されていない場合，一般的な関数 $a(z)$ と表し，それ自体はユニットの反応 y に置き換えられます。すなわち，$y = a(z)$ です。

実際に，活性化関数をコンピュータで計算する際に，具体的な計算式を設定する必要があります。活性化関数は，それをグラフで表したときに，滑らかに増加するという条件を満たせば，何でも構いません[15]。そこで，活性化関数の代表的な計算式として，以下のように定義される**シグモイド関数** $\sigma(z)$ というものが使われます。

---

[15] 活性化関数は，数学的に単調増加かつ微分可能であれば良い。

$$\sigma(z) = \frac{1}{1+e^{-z}}, \qquad (e = 2.718281\cdots).$$

なお，e は自然対数の底です。シグモイド関数は，活性化関数が求める条件を満し，かつ，計算しやすい特徴を備えています[16]。

ここまでの説明をまとめると，ニューラルネットワークは，複数の入力値に情報を加味することで一つの潜在変数にまとめて，それをユニットである活性化関数に通して，何らかの値を出力させます。ところで，計算を容易にするため，潜在変数を構成する $-\theta$ を b へ置き換えて，この $b(<0)$ を**バイアス**とよぶことにします。予想される b が大きい負値であるほど，バイアスは大きいと考えます。先述した活性化関数とその制約となる潜在変数を，以下のようにまとめたものが，ニューラルネットワークを考えるための基礎となります。すなわち，

$$y = a(z)$$
$$\text{s.t. } z = w_1x_1 + w_2x_2 + w_3x_3 + b$$

となります。なお，式中の“s.t.”は“subject to（和訳：条件の下で）”の略です。そして，これらの式の中で，分析者の手元にあるデータは，変数である入力値 x_1, x_2, x_3 と出力値 y です。変数には，具体的な数字が代入されます。そして，データとして観察できないパラメータは，各入力値への重みである w_1, w_2, w_3 とバイアス b です。入力値に重みとバイアスが加わることで，潜在変数となります。

付図３は，最も単純なニューラルネットワークを表したものです。大きい重みを与えられた入力値の情報は，優先的に出力へと伝えられます。例えば，$w_1 = 0.1$, $w_2 = 0.05$, $w_3 = 0.06$ であるならば，全ての重みの中で w_1 が最も大きいので，それは x_1 の情報を最も重視して出力へ伝えることを意味します。ただし，バイアスが大きい場合，ユニ

---

16 シグモイド関数 $\sigma(z)$ を変数 z で微分すると，

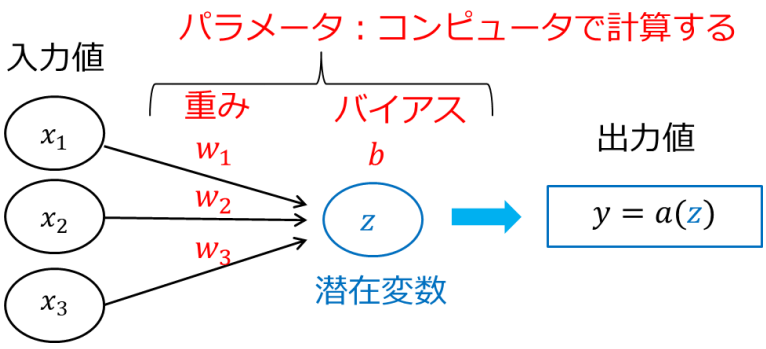
$$\sigma'(z) = -\frac{1}{(1+e^{-z})^2}(-e^{-z}) = \frac{(1+e^{-z})-1}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} - \frac{1}{(1+e^{-z})^2} = \sigma(z) - \sigma(z)^2.$$

因数分解によって，二つの項をまとめると，

$$\sigma'(z) = \sigma(z)\big(1-\sigma(z)\big)$$

というように，シグモイド関数の微分は，関数それ自体で表すことができる。このことより，シグモイド関数を用いた計算は容易である。

【付図 3】



ットの反応自体にブレーキが作動します。すなわち，バイアスは，潜在変数 z を小さくするよう作用します。以上より，ニューラルネットワークの解析とは，入力値と出力値を上手く結びつけるために，「重みとバイアスの大きさがどれくらいであればよいのか」ということを推定する作業です。

ニューラルネットワーク解析のイメージを持っていただくために，数値例を紹介します。解析の前提として，活性化関数を単純化します。**付図 2** で表した活性化関数は非線形であり，計算が複雑となります。その代わりに，線形の活性化関数を考えます。例えば，$y = 0.5z + 0.2$ と設定します。**付図 4** は，この線形の活性化関数を表しています。

次に，手元にあるデータは，入力値の集合 $X = \{x_1,\ x_2\}$ と出力値の集合 $Y = \{y\}$ で構成されているとします。各データの数字は，$X = \{1,\ 2\}$ と $Y = \{1\}$ であるとしましょう。なお，この活性化関数の下では，出力値が $y \geq 0.2$ であればユニットの反応は良く，$y < 0.2$ であれば，ユニットの反応は鈍いということにします。データとして手元にある数値を，活性化関数と制約である潜在変数に代入すると，

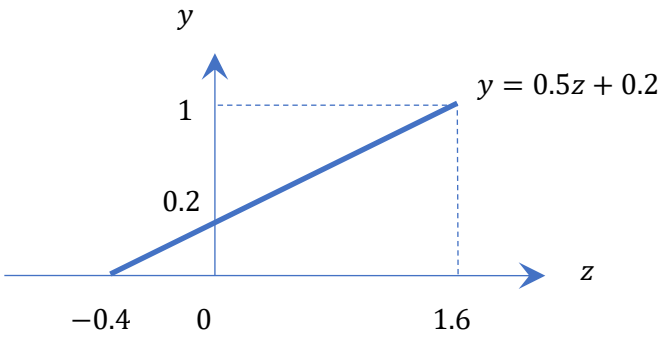
$$1 = 0.5z + 0.2$$
$$\text{s.t. } z = w_1 + 2w_2 + b$$

と表されます。活性化関数の解となる潜在変数を z^* と定義して，活性化関数を z について解くと $z^* = 1.6$ を得ます。このため，以下の制約式

$$1.6 = w_1 + 2w_2 + b$$

を満たすように，パラメータ集合 $\{w_1, w_2, b\}$ の数値を探します。この答えは数多くあるため，コンピュータを使って様々な数値を何回も代入することで，結果を更新していきます。この制約を満たすパラメータ値の一つとして，$\{0.75,\ 1.5,\ -2.15\}$ があります。

【付図 4】



これらの値を制約式に代入すると，

$$0.75 + (2 \times 1.5) - 2.15 = 1.6$$

であることを確認できます。

ところで，**付図 4** より，潜在変数が $z \geq 0$ であるときユニットの反応は良く，潜在変数が $z < 0$ であるときユニットの反応は鈍いことが分かります。このことより，ユニットの反応の良しあしを決定付けるような入力値の集合 $X = \{x_1, x_2\}$ を，平面上の境界線として図示することができます。その境界線は，パラメータ値を代入した潜在変数の制約式である $z = 0.75x_1 + 1.5x_2 - 2.15$ に，ユニットの反応の良しあしの境となる $z = 0$ を代入することによって得られます。すなわち，

$$0.75x_1 + 1.5x_2 - 2.15 = 0$$

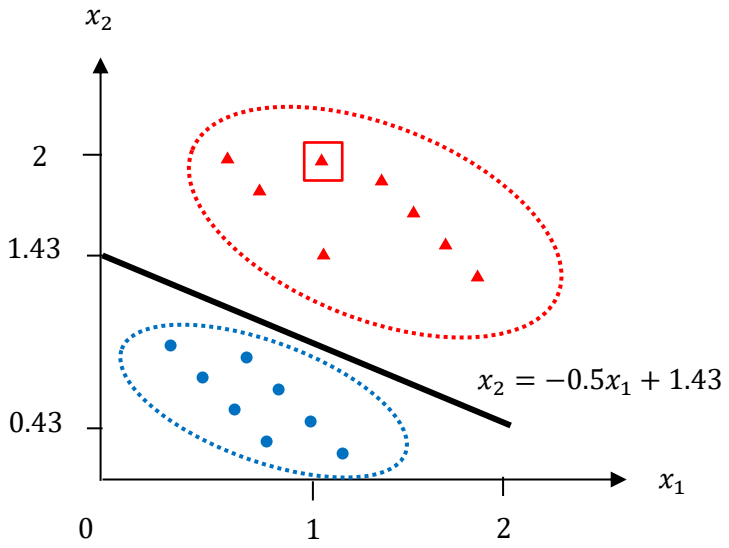
であり，変形して

$$x_2 = -0.5x_1 + 1.43$$

です。

付図 5 は，推定されたパラメータの下で，ユニットの反応の良し悪しを分けるような入力データの分布例を表しています。境界線の上方にあるデータ群（赤色）の下ではユニットの反応は良く，境界線の下方にあるデータ群（青色）の下ではユニットの反応は鈍いことを意味しています。

【付図 5】



入力値 $X = \{1, 2\}$ は，**付図 5** の中で赤い四角で囲まれた▲に位置するため，ユニットが上手く反応するような入力値であることが分かります。推定の結果，どのような入力値のベクトル（この場合，x_1, x_2の 2 変数）が，ユニットの反応を高めるのかどうかということまで分かるのです。これをもって，推定作業が完了します[17]。

---

[17] 本文第 7 章において紹介した RNN における入出力とパラメータの関係は，以下の通りである。RNN では，長さ T の入力（単語）の系列 $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_T})$ が与えられた場合，t 番目の単語を受けた隠れ層の第 ℓ 層目のユニットの状態ベクトル $\boldsymbol{h}_t^{(\ell)}$ を，以下のように再帰的に更新する。

$$\boldsymbol{h}_t^{(\ell)} = \boldsymbol{a}^{(\ell)}\left(\boldsymbol{W}^{(\ell)}\begin{bmatrix}\boldsymbol{h}_t^{(\ell-1)} \\ \boldsymbol{h}_{t-1}^{\ell}\end{bmatrix} + \boldsymbol{b}^{(\ell)}\right)$$

ここで，推定の対象となるパラメータは，$\boldsymbol{W}^{(\ell)}$（重み行列と呼ばれる）に縮約されている。なお，$\boldsymbol{a}^{(\ell)}$ は p43 で紹介した活性化関数であり，$\boldsymbol{b}^{(\ell)}$ はバイアス・ベクトルとよばれるものであり，これらも推定の対象となる。より詳細な解説は坪井他(2017)を参照していただきたい。

参考文献

生田祐介・木下祐輔・松林洋一（2020）「テキストデータを利用した新しい景況感指標の開発と応用（上）―入門編：基礎的概念と分析手法の解説―」, APIR Discussion Paper Series No. 47。

石田基広・金明哲編著（2012), 『コーパスとテキストマイニング』共立出版。

和泉潔・後藤卓・松井藤五郎(2009)「テキスト情報による金融市場の逐次外挿予測」『人工知能学会ファイナンスにおける人工知能応用研究会』SIG-FIN-03-02 pp6-14。

大高一樹・菅和聖(2018)「機械学習による景気分析－「景気ウォッチャー」調査のテキストマイニング」, 日本銀行ワーキングペーパーシリーズ No.18-J-8。

奥村学(2010)『自然言語処理の基礎』コロナ社。

奥村学(2010)『言語処理のための機械学習入門』コロナ社。

岡崎陽介・敦賀智裕（2015)「ビッグデータを用いた経済・物価分析について―研究事例のサーベイと景気ウォッチャー調査のテキスト分析の試み―」BOJ Reports & Research Papers。

涌井良幸・涌井貞美『ディープラーニングがわかる数学入門』技術評論社 2017 年

関和広・生田祐介・松林洋一（2020）「ニュース記事に基づく景気指標 S-APIR の開発」, 第 24 回人工知能学会金融情報学研究会。

神嶌敏弘編『深層学習』近代科学社 2015 年

黒橋禎夫・柴田知秀(2016)『自然言語処理概論』サイエンス社。

小林雄一郎(2017)『R によるやさしいテキストマイニング[機械学習編]』オーム社。

五島圭一・高橋大志・山田哲也（2019）「自然言語処理による景況感ニュース指数の構築とボラティリティ予測への応用」IMES Discussion Paper Series No.2019-J-3。

塩野剛士(2018)「人工知能とテキストデータを活用した数量分析」IMES Discussion Paper Series No.2018-J-9。

巣籠悠輔『詳解ディープラーニング：TensorFlow・Keras による時系列データ処理 』マイナビ出版 2017 年

高村大也・乾孝司・奥村学（2006)「スピンモデルによる単語の感情極性抽出」『情報処理学会論文誌』Vol.47 No.2 pp627-637。

高村大也(2010)『言語処理のための機械学習入門』コロナ社。

土屋誠司(2015)『はじめての自然言語処理』森北出版株式会社。

坪井祐太・海野裕也・鈴木潤（2017)『深層学習による自然言語処理』講談社。

照井伸彦(2018)「ビッグデータ統計解析入門－経済学部/経営学部で学ばない統計学－」日本評論社。

松本裕治・奥村学(2017)『コーパスと自然言語処理』朝倉書店。

山本裕樹・松尾豊（2016)「景気ウォッチャー調査の深層学習を用いた金融レポートの

指数化」，第 30 回人工知能学会全国大会，3L3-OS-16a-2。
Cho, K., Merrienboer, van B., Bahdanau, D., and Bengio, Y., (2014) “On the properties of neural machine translation: Encoder–decoder approaches,” in Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111.
Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., (2019) “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186.
Manevitz, L. M. and Yousef, M., (2002) “One- class SVMs for Document Classification,” The Journal of Machine Learning Research, Vol. 2, pp. 139–154.
Manning, C. D., Raghavan, P., and Schutze, H, (2008) “Introduction to information retrieval,” Cambridge University Press, NY.
Takamura, H., Inui, T., and Okumura, M., (2005) “Extracting semantic orientations of words using spin model,” in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005), pp. 133–140.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I., (2017) “Attention is all you need,” in Proceedings of the 31st International Conference on Neural Information Processing Systems.