

学修番号 14890522

## 修士論文

# 単語分散表現を用いた 多層 Denoising Auto-Encoder による評価極性分類

張 培楠

2016 年 2 月 24 日

首都大学東京大学院  
システムデザイン研究科 情報通信システム学域

文部省

張 培楠

審査委員：

小町 守 准教授 (主指導教員)

山口 亨 教授 (副指導教員)

高間 康史 教授 (副指導教員)

# 単語分散表現を用いた 多層 Denoising Auto-Encoder による評価極性分類\*

張 培楠

## 内容梗概

ウェブが普及し、テキストによる評価情報が注目を集めている。中でもソーシャルメディアやブログといったユーザ発信サービスの使用率は年々上昇し、特定のショッピングサイトに記述されていない商品やサービスに関する莫大な極性情報を含んでいる。そのため、極性を含む評価（肯定・否定）情報を適切に分類することは必要不可欠な技術といえる。

このような評価極性分類タスクにおいての一般的な目標は、ある文もしくは文章を与え、それをポジティブ（望ましい）とネガティブ（望ましくない）に分類することである。その際に機械学習で用いられる最も基礎的な素性・特徴量（以下素性と記述）は単語（Bag-of-Words）素性であり、例えば「便利」という単語にはポジティブに分類される重みが学習されやすく、「不便」という単語ネガティブに分類される重みが学習されやすい。

しかし、単語素性のみを使った分類では限界がある。まず、従来のヒューリスティックに分類ルールを作成する手法や極性辞書を用いた手法は言語に依存していたり、極性辞書を作る必要があったりと人的なコストが大きい。さらに日々増加する新語や未知語に対応することが困難であることが予想される。これらによってデータスパースネス問題が引き起こされる可能性がある。また、 $n$  項関係を考慮する上で多項式カーネルを用いる必要があるが、それでは素性が離散的になってしまい上手く学習できない。したがって、これらのことにより表層の素性だけでは不十分といえる。

さらに、単語素性は統語構造を考えないため「良いデザインだけど不便」と「不便だけど良いデザイン」が同じ極性になってしまうが、主節が主な意見情報を担っ

---

\*首都大学東京大学院 システムデザイン研究科 情報通信システム学域 修士論文, 学修番号 14890522, 2016 年 2 月 24 日.

ていることを考慮すると、前者はややネガティブ、後者はややポジティブな意見情報を表現していると考えられるが、このような違いを Bag-of-Words 素性では表現できない。

この統語構造の問題を解決するため、Nakagawa らは係り受け関係を用いて構文木を作成し、その依存部分木に極性を割り当てる手法を提案した。しかしそれには複雑な係り受けに基づく素性テンプレートを設計するため、専門的知識が必要かつモデルが複雑になってしまう。

そこで我々は極性辞書のような語彙知識を用いなくても単語の意味を表現でき、かつデータスパースネスに頑健な分散表現と、複雑な素性やモデル設計を必要とせず、かつ高い表現力を持つ深層ニューラルネットワークに注目した。

したがって、本研究では単語の分散表現と深層ニューラルネットワークの一種である多層 Denoising Auto-Encoder (以下 SdA と略す) を用いて評価極性分類タスクに取り組む。

単語の分散表現では、単語をベクトルとして表現する。しかし分散表現では単語ベクトルを従来の 1-of-K のような疎なベクトルではなく、単語自体の意味を表す密なベクトルとして扱う。つまり意味が近い単語同士のベクトルの距離は近いということになる。極性分類タスクにおいて単語の意味は極性を大きく左右するため、記号的な意味しかない 1-of-K 表現よりも、より意味を捉えることができる分散表現の方が望ましい。その単語分散表現を学習する手法として Mikolov らの Skip-gram や CBOW のモデルが近年大きな成果を収めている。

SdA は Denoising Auto-Encoder (以下 dA と略す) を多層に重ねたもので、高い汎化能力を持ったニューラルネットワークである。積み重ねられた dA によって抽出される特徴は層が深くなっていくのに連れてより抽象的な表現を獲得できる。この手法は音声認識や画像処理、分野適応などで高い表現能力を証明している。

以上のことを踏まえて、本研究の主要な貢献を以下の 2 点とする。

- 評価極性分類タスクにおいて、大規模コーパスから学習した単語分散表現を用いることと、多層 (3 層以上) の SdA を用いることが分類性能に大きく寄与することを示した。
- 日本語の評価極性分類タスクに対し、複雑なモデルを設計することなく、現時点における世界最高精度を達成した。



# Japanese Sentiment Classification with Stacked Denoising Auto-Encoder using Distributed Word Representation\*

Peinan Zhang

## Abstract

As the popularity of social media continues to rise, serious attention is being given to review information nowadays. Reviews with positive/negative ratings, in particular, help (potential) customers to compare products and to make purchasing decisions. Consequently, automatic classification of the polarities (such as positive and negative) of reviews is extremely important.

The general goal of our task is to classify the input sentences or articles into positive or negative labels. The most basic feature for text classification in machine learning is bag-of-words feature. For instance, the word “convenient” has the weight that tends to be learned as positive and conversely the word “inconvenient” are prone to be learned as negative.

However, the bag-of-words feature has numbers of shortcomings. To begin with, huge human labor is necessary for making polarity dictionaries and building classification rules, and thus the outcome cannot cope with the new words that emerge daily. This will lead to a data sparseness problem. In addition, although the polynomial kernel is often used to consider n-ary relations, it will not learn well because of these discrete features.

Furthermore, the bag-of-words feature cannot take syntactic structures into account. This leads to mistakes such as “a great design but inconvenient” and “inconvenient but a great design” being deemed to have the same meaning, even

---

\*Master’s Thesis, Department of Information and Communication Systems, Graduate School of System Design, Tokyo Metropolitan University, Student ID 14890522, February 24, 2016.

though their nuances are different; the former is somewhat negative whereas the latter is slightly positive.

To solve this syntactic problem, one of the previous studies proposed a sentiment analysis model that used dependency trees with polarities assigned to their subtrees. However, the proposed model requires specialized knowledge to design complicated feature templates.

In this study, we propose an approach that uses distributed word representation to overcome the first problem and deep neural networks to alleviate the second problem. The former is an unsupervised method capable of representing a word's meaning without using hand-tagged resources such as a polarity dictionary. In addition, it is robust to the data sparseness problem. The latter is a highly expressive model that does not utilize complex feature engineering or models.

Therefore, we work on the sentiment classification task with distributed word representation and stacked denoising auto-encoder, which is one of the deep neural networks.

Distributed word representations, or word embeddings, represent words as vectors. Distributed representations of word vectors are not sparse but dense vectors that can express the meaning of words. Sentiment classification tasks are significantly influenced by the data sparseness problem. As a result, distributed word representation is more suitable than traditional 1-of-K representation, which only treats words as symbols. In our proposed method, to learn the word embeddings, we employ a state-of-the-art word embedding technique called word2vec. Although several word embedding techniques currently exist, word2vec is one of the most computationally efficient and is considered to be state-of-the-art.

A stacked denoising auto-encoder (SdA) is a deep neural network that extends a stacked auto-encoder with denoising auto-encoders (dA). Stacking multiple layers and introducing noise to the input layer adds high generalization ability to auto-encoders. This method is used in speech recognition, image processing

and domain adaptation; further, it exhibits high representation ability.

Our research makes the following two main contributions:

- We show that distributed word representation learned from a large-scale corpus and stacked denoising auto-encoder with multiple layers (more than three layers) contributes significantly to classification accuracy in sentiment classification tasks.
- We achieve state-of-the-art performance in Japanese sentiment classification tasks without designing complex features and models.

# 目次

図目次	ix
第 1 章 はじめに	1
第 2 章 関連研究	3
2.1 評価極性分類 . . . . .	3
2.2 深層学習 . . . . .	4
第 3 章 単語分散表現を用いた	
多層 Denoising Auto-Encoder による評価極性分類	6
3.1 分散表現 . . . . .	6
3.1.1 単語の分散表現 . . . . .	6
3.1.2 文の分散表現 . . . . .	7
3.2 Auto-Encoder . . . . .	8
3.2.1 Denoising Auto-Encoder . . . . .	9
3.2.2 Stacked Denoising Auto-Encoder . . . . .	10
第 4 章 評価極性分類実験	12
4.1 比較手法 . . . . .	12
4.2 使用データ・ツール . . . . .	13
4.3 実験結果 . . . . .	14
第 5 章 考察	18
5.1 手法 . . . . .	18



5.2	パラメータ . . . . .	19
5.3	活性化関数, 最適化手法及びデータサイズによる正解率の変化 . . .	21
5.3.1	活性化関数 . . . . .	21
5.3.2	最適化手法 . . . . .	21
5.3.3	データサイズによる正解率の変化 . . . . .	23
5.4	事例分析 . . . . .	23
第 6 章	おわりに	26
謝辞		27
参考文献		28
発表リスト		32

# 目次

3.1	文の分散表現 . . . . .	7
3.2	Auto-Encoder . . . . .	9
3.3	Stacked Denoising Auto-Encoder . . . . .	10
4.1	信頼区間と各手法の正解率 . . . . .	15
4.2	訓練データの増加にともなう正解率の変化 . . . . .	17
5.1	中間層数の変化に伴う学習時間 . . . . .	19
5.2	中間ノード次元の変化に伴う学習時間 . . . . .	20
5.3	活性化関数 . . . . .	22
5.4	AE において 100 Epoch 実行した際の各最適化手法の収束度 . . . . .	23

## 第1章 はじめに

ウェブが普及し、テキストによる評価情報が注目を集めている。また、ソーシャルメディアやブログといったユーザ発信サービスの使用率は年々上昇し、特定のショッピングサイトに記述されていない商品やサービスに関する莫大な極性情報を含んでいる。そのため、こういった極性を含む評価（肯定・否定）情報を適切に分類することは必要不可欠な技術といえる。

このような評価極性分類タスクにおいての一般的な目標は、ある文もしくは文章を与え、それをポジティブ（望ましい）とネガティブ（望ましくない）に分類することである。その際に機械学習で用いられる最も基礎的な素性は単語（Bag-of-Words）素性であり [18,27]、例えば「便利」という単語にはポジティブに分類される重みが学習されやすく、「不便」という単語はネガティブに分類される重みが学習されやすい。

しかし、単語素性のみを使った分類では限界がある。まず、従来のヒューリスティックに分類ルールを作成する手法や極性辞書を用いた手法は言語に依存していたり、極性辞書を作る必要があり人的なコストが大きい。さらに日々増加する新語や未知語に対応することが困難であることが予想される。したがってデータスパースネス問題が引き起こされる可能性があるため、表層の素性だけでは不十分といえる。

また、統語構造を考えないため「良いデザインだけど不便」と「不便だけど良いデザイン」が同じ極性になってしまうが、主節が主な意見情報を担っていることを考慮すると、前者はややネガティブ、後者はややポジティブな意見情報を表現していると考えられるが、このような違いを Bag-of-Words 素性では表現できない。

この統語構造の問題を解決するため、Nakagawa ら [17] は係り受け関係を用いて構文木を作成し、その依存部分木に極性を割り当てる手法を提案した。しかしそれには複雑な係り受けに基づく素性テンプレートを設計するため、専門的知識が必要かつモデルが複雑になってしまう。

そこで我々は極性辞書のような語彙知識を用いなくても単語の意味を表現でき、かつデータスパースネスに頑健な分散表現と、複雑な素性やモデル設計を必要とせず、かつ高い表現力を持つ深層ニューラルネットワークに注目した。したがっ

て本研究では単語の分散表現と多層 Denoising Auto-Encoder (以下 SdA と略す) を用いて評価極性分類タスクに取り組む。

単語の分散表現では、単語をベクトルとして表現する。しかし分散表現では単語ベクトルを従来の 1-of-K のような疎なベクトルではなく、単語自体の意味を表す密なベクトルとして扱う。つまり意味が近い単語同士のベクトルの距離は近いということになる。極性分類タスクにおいて単語の意味は極性を大きく左右するため、記号的な意味しかない 1-of-K 表現よりも、より意味を捉えることができる分散表現の方が望ましい。その単語分散表現を学習する手法として Mikolov ら [15,16] の word2vec が近年大きな成果を収めている。

SdA は Denoising Auto-Encoder (以下 dA と略す) を多層に重ねたもので、高い汎化能力を持ったニューラルネットワークである。積み重ねられた dA によって抽出される特徴は層が深くなっていくのに連れてより抽象的な表現を獲得できる。この手法は音声認識 [5] や画像処理 [26,28]、分野適応 [2] などで高い表現能力を証明している。

本研究の主要な貢献は以下の 2 点である。

- 評価極性分類タスクにおいて、大規模コーパスから学習した単語分散表現を用いることと、多層 (3 層以上) の SdA を用いることが分類性能に大きく寄与することを示した。
- 日本語の評価極性分類タスクに対し、複雑なモデルを設計することなく、現時点における世界最高精度を達成した。

最後に本論文の章立てを説明する。本章をこの論文の導入とし、続く第 2 章では評価極性分類と深層学習の二つの側面から関連研究について見ていき、本研究との差異について述べる。第 3 章では、本研究での提案手法を大きく分散表現と Auto-Encoder に分けて具体的に説明する。分散表現では単語の分散表現と文の分散表現、Auto-Encoder ではその原理とその派生である Denoising Auto-Encoder と Stacked Denoising Auto-Encoder について記述する。第 4 章では主に実験設定や手法、使用データ・ツール、結果などの実験に関連する内容を記す。第 5 章では実験結果を元に提案手法について具体例を用いながら考察をしていく。第 6 章では本研究の総括、そして今後の展望について述べる。



## 第 2 章 関連研究

ここでは本研究に関連する研究を，評価極性分類タスクと深層学習（単語の分散表現・多層ニューラルネットワーク）に分けて説明する．

### 2.1 評価極性分類

評価極性分類タスクについてはこれまでに多くの研究がなされてきた．

Nakagawa ら [17] は，評価表現の依存構造木を考え，個々の部分依存構造木に対する評価極性を隠れ変数で表し，隠れ変数間の相互作用を考慮した日本語と英語の評価極性分類を条件付き確率場（CRF）[14]で行った．これにより，評価極性の反転などの統語構造を考慮することを可能とした．しかし，12 種という非常に多くの素性テンプレートを作る必要があり，複雑なモデルである．それに対して，本研究での分類手法は素性エンジニアリングする必要がないため，比較的簡単なモデルである．

同じことが Choi ら [3] にも言える．Choi らは，係り受け関係を考慮した手法で評価極性の分類を行っているが，単語単位の極性からは人手によってルールを作成し，評価表現全体の極性を構成している．また，これらのルールは英語に基づいているため，日本語での分類を考えた際には新たに日本語のルールを作成しなければならない．しかし本研究では文の分散表現を単語の分散表現の平均で計算しているためため言語に依存しない手法である．

Ikeda ら [10] は構成する単語の周辺を見ることで評価極性を学習する手法で文分類を行ったが，評価表現辞書を使うため，辞書に登録されていない単語にはうまく対処できない．加えて，辞書を作成する必要があり，人的リソースがかかる上新語に迅速に対応できない．一方，本研究は極性辞書を用いないので未知語があっても大規模な言語データがあれば頑健に動作する．



## 2.2 深層学習

深層学習の大きな特徴のひとつとして、内部に素性抽出器を持つことで従来のように素性を人手によって設計する必要がないことがあげられる。これにより、素性エンジニアリングでの時間や労力を節約できる他、人間では作り得ないような有効な素性を取り入れることができるようになる。

Collobert ら [4] は入力である任意の長さの文を固定次元に変換するために、単語ベクトルを3つずつ畳み込むことで局地的な特徴を得たのち、各次元の最大値を取得して新たなベクトルにプーリングした。このプーリングされたベクトルを従来のニューラルネットワークを用いて単語分散表現の表現学習を行った。これは本研究で使用する表現学習とは異なる手法で単語分散表現を作成している。

Glorot ら [7] は分野適応タスクに SdA を使用した。Amazon の4つの分野の商品レビューに対しての評価を、それぞれ学習させた学習器を別の分野でテストを行い、その性能を測った。この研究には感情分類タスクも含まれているが、入力に Collobert ら [4] の手法で表現学習した最もよく出現する動詞のみを使用するため、文ベクトルを作成する本研究とは異なる。

Collobert ら [4] や Mikolov ら [15, 16] の表現学習を用いて単語の分散表現を作成し、Twitter などのマイクロブログの極性分類を行った研究として、dos Santos ら [6] と Tang ら [23] が挙げられる。両者とも感情分類タスクであるが、前者は文ベクトルを文字単位の畳み込みネットワークで生成したのに対し、後者は単語の分散表現自体を極性として表現したモデルを使用した。本研究では単語の分散表現を用いて単語単位のベクトルで文ベクトルを生成するためこれらの研究とは異なる。

Kim ら [12] は、Convolutional Neural Networks を用いて文の評価極性分類を行った。Mikolov らの CBOW で表現学習した単語分散表現を低い次元の層へ畳み込むことで素性マップを作成し、そしてそれらの素性マップをさらにマックスプーリングしたのち、softmax によって分類を行った。また、Kim らは素性マップを作成する手法に、異なる Window サイズで畳み込んだ素性マップを同じ層にプーリングすることで正解率の向上に成功した。この研究は本研究と同じ評価極性分類のタスクであり、構築したモデルも似通っているため比較実験を第4章で行った。

Socher ら [20, 21] は共通の Auto-Encoder を再帰的に配置し、入力するベクトルを連結 (concatenate) させることで単語の順番といった構文的な情報を考慮し

たモデルを作成した。また、構文木に AE を配置することで、フレーズ単位の極性を表現することを可能とした。この再帰的なニューラルネットワークを使用して文ベクトルを作成したが、これは本研究での文ベクトルを作成する手法とは異なる。

Tai ら [22] は Long Short-Term Memory (以下 LSTM) を木構造に配置することによって評価極性分類のタスクにおいて精度の向上を図った。Reccurent Neural Network の拡張である LSTM は、forget gate を導入することにより構造がより複雑化したが、長い入力进行处理する際の勾配消失・爆発問題に対処できることで長期依存を学習できるようになった。この LSTM を木構造に構築することで Stanford Sentiment Treebank を用いた評価極性分類タスクにおいて高い精度を得ることができた。この手法を適応させるためには、構文木の部分木全てにアノテーションされた学習データが必要となる。しかし日本語データでその条件を満たすものが存在しないため本研究で比較することはできなかった。

## 第 3 章 単語分散表現を用いた

### 多層 Denoising Auto-Encoder による評価極性分類

本研究では、日本語の文が与えられた時に、その極性をポジティブあるいはネガティブの 2 値に分類するタスクを考える。

提案手法は入力された文から文ベクトルを作成し、学習器への入力とする。文ベクトルは文に含まれる単語ベクトルの平均によって得る。単語ベクトルは分散表現を用いる。

3.1 節では分散表現、3.2 節では多層ニューラルネットワークの説明をしていく。

#### 3.1 分散表現

伝統的な単語のベクトル表現の方法に、Bag-of-Words を作る時に用いる 1-of-K 表現がある。1-of-K 表現による単語ベクトルの次元は語彙数と等しく、各次元が一つの単語に対応づく。1-of-K 表現では表層的に異なった単語同士は全く別の記号として扱うが、例えば「犬」と「猫」のようにどちらもペットとして飼われる動物であれば、似たような文脈で使用されることから、同じようなベクトルで表現したい。それを解決するのが分散表現である。

分散表現を学習するタスクは表現学習と呼ばれ、言語の意味的な類似度を捉える必要のある問題で近年注目を集めている。単語の表現学習では、各単語に低次元で密な実数値ベクトルを割り当て、各単語ベクトル表現がその単語の特徴を捉えるように大量のテキストコーパスからベクトル空間を学習する [1]。

##### 3.1.1 単語の分散表現

出現する語彙の数を  $|V|$ ，単語を表現するベクトルの次元を  $d$ ，1-of-K ベクトルを  $u \in \mathbb{R}^{|V|}$ ，単語ベクトルの集合の行列を  $L \in \mathbb{R}^{d \times |V|}$  とすると、ターゲットとしている  $k$  番目の単語  $w_k$  のベクトルは式 3.11 のように表すことができる。

$$w_k = Lu_k \quad (3.11)$$

単語の分散表現の学習には 2.2 節で挙げた Collobert ら [4] の手法以外に、



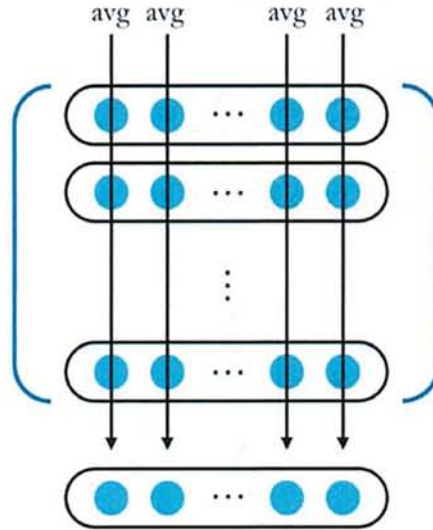


図 3.1 文の分散表現

word2vec で使用する Continuous Bag-of-Words (CBOW) もしくは Skip-gram が使用される手法もある [15, 16]. 前者は単語周辺の文脈から決めたウィンドウサイズ分の単語を使用して注目している単語を推定するモデルで、後者は CBOW とは逆に注目している単語からウィンドウサイズ分の周辺の単語を推定するモデルである. Mikolov らの研究では Skip-gram の方がよりよい性能を発揮したと示している. そのため本研究では Skip-gram を使用した.

### 3.1.2 文の分散表現

本研究では, コーパスにある  $|M|$  個の文を用いて, 本タスクで使用する文行列  $S \in \mathbb{R}^{|M| \times d}$  を作成する (図 3.1).

まず単語ベクトルを用いて文ベクトルを作成する.  $|N|$  単語から構成される  $1 \leq i \leq M$  番目の文が与えられた時に, その文に含まれている単語の特徴ベクトルから文ベクトル  $S^{(i)} \in \mathbb{R}^d$  を作成する. この時, 文ベクトル  $S^{(i)}$  の  $1 \leq j \leq d$  番

目の要素  $S_j^{(i)}$  は式 3.12 のように average で算出する [13].

$$S_j^{(i)} = \frac{1}{N} \sum_{n=1}^N w_n^{(i)} \quad (3.12)$$

次に、求める文行列  $S$  を式 3.13 のように算出する.

$$S = \begin{bmatrix} S^{(1)T} \\ S^{(2)T} \\ \vdots \\ S^{(M)T} \end{bmatrix} \quad (3.13)$$

## 3.2 Auto-Encoder

ニューラルネットワークにおいて、層がより深くなっていくのに連れて、誤差逆伝搬で学習させるには時間がかかる上に過学習もしやすくなる傾向がある. したがって、各層で事前学習によって特徴抽出しやすい初期値を持たせるために Auto-Encoder を用いる手法が提案されている.

Auto-Encoder は、Hinton ら [8] によって考案された教師なし学習のいち手法で、ニューラルネットワークを使用して入力の次元を中間層で圧縮するアルゴリズムからなっている. これは入力と同じになるような出力の中間層での表現を学習し、その中間層の次元は入出力のものより小さくすることで次元圧縮を可能としている (図 3.2). このとき、入力を中間層にエンコードする式を式 3.21, 中間層から出力層にデコードする式を式 3.22 とする.

$$y = s(Wx + b) \quad (3.21)$$

$$z = s(W'y + b') \quad (3.22)$$

このとき、 $s(*)$  は  $\tanh$  や Sigmoid 関数で、 $W, W'$  は重み行列、 $b, b'$  はバイアス項を表している.

$d$  次元の入力ベクトル  $x$  と出力ベクトル  $z$  の差を表す損失関数は、式 3.23 のように交差エントロピー関数で表されており、これを確率的勾配降下法 (SGD) や



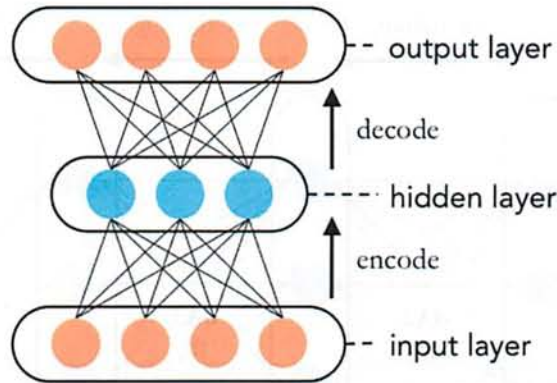


図 3.2 Auto-Encoder

AdaDelta などの最適化手法により最小化していく。具体的な最適化手法の比較は 4 章および 5 章で触れる。

$$L_H(x, z) = - \sum_{k=1}^d [x_k \log z_k + (1 - x_k) \log(1 - z_k)] \quad (3.23)$$

### 3.2.1 Denoising Auto-Encoder

従来の多層パーセプトロンでは損失関数に正則化項を使用することがほとんどだが、Auto-Encoder では Denoising を使用することで正則化と同等の役割を果たすことができる。

Denoising Auto-Encoder は Auto-Encoder の確率的なバージョンで、入力に対して学習のたびにランダムなノイズを与えて事前学習することで、より高い汎化能力を持った初期値を決定することができる手法である。損失関数はノイズの入っていない入力进行评估するため、Denoising を用いた Auto-Encoder ではより良い特徴を抽出することが可能である [24]。似たような正則化の働きをする手法として Dropout [9] があるが、これは入力ではなく中間ノードを一定の確率で無視して学習を行う。これにより各データごとに異なるモデルを使用して学習し、推定時にはあたかも全部の結果の平均を使っていることに対応しているので過学習しにくくなる。

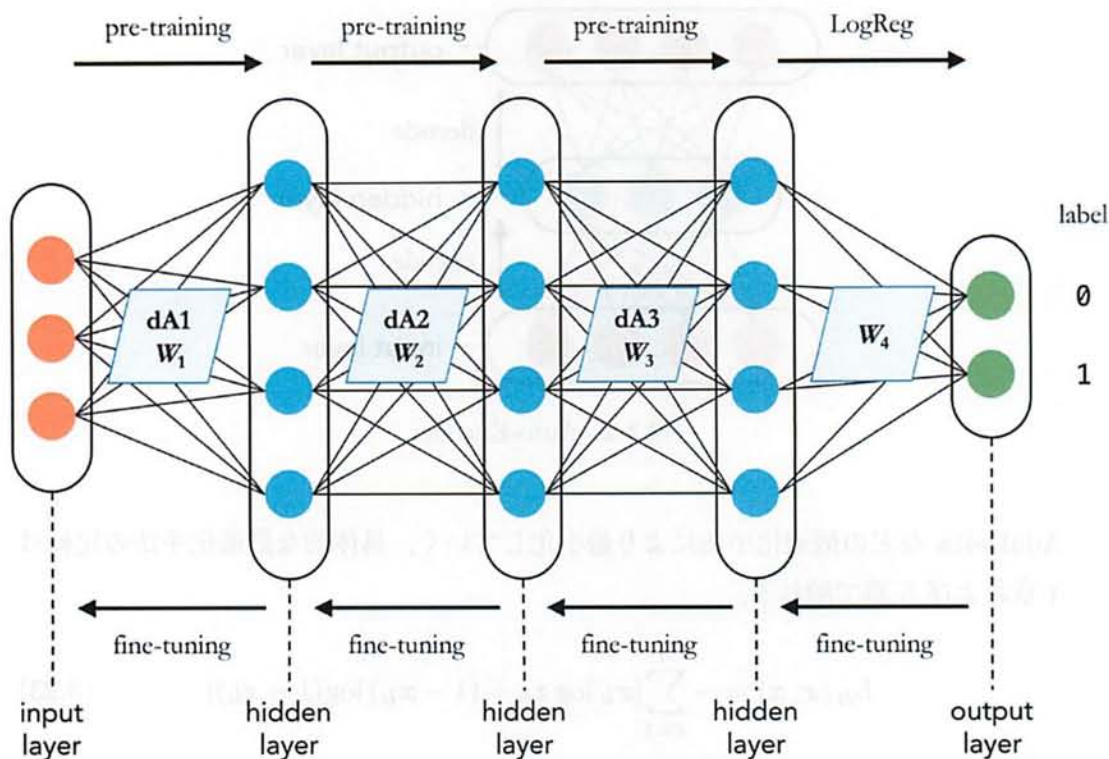


図 3.3 Stacked Denoising Auto-Encoder

### 3.2.2 Stacked Denoising Auto-Encoder

dA を多層に積み重ねることで、より表現能力を向上させた手法が Stacked Denoising Auto-Encoder である。積み重ねられた dA によって抽出される特徴は層が深くなっていくのに連れてより抽象的な表現を獲得できる [25]。

SdA による学習は 2 つの段階からなっている。まず、ニューラルネットワーク各層を dA として教師なしの事前学習 (pre-training) を行い、次にニューラルネットワーク全体で教師ありの微調整 (fine-tuning) を行う。事前学習では、入力  $A_i$  に始まり各層の dA による特徴抽出がなされ、抽出された内部表現である中間層が次の dA の入力として渡される。最後の中間層まで事前学習すると、その中間層がロジスティック回帰によって分類され、分類結果のベクトルが出力層に渡される。微調整では出力層から正解ラベルを使い誤差逆伝搬により各層での重み行列の更新

がなされる。(図 3.3)

図 3.3 において, 入力層は式 3.12 で得られる  $S_i \in \mathbb{R}^d$  で, 1 層目で重み行列  $W_1$  を持つ dA1 によって最初の中間層に変換される. なお, 中間層の数と中間ノードの次元はハイパーパラメータになっており, ここでは  $i$  層目の中間ノード次元を  $n_i$  とする. したがって  $W_1$  の次元は式 3.21 から  $n_1 \times d$  となる. 同様にして,  $l-1$  層目までの重み行列は  $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$  となる (ただし  $i > 2$ ). 最後の  $l$  層では, ロジスティック回帰によって正解ラベルと同じ次元  $d_{ans}$  の出力をするため, 重み行列  $W_l$  の次元は  $d_{ans} \times n_{l-1}$  となる.

## 第 4 章 評価極性分類実験

### 4.1 比較手法

SdA の有効性を示すために、線形分類器 (ロジスティック回帰)\*による分類との比較を行った. (LogRes-w2v)

単語の分散表現の有効性を示すために、Bag-of-Features を使った手法 (LogRes-BoF, SdA-BoF) との比較を行った. 1-of-K 行列で式 3.12 と同じ方法で文ベクトル  $S \in \mathbb{R}^{|V|}$  を作成し、それらに対して主成分分析 (PCA)<sup>†</sup>で  $d = 200$  次元まで次元削減を行った.

ここでは同じコーパスを使用した Nakagawa らの手法と本研究で示した手法、そして比較するハイパーパラメータの説明をする.

**MFS** データセットの中でもっとも大きい割合の回答 (ここではネガティブ) を常に選ぶベースライン.

**Tree-CRF** 本研究以前においての世界最高精度である隠れ変数を持つ CRF による手法. [17]

**CNN** Convolutional ニューラルネットワークを用いた手法. [12]

**LogRes-BoF** 1-of-K の単語ベクトルの average で文ベクトルを作成し, PCA により 200 次元まで次元削減を施した行列を線形分類器 (ロジスティック回帰) で分類する.

**SdA-BoF** LogRes-BoF で作成したベクトルを SdA で分類する.

**LogRes-w2v** 単語の分散表現で作成した文ベクトルを, 線形分類器 (ロジスティック回帰) で分類する.

**SdA-w2v** 3 章で説明した, 単語の分散表現で作成した文ベクトルを SdA を用いて分類を行う提案手法.

**SdA-w2v-neg** Nakagawa らと同様, 否定表現に対して処理を施して単語の分散表現を作成し, SdA を使って分類を行う.

\*ロジスティック回帰は Theano version-0.6.0, SdA は chainer version-1.4.1 を用いて実装した.

<sup>†</sup>scikit-learn version-0.10 を使用した.



また、提案手法に関するハイパーパラメータの Denoising と中間層数、中間ノード次元数は以下のように調節した。

Denoising の有効性を示すために、Denoising を施していない手法（ノイズレートが 0 %）と Denoising を行った手法（ノイズレートが 10 %, 20 %, 30 %, 40 %, 50 %）のものと比較を行った。Denoising の手法として、指定されたノイズレートで 0 を含む二項分布で乱数を生成し、それを入力ベクトルの対応する要素にかける。

多層の有効性を示すために、中間層の数が 1 層のものと、2 層以上のものとの比較を行った。

ネットワークの表現力を調べるために、中間ノードの次元数を 100, 300, 500, 700 次元で比較を行った。

## 4.2 使用データ・ツール

単語の分散表現の作成にあたって、Mikolov ら [15,16] の Skip-gram から作成される word2vec<sup>‡</sup> を使用した。学習には日本語版 Wikipedia のダンプデータ（2014 年 11 月時点）を使い、MeCab<sup>§</sup> にて分かち書きしたのち word2vec を用いてサイズ 200 次元で単語の分散表現を作成した。できあがったモデルの語彙数は、否定表現を処理しなかった手法では 426,782 語、否定表現を処理した手法では 431,456 語である。

実験には NTCIR-6 の意見分析パイロットタスクテストコレクション [19] と NTCIR-7 MOAT をランダムに混ぜ合わせた NTCIR-J を使用した。NTCIR-6 および NTCIR-7 コーパスは大規模なテストコレクションを用いた検索、アジア言語と英語の言語横断検索、質問応答、言語横断情報アクセスに焦点を当てたワークショップで作成されたコーパスである。本研究で使用した日本語のデータでは、毎日新聞と読売新聞の記事に対して 3 人のアノテータにより文単位で評価極性タグが付けられている。本研究では、同じ文に対して 3 人のアノテータのアノテーション結果の和集合を取ったものを使用した。アノテータのタグがポジティブとネガティ

---

<sup>‡</sup><https://code.google.com/p/word2vec/>

<sup>§</sup>MeCab version-0.996, IPADic version-2.7.0



表 4.1 NTCIR-6, NTCIR-7 及び NTCIR-J の極性の内訳

コーパス名	ポジティブ	ネガティブ	合計
NTCIR-6	769	1,830	2,599
NTCIR-7	228	570	798
NTCIR-J	997	2,400	3,397

ブの両方を含む場合、常に特定のアノテータのタグを使用する。両コーパスの極性（ポジティブ、ネガティブ）の内訳を表 4.1 に示す。コーパスにはニュートラル極性も含まれているが、本実験では二値分類であるためニュートラル極性は取り除いた。

また実験は 10 分割交差検定により行い、性能は正解率（Accuracy）によって評価した。

### 4.3 実験結果

ここでは上で挙げた手法での実験結果と、パラメータチューニングの比較をしていく。

まず、各手法を用いた正解率を表 4.2、信頼区間を含めた正解率を図 4.1 に示す。

Nakagawa ら [17] の提案手法である Tree-規則と Tree-CRF の正解率はそれぞれ 72.6 % と 78.9 % で、Kim ら [12] の提案手法である CNN の正解率は 80.3 であった。本研究においてのベースラインの MFS の正解率は 70.4 % で、Bag-of-Features を使用したロジスティック回帰と SdA による分類器の正解率はそれぞれ 70.8 % と 76.9 % であった。同じように、単語の分散表現を使ったロジスティック回帰と SdA による分類の正解率はそれぞれ 79.5 % と 81.7 % で、SdA-w2v と同じパラメータで、否定表現処理を施したデータに対して分類したところ、正解率が 80.9 % になった。なお、SdA での実験では、後述するパラメータチューニングを行って最も正解率が高かった組み合わせ（中間層 4 層、中間ノード 500 次元、ノイズレート 10 %）を使用している。<sup>¶</sup>

また、ハイパーパラメータごとの比較を表 4.3 のように行った。

<sup>¶</sup>開発データは使用せず、10 分割交差検定による正解率で評価した。

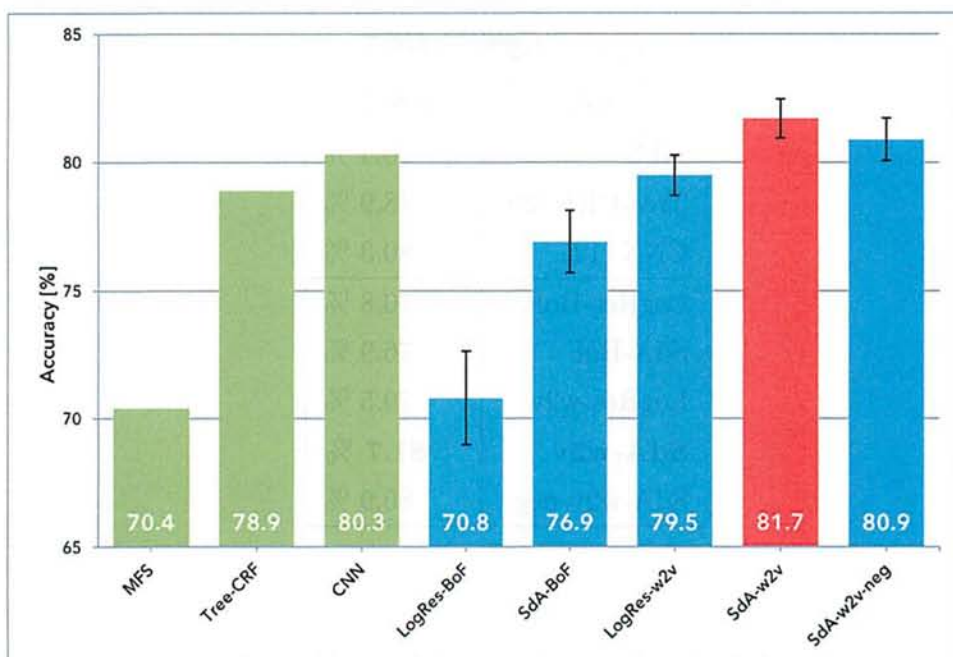


図 4.1 信頼区間と各手法の正解率

表 4.3 上段では、中間層を 3 層、中間ノードを 100 次元で固定して各層のノイズレートの変化に伴う正解率の変化を比較した。ノイズレートが 10 % の時にもっとも高い正解率が得られた。

表 4.3 中段では、中間ノードを 100 次元、ノイズレートを 0 % に固定して、層の数を変化させて正解率の変化を比較した。中間層が 4 層の時にもっとも高い正解率が得られた。

表 4.3 下段では、中間層を 3 層、ノイズレートを 0 % に固定して、中間ノードの次元を変化させて正解率を算出した。中間ノードが 500 次元の時にもっとも高い正解率が得られた。

次に、学習の際に使用した活性化関数及び最適化手法の比較を表 4.4 に示す。

最後に、テストデータを固定して訓練データの増加に伴う正解率の変化を図 4.2 に示す。

表 4.2 評価極性分類精度

手法	正解率
MFS	70.4 %
Tree-CRF [29]	78.9 %
CNN [12]	80.3 %
LogRes-BoF	70.8 %
SdA-BoF	76.9 %
LogRes-w2v	79.5 %
<b>SdA-w2v</b>	<b>81.7 %</b>
SdA-w2v-neg	80.9 %

表 4.3 ハイパーパラメータによる正解率の変化

パラメータ		正解率
ノイズレート	0 %	81.1 %
	<b>10 %</b>	<b>81.5 %</b>
	20 %	81.4 %
	30 %	80.9 %
中間層数	1	80.6 %
	2	80.4 %
	3	81.1 %
	<b>4</b>	<b>81.6 %</b>
	5	81.4 %
	6	81.1 %
中間ノード次元	100	81.1 %
	300	81.2 %
	<b>500</b>	<b>81.3 %</b>
	700	81.2 %

表 4.4 活性化関数及び最適化手法による正解率の変化

手法		正解率
活性化関数	Sigmoid	78.8 %
	<b>ReLU</b>	<b>81.6 %</b>
	tanh	73.7 %
最適化手法	<b>SGD</b>	<b>80.6 %</b>
	AdaGrad	71.9 %
	AdaDelta	78.7 %
	Adam	52.1 %

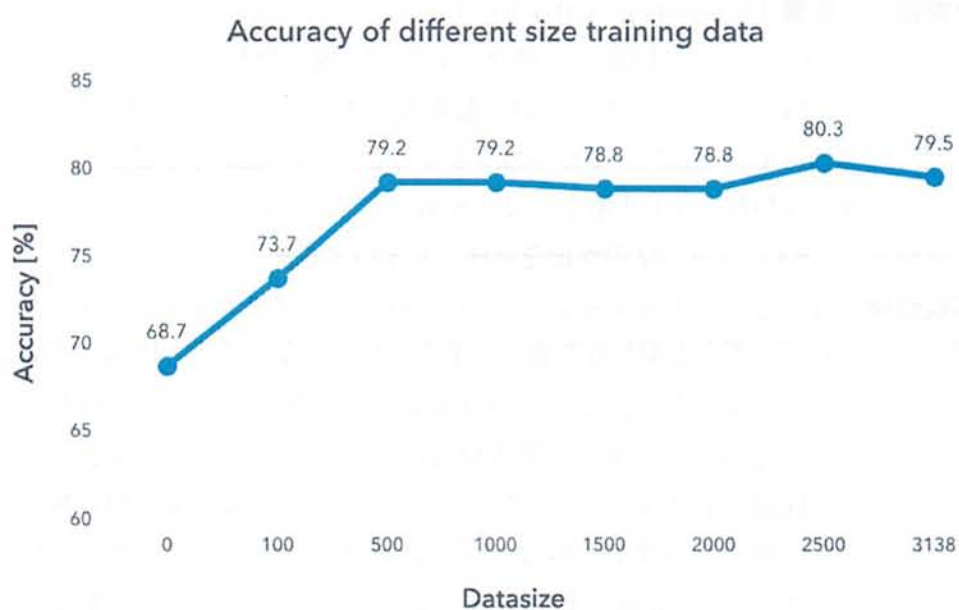


図 4.2 訓練データの増加にともなう正解率の変化



## 第 5 章 考察

本章では前章での実験及び実験により得られた結果と実際の例文（表 5.1）を元に考察を行う。

### 5.1 手法

**Bag-of-Features vs. 単語分散表現** 学習手法を線形分類器（ロジスティック回帰）で固定した場合、正解率は Bag-of-Features では 70.8 % で単語分散表現では 79.5 % であった。SdA で学習した場合、Bag-of-Features は 76.9 % で単語分散表現は 81.7 % であった。これらの結果から、Bag-of-Features よりも単語分散表現を使用した方が正解率は 4.8 ~ 8.7 ポイントもの向上が見られた。このことから、このタスクにおける単語分散表現の貢献は非常に大きいといえる。

**線形分類器 vs. 多層 Denoising Auto-Encoder** 学習に使用する単語ベクトルを Bag-of-Features に固定した場合、正解率は線形分類器では 70.8 % で SdA では 76.9 % であった。単語分散表現を使用した場合、線形分類器は 79.5 % で SdA は 81.7 % であった。このことから、SdA で学習したほうが従来の線形分類器よりも正解率に 2.2 ~ 6.1 ポイントの改善が見られた。したがって、SdA は高い学習性能を発揮したといえる。

**否定表現処理** 表 4.2 から分かるように、同じパラメータでチューニングしたにもかかわらず、否定表現処理を施した手法はしていない手法と比べて正解率が 0.8 ポイント低下した。これは Nakagawa らの報告とは違うものとなった。考えられる原因として、否定表現処理をすることで引き起こされるスパースネス問題が挙げられる。そこでコーパスに含まれる否定表現の数を調べたところ、異なり語数が 326 語で延べ語数は 1,239 語であった。これはコーパス全体の異なり語数の約 3.8 %、延べ語数の約 1.0 % で正解率に大きな影響を及ぼすと考えにくい。

**Convolutional Neural Network** 実験結果より、SdA は 81.7 % で CNN の 80.3 % を 1.4 ポイントで上回った。SdA と CNN はともに全結合型の



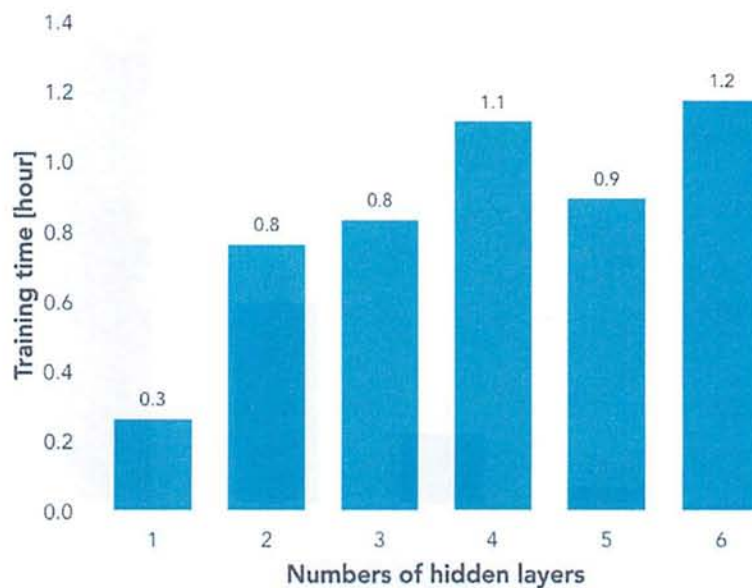


図 5.1 中間層数の変化に伴う学習時間

ニューラルネットワークであり，CNN の分類能力の高さは [12] から分かる．しかし，畳み込んだ素性マップを単純に低い次元の層にマックスプーリングする CNN とは異なり，SdA は深い層においても繰り返し次元圧縮を行うことでより抽象的な表現を獲得できていると考えられる．

## 5.2 パラメータ

表 4.3 にあるように，パラメータを変えることによって得られた結果の考察を行う．

**denoising** ノイズレートが 0 % のまったく denoising を施さない手法（正解率 81.1 %）と比べると，10 % と少しのノイズを与えたほう（正解率 81.5 %）が正解率が 0.4 ポイントの向上が見られた．しかし過剰にノイズを付与すると正解率が下がることも確認できた．これは強すぎるノイズを与えてしまうと入力を復元できなくなるという原因が考えられる．

**中間層** 中間層数が 1 のただの AE（正解率 80.6 %）よりも，多層に積み重ねた手法のほう（正解率 81.6 %）が正解率を 1.0 ポイント改善することが見て

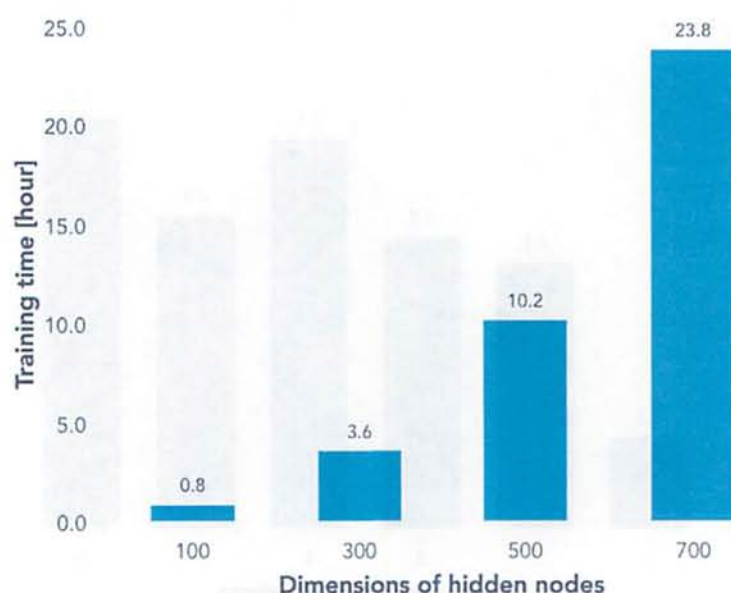


図 5.2 中間ノード次元の変化に伴う学習時間

取れた。これは普通の AE では抽出できないような抽象的な特徴を捉えることができることに起因すると考えられる。

**中間ノード次元** 中間ノードの次元を変化させて実験を行ったが、200 次元の増減で正解率は  $\pm 0.1$  ポイントとわずかにしか変動しなかった。顕著な変化を得るためには更に増減幅を増やす必要があると考えられる。

また、パラメータ（中間層数、中間ノード次元）によつての学習時間の変化をそれぞれ図 5.1 と図 5.2 に示す。

図 5.1 にあるように、中間層の数を増やしていても学習時間の増加は緩やかである。それに対して、中間ノード次元を変化させた場合、図 5.2 のように 200 次元増やすごとに学習時間は倍になることが見て取れる。これは SdA の構造に起因すると考えられる。SdA において前後の層とのノードの結合は全結合になっている。1 層の  $n$  次元ノードのネットワークを考えると、結合の数は  $l \times n \times n = ln^2$  になる。中間層と結合の数の関係は線形だが、中間ノード次元と結合の数は指数的である。そのため、少しの次元数増加でかなりの時間がかかってしまうと考えられる。また、表 4.3 から分かるように、中間ノードを 200 次元増やしても正解率は 0.1 ポイントしか変化しないため、中間ノード次元を増やすよりも中間層の数を増加させたほうがより性能の向上につながる。

## 5.3 活性化関数、最適化手法及びデータサイズによる正解率の変化

表 4.4 を元に活性化関数と最適化手法の考察を行う。

### 5.3.1 活性化関数

従来の多層ニューラルネットワークでは活性化関数に Sigmoid 関数 (式 5.31) や双曲線関数の一種である tanh 関数 (式 5.32) を使用するのが一般的であった。これらの非線形関数は滑らかである上値域が有界であるため、数值的に不安定になりにくく表現力が高い。さらに非線形関数でありながら比較的単純であるため誤差逆伝播法などによる微分の計算も容易である。しかしその一方で、大きい入力値を取った場合に微分係数が小さくなることから出力値が指数的に減少する問題が挙げられる。この問題に対処したのが Rectified Linear Unit (ReLU) またはランプ関数 (式 5.33) と呼ばれる関数である。この関数は値域が有界ではない欠点があるものの、正の値を取るユニットについて勾配が減衰しないため最適化では有利である。また、上記の二つよりも単純であるため、計算速度も速いという利点が挙げられる。これらのグラフを図 5.3 にまとめて表示する。

表 4.4 より、本研究においても ReLU の正解率は他を上回ることが見て取れる。

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5.31)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5.32)$$

$$\text{ReLU}(x) = \max(0, x) \quad (5.33)$$

### 5.3.2 最適化手法

第 3 章でも触れたように、ニューラルネットワークでの誤差を最小化するために複数の最適化手法を試した。その結果は表 4.4 で示している。

最も基本的で一般的な最適化手法として確率的勾配降下法 (SGD) が挙げられる。これは各要素についての平均損失の勾配を用いてパラメータを反復更新していく手

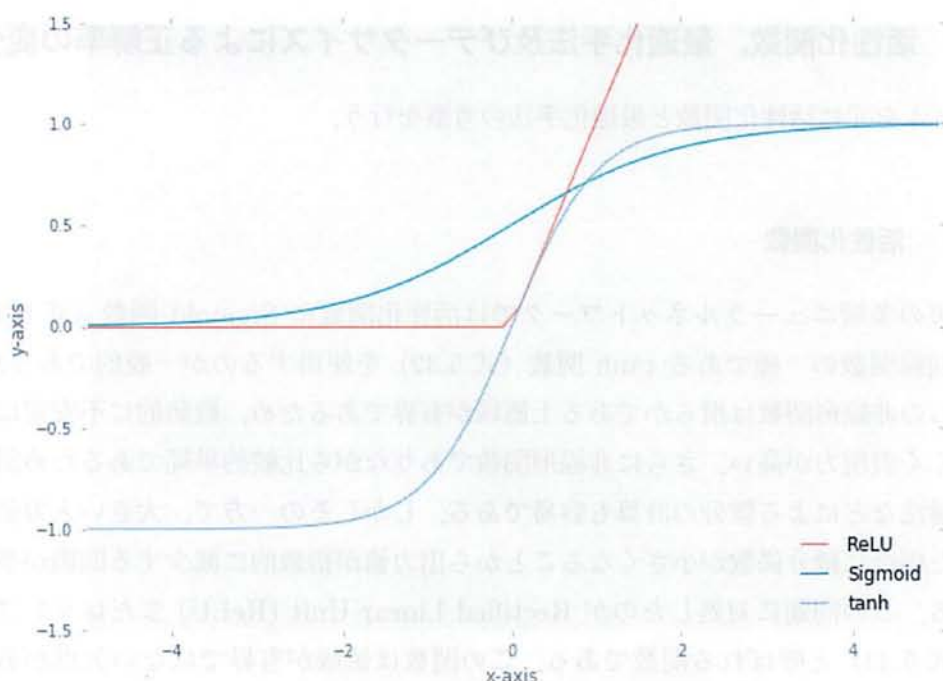


図 5.3 活性化関数

法である。この手法は目的関数が凸関数では場合でも局所解に収束しやすいが、学習率の調節が重要で大きすぎると発散し、小さくすぎると収束まで遅くなる問題が挙げられる。そのため、SGD よりも高い精度で高速に収束する手法が多く改良されている。中でもよく使用されるものとして AdaGrad と AdaDelta, Adam が知られている。

本研究ではこれらの 4 つの最適化手法を使用して比較を行い、その結果を表 4.4 に示している。表から見て取れるように、本来の SGD が最も良い性能を発揮しており、SGD の改良手法である AdaGrad, AdaDelta, Adam はそれを下回っている。また、図 5.4 にて、各最適化手法の AE において 100 Epoch 実行した時の誤差の収束度を示している。ここでは Adam が最も速く収束し、続いて AdaGrad, AdaDelta 最後に SGD が並んでいる。そして収束した際の誤差も Adam が最も小さく、SGD が最も大きい結果となっている。それにもかかわらず、本実験では SGD が最もよい精度であることは当初の予想に反していた。



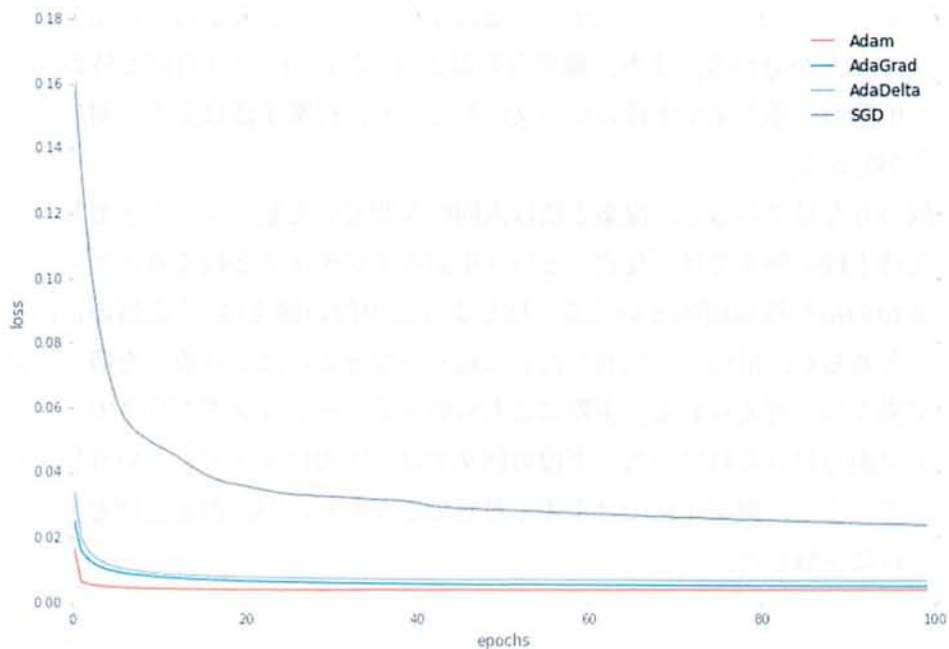


図 5.4 AE において 100 Epoch 実行した際の各最適化手法の収束度

### 5.3.3 データサイズによる正解率の変化

テストデータを固定し，訓練データを約 500 文刻みで増やしていき，それによる正解率の変化を測った．図 4.2 から訓練データが 0 文の時の正解率が最も低く，500 文までは線形に正解率が高くなっていくのが見て取れる．500 文以降は細かい変動はあるもののデータ量の増加に伴い徐々に正解率が高くなっている．このことから，より多くのデータを使用することが正解率の向上に繋がることが言える．

## 5.4 事例分析

事例分析を表 5.1 を参考に行っていく．BoF, Log, AE, Neg, Prop, ans はそれぞれ Bag-of-Features, LogRes, Auto-Encoder ( 1 層の SdA ), 否定処理, 提案手法, 正解ラベルに対応する．値には予測値のポジティブとネガティブを指す P と N が表示される．

一番上の正答例に注目すると，「フジモリズム」といった造語に対しても，BoF

などが苦手としたデータスパースネス問題に提案手法では影響されずに頑健に分類できていることが分かる。また、線形分類器と 1 層の AE では有効な特徴の抽出が困難な中段の二重否定や下段の長い文に対しても、提案手法は上手く対応できていることが分かる。

誤答例の方を見てみると、提案手法は人間的な間違いを犯しているとも見て取れる。例えば上段の例文では「反省」という単語をポジティブと捉えるかネガティブと捉えるかの境界線は曖昧といえる。同じように中段の例文は、「悲劇の再演を防止する」とあるが、最後に「批判した」に続くのでどちらにより重みを置くかは人によって異なると考えられる。実際にこちらのアノテーションタグはネガティブとポジティブ両方付けられていた。下段の例文では「圧力に屈せず」という極性の反転が起きているが、提案手法では上手く対処できなかったが、否定処理を施した手法ではこれに正解した。

表 5.1 事例分析のための正答例と誤答例

正答例						
BoF	Log	AE	Neg	Prop	ans	例文
N	N	N	N	P	P	同 25 日の毎日新聞との単独会見では、貧困率などの細かい数字を挙げて 10 年間の政権の成果を強調し「フジモリズムはペルー全土に根付いている」と胸を張った。
N	P	N	N	P	P	牛で成功したクローン技術を人へ応用するのは難しいことではない。
P	P	P	P	N	N	世界経済は、好調を続ける米国とは対照的に、日本や欧州では回復への足取りが重く、日本も今年上半期まで明確な回復軌道が示されなければ、各国が一層の景気回復策を求めてくる可能性がある。
誤答例						
BoF	Log	AE	Neg	Prop	ans	例文
N	N	N	N	P	N	もう少し配慮した書き方があったかなとも思う」と反省を口にした。
N	N	N	N	N	P	教育省の談話は「歴史教科書は歴史の真相を反映すべきであり、そうしてこそ若い世代に正しい歴史観をもたせ、悲劇の再演を防止できる」と批判した。
P	N	N	P	N	P	同市は圧力に屈せず、この宣言を守り抜いてもらいたい。

第 6 章 おわりに

本研究では日本語の評価極性分類タスクに対して、単語分散表現のひとつである word2vec を用いた大規模コーパスからの単語の表現学習と多層 Denoising Auto-Encoder を適用し、高い精度を得ることができた。本手法では辞書のような語彙知識や複雑なモデル設計、そして素性エンジニアリングなどの専門家による高度な知識を必要としないため、人的コストが低く一般化しやすい利点が存在する。また、本論文では実験していないが、他の言語の学習データを用いることによりその言語に適した評価極性分類も可能である。

本研究の展望として、ポジティブ・ネガティブの二値分類であった本タスクを、怒り・悲しみ・嬉しさなどの多値分類タスクに応用することと、文の分散表現を作成するにあたって Recurrent Neural Network [11] や Recursive Neural Network により大域的な情報の獲得を試みるなどが挙げられる。



# 謝辞

まず、本論文の執筆にあたって、指導教員である小町守准教授に心より感謝の意を表したいと思います。自分の学部4年時に、自然言語処理のしの字も知らずにこの新設された研究室に配属された頃から、大変熱意を持って研究のことに始まり生活のことまで相談に乗っていただき、そして進むべき道筋を示し、導いていただきました。時には厳しく、時には優しくご指導をしてくださったからこそ、今日日の自分がいることは考えるまでもありません。ともに研究について悩んだり、新しい知識に立ち向かったり、大学院入試の問題を解いたり、またはマリオカートを遊んだりした記憶は今もなお鮮明に脳裏に残っています。この研究室だったからこそ、私は自然言語処理に出逢い、遠い夢だった人工知能に取り組み、まだまだ白面郎ではありますが遠い夢のまま終わらせないことができました。また、教育熱心な小町先生のおかげでご自身の多様な体験から生まれた知見や、インターンシップを通して多くの経験を得ることができました。私は、この研究室の第一期生であることに誇りを持っています。これからの、小町研究室のますますの躍進を切に願っています。

また、的確な添削をいただいた高間教授、有意義な議論をさせていただいた山口教授、ともに研究室の楽しい時間を過ごした同期の方々、先輩、後輩そして心を支えてくれた家族に、この場を借りてお礼を申し上げます。充実した三年間をありがとうございました。

## 参考文献

- [1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, Vol. 3, pp. 1137–1155, 2003.
- [2] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of The 29th International Conference on Machine Learning*, pp. 767–774, 2012.
- [3] Yejin Choi and Clarie Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 793–801, 2008.
- [4] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, 2008.
- [5] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 30–42, 2012.
- [6] Cicero Nogueira dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics*, pp. 69–78, 2014.
- [7] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 513–520, 2011.
- [8] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, Vol. 313, No. 5786, pp. 504–507, 2006.
- [9] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever,

- and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [10] Daisuke Ikeda, Hiroya Takamura, Lev-Arie Ratinov, and Manabu Okumura. Learning to shift the polarity of words for sentiment classification. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pp. 296–303, 2008.
  - [11] Ozan Irsoy and Claire Cardie. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 720–728, 2014.
  - [12] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014.
  - [13] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1188–1196, 2014.
  - [14] Andrew McCallum. Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 403–410, 2002.
  - [15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
  - [16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pp. 3111–3119, 2013.
  - [17] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using crfs with hidden variables. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 786–794, 2010.



- [18] Bo Pang, Lee Lillian, and Vaithyanathan Shivakumar. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 79–86, 2002.
- [19] Yohei Seki, David Kirk Evans, Lun-Wei Ku, Hsin-Hsi Chen, Noriko Kando, and Chin-Yew Lin. Overview of opinion analysis pilot task at ntcir-6. In *Proceedings of NTCIR-6 Workshop Meeting*, pp. 265–278, 2007.
- [20] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 151–161, 2011.
- [21] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.
- [22] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, Vol. abs/1503.00075, , 2015.
- [23] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 1555–1565, 2014.
- [24] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, 2008.
- [25] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The*



- Journal of Machine Learning Research*, Vol. 11, pp. 3371–3408, 2010.
- [26] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Advances in Neural Information Processing Systems 26*, pp. 809–817. 2013.
- [27] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pp. 90–94, 2012.
- [28] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems 25*, pp. 341–349. 2012.
- [29] 中川哲治, 乾健太郎, 黒橋禎夫. 隠れ変数を持つ条件付き確率場による依存構造木の評価極性分類. 研究報告自然言語処理 (NL) , Vol. 2009, No. 10, pp. 1–7, 2009.

## 発表リスト

- 張 培楠, 小町 守. 単語分散表現を用いた多層 Denoising Auto-Encoder による評価極性分類. 第 222 回自然言語処理研究会, 2015.
- Peinan Zhang and Mamoru Komachi. Japanese Sentiment Classification with Stacked Denoising Auto-Encoder using Distributed Word Representation. In *Proceedings of The 29th Pacific Asia Conference on Language, Information and Computation*, 2015.