深層学習による論文概要からの キーワード提示に関する研究

林　坂　弘 一 郎

概要

本研究では深層学習やディープラーニングとも呼ばれる多層ニューラルネットワークを用いて論文概要からその論文の主要キーワードを自動提示するための方法を提案する．具体的には論文概要の索引語を入力層，キーワードを出力層とする多層ニューラルネットワークを構築し，論文概要の索引語頻度と文書頻度から得られる TF-IDF 値を入力層に，キーワードを出力層に与える．多数の訓練データをこの多層ニューラルネットワークに与え学習を行う．さらに，新たに与えられたテストデータに対してキーワード提示の推論処理を行う．

キーワード：多層ニューラルネットワーク，深層学習，キーワード提示，PDF データベース，自然言語処理，人工知能

1.　まえがき

筆者 [1] は研究者自身が Web などから収集した学術論文の文献 PDF ファイルを一元管理するための文献 PDF データベースシステムを Perl 言語によって開発した．さらに，筆者 [2] は文献 PDF ファイルを一元管理するだけでなく，複数のユーザが文献情報を共有できる文献 PDF データベースシステムを PHP フレームワークのひとつである Laravel フレームワーク上で新たに開発した．これらのシステムは 3 層クライアント・サーバ方式が採用されており，PDF ファイルを Web ブラウザを利用してシステムにアップロードするだけで全文検索ができるようになるだけでなく，著者や題目，概要，キーワードなどの詳細情報を登録することで，研究者は短時間で効率的に必要な文献を検索，活用できるようになっている．しかしながら，文献ごとにキーワード等の情報を一つひとつ登録することは利用者にとっての大きな負担となっていた．

これまでに自然言語処理に基づいたキーワード自動抽出や主題抽出は様々な手法 [3–8] が提案されている．木本 [3] は単語に関する性質は取り入れず，文章を完全に理解しないでも，キーワードについて文章中，ならびに，シソーラス中での語の特徴を抽出して，その特徴を利用して必要キーワードを抽出する方式と，キーワードの相対的な重要度を評価する方式とを提案した．大澤ら [4] は文書が著者の独自の主張をするために構成されているという仮定に基づき，共起グラフの分割・統合操作によってキーワードを抽出する手法を提案した．永田ら [5] は単語の出現頻度と共起確率などから算出されたスコアに基づき，小学生が情報発信の学習を行う際に，キーワードを適応的に提示することで，発信するメッセージの内容の推敲を促す学習支援システムを提案している．

佐藤ら [6] は Web 上のニュース記事に対する個人ブログの主張を抽出することを目的とした形態素による主張抽出ルールを提案した．近藤ら [7] は Wikipedia の見出し語を興味キーワードとして抽出し，単語の出現頻度に基づく興味キーワードのスコアリング方法を提案した．さらに白井 [8] はマイクロブログ，ニュース記事に代表される文書ストリームデータに対して，ストリーム中での特徴の変動を考慮したマルチラベル分類を提案している．

一方でニューラルネットワークを用いた機械学習に関する研究は様々な分野で行われている．特に Google が機械学習用のライブラリである TensorFlow [9] を 2015 年 11 月に公開したこともあり，近年ではニューラルネットワークの隠れ層を多数にした深層学習やディープラーニングと呼ばれる多層ニューラルネットワークに基づいた画像認識，文字認識，音声認識などの人工知能の研究が盛んに行われている．

本論文では上述したような文献 PDF データベースシステムに文献情報を登録する際の利用者の負担を軽減することを目的として，論文の概要データからその論文の主要なキーワードを自動提示するための方法を提案する．具体的には，概要の索引語を入力層，キーワードを出力層とする多層ニューラルネットワークを構築し，この入力層に論文概要で用いられる索引語の重みを与える．多数の訓練データをこの多層ニューラルネットワークに与えて学習を行うことで，新たに与えられた論文概要データに対するキーワード提示の推論処理を行う．

本論文の構成は次のとおりである．**2.**では論文概要の文書集合とそのキーワードから多層ニューラルネットワークに入力するための特徴データを単語の出現頻度による重み付けによって生成する方法を議論する．**3.**では生成された論文概要に関する特徴データからキーワードを提示するための深層学習の手法を議論する．さらに **4.**では電子情報通信学会論文誌 D の概要とキーワードの特徴データから多層ニューラルネットワークの学習を行い，キーワード提示の推論処理を行った分析例を示す．

2.　索引後の抽出と重み付けによるデータ生成

ここでは，論文の概要とキーワードの組み合わせから深層学習に必要となる特徴データを生成する手順について議論する．

いま，n 個の論文に対するそれぞれの日本語概要（以降は文書と呼ぶ） $D_1, D_2, \cdots, D_n$ があるとし，文書 D_j $(j = 1, \cdots, n)$ には k_j $(= 1, 2, \cdots)$ 個のキーワードが指定されているものとする．まず文書 $D_1, D_2, \cdots, D_n$ を形態素解析によって単語に分割するとともに，分割された単語の品詞を取得する．次に文書集合全体を形態素解析することによって得られた単語のリストからストップワードを削除する．ここでストップワードとは，「する」，「ある」，「こと」，「もの」のように高頻度で文中に出現し，語彙内容に乏しい単語である．さらに品詞を名詞，動詞，形容詞，副詞に限定して意味がありそうな単語だけを抽出し，その重複を取り除く．これによって文書集合全体から抽出された m 個の索引語のリストを $v_1, v_2, \cdots, v_m$ とする．

本論文では索引語 v_i $(i = 1, 2, \cdots, m)$ の文書 D_j における重み d_{ij} を次式 [10] によって決定し，これを次節で用いる論文概要の特徴データセットとする．

$$d_{ij} = \frac{l_{ij} g_i}{n_j}. \tag{1}$$

ここで，l_{ij} は索引語 v_i の文書 D_j における局所的な重みである．局所的な重み l_{ij} には，索引語 v_i の文書 D_j での出現頻度 f_{ij}，すなわち索引語頻度 (term frequency: TF) をそのまま利用し，$l_{ij} = f_{ij}$ とすることができる．しかしながら，索引語頻度による重み付けは出現頻度の高い索引語に過大な重み付けを与える傾向がある [11]．したがって本論文では出現頻度の高い索引語の影響を少なくするために l_{ij} には対数化索引語頻度 (logarithmic TF)

$$l_{ij} = \log(1 + f_{ij}) \tag{2}$$

を利用する．また， 式 (1) の g_i は文書集合全体での索引語 v_i の大域的な重みであり，本論文では文書頻度の逆数 (inverse document frequency: IDF)

$$g_i = \log \frac{n}{n_i} \tag{3}$$

を利用する．なお，n_i は索引語 v_i を含む文書数である．ここでも対数化を行うことで IDF の値の変化を小さくしている [11]．さらに，式 (1) の n_j は文書の長さによる影響を排除するために利用される文書正規化係数であり，本論文ではコサイン正規化

$$n_j = \sqrt{\sum_{i=1}^{m} (l_{ij} g_i)^2} \tag{4}$$

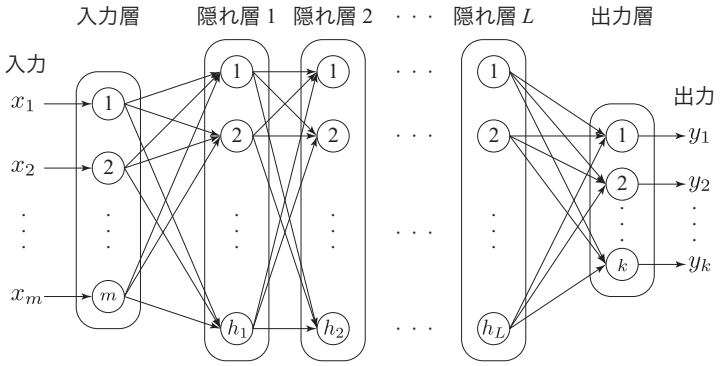

図１　多層ニューラルネットワークの構成図

を使用する．

一方で，文書集合全体で指定されているキーワードの集合を $\kappa_1, \kappa_2, \cdots, \kappa_k$ とする．本論文ではキーワード κ_i が文書 D_j に指定されているかどうかを表すための変数

$$t_{ij} = \begin{cases} 1 & \kappa_i \text{ が } D_j \text{ に指定されている} \\ 0 & \text{それ以外} \end{cases} \tag{5}$$

を利用する．また，文書 D_j に指定されたキーワードの個数を

$$\tau_j = \sum_{i=1}^{k} t_{ij} \tag{6}$$

と書くことにする．

3.　深層学習

ここでは，式 (1) および式 (5) によって生成された論文概要に関する特徴データからキーワード提示を行うための多層ニューラルネットワークについて述べる．

ニューラルネットワークは，脳内に存在するニューロンと呼ばれる神経細胞の構造を計算機上で表現した数理モデルである．図 1には本論文で採用する多層ニューラルネットワークの構成図を示す．

図 1に示すとおり，入力層のニューロン数 m は前述した索引語リストの総数と一致させ，隠れ層 l $(= 1, 2, \cdots, L)$ のニューロン数を h_l とする．また，多層ニューラルネットワーク入力層の出力ベクトルを $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)^{\mathrm{T}}$，隠れ層 l $(= 1, 2, \cdots, L)$ の出力ベクトルを $\boldsymbol{a}^{(l)} = \left(a_1^{(l)}, a_2^{(l)}, \cdots, a_{h_l}^{(l)}\right)^{\mathrm{T}}$ とする．さらに，隠れ層 l でのバイアスを $\boldsymbol{b}^{(l)} = \left(b_1^{(l)}, b_2^{(l)}, \cdots, b_{h_l}^{(l)}\right)^{\mathrm{T}}$

とする．このとき，入力層の出力 $\boldsymbol{x}$ と隠れ層 1 の出力 $\boldsymbol{a}^{(1)}$ の関係は

$$\boldsymbol{a}^{(1)} = f^{(1)}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right) \tag{7}$$

と書くことができる．ここで $\boldsymbol{W}^{(1)}$ は隠れ層 1 における入力信号の重みを表すパラメータ行列

$$\boldsymbol{W}^{(1)} = \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & \cdots & w_{m1}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & \cdots & w_{m2}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1h_1}^{(1)} & w_{2h_1}^{(1)} & \cdots & w_{mh_1}^{(1)} \end{pmatrix} \tag{8}$$

である．

次に，隠れ層 l の出力 $\boldsymbol{a}^{(l)}$ は $l-1$ 層の出力 $\boldsymbol{a}^{(l-1)}$ を用いて

$$\boldsymbol{a}^{(l)} = f^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)}\right), \quad l = 2, 3, \cdots, L \tag{9}$$

と書くことができる．ここで，重み付けパラメータ行列 $\boldsymbol{W}^{(l)}$ は

$$\boldsymbol{W}^{(l)} = \begin{pmatrix} w_{11}^{(l)} & w_{21}^{(l)} & \cdots & w_{h_{l-1}1}^{(l)} \\ w_{12}^{(l)} & w_{22}^{(l)} & \cdots & w_{h_{l-1}2}^{(l)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1h_l}^{(l)} & w_{2h_l}^{(l)} & \cdots & w_{h_{l-1}h_l}^{(l)} \end{pmatrix}, \quad l = 2, 3, \cdots, L \tag{10}$$

である．なお，式 (7) および式 (9) において，活性化関数 $f^{(l)}(\cdot)$ $(l = 1, 2, \cdots, L)$ には シグモイド関数 $\sigma(x)$，ハイパボリックタンジェント $\tanh x$，ReLU (Rectified Linear Unit / 正規化線形関数) $\mathrm{relu}(x)$ のいずれかを利用する．それぞれの活性化関数の定義は次のとおりである．

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{11}$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{12}$$

$$\mathrm{relu}(x) = \max(0, x). \tag{13}$$

さらに，出力層のニューロン数 k は前述したキーワード集合の個数と一致させる．出力層の出力を $\boldsymbol{y} = (y_1, y_2, \cdots, y_k)^{\mathrm{T}}$ とすると，隠れ層 L の出力 $\boldsymbol{a}^{(L)}$ と $\boldsymbol{y}$ との関係も同様に

$$\boldsymbol{y} = f^{(L+1)}\left(\boldsymbol{W}^{(L+1)}\boldsymbol{a}^{(L)} + \boldsymbol{b}^{(L+1)}\right) \tag{14}$$

と書くことが可能である．ここで

$$\boldsymbol{W}^{(L+1)} = \begin{pmatrix} w_{11}^{(L+1)} & w_{21}^{(L+1)} & \cdots & w_{h_L1}^{(L+1)} \\ w_{12}^{(L+1)} & w_{22}^{(L+1)} & \cdots & w_{h_L2}^{(L+1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1k}^{(L+1)} & w_{2k}^{(L+1)} & \cdots & w_{h_Lk}^{(L+1)} \end{pmatrix}, \tag{15}$$

$$\boldsymbol{b}^{(L+1)} = \left(b_1^{(L+1)}, b_2^{(L+1)}, \cdots, b_k^{(L+1)}\right)^{\mathrm{T}} \tag{16}$$

である．また，文字認識のような分類問題では，式 (14) の活性化関数 $f^{(L+1)}(\cdot)$ に対して一般的にソフトマックス関数が利用される．しかしながら，本論文で取り扱うキーワード提示問題では，一つの文書に対して複数のキーワードを提示することを考えるため，活性化関数 $f^{(L+1)}(\cdot)$ には回帰問題に対して一般的に利用される恒等関数を用いることとする．

次に式 (7)，式 (9) および式 (14) で与えた多層ニューラルネットワークに対する学習について議論する．本論文では多層ニューラルネットワークの学習に関して，n_{tr} $(< n)$ 個の訓練データを抽出し，$\boldsymbol{x}$ には式 (1) で求めた重みを入力する．また，学習に用いる指標となる損失関数には 2 乗誤差

$$E = \frac{1}{n_{tr}} \sum_{j=1}^{n_{tr}} \sum_{i=1}^{k} (y_{ij} - t_{ij})^2 \tag{17}$$

を考える．ここで，y_{ij} は文書 D_j の入力データに対して出力層の i 番目のニューロンが出力した値であり，t_{ij} は式 (5) のとおりキーワード κ_i が文書 D_j に指定されているかどうかを示す．学習においては式 (17) の損失関数値を最小にするような重み行列 $\boldsymbol{W}^{(l)}$ $(l = 1, 2, \cdots, L+1)$ とバイアス $\boldsymbol{b}^{(l)}$ $(l = 1, 2, \cdots, L+1)$ を求めることになる．

さらに学習ができれば，n_{te} $(= n - n_{tr})$ 個のテストデータの重み $\boldsymbol{x}$ を入力し，出力 $\boldsymbol{y}$ を得る．これによって，新たに与えられた文書であるテストデータに対してキーワードの提示を行うことができる．

4.　分析例

ここでは，電子情報通信学会論文誌 D の概要とキーワードのデータから多層ニューラルネットワークの学習を行った分析例を示す．

4.1　データセットの生成

本論文では電子情報通信学会論文誌 D の 2007 年 1 月号から 2017 年 12 月号に掲載され，かつキーワードが指定された 2556 編の論文および研究速報から次のような手順で分析用データを生成した．全 2556 編の文書に指定されたキーワードは 7315 種類であった．これらのキーワードについて指定された論文数を調査したところ表 1に示すとおりとなった．表 1は，2 種類のキーワード（「音声認識」および「FPGA」）が 28 編の論文で指定されており，58 種類のキー

表 1　指定された論文数とキーワード数

指定論文数	28	23	19	18	17	16	15	14	13	12	11	10
キーワード数	2	2	1	3	3	5	2	5	9	5	12	9
累積値	2	4	5	8	11	16	18	23	32	37	49	58

指定論文数	9	8	7	6	5	4	3	2	1
キーワード数	8	12	29	44	65	131	255	773	5978
累積値	66	78	107	151	216	347	602	1375	7353

表 2　生成したデータセット

Dataset	指定論文数	キーワードの種類 (k)	文書数 (n)	索引語数 (m)	訓練データ数 (n_{tr})	テストデータ数 (n_{te})
DS-10	≥ 10	58	698	5595	600	98

ワードが 10 編以上の論文で指定されていることを意味している．その一方で，表 1から 5978 種類のキーワードは 1 編の論文でしか指定されていないキーワードであることも分かる．

本論文では特定のキーワードが指定されている文書数が 10 編以上となるようなキーワードを抽出し，それらのキーワードが指定された文書を抽出することで表 2に示すデータセットを次の手順で生成した．生成したデータセットでは 10 編以上の論文で指定されているキーワードの種類は $k = 58$ であったため，この 58 種類のいずれか一つでも指定している論文を抽出したところ $n = 698$ となった．この 698 の論文の概要について MeCab [12] による形態素解析を実行し，ストップワードの削除と品詞による単語のフィルタリングを行った．この結果 $m = 5595$ の索引語リストが生成された．さらに，単語に分割されたそれぞれの概要について式 (1) の重みを算出することで分析用データを生成した．

4.2　深層学習

表 2に示した 698 件の分析用データは，$n_{tr} = 600$ 件の訓練データと $n_{te} = 98$ 件のテストデータにランダムに分割する．すなわち，600 件の訓練データでニューラルネットワークの学習を行い，98 件のテストデータでニューラルネットワークの予測精度を検証する．なお，分析プログラムは TensorFlow [9] を用いて Python 3 によって記述し，NVIDIA 社 [13] の GPU (GeForce 1070Ti) を搭載した計算機上で実行した．

学習では式 (17) で与えられる損失関数を最小にする重みとバイアスを決定する．この際，重

みとバイアスに関するパラメータの初期値はランダムに与え，TensorFlow の勾配降下法による最適化アルゴリズムを利用する．さらに，学習にはミニバッチを利用する．すなわち，n_{tr} 個の訓練データを b (>1) 個のミニバッチに分割し，分割されたそれぞれのミニバッチを用いて最適化を繰り返す．この際，最適化を b 回繰り返せばすべての訓練データが一度利用されたことになる．なお，この b 回の繰り返しをエポック (epoch) と呼び，本論文では予備実験の結果，$b=4$ として学習を行った．

4.3　学習結果の考察

本研究では多層ニューラルネットワークの学習を行うに当たり，隠れ層の数 $L=1,2,3,4$ に対して，隠れ層における活性化関数に式 (11)–(13) で与えたシグモイド関数，ハイパボリックタンジェント，正規化線形関数を利用し，活性化関数ごとに隠れ層におけるニューロン数を変化させた．なお，ニューロン数はすべての隠れ層で等しくなるよう，$h=h_1=\cdots=h_L$ とした．ランダムに抽出した異なる 100 通りの訓練データ 600 件の組み合わせについてミニバッチで学習を行い，エポックごとに損失関数値を算出した．算出された損失関数値の平均値の推移を図 2–4に示す．図 2–4より，エポックの初期では損失関数値の平均値が振動することがあるが，エポックが進むにつれて損失関数値の平均値が小さくなっている，すなわち，学習が進んでいることが読み取れる．また，図 2および図 3より，シグモイド関数やハイパボリックタンジェントの場合は $h=50$ や $h=2500$ のときには学習が進まない一方で，$h=100$ や $h=250$ のときには学習が進んでいる様子が確認できるなど，h の選択による学習の進み方に大きなばらつきが存在する．これに対して，図 4の正規化線形関数の場合は $h=50$ のときを除いて，$h=2500$ であっても学習が進んでおり，さらに h の選択による学習の進み方に関するばらつきが比較的小さいことが分かる．

図 5–7には訓練データに対するエポックごとの提案キーワード正答率の推移を示す．なお，ここでは，文書 D_j に指定されたキーワードの個数 τ_j が既知であるとして，式 (14) によって得られたニューラルネットワークの出力から出力値 y_i $(i=1,\cdots,k)$ の大きな順に τ_j 個をキーワードとして提示することを考えた．図 5–7より，隠れ層のニューロン数 h を適切に選ぶことができれば学習が進むことで訓練データには 100% に近い精度でキーワードを提示できることが読み取れる．さらに，隠れ層の数を固定して図 5–7を比較すると，テストデータに対する正答率が最も早く上昇し収束するのは活性化関数に正規化線形関数を採用したときであることも読み取れる．

一方で，図 8–10にはテストデータに対するエポックごとの提案キーワードの正答率を示す．

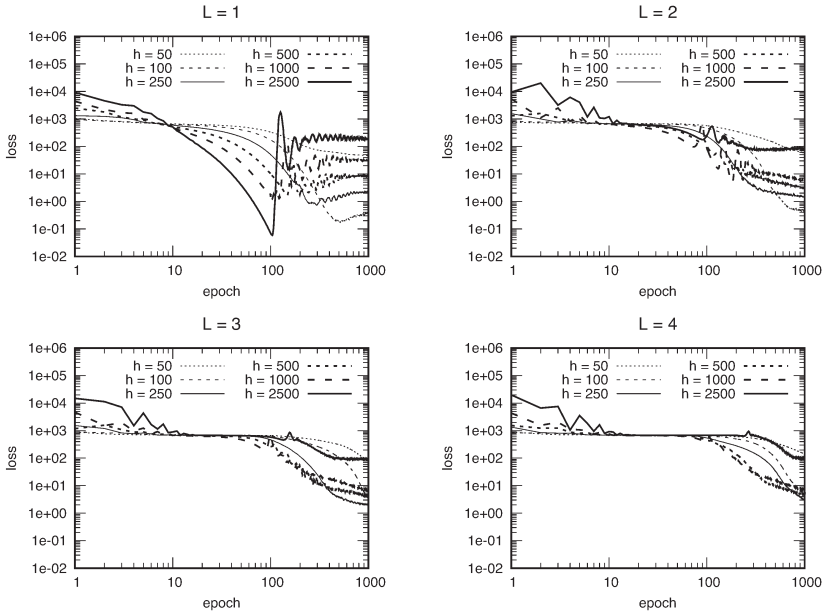

図２　損失関数値の平均値の推移（シグモイド関数）

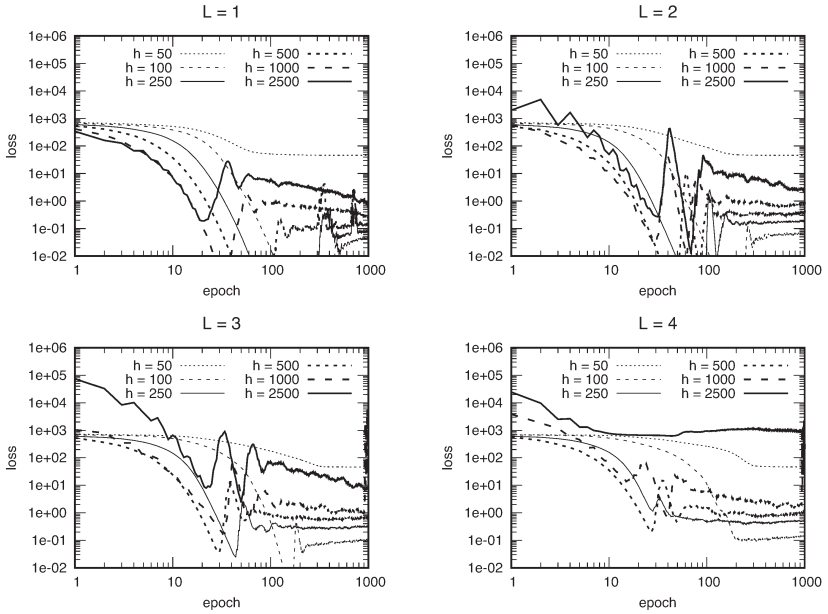

図３　損失関数値の平均値の推移（ハイパボリックタンジェント）

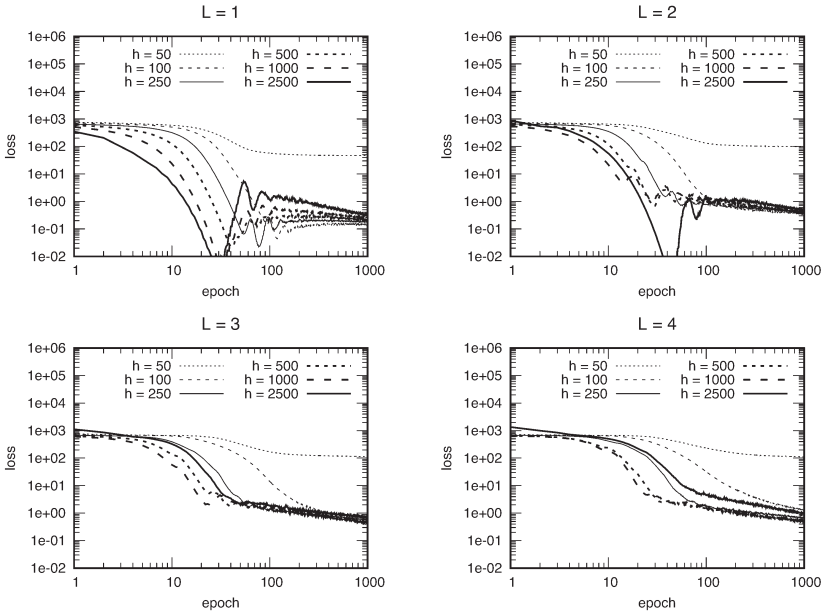

図４　損失関数値の平均値の推移（正規化線形関数）

図 8–10から h の選択によるばらつきは訓練データと比べて大きくなるが，h を適切に選べば学習が進み 55% から 58% 程度の精度でキーワードを提示できることが分かる．また，多くのケースにおいて $h = 250$ または $h = 500$ のときに正答率が高いことが分かる．

次に，テストデータにおける指定キーワードの個数 τ_j が未知であるものとして，ニューラルネットワークの出力値 y_i がしきい値 α 以上であるときに，キーワード κ_i を提示することを考えた．表 3–5にはテストデータにおけるしきい値 α の変化に対する正答率の変化を示す．ここで，3 種類の活性化関数について，後述する 2 種類の正答率がどちらも 50% を超え，かつ，正答率の合計が最大になるような L と h の組み合わせを選択して表 3–5に掲載した．つまり，活性化関数にシグモイド関数やハイパボリックタンジェントを採用した場合には，隠れ層の数を $L = 2$ とし，隠れ層ごとのニューロン数を $h = 500$ としたときにキーワード提示の精度が最もよく，正規化線形関数では，$L = 4, h = 250$ のときに精度が最も良いことになる．

表 3–5においてテストデータ中の指定キーワード総数はそれぞれ，11196, 11193, 11192 である．これらの表より，しきい値 α が小さいとキーワード提示数が多くなり，α を大きくすると提示数が減少し，$\alpha = 0.2$ でニューラルネットワークによって提示されたキーワード数がテストデータの指定キーワード数に最も近くなる．

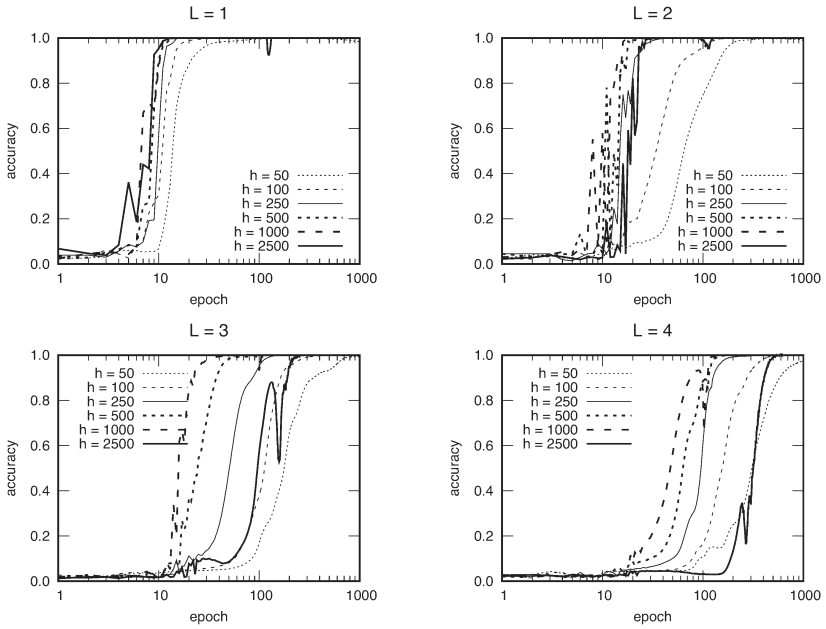

図 5　訓練データに対する正答率の推移（シグモイド関数）

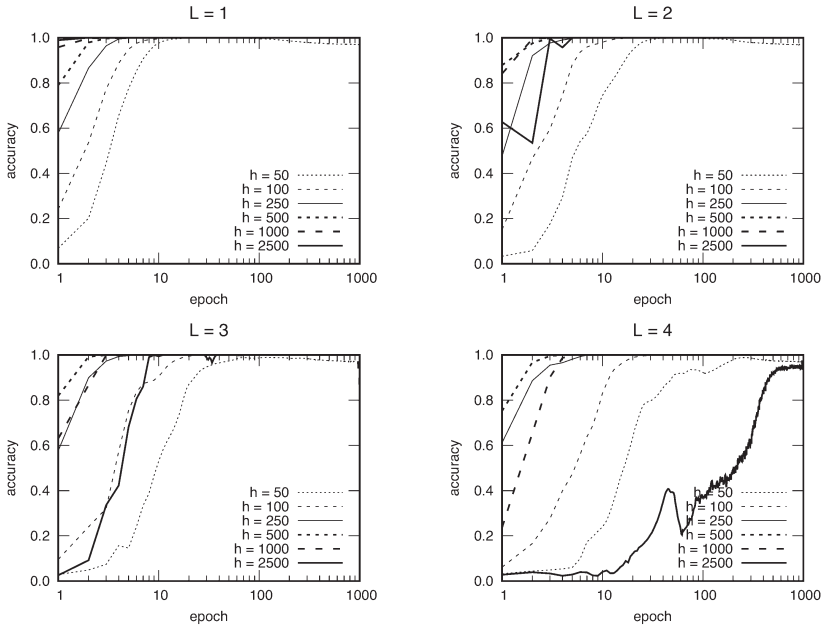

図 6　訓練データに対する正答率の推移（ハイパボリックタンジェント）

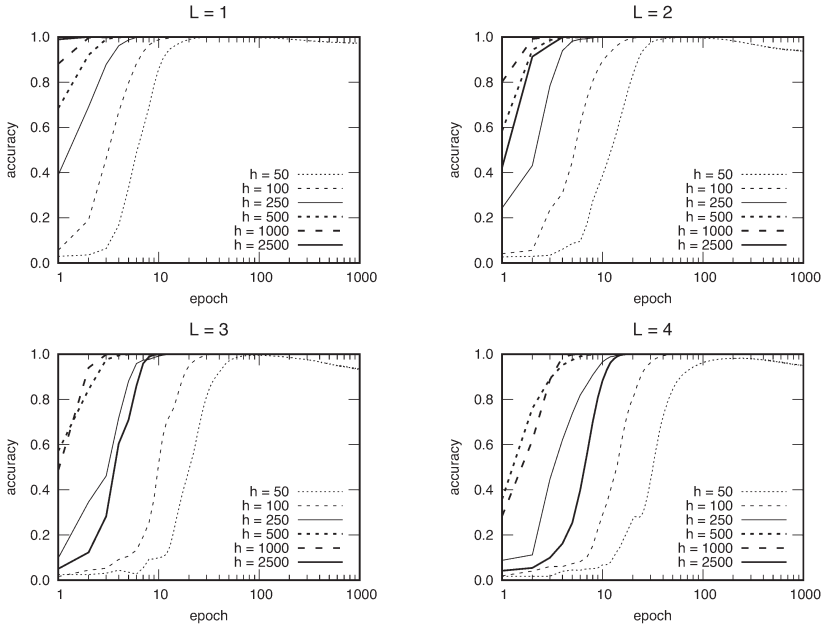

図７　訓練データに対する正答率の推移（正規化線形関数）

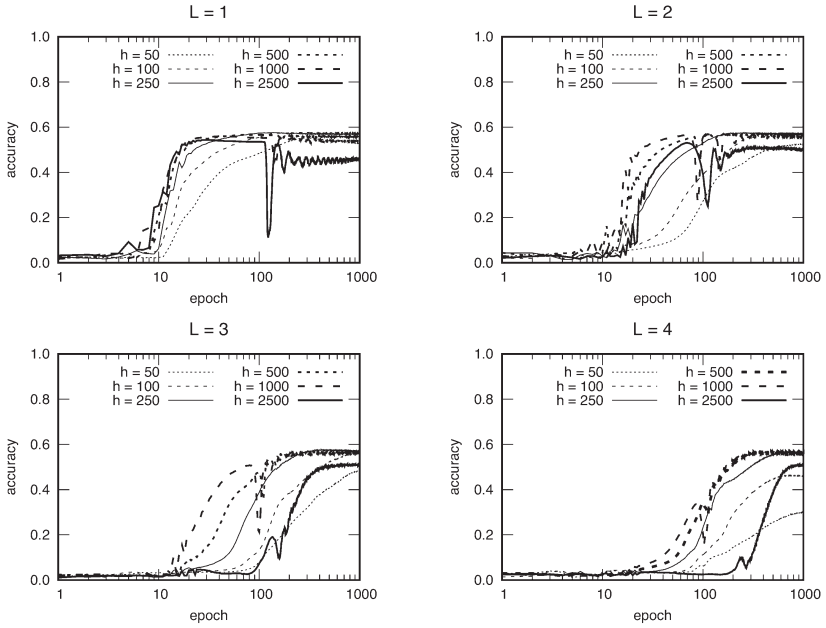

図８　テストデータに対する正答率の推移（シグモイド関数）

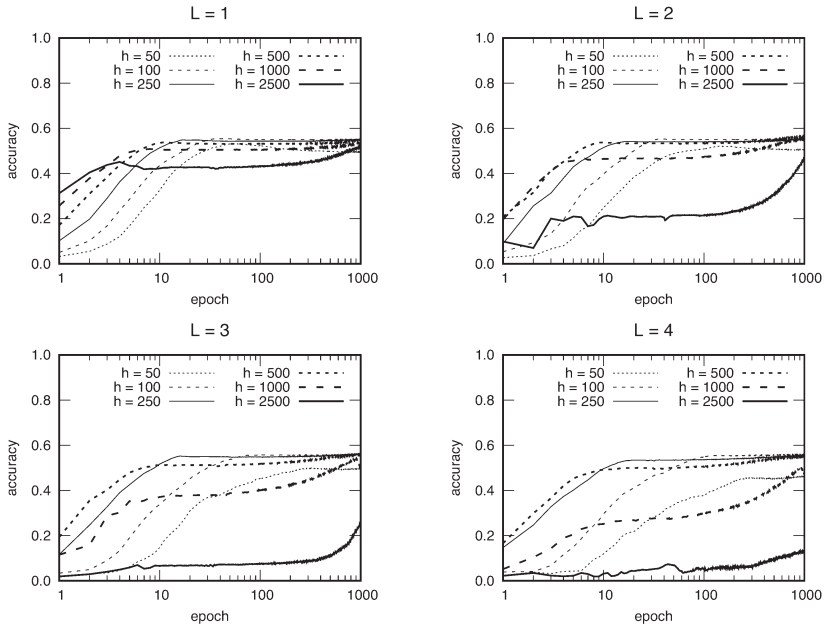

図 9　テストデータに対する正答率の推移（ハイパボリックタンジェント）

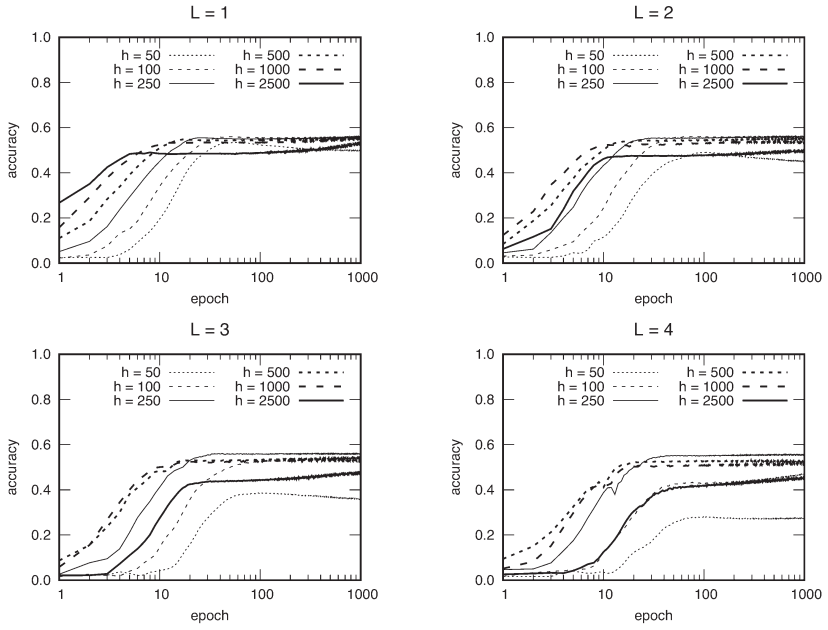

図 10　テストデータに対する正答率の推移（正規化線形関数）

表３　しきい値 α に関する正答率の変化 （シグモイド関数，$L = 2, h = 500$）

α	提示数	正答数	誤答数	正答率 (1)	正答率 (2)
0.1	36213	8308	27905	22.94%	74.21%
0.2	10873	6109	4764	**56.19%**	**54.56%**
0.3	5379	4216	1163	78.38%	37.66%
0.4	3108	2768	340	89.06%	24.72%
0.5	1759	1637	122	93.06%	14.62%
0.6	911	865	46	94.95%	7.73%
0.7	457	443	14	96.94%	3.96%
0.8	162	152	10	93.83%	1.36%
0.9	55	54	1	98.18%	0.48%

表４　しきい値 α に関する正答率の変化 （ハイパボリックタンジェント，$L = 2, h = 500$）

α	提示数	正答数	誤答数	正答率 (1)	正答率 (2)
0.1	52970	8525	44445	16.09%	76.16%
0.2	12487	6420	6067	**51.41%**	**57.36%**
0.3	5878	4561	1317	77.59%	40.75%
0.4	3484	3066	418	88.00%	27.39%
0.5	1952	1795	157	91.96%	16.04%
0.6	956	906	50	94.77%	8.09%
0.7	499	485	14	97.19%	4.33%
0.8	195	181	14	92.82%	1.62%
0.9	53	51	2	96.23%	0.46%

表５　しきい値 α に関する正答率の変化 （正規化線形関数，$L = 4, h = 250$）

α	提示数	正答数	誤答数	正答率 (1)	正答率 (2)
0.1	30341	7744	22597	25.52%	69.19%
0.2	9993	5888	4105	**58.92%**	**52.61%**
0.3	5609	4441	1168	79.18%	39.68%
0.4	3528	3123	405	88.52%	27.90%
0.5	2133	1938	195	90.86%	17.32%
0.6	1262	1164	98	92.23%	10.40%
0.7	625	593	32	94.88%	5.30%
0.8	245	233	12	95.10%	2.08%
0.9	97	95	2	97.94%	0.85%

また，各表において正答率 (1) はニューラルネットワークによって提示されたキーワード総数を分母として求めた正答率であり，正答率 (2) はテストデータで指定されたキーワード総数を分母として計算した正答率である．これらの表より，正答率 (1) は α を大きくすると上昇する，すなわち，提示されるキーワード数は減少するものの，提示されたキーワードはより正確なものとなることを意味している．一方で，正答率 (2) は α を小さくすれば上昇する．すなわち，提示されるキーワードに誤答が含まれているものの，テストデータで指定されたキーワードをより網羅できるようなることを意味している．さらに，表 3–5において，$\alpha = 0.2$ のときに正答率 (1)，正答率 (2) ともに 50% を超えていることが読み取れる．

5.　むすび

本論文では，文献 PDF データベースにおける文献情報の登録作業に関わる利用者負担を軽減することを目的として，論文の概要データからその論文の主要なキーワードを自動提示するための深層学習を提案した．本論文では概要の文章を形態素解析によって単語に分割し，その出現頻度に基づいた索引語の重みを算出した．さらにこの重みを多層ニューラルネットワークに入力し学習を進めることで，新たな論文概要に対してキーワードを提示する手法を提案した．さらに電子情報通信学会論文誌 D の概要データを用いた分析によって 55% 以上の精度でキーワードを提示できることを示した．

本論文では多層ニューラルネットワークの学習において，パラメータの初期値はランダムに与え，勾配降下法による最適化を利用した．今後はデータ生成における索引語に関する重みデータの生成方法，さらに深層学習におけるパラメータ初期値の分布，勾配降下法における収束判定，ミニバッチのサイズなどに関する検討を行い，提案手法の一層の精度向上を図る．さらに，著者情報や論文タイトルなどの詳細情報を本文から自動的に抽出する技術へと発展させたい．

参考文献

[1] 林坂弘一郎：研究者向け文献 PDF データベースシステムの開発, 神戸学院大学経営学論集, Vol. 13, No. 1, pp. 19–42 (2016).

[2] 林坂弘一郎：共有可能な PDF データベースシステムの開発, 日本経営システム学会誌, Vol. 35, No. 1, pp. 43–50 (2018).

[3] 木本晴夫：日本語新聞記事からのキーワード自動抽出と重要度評価, 電子情報通信学会論

文誌 D, Vol. J74-D-I, No. 8, pp. 556–566 (1991).

[4] 大澤幸生, Benson, N. E., 谷内田正彦：KeyGraph: 語の共起グラフの分割・統合によるキーワード抽出, 電子情報通信学会論文誌 D, Vol. J82-D-I, No. 2, pp. 391–400 (1999).

[5] 永田亮, 須田幸次, 掛川淳一, 森広浩一郎, 正司和彦：小学生を対象としたメッセージ推敲のための適応型キーワード提示システム, 電子情報通信学会論文誌 D, Vol. J91-D, No. 2, pp. 200–209 (2008).

[6] 佐藤大輔, 中川博之, 田原康之, 大須賀昭彦：閲覧中のニュース記事に対するブログ記事から主張を抽出して提示するシステムの提案, 電子情報通信学会論文誌 D, Vol. J94-D, No. 11, pp. 1773–1782 (2011).

[7] 近藤光正, 中辻真, 田中明通：Wikipedia に基づく Web 閲覧履歴からの潜在的興味キーワード抽出, 電子情報通信学会論文誌 D, Vol. J96-D, No. 5, pp. 1199–1211 (2013).

[8] 白井匡人, 三浦孝夫：トピックモデルに基づく文書ストリームのマルチラベル分類, 電子情報通信学会論文誌 D, Vol. J99-D, No. 4, pp. 392–402 (2016).

[9] TensorFlow, `https://www.tensorflow.org/`（2018 年 7 月 17 日確認）.

[10] Manning, C. D., Raghavan, P. and Schütze, H.: *Introduction to Information Retrieval*, Cambridge University Press (2008).

[11] 北研二, 津田和彦, 獅々堀正幹：情報検索アルゴリズム, 共立出版 (2002).

[12] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis, in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vol. 4, pp. 230–237 (2004).

[13] NVIDIA, `http://www.nvidia.com/page/home.html`（2018 年 7 月 17 日確認）.