DEIM Forum 2015 D5-6

ペアワイズ分類における深層学習の検証

古堂　和音† 　佐藤　晴彦 † 　小山　聡 † 　栗原　正仁 †

† 北海道大学院情報科学研究科　〒 060-0814 札幌市北区北 14 条西 9 丁目
E-mail: †{f-chord.133211,haru,oyama,kurihara}@complex.ist.hokudai.ac.jp

あらまし　本研究では，ペアワイズ分類に Dropout 正則化付き深層学習を用いた際の性能について検証する．ペアワイズ分類は与えられた二つのデータ間にある関係が成り立つか否かを判定する問題であり，その代表的な例として著者同定等のエンティティ解決問題がある．従来，ペアワイズ分類においては，異なる素性の組み合わせを計算するカーネルを用いた SVM による方法等，機械学習によって分類器を獲得する方法が提案されている． 一方近年，様々な問題に対して，深層学習を用いる方法が SVM 等の従来の方法を上回る分類精度を示すことが報告されている．そこで本研究では，ペアワイズ分類に対して深層学習を用いる方法を提案する．その際，特に，スパースなペアデータを対象とする深層学習における Dropout 正則化の有効性について検証する．

キーワード　ペアワイズ分類, エンティティ解決，深層学習, ニューラルネットワーク, 正則化

1. はじめに

現代のウェブ時代には，正体の分からない人物のアイデンティティの推定や二つのエンティティのマッチングといったことが重要になる．そのような問題を解決する分類がペアワイズ分類である．ペアワイズ分類は，与えられた二つの例が同じクラスに属するか否かを判定する分類である．顕著な例として同姓同名の問題があり，ユーザーが情報発信者の信頼性を判定したり，サービス提供者が適切なサービスを提供するという点でも重要な問題である．エンティティのマッチングや重複発見は，データベースのコミュニティで長い間研究されてきたが，近年はこれを機械学習を用いて行う方法が考えられている．ある二つの例の組が与えられたとき，それら二つの例に同じ素性が出現しているかどうかで，二つのエンティティを比較する方法がある．この方法は，同じクラスに属する二つの例が共通の素性を多く含む場合は有効であるが，共通な素性が僅かしか持たない場合は，判断することは難しくなる．例えば，同姓同名の論文著者マッチングの問題では，二つの例間に共通な素性（単語）を僅かしか持たず，上記のような問題が起こる (図 1)．この問題を回避する方法は，二例間のすべての素性の組を考慮する方法である．小山ら [1] による異なる例からの素性の組み合わせを考慮したペアワイズ分類では，カーネル SVM により，二例間の素性組による高次元性を回避しこの問題に対処している．一方で，近年，深層学習が様々な分野で注目を集めている．深層学習は，ニューラルネットワークを大規模に拡張する方法であり，特に，画像認識や音声認識等の分野では SVM 等の既存の学習器を大きく上回る分類精度を示すことが報告されている．しかし，ペアワイズ分類に対して深層学習を扱った研究は少なく，まだ十分に検討されていないのが現状である．ペアワイズ分類は重複発見やエンティティのマッチング，更には一般のクラスタリングにおいても重要な要素技術であり，この問題に対し深層学習を検証することは興味深い．本研究の目的は，ペアワイズ分類に対し深層学習を用いる方法を提案し，その性能を検証することである．また，深層学習に用いる Dropout 正則化や，Adagrad,Adadelta 最適化の有効性について検証する．

2. ペアワイズ分類

2.1 定　式　化

ここではペアワイズ分類の定義と定式化を行う．ペアワイズ分類は二つの例が与えられた時，それらが同じクラスか否かを分類する問題である．また，二つの例の組 $\boldsymbol{x}^{\boldsymbol{\alpha}}$ と $\boldsymbol{x}^{\boldsymbol{\beta}}$ からなる新たな素性をペアインスタンスと呼ぶ．ペアワイズ分類は，このペアインスタンスが同じクラスに属するか否かを判別する問題と定義される．これは以下のように定式化できる．

$$f(\boldsymbol{x}^{\boldsymbol{\alpha}}, \boldsymbol{x}^{\boldsymbol{\beta}}) = \begin{cases} 1 & (\boldsymbol{x}^{\boldsymbol{\alpha}}, \boldsymbol{x}^{\boldsymbol{\beta}}\text{が同じクラスに属するとき}) \\ -1 & (\text{それ以外のとき}) \end{cases}$$

2.2 元のデータからのペアインスタンスの作成

ペアワイズ分類に用いる類似度を固定，もしくは人手で設計することは困難である．したがって，これを機械学習によって獲得することを考える．多くの場合は，母集団から二例 $\boldsymbol{x}^{\boldsymbol{\alpha}}, \boldsymbol{x}^{\boldsymbol{\beta}}$ をサンプルしペアインスタンスを作成し，それらが同一クラスかどうかを判別して人手でラベルを与える．そして，ペアインスタンスとラベルの組みを学習データとして，機械学習器を学習させる．

2.3 類似度との関連

ペアワイズ分類と類似度との関連は深い．分類器の出力を $f(\boldsymbol{x}^{\boldsymbol{\alpha}}, \boldsymbol{x}^{\boldsymbol{\beta}}) \in [0, 1]$ のような連続値とすると，類似度を出力する関数と考えることもできる．また，この類似度に対して閾値を設定してやれば二値の分類器として使える．また二値の分類器も，連続値で出力出来る場合が多く，これを類似度と考えることもできる．

2.4 関 連 研 究

小山ら [1] は，二例間のデカルト積への仮想的な写像を行うカーネル関数を定義し，これと SVM との組み合わせによるペ

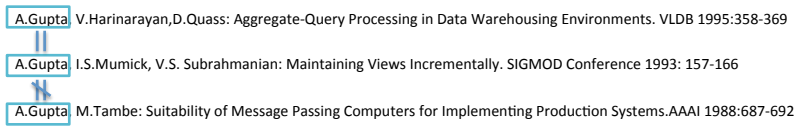

図 1　著者のマッチング

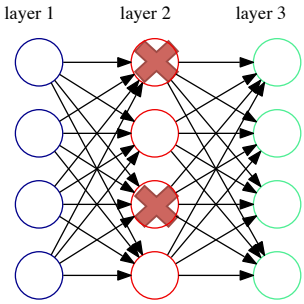

図 2　Dropout は一定確率で中間層の出力を 0 とする

アワイズ SVM を提案している．本研究では，学習結果のベースラインとしてこの手法との比較も行う．

3. 深層学習

深層学習は，既存のニューラルネットワークを多層に拡張したものである．それまで，過学習や勾配消失問題 (注1) により，多層に重ねたニューラルネットワークは学習が困難とされてきた．これらの課題は，学習手法の改善や計算機性能の向上により克服され，近年，深層学習が注目を集めるに至った．深層学習には，様々なモデルが存在する．以前までは，StackedAutoEncoder [2] 等，多層ニューラルネットワークに対し，誤差逆伝搬法に先立って事前学習を行う手法が主流であったが，近年，後述する ReLU 活性化関数や最適化手法の改良により，事前学習が必要ないという報告がされている [3]．したがって，今回用いる深層学習は，事前学習を行わず，そのもとで，Dropout 正則化，Adagrad,Adadelta 正則化の有効性を確認する．

3.1 Dropout

Dropout [4] はニューラルネットワークにおける正則化である．深層学習を含むニューラルネットワークは，過学習を起こしやすい．特に深層学習器は，通常のニューラルネットワークより表現力が高く，過学習を抑えるためには正則化が必要になる．Dropout は，学習の過程において，中間層の出力をある確率で 0 とする正則化手法である．これは，任意の n 次元中間層の出力に対し，以下のようなマスク関数

$$\boldsymbol{mask} = [m_0, m_1, ..., m_n]$$
$$p(m_i = 0) = \pi(0 < \pi \leqq 1.0)$$

をかけることで実装可能である．これによってネットワーク全体の共適応 (co-adaptation) を防ぎ，指数的な数の潜在部分ネットワークを学習する．学習後のニューラルネットワークは，これらの部分ネットワークの出力を混合した形で出力する効果をもたらす．Dropout の欠点として，学習の更新回数が増え，学習が進みにくくなることがあげられる．この欠点を克服した FastDropout [5] と呼ばれる手法も提案されている．

3.2 ReLU 活性化関数

Rectified Linear Unit 関数 (ReLU 関数) は，しきい値以下に対しては，0 を，しきい値以上では，入力値を出力する関数であり，

$$f(\boldsymbol{x}) = max(0, x) \tag{1}$$

のように書ける．通常，ニューラルネットワークの活性化関数としては，シグモイド関数が用いられることが多い．シグモイド関数を用いた深層学習器は，誤差逆伝搬において，入力層側での勾配が消えてしまう問題 (勾配消失) を抱えている．ReLU 活性化関数は，勾配消失を起こしにくく，かつネットワークの出力のスパース性を誘導する効果を持つ [3].

3.3 最適化

ここでは，近年有効性が報告されている Adagrad および Adadelta の二つの最適化手法を説明する．

3.3.1 Adagrad

ニューラルネットワークにおける誤差逆伝搬法において，各ステップにおける学習率の設定は重要であり，学習後のニューラルネットワークの汎化性能に大きな影響を与える．一般に学習率は，学習データに対し適切に定める必要がある．大きな学習率から開始し，学習が進むにつれ学習率を下げていく荷重減衰と呼ばれる方法等がある．しかし，学習率の初期値や減衰のスケジュールは人手で決定する必要がある．こういったハイパーパラメータの設定を減らし，学習率を過去の勾配情報から決定し，その時点での適切な学習率を得ようとする方法が Adagrad [6] である．パラメータ θ のステップ t における更新量は以下のように表される．

$$\Delta\theta_t = -\frac{\eta}{\sqrt{\sum_{\tau=1}^{t} g_\tau^2}} \tag{2}$$

ここで，η は事前に定めた学習率である．Adagrad では，既定の学習率を過去の勾配のＲＭＳ（二乗平均平方根）で割った値をステップでの更新量とする．

3.3.2 Adadelta

adagrad は過去の勾配情報に従うことで，適切な学習を可能

(注1)：多層ニューラルネットワークに対する誤差逆伝播学習において，誤差関数の勾配が入力層側で 0 に近い値になってしまい，その層での学習が困難になる現象

にする方法である．一方で，依然としてハイパーパラメータ η の設定が必要である．このハイパーパラメータの更新を不要にしたものが，adadelta である．更新式は以下のように示される．

Algorithm 1 Adadelta [7]

Require: $Initial Parameter x_1$

$Initialize Accumulation Variables E[g^2]_0 = 0, E[\Delta x^2]_0 = 0$

for $t = 1$ to T **do**

$Compute Gradient : g_t$

$Accumulate Gradient : E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2$

$Compute Update : \Delta x_t = \frac{-RMS[\Delta x]_{t-1}}{RMS[g]_t} g_t$

$Accumulate Updates : E[\Delta x^2]_t = \rho E[\Delta x^2]_{t-1} + (1-\rho)\Delta x_t^2$

$Apply Update : x_{t+1} = x_t + \Delta x_t$

end for

4. 評価実験

4.1 実験データとプログラム

実験データは DBLP(注2) からの引用データセットを使った．各データは，論文 ID と著者 ID とタイトルに出現する単語を表す疎行列で表現されており，それらを一旦密行列に変換してから，以下に従う方法でペアインスンタンスを作成し実験を行った．

4.1.1 ペアインスタンスの作成

ペアインスタンスの作成は，以下の方法で行った．まず，データ集合を 2 等分し，それぞれを学習データの母集団，テストデータと検証データの母集団とする．そして，各母集団の中で，可能なすべての組み合わせを作り，ペアインスタンスとした．ペアインスタンスの作成法は，二つの例を結合したものとした．また，学習結果がペアインスタンスの構成順序に依存しないように，構成順序を逆にしたペアインスタンスも学習データとした．各データセットの大きさや，データ中に現れる素性数を表 1 に示す．また，実際の学習の際には正例と負例のバランスをとるようにした．

省略された名前	論文数	学習データ数	テストデータ数	素性数
J.Anderson	178	2936	3916	1890
A.Gupta	398	8080	19701	3444
M.Sato	157	3003	3081	1724
J.Smith	389	5980	18915	3652
K.Tanaka	176	3560	3828	1832
J.Mitchell	268	10892	8911	2106

表 1　データセットの大きさ
学習データにはペアインスンタンスの構成順序を逆にしたものも含まれる

(注2)：http://www.informatik.uni-trier.de/ ley/db/

4.2 深層学習の設定

層数の違いにより識別性能の差を見るため，通常のニューラルネットワークに加え，様々な層数のニューラルネットワークで実験を行った．それぞれのユニットにおける活性化関数は，出力層以外を ReLU とし，出力層はソフトマックス関数

$$y_j = \frac{exp(\boldsymbol{a_j})}{\sum_{i=1}^{k} exp(\boldsymbol{a_i})} \quad (3)$$

$$\boldsymbol{a_i} = \boldsymbol{w_i x} + \boldsymbol{b_i} \quad (4)$$

を用いる（ここで，$\boldsymbol{w_i}$ は $\boldsymbol{w}$ の i 行目，$\boldsymbol{b_i}$ は $\boldsymbol{b}$ の i 番目の要素とする）．学習は，(i) 確率的勾配効果法 (以下，SGD)，(ii)Adagrad，(iii)Adadelta による比較により行う．SGD の学習率は 0.01 とした．合わせて，検証データに対する誤識別率をモニターし，誤識別率が最も小さなモデルを最終的な学習結果とするストッピングも行う．

4.3 評価方法

実験の評価は適合率，再現率，適合率-再現率曲線及びその曲線下の AUC により行った．再現率，適合率及び AUC は複数回実験を行った際の平均値とした．各分類器が出力する連続値に対して閾値を設定し，その閾値ごとの適合率，再現率をプロットした．

5. 結　　果

5.1 Adagrad, Adadelta, SGD の比較

はじめに，Adagrad,Adadelta,SGD を Dropout の有無に関して比較した．Anderson_J という名前で混同されている著者のペアに関して実験した．深層学習器を 300 回学習した経過を示す (図 3)．結果から，学習後の正解率が SGD よりも高く，Adagrad,Adadelta が誤差逆伝播法に対して有効であることがわかる．さらに，学習途中の誤差関数が，SGD のほうが低い値を示しており，SGD では過学習を起こしていることがわかる．また，学習が進むにつれ，Dropout を行った際に，誤差関

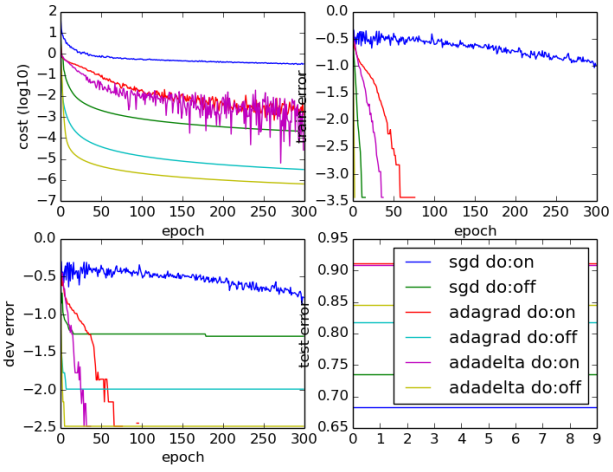

図 3　J.Anderson の学習の経過
左上:誤差関数値，右上:学習データに対する誤識別率，左下:検証データに対する誤識別率, 右下:テストデータに対する正解率, dropout の有無 (do:on , do:off)

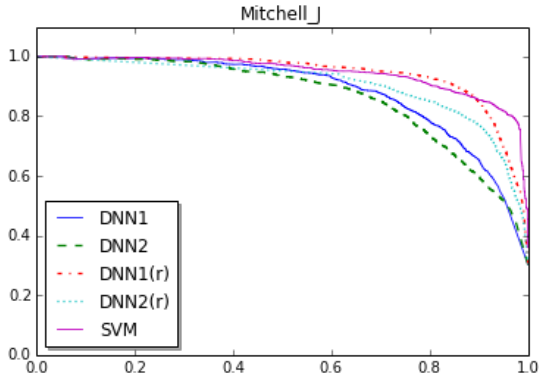

図 4　高い AUC を示した例

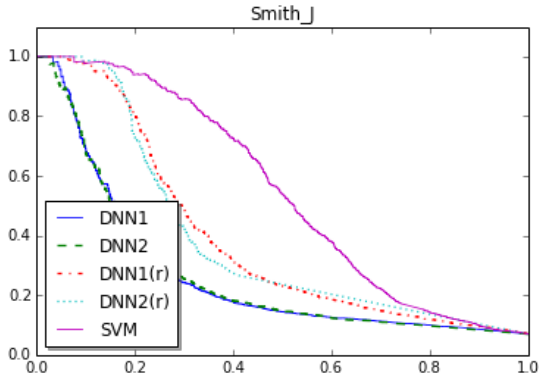

図 5　低い AUC を示した例

数の値が振動することが確認できる．

5.2　先行研究との比較実験

先行研究であるペアワイズ SVM との比較実験を行った．中間層の設定は，いくつかの設定で，比較的良い結果を示したものを用いた．深層学習器の中間層は { 入力次元-2000-1000-1000-1000-2(出力次元)} とし，活性化関数は，出力層にソフトマックス関数を，その他の層には ReLU を用いた．また，最適化は Adagrad と Adadelta により行った．異なるネットワーク重みの初期値から学習し，識別した際の適合率再現率 (表 2)，AUC(表 3) 及び正解率 (表 4) を示す．それぞれの値は，3 回実験を行った際の平均値である．正則化を行わない場合と比較して，Dropout を行った場合に高い適合率，再現率を示すことが確認できる．先行研究との正解率の比較では，6 件中 4 件で深層学習器が先行研究より高い精度を示している．また，Dropout あり Adagrad を行った深層学習器が，先行研究より高い再現率を示すことが確認できる．同じく Dropout あり Adadelta による再現率も多くのものが先行研究より高い再現率を示している．次に，高い AUC と低い AUC を示したときの，再現率適合率曲線を図 4,5 に示す．DNN1 が Dropout を用いず，Adagrad で学習したもの，DNN2 が同じく Adadelta で学習したもの，DNN1(r) が Dropout を用いて Adagrad で学習したもの，DNN2(r) が同じく Adadelta により学習した結果を示す．図 4 では，Dropout と Adagrad 学習を行った深層学習器が SVM よりも高い AUC を示したときの再現率適合率曲線が確認できる．

省略された名前	DNN Adagrad	DNN(r) Adagrad	DNN Adadelta	DNN(r) Adadelta	SVM
J.Anderson	0.7796	0.8487	0.7058	0.8327	**0.8612**
A.Gupta	0.5011	0.6342	0.4250	0.5839	**0.7068**
M.Sato	0.4601	0.6156	0.4811	0.6092	**0.7975**
J.Smith	0.3047	0.4016	0.2785	0.4059	**0.5391**
K.Tanaka	0.7310	0.8075	0.6879	0.8240	**0.9052**
J.Mitchell	0.8908	**0.9475**	0.8693	0.9127	0.9443

表 3　AUC

省略された名前	DNN Adagrad	DNN(r) Adagrad	DNN Adadelta	DNN(r) Adadelta	SVM
J.Anderson	0.1030	**0.0718**	0.1176	0.0738	0.0993
A.Gupta	0.0755	**0.0624**	0.0710	0.0823	0.0710
M.Sato	0.1643	0.1456	0.1565	0.1430	**0.1105**
J.Smith	0.0644	0.0602	0.0646	0.0586	**0.0583**
K.Tanaka	0.1516	0.1343	0.1590	**0.1107**	0.1237
J.Mitchell	0.1670	**0.0930**	0.1690	0.0987	0.1624

表 4　エラー率

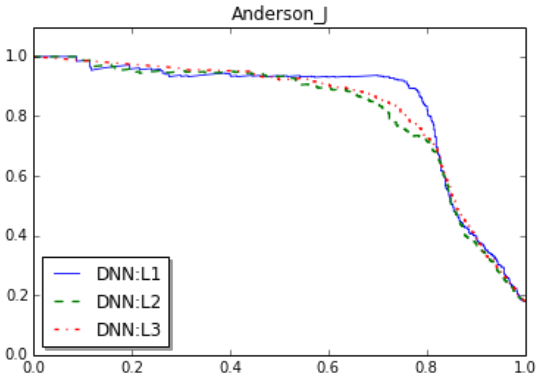

図 6　J.Anderson の適合率-再現率曲線

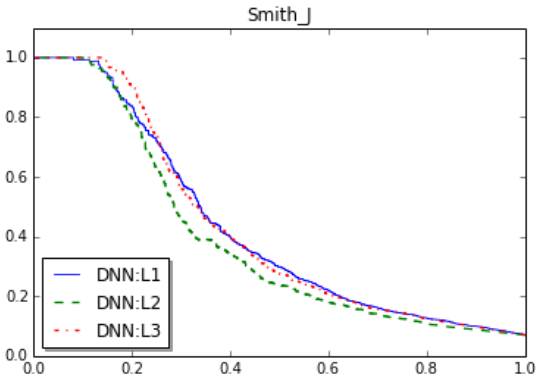

図 7　J.Smith の適合率-再現率曲線

5.3　異なる層数のニューラルネットワークによる比較

次に深層学習器の層数の違いによる結果の変化を見る．ここでは，3 種類の層数のニューラルネットワークの実験を行った結果を示す (図 6,7).DNN1:L1 が 1 層ニューラルネットワーク，DNN:L2 が 2 層ニューラルネットワーク，DNN:L3 が 3 層ニューラルネットワークを示している．中間層のノード数は浅いものから {2000}(1 層),{2000-1000}(2 層),{2000-1000-1000}(3 層) とする．ここで，中間層とは，入力層と出力層以外の層とする．図 6,(reffig:cmplayers2 で浅い学習器のほうが高い適合率を示すことが確認できる．中間層が 1 層 ({2000}) と 2 層

省略された名前	DNN[Adagrad]		DNN(r)[Adagrad]		DNN[Adadelta]		DNN(r)[Adadelta]		SVM	
	P	R	P	R	P	R	P	R	P	R
J.Anderson	0.8494	0.5116	0.9205	0.6477	0.8252	0.4225	0.8793	0.6724	0.9403	0.4775
A.Gupta	0.8327	0.2467	0.7786	0.4772	0.6440	0.2679	0.6522	0.5141	0.9074	0.3651
M.Sato	0.6211	0.2413	0.6288	0.4844	0.7002	0.2269	0.6370	0.4813	0.9638	0.4007
J.Smith	0.7586	0.1215	0.7793	0.2124	0.7656	0.1078	0.9516	0.1694	0.9542	0.1732
K.Tanaka	0.8724	0.3446	0.8763	0.4546	0.8485	0.3192	0.8621	0.5921	0.9970	0.4096
J.Mitchell	0.9771	0.4605	0.9585	0.7234	0.9573	0.4594	0.8553	0.8033	0.9731	0.4751

表 2　適合率 (P) 及び再現率 (R) を示す．深層学習 (DNN:Dropout なし，DNN(r)：Dropout あり),SVM による結果の比較を行った．

({2000-1000}) では，再現率適合率曲線に大きな変化が見られるが，２層と３層 ({2000-1000-1000}) ではあまり大きな変化は見られない．

6. 考　　察

ペアワイズ分類における深層学習器は，先行研究と比べて高い再現率を示し，いくつかのデータで高い正解率を示した．SVM の場合は，カーネルのような，問題に特化した構造を入れることでデータの特殊性に対応する一方で，深層学習器では，そのような構造なしでも学習器自身がある程度ペアの情報を抽出できている可能性があり，この点は SVM と比べ有利な点である．しかし，AUC 及び適合率に関しては低い値を示し，二例間の類似度の総合的な評価という点では，先行研究のほうが優れていると言える．学習器の層数に関しては，5.3 項で見たように，浅い層の学習器のほうが良い結果を示す例が見られた．一般的に，深層学習は，画像のピクセル等の抽象度の低い特徴量を，ラベルのような抽象的な特徴へ階層的に変換する（特徴抽出する）といったことに向くと言われる．一方で，ペアワイズ分類では,「深く，階層化された」データ構造というよりは，二例間の素性の組み合わせの共起に関する特徴さえ取り出せればよく，必ずしも階層的な特徴抽出は必要ないと考えることができる．むしろ，深層学習器の膨大なパラメータがネックとなり，学習器の汎化性能の低下につながる可能性もある．n 個の素性数を持つ２つのデータ間のすべての素性の組み合わせは n^2 通りであり，これらのすべての特徴を捉えるために，中間層のノード数を n^2 にすることも考えられるが，その場合，メモリ消費量の点で現実的な手法とはいえない．そういったことをせずに，階層化されたネットワークによって，特徴が抽出されることを期待したが，難しいことが実験により示された．他に特徴的な点は，今回用いたデータが，論文タイトル中の単語の出現に基づく疎行列である点である，タイトルの単語出現に基づくベクトルは，もともとある程度意味を持つ抽象度の高い素性であると言え，この点でこれまで成功を収めてきた画像データのピクセルのような抽象度の低い素性とは異なる．実際に，表 5 に示すように，あるペアインスタンスの次元数と実際に持っている素性数 (論文タイトル中に出現する単語数) の平均を比較してみると，非常にスパースであることがわかる．このように，抽象度の低い密なデータから抽象的な構造を見つけるのではなく，もともと抽象度の高い疎なデータを深層学習で扱えるかといったことを探ることは今後の課題である．学習に関しては，Dropout 正則化が非常に有効に働くことが 5.1 項や，表 2 から確認できる．このことから，スパースなデータの学習に関しても Dropout が有効であるといえる．また，学習の際に Adagrad や Adadelta が SGD と比較して最適解を探索する上で効果的なことも確認できる．

省略された名前	素性数	平均素性数
J.Anderson	1890	18.9
A.Gupta	3444	19.7
M.Sato	1724	22.2
J.Smith	3652	17.7
K.Tanaka	1832	20.3
J.Mitchell	2106	17.4

表 5　各学習データの素性数と，ペアインスタンス中に含まれる平均素性数

7. ま と め

本研究ではペアワイズ分類における深層学習器について検証した．先行研究であるペアワイズ分類との比較では，深層学習器がそれよりも高い識別性能を出すことは難しいことがわかった．学習の方法として，SGD よりも Adagrad,Adadelta といった最適化手法が有効であることを確認し，その後，本問題設定において Dropout 正則化が非常に有効に働くことを確認した．前述のように，先行研究と比較して，高い再現率が確認でき，一部の識別率では，先行研究を上回るものも見られた．一方で，適合率や AUC の値は，先行研究の結果を上回る例は少なかった．また，必ずしも深い層の学習器が良い結果を示すとは限らず，浅い層の学習器のほうが精度が高い結果も確認できた．今後の課題としては，疎な素性を持つデータに関する深層学習の振る舞いや，画像や音声と言った他のデータに関するペアワイズ分類における深層学習の有効性についても検証したい．

文　　献

[1] 小山聡, クリストファー D. マニング．異なる例からの素性の組合せを用いたペアワイズ分類器の学習．人工知能学会論文誌, Vol. 20, pp. 105–116, November 2005.

[2] 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. *J. Mach. Learn. Res.*, Vol. 11, pp. 3371–3408, December 2010.

[3] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *JMLR W&CP: Pro-*

ceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011), pp. 315–323, April 2011.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, Vol. 15, pp. 1929–1958, January 2014.

[5] Sida Wang and Christopher Manning. Fast Dropout Training. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Vol. 28, pp. 118–126. JMLR Workshop and Conference Proceedings, May 2013.

[6] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159, July 2011.

[7] Matthew Zeiler. ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*, December 2012. http://arxiv.org/abs/1212.5701.