解　説

自然言語処理のための深層学習

Deep Learning for Natural Language Processing

ボレガラ
ダヌシカ
Danushka Bollegala

リバープール大学
Department of Computer Science, The University of Liverpool
danushka.bollegala@liverpool.ac.uk, http://www.csc.liv.ac.uk/~danushka

keywords: Deep Learning, Natural Language Processing.

Summary

1. は　じ　め　に

深層学習は既に音声認識 [Deng 13]，画像認識 [Le 12] など様々な認識タスクにおいて素晴らしい成果をもたらしている．自然言語処理分野 [Manning 02] も例外ではない．言語モデル構築 [Bengio 03]，固有名詞抽出 [Wang 13]，構成的意味論に基づく意味構築 [Socher 12]，評判分類 [Socher 11c, Glorot 11] など様々なタスクにおいて深層学習を用いた手法は圧倒的な精度を報告している．

自然言語処理では文書で書かれたテキスト情報 (textual information) を主に処理対象としている．テキスト情報と一言でいってもその中に，電子書籍，新聞記事，ウェブページ，ブログ，評判や口コミ，ツイートなど様々な種類のテキストが含まれている．人手でルールや辞書を整備するルール型 (rule-based) 自然言語処理システムに代わって，今日では機械学習や統計的手法を用いる自然言語処理システムがその精度とコストという面で大変注目を集めている．特に，ルールだけでは十分カバーできない言語現象や，ルールを書くための専門知識を持っている，いわゆるドメインエキスパートが集められない場合はルール型の言語処理システムを構築，維持することが困難である．

画像処理の場合はピクセル，音声処理の場合は音声信号といった基本入力が決まっているのに対し，自然言語では処理対象とするテキストをどのように表現すべきかは決まった方法がなく，タスクによって様々である．例えば情報抽出の場合はテキストを単語の集合 (bag-of-words) として表現するのが主流であり，評判分析，文書自動要約，機械翻訳のようなより高度なタスクでは品詞解析，係り受け解析，照応解析, 意味ラベルを使ったより複雑な表現方法が用いられている [Koehn 09]．このような統計的自然言語処理ではテキストをどのような特徴量を使って表現するかが自然言語処理の専門家が考えなければならない最も重要な課題といっても過言ではない．

表現学習 (representation learning) では有効な特徴の組み合わせを自動的に学習する．深層学習ではネットワークの層を重ねることで特徴量のより複雑な組み合わせが考慮できる [Bengio 12a, Bengio 12b]．例えば，入力層では単語の出現（有無を表すバイナリ記法）を認識する入力ノードを用意しておくと，第２層では２単語の重み付き組み合わせからなる特徴量が生成される．例えば，単語を個別に見て判断できない否定表現を含む評判分類タスクでは単語の組み合わせを特徴量として使うことでより高い精度が得られる．従って，自然言語処理分野では単語の組み合わせを特徴量として機械学習を行う手法が昔から使われてきた．最も簡単な解決方法として文書を一単語 (unigram) だけで表現するのではなく，連続して出現する単語の組 (bigram, trigram など）として表現する手法がある．しかし，連続する長さが長くなれば組み合わせの数が膨大に増え，そのような特徴の出現頻度が減る．そのため，十分な訓練サンプル数が確保できなくなる問題が生じる．更に，どの単語の組み合わせが目的とするタスクに関して有効なのかも定かではない．

深層学習は自然言語処理におけるこの表現学習の問題を事前学習 (pre-training) を行うことで解決している．この事前学習を行うことでノード間のどの接続にどれくらい重みをつけるかを決めている．重要でない特徴の組み合わせに対する重みを下げる，あるいはゼロにすることでより簡潔かつ，タスクの達成に関連する特徴の組み合わせを優先的に残すことができる．事前学習では与えられた入力とネットワークを通じて元へ伝搬された出力の差が小さくなるように学習が行われる点では，事前学習は構造予測 (Structure Prediction) の分野で提案されている Alternating Structural Optimisation (ASO)[Ando 05] の考え方に近い．つまり，入力が正しく再現できるネットワーク構造を学習することによってデータそのものの構造を事前に学習させておき，目的とするタスクを学習する際に元の入力ではなく，そこから学習した構造を特徴とし

て使う．自然言語処理の場合で考えると，単語が文中にランダムに出現しているのではなく，その前後の文脈に依存して，その出現が決まっているので，その依存関係が事前に与えられていればもし一部の特徴が入力に出現していない場合でもそれを補うことができる．特に，特徴の出現はスパースである自然言語の場合は事前学習は重要な役割を果たすことは容易に理解できる．

次に，事後学習 (post-training) では事前学習で得られた有効な特徴の組み合わせを使って目的とするタスクを学習している．事前学習によって学習された特徴の組み合わせを事後学習で使用する方法はいくつか存在する．例えば，目的とするタスクに関するラベル付きデータ中に含まれる特徴から事前学習によって，特徴の組み合わせを学習し，その組み合わせ特徴を複合特徴として使うことができる．深層学習が自然言語処理分野で広く用いられてきた重要な理由として目的とするタスクと無関係にまず事前学習でネットワーク構造を学習させ，その学習させたネットワークを使って様々な自然言語処理タスクが同時に学習できるというこの再利用性のメリットもある．

深層学習の重要な特徴として事前学習と事後学習を分けている点が挙げられる．テキストを正しく表現するための特徴量を学習するタスクは特徴量の数が増えるに従い，複雑になる．しかし，事前学習ではラベルがついたデータを用いないため，ラベルが付けられていないデータを容易かつ膨大に集められる自然言語処理の多くタスクでは問題にならない．一方，事後学習ではネットワーク構造ではなく，目的とするタスクを学習するためにラベルが付いたデータを必要とする．従って，事前学習と事後学習を分けることで大量データをより有効に活用でき，全体として性能が向上する．

本稿では，自然言語処理分野で深層学習が応用されている例をいくつか交えながら，自然言語処理の根本的な課題と深層学習によってそれらがどのように解決されているかを解説する．具体例として言語モデル構築に関する研究事例 (2 章) と意味構築に関する研究事例 (3 章) を紹介する．最後に，4 章では自然言語処理に深層学習を適応する際に乗り越えなければ課題をいくつか紹介し，本稿をまとめる．深層学習の基礎，学習方法，実装方法については本特集号で以前数回に渡って詳しく解説されており，本稿では簡便のため省略する．

2. 深層学習と言語モデル

言語モデルとは単語が文書中に出現する過程を確率過程と見なし，ある単語がある位置に出現する確率はどれくらいかを計算するためのものである．単語の出現しやすさを予測することは自然言語処理に限らず，音声認識の分野でも様々なタスクにおいて基本となる．自然言語処理における言語モデルの応用例として機械翻訳システムが挙げられる．機械翻訳システムでは生成した翻訳文がその適用先言語においてどれくらいもっともらしいかを言語モデルを使って評価することで不自然な翻訳文が生成される可能性を減らすことができる．英語から日本への機械翻訳を例として考えると，作成した和文は日本語として不自然であれば，日本語のネーティブ話者が普段使わない単語列が出現しているということになる．すなわち，生成された和文の出現確率を言語モデルを使って計算すると低い確率となり，この「日本語としての不自然さ」が定量的に評価できるため，そのような不自然な和訳をなるべく作らないように機械翻訳システムを自動調整することが可能となる．

この言語モデルは形式的に次のように表せる．ある文書において j 番目に出現する単語を w_j として表し，1 番目から $j-1$ 番目まで連続し，出現する単語列 $w_1, w_2, \ldots, w_{j-1}$ を w_1^{j-1} として表すと，w_1^{j-1} の次に w_j が出現する条件付き確率を $P(w_j|w_1^{j-1})$ と書くことができる．そうすると，この言語モデルに従い，長さ T の単語列からなる文書が生成される確率は次式で与えられる．

$$P(w_1^T) = \prod_{j=1}^{T} P(w_j|w_1^{j-1}) \tag{1}$$

ただし実際には，あまりにも離れている単語はお互い関係しないこともあり，連続する長さを 2 単語から 5 単語までの範囲に限定することが多い．どんなに大きなコーパスであっても連続する長さが増えると，その単語列の出現頻度が減り，コーパス中に全く現れない単語の連続が生じてしまう．これはデータスパースネス問題や，ゼロ頻度問題などと名付けられており，言語モデルを構築する上で解決しなければならない根本的な問題の一つである．コーパス中に出現しなかった単語の連続に関する出現頻度を計算する方法はスムージング (smoothing) と呼ばれている．例えば trigram(連続する長さが 3 単語までに限定）の言語モデルの場合，ある trigram がコーパス中に出現しない場合，その中に含まれている bigram(連続する 2 単語) の出現確率を使って trigram の出現確率を予測するという方法がある．より短い長さの連続に関する統計情報を使うという意味でこのやり方が back-off smoothing と呼ばれている [Katz 87].

2·1 ニューラルネットワーク言語モデル

言語モデルにおいて単語列の出現確率を予測するためにニューラルネットワークを用いた有名な例として図 1 で示している Bengio らによるニューラルネットワーク言語モデル (Neural Network Language Model: NLMM)[Bengio 03] がある．NLMM ではまず単語列 w_{j-n+1}^{j-1} が与えられているときの，単語 w_j が出現する条件付き確率を出力するニューラルネットワークを学習する．NLMM では各単語 w_{j-n+1}^{j-1} を，出現したその単語の索引のみが 1 で残りの要素が全て 0 である N 次元のベクトルで表現して

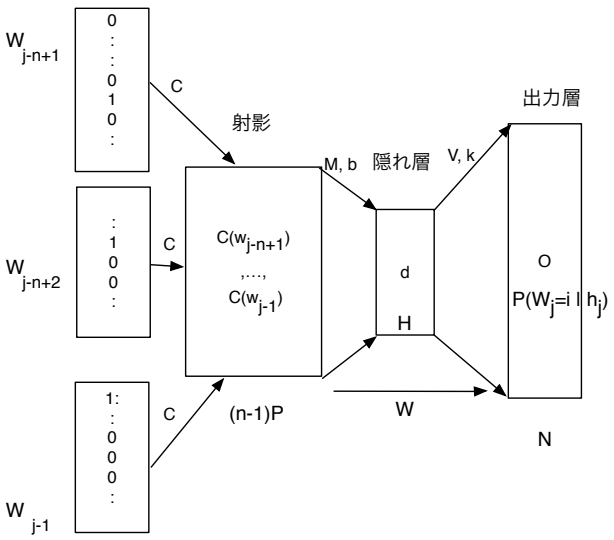

図 1 ニューラル言語モデルの構造

いる．ここでは N は語彙数である．このようなベクトル表現を 1-of-N 表現と呼ぶ．

次に，図 1 に示してあるように，これらの各 N 次元ベクトルを射影行列 $\mathbf{C}$ を使って $P < N$ 次元へ射影する．直感的には各単語の射影ベクトルはその単語の何らかの意味構造を表していると考えれば良い．例えば，分布意味論 (distributional semantics) では単語をその単語が出現する文脈を使って表現することができる．共起する文脈をコーパス中で最も良く出現する P 個の単語に限定すれば P 次元空間に全ての単語を射影することができる．NNLM では射影行列 $\mathbf{C}$ を $N \times P$ 個の自由パラメータと見なし，コーパスから学習する．射影適用後のベクトルは次の隠れ層への入力となる．具体的には，$(n-1)$ 個の単語 $w_{j-n+1}, \ldots, w_{j-1}$ それぞれに対する射影ベクトル $C(w_{j-n+1}), \ldots, C(w_{j-1})$ を連結した $(n-1)P$ 次元のベクトル，$\boldsymbol{c}$，が隠れ層への入力となる．Bengio らによるニューラル言語モデル [Bengio 03] では H 個のノードからなる隠れ層が一つしか存在しない．隠れ層の活性量に対し，tanh 関数を使って非線形性変換が行われる．隠れ層への入力を $\boldsymbol{c}_l$，隠れ層に対する重み行列を $\mathbf{M}$，j 番目の隠れノードに関するバイアスを b_j と表した場合，隠れ層の j 番目の出力ノードの出力，d_j，が次のように計算できる．

$$d_j = \tanh\left(\sum_{l=1}^{(n-1)P} \mathbf{M}_{jl}\boldsymbol{c}_l + b_j\right) \quad \forall j = 1, \ldots, H \tag{2}$$

最終的に，出力層では再び H の隠れ層の出力から出現確率 $o_i = P(w_j = i|h_j)$ は次のように計算できる．

$$o_i = b_i + \sum_{j=1}^{H} \mathbf{V}_{ij}\boldsymbol{d}_j + k_i + \sum_{l=1}^{(n-1)P} \mathbf{W}_{il}\boldsymbol{c}_l \tag{3}$$

$$P(w_j = i|h_j) = \frac{\exp(o_i)}{\sum_{r=1}^{N}\exp(o_r)} \tag{4}$$

ここでは射影行列 $\mathbf{C}$，隠れ層の重み行列 $\mathbf{M}$，バイアスベクトル $\boldsymbol{b}$，出力層の重み行列 $\mathbf{V}$，$\boldsymbol{c}$ に対する重み行列 $\boldsymbol{W}$ は全て学習すべきパラメータである．これらのパラメータをまとめて $\boldsymbol{\theta}$ として表す．コーパス中で観測された単語 w_j に対する出現確率が最大となるように逆伝搬法 (back propagation) を用いてパラメータ学習を行う．この目的関数として次式で与えられる交差エントロピーにパラメータ $\boldsymbol{\theta}$ に関するフロベニウスノルムを正則化項，$R(\boldsymbol{\theta})$ として加えたものが用いられる．

$$E = \sum_{i=1}^{N} t_i \log P(w_j = i|h_j) + \beta R(\boldsymbol{\theta}) \tag{5}$$

ここで t_i は単語列 w_{j-n+1}^{j-1} の直後に単語 $w_j = i$ が出現した場合に 1 となり，そうでない場合は 0 となる学習信号を表す二値変数である．正則化係数 β はパラメータ $\boldsymbol{\theta}$ のフロベニウスノルムに関する損失を調整するために使われている．

2·2 その他の言語モデル

ゼロ頻度問題を避けるために言語モデルは膨大なコーパスを用いて計算されることが普通である．しかし，上述したニューラル言語モデルは語彙数に比例して線形にその計算量が増える．従って，大規模なコーパスを扱うには大きな行列演算，微分の伝搬を必要とし，スムージング手法と比べ，計算時間が必要となる．Schwenk らはコーパス中の全ての単語ではなく，高出現頻度を持つ単語のみを対象に学習を行うことで計算時間を減らし，膨大なコーパスを使ってニューラル言語モデルを学習することに成功した [Schwenk 05, Schwenk 04]．Arisoy らは図 1 に更に隠れ層を追加することでより深いニューラル言語モデルを構築し，音声認識タスクにおける誤り率を下げることに成功した [Arisoy 12].

Colloberet らは [Collobert 08, Collobert 11]Bengio らと異なる手法を使った言語モデルを提案した．上述した，単語の出現確率と予測する Bengio ら [Bengio 03] の NNLM 違って, Colloberet らはある文脈において特定単語が出現するか否かを予測する二値分類タスクとして言語モデル構築問題を定式化した [Okanohara 07]．具体的には，ある単語 w が出現している文脈 s をウィキペディアから抽出し，それらを w の出現に関する正例とし，文脈 s 中の w の出現をランダムに選択した単語で置き換えた文脈 s^w を w の出現に関する負例とした. ソフトマックス層を除く，ニューラルネットワークの出力が関数 f で与えられるとすると，次のヒンジコストが最小になるようにネットワークの重みと単語の表現が学習される．

$$\sum_{s\in\mathcal{S}}\sum_{w\in\mathcal{D}} \max(0, 1 - f(s) + f(s^w)) \tag{6}$$

ただし，式 (6) では $\mathcal{S}$ は全文集合を表しており，$\mathcal{D}$ は語彙集合を表わしている．Colloberet らの研究 [Collobert 08]

で示している通り，このようにして学習された単語表現を使って単語クラスタリングを行った場合，意味的に類似している単語が同じグループに属しているため，正しい意味表現が学習できたと言える．この方法で学習された単語の表現はその後の深層学習を使った自然言語処理の研究でもネットワークを初期化するために使われている [Socher 11c].

ニューラルネットワークで自然言語処理を行う場合に一つ問題となるのは長さが異なる文をどのようにして長さが固定の入力層へ入力するかということである．単純な解決方法としてある長さの窓を事前に決めておき，その内で出現している単語のみを入力するという方法がある．しかし，この方法ではその窓の長さより広い範囲で関係をしている単語間の関係が考慮できないという欠点がある．この問題を解決するために Colloberet らは時間遅れニューラルネットワーク (Time Delay Neural Network) TDNN [Waibel 89] を用いた．この方法では文中の単語を左から右へ入力してゆき，TDNN によってその文全体にわたって畳込みが行われる．

Colloberet らは上記の方法で学習させたニューラルネットワークを用いて，自然言語処理における 6 つの基本的なタスクを同時に学習することに成功した．具体的には，品詞タグ付け，チャンキング，固有名詞抽出，意味ラベル付与，言語モデル構築，類似語判定を同時学習しているが，特に注目すべきタスクは意味ラベル付与である．実際に複数のタスクを同時に学習する際に，まずタスクを選択し，そのタスクに関する学習事例を一つランダムに選択し，それに関してネットワークの重みを更新するという手順をとっている．これらのタスクはお互い関連するため同時学習を行うことで単独に学習する場合に比べてより良い精度を得ている．事前学習によって言語の構造を学習した上で，その同じネットワークで複数のタスクが学習できるということを示した例として Colloberet らの研究は注目を浴びている．なお，画像とテキスト両方から特徴量を抽出し，深層学習によってそれらを組み合わせたマルチモーダル言語モデルも提案されており，画像検索や画像の自動アノテーションという応用がなされている [Kiros 13].

3. 深層学習と意味構築

単語や句の意味をどのように表現するかは自然言語処理分野の基本課題の一つとなっており，現在でも盛んに研究が行われている [Mitchell 08, Mitchell 09, Baroni 10a, Baroni 10b, Liang 11, Grefenstette 11, Grefenstette 13, Erk 13, Turney 13]．一つの単語の意味なら人手で作成された辞書を引けば良いが，複数の単語からなる句，文あるいは文書となればそれらの意味表現を事前に作成しておくことは非常に困難な問題である．単語そのものには潜在している意味がなく，その単語の使われ方によって意味が生まれるという分布仮説 (Distributional Hypothesis) は Firth[Firth 57] や Harris[Harris 85] によって提唱され，自然言語処理分野で注目を浴びて来た．特に，膨大なテキストコーパスに対して統計処理を行う，統計的自然言語処理の分野では，単語の意味表現は事前に与える必要がなく，分布仮説に基づき，単語が出現する文脈から自動的に構築できるという点では都合が良い．例えば，ある単語 w の意味表現としてあるコーパス中に w が出現する文脈から他の単語 w_i を抽出し，ベクトル $\boldsymbol{w}$ として表現することができる．各単語 w_i は意味ベクトル $\boldsymbol{w}$ の要素となっており，その値を何らかの共起尺度を使って計算することができる．単語 w が出現する文脈として例えばコーパス中に w が出現する位置の前後数単語を使う方法と w と何らかの係り受け関係で繋がっている単語 w_i のみを使う方法が広く使われている．

単語単位であれば大規模なコーパスを使えばそのコーパス中に出現する文脈を集めることで単語単位の意味表現が作成できるが，句単位，文単位となればどれほど大きなコーパスであっても同じ文が数回出現することはまず考えられないので分散説を単語以上の単位の意味を表現するために適用するのは無理がある [Turney 13]．そこで，自然言語処理分野では一単語の意味を表す構造に対し，何らかの演算を施すことで句や文の意味表現を構築する分散的意味構築 (Distributional Semantic Composition) の研究が行われてきた．しかし，一単語の意味をどのように表現し，それらの意味表現に対してどのような演算を行うべきかはまだ未解決課題となっており，深層学習の特徴である有効な特徴の学習と組み合わせをこのタスクに応用できないか研究されてきた．その代表的な研究事例として Socher[Socher 11a] らによる言い換え表現認識の研究を紹介する．

3·1 言い換え表現認識への応用

言い換え表現認識では与えられた 2 つの文が同じ意味を表しているかどうか判定するのが目的となる．この問題は自然言語処理おける様々な応用で重要である．例えば，文書自動要約では 2 つの文が同じ意味を表しているのであればそのどれか一つのみを要約に含むことでより簡潔な要約を作成することができる．深層学習を用いて言い換え表現認識する問題を具体的に説明するために m 個の単語 $x_1, x_2, \ldots, x_m$ からなる文を考えよう．まず，問題になるのはそれぞれの単語の意味表現として何を用いるかである．Socher らは 2·2 節で紹介した Collobert ら [Collobert 08] によって提案されたニューラル言語モデルを使って学習させたベクトルを単語の意味表現として用いた．この方法ではそれぞれの単語が N 次元のベクトルとして表現される．なお，ベクトルの要素は実数であるため深層学習で使われるシグモイド関数のような連続的な非線形演算と相性が良い．この一単語の意味表現方法以外に，Baroni による分散メモリ (Distributional Memory)[Baroni

10a] のような係り受け関係に基づく一単語の意味表現方法などもある．

図 2 では「とても美しい絵」という文に対し，再帰自己符号化器を適用し，意味表現を構築する方法を説明する．まず，それぞれの単語「とても」,「美しい」と「絵」に関してその意味表現が Collobert らの手法 [Collobert 08] を用いてベクトル $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, と $\boldsymbol{x}_3$ が与えられているとする．ここでは全ての親が子を 2 つ持つような二分構文木 (binary parse trees) を対象としている．自己符号化器では 2 つの子をを表すベクトルを $\boldsymbol{c}_1$ と $\boldsymbol{c}_2$ とすると，その親を表すベクトル $\boldsymbol{p}$ は次式で与えられる．

$$\boldsymbol{p} = f(\mathbf{W}_e[\boldsymbol{c}_1; \boldsymbol{c}_2] + \boldsymbol{b}_e) \tag{7}$$

ここでは，N 次元の縦ベクトル $\boldsymbol{c}_1$ と $\boldsymbol{c}_2$ を連結して作られた $2N$ 次元のベクトルを $[\boldsymbol{c}_1; \boldsymbol{c}_2]$ として表現する．符号化行列 $\mathbf{W}_e$ をこの合成したベクトルに適用し，更にバイアスベクトル $\boldsymbol{b}_e$ が足される．最終的に活性化関数 f をベクトルの各要素ごとに適用し，親ベクトル $\boldsymbol{p}$ を計算する．次に，親に対し，復号行列 $\mathbf{W}_d$ を適用することでその子を次のように生成する．

$$[\boldsymbol{c}_1'; \boldsymbol{c}_2'] = f(\mathbf{W}_d\boldsymbol{p} + \boldsymbol{b}_d) \tag{8}$$

生成された子ベクトルを $\boldsymbol{c}_1'$ と $\boldsymbol{c}_2'$ とする．自己符号化器では子ノードに関するベクトルを生成する場合の再現誤差 (reproduction error)E_{rec} は

$$E_{rec}(\boldsymbol{p}) = ||[\boldsymbol{c}_1; \boldsymbol{c}_2] - [\boldsymbol{c}_1'; \boldsymbol{c}_2']||^2$$

で与えられる．二分構文木では一つの個ノードに対して，2 つの親 y_1 と y_2 が存在するので，構文木 $\mathcal{T}$ に関する再現誤差 $E_{rec}(\mathcal{T})$ は,

$$E_{rec}(\mathcal{T}) = E_{rec}(y_1) + E_{rec}(y_2)$$

となる．再現誤差は凸関数ではないが，Socher らは Limited Memory BFGS アルゴリズム [Liu 89] を用いて $E_{rec}(\mathcal{T})$ が最小となるようにパラメータ $\mathbf{W}_e$, $\mathbf{W}_d$,$\boldsymbol{b}_d$ と $\boldsymbol{b}_e$ を学習することで多くの場合は良い解が得られると報告している．自己符号化器の詳細については本特集号第 3 回で既に紹介されているのでそちらも合わせて参照されたい．

図 2 は終端記号である単語から出発し，その親ノードを構文木に沿って順に生成していくプロセスを示している．再帰自己符号化器では対象とする親ノードの直下の子ノードしか生成しないが，Socher らはこの手法を更に拡張し，図 3 で示してある展開再帰自己符号化器 (unfolding recursive autoencoder) を提案した．展開再帰自己符号化器では対象とするノードより下にある全てのノードを生成するようにしている．図 3 では構文木の元のノードを表すベクトルを色塗りの丸で示しており，y_2 が対象とするノードの場合展開再帰自己符号化器によって再現されたノードに関するベクトルを色塗りの二重丸で示している．図 2 と図 3 では x_1', x_2', x_3', y_1' はそれぞれ x_1, x_2, x_3, y_1 から再現されたノードを表す．従来の自己符号化器と比べ，展開再帰自己符号化器ではある親ノードまでの全てのノードを生成するため対象とするノードは構文木上でどの深さで出現しているかが考慮できるという利点がある．なお，Socher らによる評価実験では自己符号化器を用いた場合と比べ，展開再帰自己符号化器を用いることで言い換え表現認識率が上がると報告されている．展開再帰自己符号化器を繰り返し適用することで最終的に構文木の根まで全てのノードに関する意味表現を作ることができる．

展開再帰自己符号化器を用いることで与えられた構文木に含まれている全てのノードに関してベクトルを付与することができたが，与えられた 2 つの文は言い換え表現になっているかどうか判断するためには更にその 2 つの文に関する意味表現を比較する必要がある．しかし，文の長さが異なるためそれぞれの構文木に含まれるノード数が異なり，単純には比較できない．一つの単純な比較方法として 2 つの文で根に関するベクトル同士を比較するという方法が考えられるがこの方法だと文に含まれる単語が直接比較されないという欠点がある．

長さが異なる 2 つの文から固定数の特徴を生成するために Socher らは動的プーリング (Dynamic Pooling) 方法を提案した．複数の要素を決まった数の領域に当てはめる作業がプーリングと呼ばれている．動的プーリングでは文の長さに応じて当てはめる領域の大きさを決めている．動的プーリングではまず単語間の類似度行列 $\mathbf{S}$ を作成する．それぞれの文に l 個と m 個の単語が含まれている場合はまず，それぞれの文に含まれている単語を $\mathbf{S}$ の行と列に入れておき，次に構文木上で左から右，下から上へと行きがけ順に辿り，それぞれ $(n-1)$ 個と $(l-1)$ 個の非終端記号も行列 $\mathbf{S}$ の行と列に追加しておく．次に全てのセルについて，その行や列に対応する単語や非終端記号間のベクトル対間のユークリッド距離を計算し，$\mathbf{S}$ の対応するセルに入れておく．この類似度（距離）行列は $2l-1$ 個の行と $2m-1$ 個の列からなり，2 つの構文木に含まれる単語（終端記号）と非終端記号に関する類似度情報を含んでいる．

次に，行と列をそれぞれ p 個の等間隔の領域に分割する．正確には $2l-1$ と $2m-1$ がそれぞれ p の倍数でなければ等間隔に分割できないが，その場合は $\lfloor \frac{2l-1}{p} \rfloor$ と $\lfloor \frac{2m-1}{p} \rfloor$ として分割領域数を決め，残りの行や列を作成した分割領域にできるだけ一様に追加することで対処している．こうして得られるプール行列 $M \in \mathbb{R}^{p \times p}$ の要素は元の類似度行列 $\mathbf{S}$ で対応する領域内での最小値を要素として持つ．これは 2 つの文に対する構文木上で近くにあるノード間で最も類似しているノード同士の距離を特徴量として選択することと等価である．なお，元の文の長さと無関係に常にプール行列は固定の大きさを持つため，このプール行列の要素を特徴量として言い換え表現

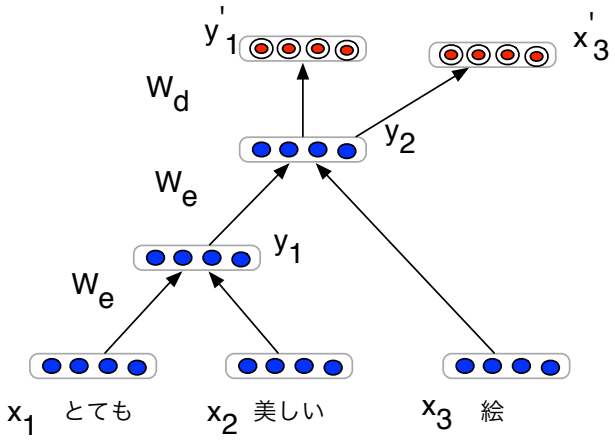

図 2 再帰自己符号化器

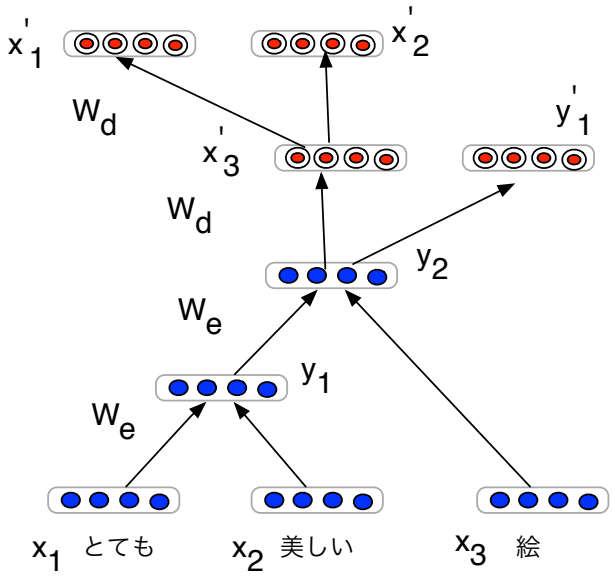

図 3 展開再帰自己符号化器

であるかどうかを判断するための分類器を学習することができる．このため Socher らはソフトマックス分類器を学習させている．

言い換え表現認識手法を評価するためのベンチマークとして広く用いられているマイクロソフトリサーチの言い換え表現コーパス [Dolan 04] 上では展開再帰自己符号化器を用いた手法が最も良い精度を報告している [Socher 11a]．なお，プール行列同士を比較することで与えられた文に意味的に近い文を検索できるようになっており，テキストの類似検索を行う際にも有効であることが示されている．

4. 今 後 の 課 題

本稿では自然言語処理分野で深層学習がどのように応用されているかを言語モデルと意味構築の研究事例を紹介しながら解説した．その他にもある単語に対して，その意味を表すベクトルとその活用を表す行列を同時に学習する Matrix Vector Recursive Neural Network (MV-RNN)[Socher 12], 評判分類への応用 [Socher 11c]，係り受け解析への応用 [Socher 11b], 系列ラベル付けへの応用 [Wang 13] などでも深層学習が使われている．今後，自然言語処理分野で深層学習が更なる発展および応用されることは期待できるが，いくつか解決すべき重要な課題も残されている．

深層学習における事前学習ではラベル付けられていないデータのみを用いて有効な特徴の組み合わせが自動的に学習される．一方，自然言語処理ではタスクによって有効な特徴が既に分かっている場合や，辞書，オントロジーなど言語資源が既に用意されている場合がある．既存の言語資源をどのように深層学習で使用するか，言語資源を全く使わないでラベルなしデータのみで有効な特徴の組み合わせが学習できるかはまだ不明である．評判分析の研究成果から分かるように十分な量のラベルなしデータがあれば言語資源を使わなくても十分な精度が得られる場合があるが，ラベルなしデータの量に限界がある言語やドメインでは既存の言語資源を無視することはできない．

自然言語処理分野で深層学習を応用する場合に解決しなければならないもう一つの重要な課題として，計算量削減がある．自然言語では単語が基本的な特徴となるため特徴量の空間が大規模となり，それらの組わせまで考慮すると大規模なニューラルネットワークを学習しなければならない．深層学習における効率的な学習方法や，分散的な学習方法は今後自然言語処理分野で深層学習を応用する際に重要になってくる．動的プーリングは長さの異なる文から固定長の特徴ベクトルを生成することができたが，文に比べて文書の長さには大きな分散があるため文書を扱うニューラルネットワークへの入力をどのようにすべきかは明らかではない．

◇ 参 考 文 献 ◇

[Ando 05] Ando, R. K. and Zhang, T.: A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data, *Journal of Machine Learning Research*, Vol. 6, pp. 1817–1853 (2005)

[Arisoy 12] Arisoy, E., Sainath, T. N., Kingsbury, B., and Ramabhadran, B.: Deep Neural Network Language Models, in *Proc. of the NAACL-HLT Workshop: Will We Ever Really Replace the N-gram Model?*, pp. 20 – 28 (2012)

[Baroni 10a] Baroni, M. and Lenci, A.: Distributional Memory: A General Framework for Corpus-Based Semantics, *Computational Linguistics*, Vol. 36, No. 4, pp. 673 – 721 (2010)

[Baroni 10b] Baroni, M. and Zamparelli, R.: Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space, in *EMNLP'10*, pp. 1183 – 1193 (2010)

[Bengio 03] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C.: A Neural Probabilistic Language Model, *Journal of Machine Learning Research*, Vol. 3, pp. 1137 – 1155 (2003)

[Bengio 12a] Bengio, Y.: Practical Recommendations for Gradient-Based Training of Deep Architectures, *arXiv* (2012)

[Bengio 12b] Bengio, Y., Courville, A., and Vincent, P.: Representation Learning: A Review and New Perspectives, *arXiv* (2012)

[Collobert 08] Collobert, R. and Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, in *ICML 2008*, pp. 160 – 167 (2008)
[Collobert 11] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuska, P.: Natural Language Processing (almost) from Scratch, *Journal of Machine Learning Research*, Vol. 12, pp. 2493 – 2537 (2011)
[Deng 13] Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M. L., Zweig, G., He, X., Williams, J., Gong, Y., and Acero, A.: Recent Advances in Deep Learning for Speech Research at Microsoft, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13)* (2013)
[Dolan 04] Dolan, B., Quirk, C., and Brockett, C.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources, in *Proc. of the 20th International Conference on Computational Linguistics* (2004)
[Erk 13] Erk, K.: Towards a semantics for distributional representations, in *10th International Conference on Computational Semantics (IWCS)*, Potsdam, Germany (2013)
[Firth 57] Firth, J. R.: A synopsis of linguistic theory 1930-55, *Studies in Linguistic Analysis*, pp. 1 – 32 (1957)
[Glorot 11] Glorot, X., Bordes, A., and Bengio, Y.: Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach, in *ICML'11* (2011)
[Grefenstette 11] Grefenstette, E., Sadrzadeh, M., Clark, S., Coecke, B., and Pulman, S.: Concrete Sentence Spaces for Compositional Distributional Models of Meaning, in *International Conference on Computational Semantics (IWCS'11)* (2011)
[Grefenstette 13] Grefenstette, E.: Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors, in *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pp. 1 – 10 (2013)
[Harris 85] Harris, Z.: Distributional Structure, *The Philosophy of Linguistics*, pp. 26 – 27 (1985)
[Katz 87] Katz, S. M.: Estimation of Probabilities from sparse data for the language model component of a speech recognizer, *IEEE Transactions on Acoustics, Speech, and Signal Processing (ASSP)*, Vol. 35, No. 3, pp. 400 – 401 (1987)
[Kiros 13] Kiros, R., Zemel, R. S., and Salakhutdinov, R.: Multimodal Neural Language Models, in *Deep Learning Workshop at NIPS'13* (2013)
[Koehn 09] Koehn, P.: *Statistical Machine Translation*, Cambridge University Press (2009)
[Le 12] Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y.: Building High-level Features using Large Scale Unsupervised Learning, in *ICML'12* (2012)
[Liang 11] Liang, P., Gordon, M. I., and Klein, D.: Learning Dependency-Based Compositional Semantics, in *ACL'11*, pp. 590 – 599 (2011)
[Liu 89] Liu, D. C. and Nocedal, J.: On the limited memory BFGS method for large scale optimization, *Mathematical Programming*, Vol. 45, pp. 503 – 528 (1989)
[Manning 02] Manning, C. D. and Schütze, H.: *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts (2002)
[Mitchell 08] Mitchell, J. and Lapata, M.: Vector-based Models of Semantic Composition, in *ACL-HLT'08*, pp. 236 – 244 (2008)
[Mitchell 09] Mitchell, J. and Lapata, M.: Language Models Based on Semantic Composition, in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 430–439, Singapore (2009)
[Okanohara 07] Okanohara, D. and Tsujii, J.: A discriminative language model with pseudo-negative samples, in *Proc. of the Annual Conference of the Association for Computational Linguistics (ACL'07)*, pp. 73 – 80 (2007)
[Schwenk 04] Schwenk, H.: Efficient training of large neural networks for language modeling, in *IJCNN*, pp. 3059 – 3062 (2004)
[Schwenk 05] Schwenk, H. and Gauvain, J.-L.: Training Neural Network Language Models On Very Large Corpora, in *Empirical Methods in Natural Language Processing*, pp. 201 – 208 (2005)
[Socher 11a] Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., and Manning, C. D.: Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection, in *NIPS'11* (2011)
[Socher 11b] Socher, R., Lin, C. C.-Y., Ng, A., and Manning, C.: Parsing Natural Scenes and Natural Language with Recursive Neural Networks, in *ICML'11* (2011)
[Socher 11c] Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D.: Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions, in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 151–161, Edinburgh, Scotland, UK. (2011), Association for Computational Linguistics
[Socher 12] Socher, R., Huval, B., Manning, C. D., and Ng, A. Y.: Semantic Compositionality through Recursive Matrix-Vector Spaces, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1201–1211, Jeju Island, Korea (2012), Association for Computational Linguistics
[Turney 13] Turney, P. D.: Distributional Semantics Beyond Words: Supervised Learning of Analogy and Paraphrase, *Transactions of Association for Computational Linguistics*, Vol. 1, pp. 353 – 366 (2013)
[Waibel 89] Waibel, A., Hanazawa, T., Hinton, G. E., and Shikano, K.: Phoneme Recognition Using Time-Delay Neural Networks, *IEEE Transactions on Acoustics, Speech, and Signal Processing (ASSP)*, Vol. 37, No. 3, pp. 328 –339 (1989)
[Wang 13] Wang, M. and Manning, C. D.: Effect of Non-linear Deep Architecture in Sequence Labeling, in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 1285–1291, Nagoya, Japan (2013), Asian Federation of Natural Language Processing

著 者 紹 介

ボレガラ ダヌシカ(正会員)

2005 年東京大学工学部電子情報工学科卒．2007 年同大学院情報理工学系研究科修士課程修了．2009 年同研究科博士課程修了．博士 (情報理工学)．東京大学大学院情報理工学研究科助教，講師を経て，現在，英国リバープール大学准教授 (Senior Lecturer)．専門分野は自然言語処理とウェブマイニング．WWW, IJCAI, AAAI, ACL, EMNLP を中心に研究成果を発表している．人工知能学会正会員