

固有値分解に基づくニューラルテンソルネットワークのパラメタ削減

石原 敬大¹, 林 克彦², 真鍋 陽俊¹, 新保 仁¹¹ 奈良先端科学技術大学院大学 情報科学研究科² 大阪大学 産業科学研究所¹{ishihara.takahiro.in0, manabe.hitoshi.me0, shimbo}@is.naist.jp²katsuhiko-h@sanken.osaka-u.ac.jp

1 はじめに

ニューラルテンソルネットワーク (Neural Tensor Network; NTN)[7] は 3 次テンソルを用いた写像により入力素性間の積を考慮するため, 通常の行列積を用いたニューラルネットワーク (NN) よりも高い表現力を持つ. この性質により, NTN は二項関係や木構造を扱うタスクで優れた性能を示している [7, 8]. しかし, NTN は多くのパラメタを持つため長い学習時間が必要であり, かつ, 過学習しやすいという問題がある [10].

NTN のパラメタを削減するため, 文献 [5] ではその 3 次テンソルの各スライス行列を 2 つの行列の積として低ランク近似する方法が提案されている. しかし, 本稿において実験的に明らかとするように, この手法は必ずしも NTN の良い近似とはならず, また, ランク数をハイパーパラメタとして決定する必要があるという問題がある.

本稿では NTN における 3 次テンソルの次元を削減するための手法を新たに 2 つ提案する. 提案手法では 3 次 ($n \times n \times k$) テンソルの各スライス行列に 1. 対称行列 2. 正規行列の制約を課し固有値分解を行うことで, それぞれ NTN のパラメタ数を $O(n^2k)$ から $O(nk)$ へと削減する. さらに, 2 つの手法を再帰型ニューラルテンソルネットワーク (Recursive NTN; RNTN) へ拡張したモデルも提案する.

知識グラフ補完および命題論理における推論 [2] の実験の結果から, パラメタ数が大幅に削減されている提案手法で, 通常の (R)NTN より優れた精度, 及び, 高速な学習を実現できることが明らかとなった*1.

2 モデル

本節ではベースラインとなるモデルと提案モデルについて説明する. 各モデル間のパラメタ数比較は表 1 にまとめる. また, それらを再帰型に拡張する方法についても説明する.

2.1 ベースライン

NN: 2 つの入力ベクトル $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ に対する行列積を用いた一層の NN を以下のように定式化する:

$$f(\mathbf{V} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{b}).$$

表 1: モデル間のパラメタ数比較

モデル	パラメタ数
NN	$(2n + 1)k$
NTN	$(n^2 + 2n + 1)k$
NTN-SMD	$(2mn + 2n + 1)k$
NTN-Diag	$(3n + 1)k$
NTN-Comp	$(6n + 1)k$

ここで, $\mathbf{V} \in \mathbb{R}^{k \times 2n}$, $\mathbf{b} \in \mathbb{R}^k$ であり, f は活性化関数を表す. パラメタ数は $(2n + 1)k$ となる.

NTN: 文献 [7] では, 活性化関数内に以下のような 3 次テンソル $\mathbf{W}^{[1:k]} \in \mathbb{R}^{n \times n \times k}$ を用いた写像が追加されている:

$$\mathbf{x}_1^T \mathbf{W}^{[1:k]} \mathbf{x}_2 = \begin{pmatrix} \mathbf{x}_1^T \mathbf{W}^1 \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_1^T \mathbf{W}^k \mathbf{x}_2 \end{pmatrix}. \quad (1)$$

これにより NTN を次のように定式化する:

$$f(\mathbf{x}_1^T \mathbf{W}^{[1:k]} \mathbf{x}_2 + \mathbf{V} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{b}).$$

パラメタ数は $(n^2 + 2n + 1)k$ となる.

NTN-SMD: 行列 $\mathbf{W} \in \mathbb{R}^{n \times n}$ のパラメタ数を削減する方法として, \mathbf{W} を低ランク行列の積として近似する方法が知られている [1]:

$$\mathbf{W} \simeq \mathbf{S}\mathbf{T}. \quad (2)$$

ここで, $\mathbf{S} \in \mathbb{R}^{n \times m}$, $\mathbf{T} \in \mathbb{R}^{m \times n}$ ($m \ll n$) である. 式 (1) における双線形項 $\mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2$ の \mathbf{W}^i を式 (2) により置き換えることで, その近似 $\mathbf{x}_1^T \mathbf{S}^i \mathbf{T}^i \mathbf{x}_2$ を得る. 本稿ではこの手法を単純行列分解 (Simple Matrix Decomposition; SMD) と呼ぶ. SMD により NTN における 3 次テンソル $\mathbf{W}^{[1:k]}$ の各スライス行列 \mathbf{W}^i のパラメタ数を n^2 から $2nm$ に削減することができる. そのモデルを文献 [5] に従って, 次の様に定式化する:

$$f(\mathbf{x}_1^T \mathbf{S}^{[1:k]} \mathbf{T}^{[1:k]} \mathbf{x}_2 + \mathbf{V} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{b}).$$

パラメタ数は $(2mn + 2n + 1)k$ となる.

*1 実験のコードは <https://github.com/tkhrshhr> で公開する.

2.2 提案手法

本項では NTN の 3 次テンソルの各スライス行列 \mathbf{W}^i に固有値分解を可能にする 2 つの制約を課す。それにより NTN のパラメタ数を大幅に削減する。

NTN-Diag: 自然言語処理においてデータが対称な構造をもつことはめずらしくない。たとえば, ある知識グラフ上で 2 項関係 (*Bob, is_relative_of, Alice*) が成り立つと, (*Alice, is_relative_of, Bob*) もまた成り立つ。また, 英語において句 “dog and cat” は “cat and dog” は同じ意味をもつ。このような対称な構造を扱う際は, $\mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2$ の値が $\mathbf{x}_2^T \mathbf{W}^i \mathbf{x}_1$ と等しくなるように, 3 次テンソルの各スライス行列を対称行列とすることが自然である。

$\mathbf{W} \in \mathbb{R}^{n \times n}$ が対称行列であるとき, 以下のように固有値分解を行うことができる。

$$\mathbf{W} = \mathbf{O} \mathbf{W}' \mathbf{O}^T.$$

ここで $\mathbf{O} \in \mathbb{R}^{n \times n}$ は直交行列であり, $\mathbf{W}' \in \mathbb{R}^{n \times n}$ は対角行列である。これを用いて双線形項 $\mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2$ を以下のように置き換えることができる:

$$\begin{aligned} \mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2 &= \mathbf{x}_1^T \mathbf{O} \mathbf{W}'^i \mathbf{O}^T \mathbf{x}_2 \\ &= \mathbf{y}_1^T \mathbf{W}'^i \mathbf{y}_2 \\ &= \langle \mathbf{y}_1, \mathbf{w}^i, \mathbf{y}_2 \rangle. \end{aligned} \quad (3)$$

ここで $\mathbf{y}_1 = \mathbf{O}^T \mathbf{x}_1$, $\mathbf{y}_2 = \mathbf{O}^T \mathbf{x}_2$, $\mathbf{w}^i = \text{diag}(\mathbf{W}'^i) \in \mathbb{R}^n$ であり, $\langle \mathbf{y}_1, \mathbf{w}^i, \mathbf{y}_2 \rangle$ は $\mathbf{y}_1, \mathbf{w}^i$ および \mathbf{y}_2 による 3 重内積 ($\langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle = \sum_{j=0}^n a_j b_j c_j$) である。結果として, 対称行列の制約により行列のパラメタ数は n^2 から n に削減されている。

NTN における 3 次テンソル $\mathbf{W}^{[1:k]}$ の各スライス行列 \mathbf{W}^i を対称行列であると仮定し, 式 (3) によって各双線形項 $\mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2$ を 3 重内積に置き換えたい。しかし, 一般に $\mathbf{W}^{[1:k]}$ における各スライス行列から導かれる直交行列は同一であるとは限らない。そこで, 文献 [4] に従い, それらが可換族を成すと仮定する*2。これにより, 次のモデルを得る:

$$f\left(\begin{pmatrix} \langle \mathbf{x}_1, \mathbf{w}^1, \mathbf{x}_2 \rangle \\ \vdots \\ \langle \mathbf{x}_1, \mathbf{w}^k, \mathbf{x}_2 \rangle \end{pmatrix}\right) + \mathbf{V} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{b}.$$

パラメタ数は $(3n+1)k$ となる。もし, \mathbf{V} が零行列であれば $\mathbf{x}_1, \mathbf{x}_2$ を交換可能なモデルとなる。

NTN-Comp: 一方で, 自然言語処理におけるデータの構造は多くの場合対称ではないため, 各スライス行列が対称行列と仮定するのは強い制約となる。固有値分解可能で, 対称行列より弱い制約の行列として正規行列がある。正規行列は, 対称行列, 反対称行列などを含むため, より柔軟なモデル化が可能となる。

複素正規行列は, ユニタリ行列によって対角化可能である。この性質を用いて実正規行列 \mathbf{W} を以下のように分解することができる [9]:

$$\mathbf{W} = \Re(\mathbf{O} \mathbf{W}' \mathbf{O}^*).$$

*2 $\mathbf{W}^i \mathbf{W}^j = \mathbf{W}^j \mathbf{W}^i, \forall i, j \in \{1, 2, \dots, k\}$ であると仮定する。これにより, $\forall i, j \in \{1, 2, \dots, k\}$ に対して, \mathbf{W}_i と \mathbf{W}_j はお互いに全ての固有ベクトルを共有する。

ここで, $\mathbf{O} \in \mathbb{C}^{n \times n}$ はユニタリ行列であり, $\mathbf{W}' \in \mathbb{C}^{n \times n}$ は対角行列である。また, \mathbf{O}^* は \mathbf{O} の共役転置であり, $\Re(\cdot)$ は複素数, 複素ベクトルまたは複素行列の実部をとる操作を表す。式 (3) と同様の手続きを用いて 3 次テンソルの各スライス行列 \mathbf{W}^i を代替することで, 以下のように双線形項を変形することができる:

$$\begin{aligned} \mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2 &= \Re(\langle \mathbf{y}_1, \mathbf{w}^i, \overline{\mathbf{y}_2} \rangle) \\ &= \langle \Re(\mathbf{y}_1), \Re(\mathbf{w}^i), \Re(\mathbf{y}_2) \rangle \\ &\quad + \langle \Re(\mathbf{y}_1), \Im(\mathbf{w}^i), \Im(\mathbf{y}_2) \rangle \\ &\quad + \langle \Im(\mathbf{y}_1), \Re(\mathbf{w}^i), \Im(\mathbf{y}_2) \rangle \\ &\quad - \langle \Im(\mathbf{y}_1), \Im(\mathbf{w}^i), \Re(\mathbf{y}_2) \rangle. \end{aligned} \quad (4)$$

ここで, $\mathbf{y}_1, \overline{\mathbf{y}_2} \in \mathbb{C}^n$ であり, $\mathbf{w}^i = \text{diag}(\mathbf{W}'^i) \in \mathbb{C}^n$ である。また, $\overline{\mathbf{y}_2}$ は \mathbf{y}_2 の共役ベクトルである。 $\langle \mathbf{y}_1, \mathbf{w}, \overline{\mathbf{y}_2} \rangle$ は複素ベクトルの 3 重エルミート内積を表す。 $\Im(\cdot)$ は複素数, 複素ベクトル, 複素行列の虚部をとる操作である。結果として, 正規行列の制約により行列のパラメタ数は n^2 から $2n$ に削減される。

$\mathbf{W}^{[1:k]}$ における各スライス行列が実正規行列であり, NTN-Diag の導出と同様に可換族を成すと仮定する。すると, 式 (4) を用いることで, 各双線形項を 3 重エルミート内積で置き換えることができる:

$$f\left(\begin{pmatrix} \Re(\langle \mathbf{x}_1, \mathbf{w}^1, \overline{\mathbf{x}_2} \rangle) \\ \vdots \\ \Re(\langle \mathbf{x}_1, \mathbf{w}^k, \overline{\mathbf{x}_2} \rangle) \end{pmatrix}\right) + \Re\left(\mathbf{V} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}\right) + \mathbf{b}.$$

ここで $\mathbf{x}_1, \mathbf{x}_2, \mathbf{w}^i \in \mathbb{C}^n (\forall i \in \{1, \dots, k\})$, $\mathbf{V} \in \mathbb{C}^{k \times 2n}$ である。パラメタ数は $(6n+1)k$ となる。

2.3 再帰型ニューラルテンソルネットワーク

再帰型ニューラルテンソルネットワーク (RNTN) の構造として 2 分木を考える。このとき, RNTN の各ノードは 2 つの子ノードの出力ベクトルを受け取る。上に述べたモデルのうち, NTN-Comp 以外は $\mathbb{R}^n \times \mathbb{R}^n$ から \mathbb{R}^k への写像であるので, k が n に等しい時, 各ノードは子ノードの出力ベクトルをそのまま入力ベクトルとして扱うことができる。一方で NTN-Comp は $\mathbb{C}^n \times \mathbb{C}^n$ から \mathbb{R}^k への写像であり, 各ノードは子ノードの出力をそのまま入力ベクトルとすることはできない。そこで, 本稿では再帰型 NTN-Comp について k を $2n$ に設定し, 子ノードの出力ベクトル $\mathbf{y}' = (y'_1, \dots, y'_{2n}) \in \mathbb{R}^{2n}$ を複素ベクトル $\mathbf{y} \in \mathbb{C}^n$ の実部と虚部の連結として扱う。すなわち,

$$(y'_1, \dots, y'_n) = \Re(\mathbf{y}), (y'_{n+1}, \dots, y'_{2n}) = \Im(\mathbf{y}).$$

とする。この手法は, \mathbb{C}^n を \mathbb{R}^{2n} に変形し, 式 (4) を実ベクトル空間上に定義できるという事実から自然に導かれる。

3 巡回行列に基づく写像との関係

巡回行列を用いてニューラルネットワークにおける行列のパラメタ数を削減し, 計算効率を向上させる手法が知られている

表 2: 知識グラフ補完タスクの精度比較.

モデル	MRR		Hits@ (Filter)		
	Filter	Raw	1	3	10
NN	0.111	0.106	7.0	11.7	18.3
NTN	0.754	0.530	69.3	79.5	86.3
NTN-SMD ($m = 1$)	0.243	0.216	15.9	26.1	40.9
NTN-SMD ($m = 2$)	0.224	0.199	15.1	23.8	37.2
NTN-SMD ($m = 3$)	0.299	0.255	20.4	32.4	49.2
NTN-SMD ($m = 10$)	0.533	0.413	42.2	59.4	74.5
NTN-SMD ($m = 25$)	0.618	0.463	52.1	67.8	80.0
NTN-Diag	0.824	0.590	74.8	89.6	92.7
NTN-Comp	0.857	0.610	80.1	90.9	93.1

[3]. $\mathbf{w}^T = (w_1, \dots, w_n)$ による巡回行列

$$C(\mathbf{w}) = \begin{pmatrix} w_1 & w_n & \dots & w_3 & w_2 \\ w_2 & w_1 & \dots & w_4 & w_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n-1} & w_{n-2} & \dots & w_1 & w_n \\ w_n & w_{n-1} & \dots & w_2 & w_1 \end{pmatrix}$$

はフーリエ行列 \mathfrak{F} を用いて, $\mathfrak{F}^{-1} \text{diag}(\mathfrak{F} \mathbf{w}) \mathfrak{F}$ に分解できる. この手法を双線形項の変形に使う場合, 節2.2で説明した正規行列を用いた手法と同等となる. なぜなら, これを式(1)における双線形項に適用すると, $\mathbf{x}_1^T \mathbf{W}^i \mathbf{x}_2 = \mathbf{x}_1^T \mathfrak{F}^{-1} \text{diag}(\mathfrak{F} \mathbf{w}) \mathfrak{F} \mathbf{x}_2 = \Re(\langle \mathbf{x}'_1, \mathbf{w}', \overline{\mathbf{x}'_2} \rangle)$ となるからである. ここで, $\mathbf{x}'_1 = \mathfrak{F} \mathbf{x}_1, \mathbf{x}'_2 = \frac{1}{n} \overline{\mathfrak{F} \mathbf{x}_2}, \mathbf{w}' = \text{diag}(\mathfrak{F} \mathbf{w}) \in \mathbb{C}^n$ である.

4 実験

4.1 知識グラフ補完

各モデルの性能を知識グラフ補完 (Knowledge Graph Completion; KGC) により評価する. KGC とはエンティティ $s, r \in \mathcal{E}$ と二項関係 $r \in \mathcal{R}$ が与えられた時, 3つ組 (s, r, o) で表される事実が真か偽か判定するタスクである. データセットには Wordnet(WN18)を用いる.

本稿では文献 [7] に従い, 3つ組 (s, r, o) に対する評価関数 $\Phi: (s, r, o) \mapsto \mathbb{R}$ を設計することで KGC を行う. 文献 [7] に従い, NTN については評価関数 Φ を

$$\mathbf{u}_r^T f \left(\mathbf{x}_s^T \mathbf{W}_r^{[1:k]} \mathbf{x}_o + \mathbf{V}_r \begin{bmatrix} \mathbf{x}_s \\ \mathbf{x}_o \end{bmatrix} + \mathbf{b}_r \right)$$

とする. ここで \mathbf{u}_r は f の出力を \mathbb{R} に写像するための k 次元ベクトルである. 活性化関数 f 内の写像を変えることでモデル間の性能を比較する. 損失関数には

$$\sum_{i=1}^N \sum_{c=1}^C \max(0, 1 - \Phi(T^i) + \Phi(T_c^i)) + \lambda \|\Omega\|_2^2$$

を用いる. ここで T^i は訓練データ中の i 番目の3つ組を表す. T_c^i は正例 T^i について, C 個ランダムに選ばれる負例のひとつである. Ω はパラメタの集合とする. ハイパーパラメタは先行

表 3: Natural Logic における命題論理の式間の関係.

関係	記号	付値による定義
Entailment	$A \sqsubset B$	$A \subset B$
Reverse entailment	$A \supset B$	$A \supset B$
Equivalence	$A \equiv B$	$A = B$
Alternation	$A B$	$A \cap B = \emptyset \wedge A \cup B \neq \mathcal{D}$
Negation	$A \wedge B$	$A \cap B = \emptyset \wedge A \cup B = \mathcal{D}$
Cover	$A \smile B$	$A \cap B \neq \emptyset \wedge A \cup B = \mathcal{D}$
Independence	$A \# B$	その他

表 4: 式のペアとその関係の例.

$not\ p_3$	\wedge	p_3
$(p_1\ or\ (p_2\ or\ p_4))$	\sqsubset	$(p_3\ or\ p_2)$
$(p_1\ or\ (p_2\ or\ p_4))$	\sqsupset	$(p_2\ and\ not\ p_4)$

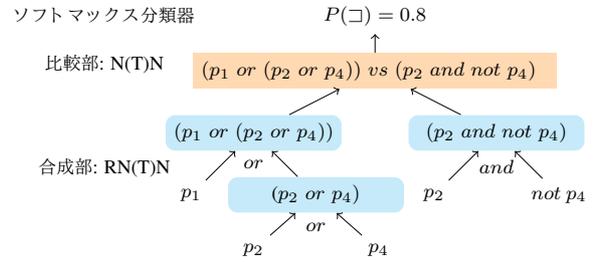


図 1: 合成部と比較部. 否定結合子のついたりテラル $not\ p_4$ は埋め込みベクトルとして扱われる.

研究 [7, 10] に従い, 埋め込みベクトルの次元数 n を 100, 正則化係数 λ を 0.0001 とし 3 次テンソルのスライスサイズ k を 4 とした. エポック数は 100 で訓練し, 最適化には初期学習係数を 0.1 とし Adagrad を使用した.

評価方法は先行研究に従う. 実験結果を表 2 に示す. NTN の精度は NN を大幅に上回っており, KGC における NTN の有効性がわかる. NTN-SMD は分解した行列のランク m が大きくなるほど精度は向上しているが, その精度は NTN に及ばず, 十分な近似とはなっていない. 一方 NTN-Diag は 3 次テンソルの各スライス行列が対称行列であるという強い制約を持つにも関わらず, NTN における過学習を緩和し, 高い精度を達成している. NTN-Comp は制約を真正規行列へと緩めることで, 精度をさらに高められることが確認できる.

4.2 命題論理における推論

文献 [2] に従い, 命題論理の式間の関係を Natural Logic [6] に基づいて分類するタスクで各再帰型モデルの性能を評価する. 命題論理の式間の関係を表 3 に, 具体的な事例を表 4 に示す.

本実験のモデルは図 1 に示すように合成部と比較部からなる. 合成部は左右の式を表すベクトルを RN(T)N により学習する. 比較層では単一層の N(T)N がそれらのベクトルを受け取り, ソフトマックス分類器がその出力を受け取る. 損失関数として L2 正則化付き交差エントロピーを用いる. また, 合成部におい

表 5: 開発データおよび最大論理結合子数別の評価データの正解率. 括弧内の数字は 5 試行の標準偏差を表す.

	開発	1	2	3	4	5	6	7	8	9	10	11	12	平均
# の割合	53.3	56.0	53.0	53.4	53.2	55.9	56.5	56.5	57.8	56.5	57.7	56.8	59.9	56.1
RNN	95.0 (0.8)	98.0	97.5	95.5	93.3	89.9	86.1	82.8	79.9	74.8	73.2	71.8	71.7	84.5
RNTN	97.2 (0.4)	99.9	99.5	98.2	95.7	92.7	88.5	84.7	81.2	78.1	77.5	74.4	74.4	87.0
RNTN-SMD ($m = 1$)	90.1 (3.4)	93.7	92.5	90.9	89.1	86.9	84.1	81.7	79.8	76.1	75.7	75.3	75.1	83.4
RNTN-SMD ($m = 2$)	91.4(4.6)	93.0	93.4	91.7	90.3	88.2	85.5	82.7	81.4	77.6	77.0	75.4	75.8	84.3
RNTN-SMD ($m = 4$)	88.6 (7.1)	90.2	90.3	89.4	87.6	86.0	83.6	81.2	79.6	76.5	75.2	74.6	75.7	82.4
RNTN-SMD ($m = 8$)	82.7(10.2)	86.8	84.9	83.5	82.5	81.1	79.1	76.6	75.6	72.4	71.3	70.9	71.2	77.9
RNTN-SMD ($m = 16$)	81.8 (11.7)	86.6	83.9	82.4	81.4	80.2	78.6	76.5	75.5	73.1	72.7	72.2	73.3	78.0
RNTN-Diag	98.1 (0.1)	99.9	98.9	98.5	97.4	94.9	91.5	87.6	85.0	80.3	78.5	77.1	75.2	88.7
RNTN-Comp	97.5 (0.1)	99.3	98.1	98.0	96.9	94.3	90.6	86.1	83.5	79.2	76.6	74.5	74.6	87.6

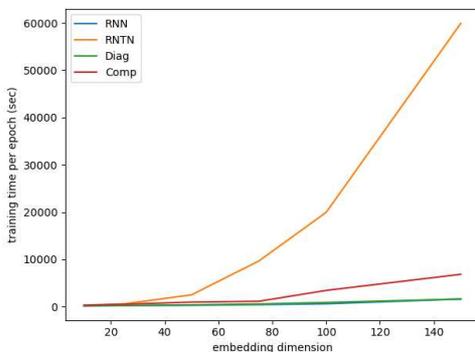


図 2: 1 エポックにかかる訓練時間の比較.

て論理結合子 *and* と *or* に別のパラメタを用意した.

本稿では式を選言標準系または連言標準系に限定し, 訓練, 開発, 評価データを生成した. 事例数はそれぞれ 62, 589, 13, 413, 55, 150 である. 訓練, 開発データの式は最大 4 個の論理結合子を含み, 評価データでは最大 12 個まで含む. データセット中のすべての式は共有する 6 個の命題変数のうち最大 4 個で構成される. ハイパーパラメタ及び最適化手法は文献 [2] に従い, 埋め込みベクトルの次元 n は 25 に (NN については 45), 比較層の出力ベクトルの次元 k は 75 に設定した. 最適化手法には Adadelta を用い, 正則化係数 λ は $\{0.0001, 0.0003, 0.0005, 0.0007, 0.0009, 0.001\}$ から探索した. 各正則化係数についてパラメタの初期化と訓練事例のシャッフルをランダムに 5 回行い, 正解率の平均が高いものを選んだ.

開発データ及び評価データの実験結果を表 5 に示す. 平均としては RNTN-Diag が最も良く, 次いで RNTN-Comp が良い精度を示している. 論理結合子 *and*, *or* が対称性を持つため, NTN-Diag でも十分な精度が得られたと考えられる. RNTN-SMD については, いずれの m に対しても平均としては RNTN を下回っている. さらに, RNTN-SMD の開発データの精度は試行によってばらつきが大きく, パラメタの初期化や訓練事例のシャッフルに対して不安定であった.

RNTN-Diag, RNTN-Comp の RNTN, RNN に対する CPU における訓練時間の比較を図 2 に示す. $O(n^2k)$ のパラメタ数

を持つ RNTN は埋め込みベクトルの次元数が上がるにつれ, 二次関数的に訓練時間が増加している. 一方で RNTN-Diag, RNTN-Comp はパラメタ数が $O(nk)$ であるため, RNN と近い訓練時間に抑えられている.

5 まとめ

(R)NTN の 3 次テンソルの各スライス行列に制約を課し, 固有値分解を行うことで大幅にパラメタ数を削減したモデルを提案した. 知識グラフ補完および命題論理における推論の実験により, 提案モデルはもともとの (R)NTN より高い精度を達成でき, 高速な学習が可能であることがわかった. 今後の課題として, 提案した RNTN の表現力を多様な構成性を持つデータで検証することが挙げられる.

参考文献

- [1] Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Corinna Cortes, and Mehryar Mohri. Polynomial semantic indexing. In *NIPS*, 2009.
- [2] Samuel R Bowman, Christopher Potts, and Christopher D Manning. Recursive neural networks can learn logical semantics. *CVSC*, 2015.
- [3] Yu Cheng, Felix X Yu, Rogerio S Feris, Sanjiv Kumar, Alok Choudhary, and Shi-Fu Chang. An exploration of parameter redundancy in deep networks with circulant projections. In *ICCV*, pages 2857–2865, 2015.
- [4] Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. In *ICML*, pages 2168–2178, 2017.
- [5] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *IJCAI*, pages 1284–1290, 2015.
- [6] Bill MacCartney and Christopher D Manning. An extended model of natural logic. In *IWCS*, pages 140–156, 2009.
- [7] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934, 2013.
- [8] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013.
- [9] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080, 2016.
- [10] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *ICLR*, 2015.