ニューラルネットワークの潜在表現について

— Word2Vec による単語埋込 —

後編

本谷 秀堅

名古屋工業大学

再掲：言語データ

言語データ

- シンボルの系列
- 離れたシンボル間に強い関係
- シンボル間の非対称な関係（例：対義語・類義語・包含関係）

単語を特徴ベクトルへと変換

- 単語をベクトルで表現する。
- AI にとって都合の良いベクトルで表現したい

AI にとって都合の良い表現

似た概念の単語は似たベクトルのほうが都合が良い

例えば RNN による品詞の推定。最後の層は MLP+Softmax 関数

AI にとって都合の良い表現

似た概念の単語は似たベクトルのほうが都合が良い

例えば RNN による会話生成。最後の層は MLP+Softmax 関数

One-hot-vector の不便さ

各単語を語彙数 K 次元の one-hot-vector で表現する
以下の例では $K = 10$

$$\boldsymbol{x}_{\text{pen}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_{\text{spoon}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{x}_{\text{walk}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \boldsymbol{x}_{\text{run}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

次のことに気付く

- どのベクトル間の距離も $\sqrt{2}$
- どのベクトルも互いに直交

単語間の関係が距離にも方向にも反映されていない

AI に都合の良いベクトル

K より低次元で、方向が概念を表すベクトル
例えば

$$\boldsymbol{x}_{\text{pen}} = \begin{bmatrix} 0.2 \\ -0.6 \\ 0.3 \end{bmatrix}, \quad \boldsymbol{x}_{\text{spoon}} = \begin{bmatrix} 0.2 \\ -0.7 \\ 0.3 \end{bmatrix}, \boldsymbol{x}_{\text{walk}} = \begin{bmatrix} -0.8 \\ 0.0 \\ 0.2 \end{bmatrix}, \boldsymbol{x}_{\text{run}} = \begin{bmatrix} -0.7 \\ 0.1 \\ 0.3 \end{bmatrix}$$

- 「pen」と「spoon」は似た方向
- 「walk」と「run」は似た方向
- pen, spoon と walk, run は互いに離れている。

単語間の関係が方向・距離に表れている
異なる単語でも概念が近いと Softmax 関数への入力も近い値に

- 都合の良いベクトルを単語ごとに学習で求める

自己教師学習 (self-supervised learning)

自己教師学習により（AI に都合の良い）表現を獲得する

- 教師あり学習の一種
- 教師信号を自動生成する（多量のデータで学習可）

- プレテキスト タスク（pretext task)
  - 文章の一部を隠して、残りから予測
  - 文章の後半を隠して、前半から予測
  - 画像の一部を隠して、残りから予測
  - カラー画像を白黒にして、カラーを復元
  - ...

想定する AI のタスク：識別や回帰

単語ベクトルの自己教師学習

Word2Vec (2013)

- CBoW (Continuous Bag-of-Words)
  - 文中で前後の単語から真ん中の単語を予測
- Skip-Gram
  - 文中の各単語について前後の単語を予測

以下の説明は"Rong, word2vec Parameter Learning Explained, 2012, `https://arxiv.org/abs/1411.2738` による

CBoW: Continuous Bag of Words

集合 ← カバン　　単語

I swam across the river to get to the other bank.

プレテキストタスク
前後2単語から、中央の単語を予測せよ

学習データ 1

across the [redacted] to get

x_1 x_2 t x_3 x_4

学習データ 2

the river [redacted] get to

x_1 x_2 t x_3 x_4

CBoW のアーキテクチャ

CBoW のアーキテクチャ (シンプル版)

入力が 1 単語だけだと、こうなる

- 1 層目も 2 層目も全結合（MLP)
- V: 語彙数, D: 中間層のユニット数

式表現（シンプル版）

1 層目から 2 層目

$$\boldsymbol{h} = \boldsymbol{W}\boldsymbol{x}$$

2 層目から 3 層目

$$\boldsymbol{u} = \boldsymbol{W}'\boldsymbol{h}$$

$$y_j = \mathrm{SM}_j(\boldsymbol{u}) = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})}$$

出力層の上から j 番目 = 辞書の上から j 番目の単語

単語数

事後確率の推定値

前段の全結合層

1つ目のMLP

結線の重みの行列表現

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \boldsymbol{w}_2^\top \\ \vdots \\ \boldsymbol{w}_D^\top \end{bmatrix} \quad \text{where} \quad \boldsymbol{w}_i^\top = [w_{i1}, w_{i2}, \ldots, w_{iV}]$$

恒等

- 中間層の活性化関数は~~口頭~~写像 $f(u) = u$
- 中間層の i 番目のユニットの出力 h_i (i=1, 2, . . . , D)

$$h_i = \boldsymbol{w}_i^\top \boldsymbol{x}$$

V 次元の one-hot ベクトル $\boldsymbol{x}$ が D 次元のベクトル $\boldsymbol{h}$ に変換された

$(D \ll V)$

$$\boldsymbol{h} = \underbrace{[h_1, h_2, \ldots, h_D]^\top}_{\text{実数値へ.}}$$

0,1 ⟶ 実数値へ.

後段の全結合層

結線の重みの行列表現

2つ目のMLP

$$\boldsymbol{W}' = \begin{bmatrix} (\boldsymbol{w}'_1)^\top \\ (\boldsymbol{w}'_2)^\top \\ \vdots \\ (\boldsymbol{w}'_V)^\top \end{bmatrix} \quad \text{where} \quad (\boldsymbol{w}'_j)^\top = [w'_{j1}, w'_{j2}, \ldots, w'_{jD}]$$

最終層の j 番目のユニットの内積計算

$$u_j = (\boldsymbol{w}'_j)^\top \boldsymbol{h}$$

最終層の活性化関数はソフトマックス関数（多クラス識別）

$$y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

y_j は隠された単語が、辞書の j 番目である事後確率の推定値

$$y_j = P(t_j = 1|\boldsymbol{x})$$

学習の損失関数　最尤推定 (W, W')

各問題に対する尤度 $L(\boldsymbol{W}, \boldsymbol{W}')$

語い数

隠した単語のみ 1 あとは 0

$$L(\boldsymbol{W}, \boldsymbol{W}') = \prod_{j=1}^{V} P(t_j = 1 | \boldsymbol{x})^{t_j}$$

前後と単語

辞書の j^* 番目の単語が正解だったとする。

$$t_j = \begin{cases} 1, & j = j^*, \\ 0, & j \neq j^*. \end{cases}$$

損失関数は交差エントロピー（下記を全学習データについて足す）

$$E(\boldsymbol{W}, \boldsymbol{W}') = -\log L = -\sum_{j=1}^{V} t_j \left(u_j - \log \sum_{j'=1}^{V} \exp(u_{j'}) \right)$$

損失関数の微分 (1/4)

（再掲）損失関数は交差エントロピー

$$E(\boldsymbol{W}, \boldsymbol{W}') = -\log L = -\sum_{j=1}^{V} t_j \left(u_j - \log \sum_{j'=1}^{V} \exp(u_{j'}) \right)$$

$t_{j^*} = 1$

正解は $j = j^*$

$$= -\left(u_{j^*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) \right)$$

損失関数の微分 (2/4)

正解で1 他0　各単語の事後確率の推定値

最終層の内積で微分

$$\frac{\partial E}{\partial u_j} = -(t_j - y_j) \equiv e_j$$

正解と推定値のズレ

最終の結線の重み：$(\boldsymbol{w}'_j)^\top = [w'_{j1}, w'_{j2}, \ldots, w'_{jD}]$

$u = w'h$

$$\frac{\partial E}{\partial w'_{ji}} = \frac{\partial E}{\partial u_j}\frac{\partial u_j}{\partial w'_{ji}} = e_j \cdot h_i$$

Eが増える方向

勾配

出力層 j番目のエラー

学習率を η で表すと

$$(w'_{ji})^{(\text{new})} = (w'_{ji})^{(\text{old})} - \eta \cdot e_j \cdot h_i$$

中間層 i番目の出力

後段のパラメータの更新式

$$(\boldsymbol{w}'_j)^{(\text{new})} = (\boldsymbol{w}'_j)^{(\text{old})} - \eta \cdot e_j \cdot \boldsymbol{h}$$

損失関数の微分 (3/4)

中間層の出力 h_i で微分（中間層から最終層は全結合）

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} e_j w'_{ji}$$

中間層は入力の $\boldsymbol{W}$ による線形変換： $h_i = \boldsymbol{w}_i^\top \boldsymbol{x}$

$$\frac{\partial E}{\partial w_{ik}} = \frac{\partial E}{\partial h_i} \frac{\partial h_i}{\partial w_{ik}} = \sum_{j=1}^{V} e_j w'_{ji} \cdot x_k$$

$h = W^T x$

損失関数の微分（4/4）

行列の形で表すことにすると

$$\frac{\partial E}{\partial \boldsymbol{W}} = \frac{\partial E}{\partial \boldsymbol{h}} \boldsymbol{x}^\top \in \mathbb{R}^{V \times D}$$

ただし、$\boldsymbol{x} \in \mathbb{R}^V$, $\partial E / \partial \boldsymbol{h} \in \mathbb{R}^D$。$\boldsymbol{x}$ は一つの成分だけが 1 であとはゼロ。

前段のパラメータの更新式

$$\boldsymbol{W}^{(\mathrm{new})} = \boldsymbol{W}^{(\mathrm{old})} - \eta \frac{\partial E}{\partial \boldsymbol{W}}$$

$D \times V$ の行列 $\boldsymbol{W}$ のうち、$\boldsymbol{x}$ が非ゼロの成分に対応する列ベクトルだけが更新される。

学習により
j番目の単語の
埋め込みベクトル
を求めている。

x
h
w_{1i}
w_{2i}
w_{3i}
更新されるのは
その単語の
埋め込みベクトル
$\frac{\partial E}{\partial W} =$
$\frac{\partial E}{\partial h}$
x^T
正解
=
ϕ
ϕ

new
OLD
w
=
w
- η
∅
∅
更新

CBoW のアーキテクチャ（正式版: C 単語入力）

Bag

集合

単語の順序は無関係

コンテクスト（文脈）

前後と単語を

とんとんの単語の埋め込みの平均で表現

$$h = \frac{1}{c}(h_1 + h_2 + \cdots + h_c)$$

式表現（正式版: C 単語入力

$h = \frac{1}{C}(Wx_1 + Wx_2 + \cdots + Wx_C)$

1 層目から 2 層目

$$\boldsymbol{h} = \frac{1}{C}\boldsymbol{W}\underbrace{(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \ldots, \boldsymbol{x}_C)}_{\text{全単語}}$$

2 層目から 3 層目

1単語入力と同じ

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{W}'\boldsymbol{h} \\ y_j &= \mathrm{SM}_j(\boldsymbol{u}) = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})} \end{aligned}$$

y_j は隠された単語が、辞書の j 番目である事後確率の推定値

$$y_j = P(t_j = 1 | \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_C)$$

学習の損失関数

各問題に対する尤度 $L(\boldsymbol{W}, \boldsymbol{W}')$

$$L(\boldsymbol{W}, \boldsymbol{W}') = \prod_{j=1}^{V} P(t_j = 1 | \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_C)^{t_j}$$

以下は入力が１単語のときと同じ

$$E(\boldsymbol{W}, \boldsymbol{W}') = -\left(u_{j^*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) \right)$$

損失関数の微分 (3′/4)

中間層の出力 h_i で微分（中間層から最終層は全結合）

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} e_j w'_{ji}$$

中間層は入力の $\boldsymbol{W}$ による線形変換： $h_i = \boldsymbol{w}_i^\top (\boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_C)/C$

$$\frac{\partial E}{\partial w_{ik}} = \frac{\partial E}{\partial h_i} \frac{\partial h_i}{\partial w_{ik}} = \frac{1}{C} \sum_{j=1}^{V} e_j w'_{ji} \cdot (x_{1k} + x_{2k} + \cdots + x_{Ck})$$

ただし、$\boldsymbol{x}_\ell = [x_{\ell 1}, x_{\ell 2}, \ldots, x_{\ell V}]^\top$

損失関数の微分（4′/4）

行列の形で表すことにすると

x^T（1単語）

$$\frac{\partial E}{\partial \boldsymbol{W}} = \frac{1}{C}\frac{\partial E}{\partial \boldsymbol{h}}\underbrace{(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_C)^\top}_{(C\text{単語})} \in \mathbb{R}^{V\times D}$$

Bagの中の単語の数

ただし、$\boldsymbol{x}_\ell \in \mathbb{R}^V$, $\partial E/\partial \boldsymbol{h} \in \mathbb{R}^D$。$\boldsymbol{x}_\ell$ は一つの成分だけが 1 であとはゼロ。

前段のパラメータの更新式

$$\boldsymbol{W}^{(\mathrm{new})} = \boldsymbol{W}^{(\mathrm{old})} - \eta\frac{\partial E}{\partial \boldsymbol{W}}$$

$D \times V$ の行列 $\boldsymbol{W}$ のうち、$\boldsymbol{x}_\ell$ $(\ell = 1, 2, \ldots, C)$ が非ゼロの成分に対応する列ベクトルだけが更新される。(入力された C 個の単語に対応する $\boldsymbol{W}$ の列ベクトルそれぞれが、同じ量だけ $(\partial E/\partial \boldsymbol{h})$ 更新される。)

CBoW で得られた単語の埋込

I have a

単語間の類似度

単語 1 と単語 2 の埋め込みベクトルへの変換

$$\boldsymbol{h}_1 = \boldsymbol{W}\boldsymbol{x}_1, \quad \boldsymbol{h}_2 = \boldsymbol{W}\boldsymbol{x}_2$$

単語 1 と単語 2 のコサイン類似度

$$S(\mathrm{word}_1, \mathrm{word}_2) = \cos\theta_{12} = \frac{\boldsymbol{h}_1^\top \boldsymbol{h}_2}{\|\boldsymbol{h}_1\|\|\boldsymbol{h}_2\|}$$

単語の埋込ベクトル ($\boldsymbol{h}$)

$\cos\theta$ が大 → 似ている.

- この例では $\boldsymbol{h}$ の次元は $D = 500$

King

Queen

単語間の類似度

Word Similarity Matrix Heatmap
king
queen
1
0.65
0.65
1
king
queen
1.00
0.95
0.90
0.85
0.80
0.75
0.70

単語の埋込ベクトル ($\boldsymbol{h}$)

単語間の類似度

Word Similarity Matrix Heatmap
king
queen
boy
girl
man
woman
father
mother
rabbit
1 0.65 0.25 0.17 0.23 0.13 0.25 0.16 0.18
0.65 1 0.21 0.35 0.17 0.32 0.15 0.29 0.21
0.25 0.21 1 0.85 0.68 0.6 0.57 0.59 0.26
0.17 0.35 0.85 1 0.59 0.75 0.44 0.65 0.24
0.23 0.17 0.68 0.59 1 0.77 0.42 0.4 0.25
0.13 0.32 0.6 0.75 0.77 1 0.38 0.61 0.22
0.25 0.15 0.57 0.44 0.42 0.38 1 0.79 0.11
0.16 0.29 0.59 0.65 0.4 0.61 0.79 1 0.18
0.18 0.21 0.26 0.24 0.25 0.22 0.11 0.18 1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Boy
King
Queen
Boy
Girl
Man
Woman
Father
Mother
Rabbit

単語の埋込ベクトル ($\boldsymbol{h}$)

単語間の類似度

Japan
Tokyo
Korea
Seoul
France
Paris
Germany
Berlin
Nagoya
rabbit

単語の埋込ベクトル ($\boldsymbol{h}$)

単語間の類似度

黒白青赤 話歌聞聞 見聞 見聞 歌聞

考察：Continuous Bag of Words は何をしたのか

- コンテクスト（文脈）：与えられた文章全体
  - CBoW では前後あわせて C 単語 $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_C)$ がコンテクスト

分布仮説

単語の意味はその単語の周辺に現れる単語、すなわちコンテクストによって決まっている。

- I swam across the river to get to the other bank.
- I walked across the road to get cash from the bank.

「コンテクスト」もベクトルで表現して、単語の予測に使った。

脱線：ベクトルの表現能力

1 本の 500 次元のベクトルが「コンテクスト」を表現できるのか。

- 500 次元の 2 値ベクトル 1 本で表現できる量

$$2^{500} \quad \text{通り}$$

- 参考：地球を構成する原子の総数の概算

$$8.2 \times 2^{51} \quad \text{個}$$

考察：Continuous Bag of Words は何をしたのか

- CBoW では全単語ベクトルの平均でコンテクストを表現していた
  - Bag of Words: 単語群を入れたカバン（bag)
  - 単語の順序は気にしない

$$\boldsymbol{h} = \frac{1}{C}\boldsymbol{W}(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_C) = \frac{1}{C}(\boldsymbol{W}\boldsymbol{x}_1 + \boldsymbol{W}\boldsymbol{x}_2 + \cdots + \boldsymbol{W}\boldsymbol{x}_C)$$

  - 各単語の埋め込みベクトルの平均でコンテクストを表現

$$= \frac{1}{C}(\boldsymbol{h}_1 + \boldsymbol{h}_2 + \cdots + \boldsymbol{h}_C)$$

- 単語の one-hot-vector の平均でコンテクストを表現する場合と比較せよ。

考察：Continuous Bag of Words は何をしたのか

- CBoW はマスクされた単語をコンテクストから予測
  - コンテクストは埋め込みベクトルの和で表現
- 予測は線形多クラス識別

$$y_j = \mathrm{SM}_j(\boldsymbol{u}) = \frac{\exp((\boldsymbol{w}'_j)^\top \boldsymbol{h})}{\sum_{j'=1}^{V} \exp((\boldsymbol{w}'_{j'})^\top \boldsymbol{h})}$$

- コンテクストの埋め込みベクトル $\boldsymbol{h}$ を用いれば単語予測が線形識別可となるように学習

考察：Continuous Bag of Words で何ができたのか

- 関係の強い単語どうしが同じ方向を向く
- アナロジーがベクトルの加減算で解けるように配置

例：日本における東京はフランスにおける何か？

$$\boldsymbol{h}_{\text{Paris}} = \underline{\boldsymbol{h}_{\text{Tokyo}} - \boldsymbol{h}_{\text{Japan}}} + \boldsymbol{h}_{\text{France}}$$

seq2sec

RNN を使ったコンテクストの表現

Seq2Seq: RNN で系列変換

- 前段の RNN: エンコーダ
  - 入力文を特徴ベクトルに
- 後段 RNN: でコータ
  - 入力文の特徴ベクトルから異なる系列を自己復号

https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

復習：系列変換

例題：自動翻訳　「This is a pen」→ 「これはペンです」

- 入力：日本語の単語系列: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$
- 出力：英語の単語系列: $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$

翻訳のために $p(\boldsymbol{y}|\boldsymbol{x})$ を求めたい。Seq2Seq で実現

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$$

条件付き言語モデル：

$$p(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T|\boldsymbol{x}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{<t}, \boldsymbol{x})$$

- 交差エントロピーを最小にするように学習
- エンコーダは入力文 $\boldsymbol{x}$ を 1 つのベクトル $\boldsymbol{z}$ へと変換する
- デコーダは $\boldsymbol{z}$ を入力とし、出力文 $\boldsymbol{y}$ を出力する

RNN による文章の符号化

隠れ変数 z は入力された文章のベクトルによる表現

https://arxiv.org/pdf/1409.3215.pdf (Fig.2)

意味の似た文章は似た方向のベクトルへと変換される

RNN による文章の符号化

RNN による文章=コンテクスト符号化の懸念事項

- 文章が長くなると、出力特徴ベクトルに文冒頭の影響が、なかなか及ばない