単語分散表現, word2vec

白坂貴規

平成 32 年 3 月 12 日

1　自然言語と単語の表現

コンピュータに自然言語を理解させる手法には三種類あり, シソーラスによる手法, カウントベースの手法, 推論ベースの手法である.

1.1　シソーラス

シソーラスとは, (基本的には) 類似辞書であり「同じ意味の単語」や「意味の似た単語」が同じグループに分類されている. また, 自然言語処理 (NLP) において利用されるシソーラスでは単語間で「上位と下位」「全体と部分」などのより細かい関連性が定義されている場合がある. NLP において最も有名なシソーラスは 1985 年にプリンストン大学で開発がスタートした WordNet である.

1.2　シソーラスの問題点

シソーラスには多くの問題点がある.

- 時代の変化に対応するのが困難
- 人の作業コストが高い
- 単語の細かなニュアンスを表現できない

これらの理由に対処したのが, 以下のカウントベースの手法と推論ベースの手法である.

2　カウントベースの手法

2.1　単語の分散表現

単語の意味を捉えたベクトル表現を単語の分散表現とよぶ. これは, 単語の意味は周辺の単語によって形成されるという分布仮説によって成り立っている. たとえば「I drink water.」「We drink water.」のように drink の周辺には飲み物が出現する. 周辺の単語を左右何単語まで含めるかをウィンドウサイズとよぶ. ここでは「You say goodbye and I say hello.」という具体例を用いてコンテキストに含まれる単語の頻度を数える.

	you	say	goodbye	and	i	hello	.
you	0	1	0	0	0	0	0
say	1	0	1	0	1	1	0
goodbye	0	1	0	1	0	0	0
and	0	0	1	0	1	0	0
i	0	1	0	1	0	0	0
hello	0	1	0	0	0	0	1
.	0	0	0	0	0	1	0

図 1: 共起行列の表

これは共起行列と呼ばれ, それぞれの行を参照することにより各単語のベクトルを得ることができる.

2.2　ベクトル間の類似度

様々な手法があるが, 単語のベクトル表現の類似度に関しては次式で定義されるコサイン類似度がよく用いられる.

$$similarity(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|} \tag{1}$$

2.3　単語の分散表現の改良

以上の共起行列では単に共起する回数に着目しているので,「drive」と「car」よりも「the」と「car」の方が強い関連性を持ってしまう. これを改善するために以下で定義する相互情報量 (Pointwise Mutual Information) という指標を使う.

$$PMI(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} \tag{2}$$

ここで, $P(x)$ は x が起こる確率, $P(y)$ は y が起こる確率, $P(x,y)$ は x と y が同時に起こる確率を表す. また, PMI はその値が高いほど関連性が高いことを示す. 例えば 10000 個の単語からなるコーパスで「the」が 1000 回,「car」が 20 回,「drive」が 10 回出現し,「the」と「car」が 10 回共起し,「drive」と「car」が 5 回共起したとする. その時, PMI(”the”, ”car”) ≈ 2.32, PMI(”car”, ”drive”) ≈ 7.97 となり正しく評価できている. また, 実践上では負の PMI は 0 にする正の相互情報量を用いる.

2.4 次元削減

次元削減には様々な手法があるがここでは特異値分解 (Singular Value Decomposition: SVD) を行う. 任意の行列 $\boldsymbol{X}$ を, $\boldsymbol{U}$, $\boldsymbol{S}$, $\boldsymbol{V}$ の 3 つに分解する. つまり,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{3}$$

ここで U と V は直行行列でありその列ベクトルは互いに直交する. S は対角行列であり, 対応する軸の重要度を示す特異値が降順に並んでいるので特異値が大きいものに対応する軸のみ残すことで次元削減が可能になる.

2.5 カウントベースの手法の問題点

現実的にはコーパスで扱う語彙数は非常に巨大で語彙数が 100 万をゆうに超えると言われている. その際カウントベースの手法では 100 万 ×100 万の巨大な行列をつくることになり, $n \times n$ の行列に対して $O(n^3)$ の計算量がかかる特異値分解は現実的ではない.

3 推論ベースの手法

推論ベースの手法では, 周囲の単語 (コンテキスト) が与えられたときにどのような単語が出現するかを推測し, 学習する. その学習の結果として単語の分散表現を得られるというのが推論ベースの手法である.

3.1 word2vec

単語は one-hot ベクトル化することで入力されるニューロンを固定長のベクトルにすることが可能. これをモデルへの入力とするが, continuous bag-of-words(CBOW) と呼ばれるモデルを使う. 以下にその概要を示した.

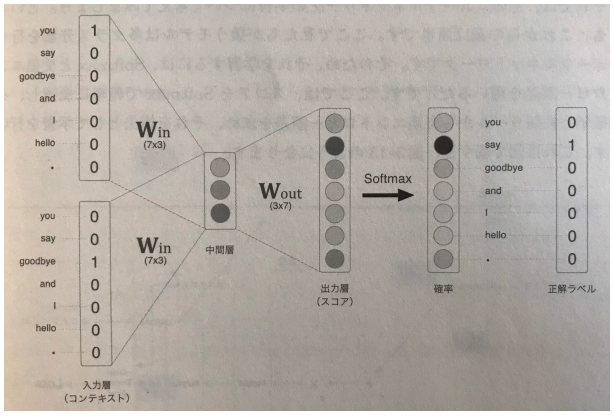

図 2: CBOW モデルのネットワーク構造

注意すべきは中間層にあるニューロンは各入力層の全結合による変換後の値が相加平均されていることである. 最終的な出力は各単語のスコアであり, Softmax 関数を適用することで単語の出現確率が求められる. 損失関数は, 多くの場合交差エントロピー誤差を用いる. 最終的に利用する単語の分散表現は, 多くの研究では入力側の重み W_{in} だけを利用する.

3.2 CBOW モデルの逆伝播

CBOW モデルの逆伝播を図解すると以下の通りである.

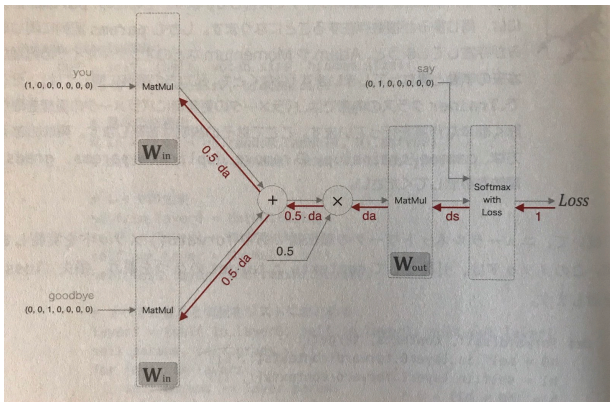

図 3: CBOW モデルの逆伝播

Softmax ノードと交差エントロピー誤差を結合した場合, 逆伝播は出力層のインデックスを k とすると,

$$ds = (y_k - t_k) \tag{4}$$

となる.[2][3] よって, 中間層のインデックスを j とすると, $\boldsymbol{W_{out}}$ の誤差逆伝播は,

$$\frac{\partial E}{\partial w_{jk}^{out}} = x_j(y_k - t_k) \tag{5}$$

である. [3] また,

$$\frac{\partial E}{\partial x_j} = \sum_k (y_k - t_k) w_{jk}^{out} \tag{6}$$

である [3]. これを用いて

$$\frac{\partial E}{\partial W_{in}} = 0.5 \times \frac{\partial E}{\partial X} \times W_{in}^T \tag{7}$$

で求められる.

3.3 CBOW モデルの確率的表現

コーパスを $w_1, w_2, \ldots, w_T$ で表現するとする. コンテキストとして w_{t-1}, w_{t+1} が与えられたときに, ターゲットが w_t となる確率は

$$P(w_t | w_{t-1}, w_{t+1}) \tag{8}$$

で表される. これは「w_{t-1}, w_{t+1} が与えられたときに w_t が起こる確率」を表しているので CBOW モデルを表現しているといえる. 損失関数である交差エントロピーも $L = -\sum_k t_k \log y_k$ であり, t_k が one-hot ベクトルの教師ベクトルであることに注意すると

$$L = -\log P(w_t | w_{t-1}, w_{t+1}) \tag{9}$$

となる. これは単に確率に対数をとって -1 倍したものである. これを一つのサンプルデータからコーパス全体に拡張することで損失関数は以下のようになる.

$$L = -\frac{1}{T} \sum_{t=1}^{T} \log P(w_t | w_{t-1}, w_{t+1}) \tag{10}$$

CBOW モデルの学習で行うことは, この損失関数をできる限り小さくすることである. そしてその時の重みパラメータが目的とする単語の分散表現である.

3.4 skip-gram モデル

word2vec では CBOW モデルの他に skip-gram モデルが提案されている. CBOW モデルは周囲の単語から中央の単語を予測するのに対し, skip-gram モデルでは中央の単語から周囲の単語を予測する.

CBOW モデルと同様に確率の表記で表現する. まず, 単語の予測は次の確率で表現される.

$$P(w_{t-1}, w_{t+1} | w_t) = P(w_{t-1} | w_t) P(w_{t+1} | w_t) \tag{11}$$

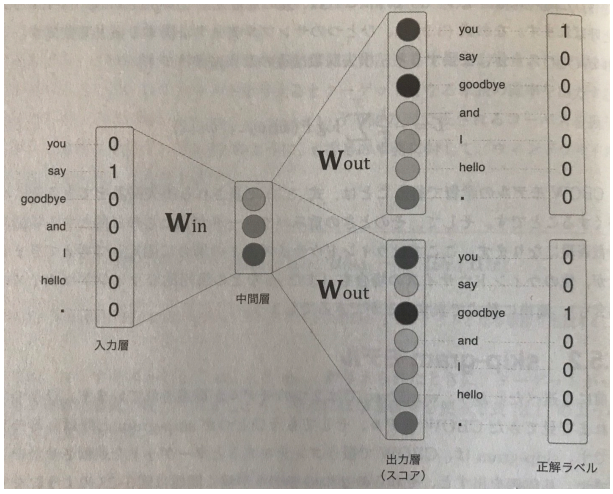

図 4: skip-gram モデルのネットワーク構造

これを交差エントロピー誤差に適用することで skip-gram モデルの損失関数が次のように導ける.

$$\begin{aligned} L &= -\log P(w_{t-1}, w_{t+1} | w_t) \\ &= -\log P(w_{t-1} | w_t) P(w_{t+1} | w_t) \\ &= -(\log P(w_{t-1} | w_t) + \log P(w_{t+1} | w_t)) \end{aligned} \tag{12}$$

これをコーパス全体に拡張することにより

$$L = -\frac{1}{T} \sum_{t=1}^{T} (\log P(w_{t-1} | w_t) + \log P(w_{t+1} | w_t)) \tag{13}$$

が導かれる. コーパスが大規模になるにつれて低頻出の単語や類推の性能の点において skip-gram モデルの方が優れている. 一方で, 学習速度の点では CBOW モデルの方が優秀である.

4 カウントベース v.s. 推論ベース

単語の分散表現の更新作業が発生した場合, カウントベースの手法はゼロから計算を行う必要があるのに対し, 推論ベースはパラメータを再学習するときに初期値を以前のものに設定することで効率的に学習が行える. カウントベースの手法では主に単語の類似性がエンコードされる推論ベースの手法では単語の類似性に加えて複雑な単語間のパターンも捉えられる (word2vec は「king - man + woman = queen」が解ける.) しかし, 推論ベースとカウントベースは優劣がつけられれないことが報告されている.

参考文献

[1] 斎藤康毅. ゼロから作る Deep Learning 2. p57-129

[2] 【機械学習】誤差逆伝播法による速度改善（その2）. https://qiita.com/m-hayashi/items/fa4749f8080e542787d2

[3] 【機械学習 誤差逆伝播法】word2vec メモ (1) https://qiita.com/sand/items/85ea76f9c26aabb849e7

[4] Improving Distributional Similarity with Lossons Learned from Word Embeddings https://www.aclweb.org/anthology/Q15-1016/