

卒業論文

計算問題の特徴分布に基づく類題選出による
自己学習支援

Self Learning Support by Automatic Selection of
Calculation Exercises based on Feature Distribution of
Exercises

成蹊大学理工学部情報科学科

S152114 宮地 雄也

要旨

目次

第 1 章	序論	1
第 2 章	分散表現	3
2.1	Continuous Bag-of-Words Model	4
2.2	Skip Gram	5
第 3 章	LSTM(Long short-term memory)	6
第 4 章	提案手法（章題は変える）	7
4.1	システム全体の流れ	7
4.2	計算式の特徴量抽出	7
4.3	判定機	8
第 5 章	結果とその検討	9
5.1	実験	9
5.2	評価	10
5.3	分析	10
5.4	検討	10
第 6 章	関連研究	11
第 7 章	結論と今後の課題	12
	参考文献	14

第 1 章

序論

昨今、小・中学生の理系離れが問題視されている。平成 30 年度全国学力・学習状況調査（全国学力テスト）の結果では平均正答率は小学校では算数 B が 51.7%，中学校数学では 47.6% とどちらも最も低く、ついで国語，理科の順で正答率が低い。小中どちらとも理系教科の習熟度が低いことを示している。この要因の一つに，数学は一つの計算方法が様々な分野に横断していくことが一度，苦手を生んでしまったらそこからの分野の理解度が下がり，次の分野での応用がきかないために連鎖的に苦手が蓄積することが原因ではないかと考えた。各単元のちょっとした積み残しが，後々，尾を引いていることが全国学力テストの結果から見て取れる。この状況を打破するには子供一人一人の苦手と向き合い，苦手と感じる前に理解していくしかない。しかしながら，生徒と向き合うべき教師の労働時間は過酷を極めており，ベネッセ教育総合研究所の調査ではし小中高の教員の指導時間は増加の一途を辿っている。[2] 下の表は [2] での調査の結果の抜粋である。(1.1)

	調査年	25 歳以上	26～30 歳	31～40 歳	41～50 歳	51～60 歳
出勤時間	2010	7:44	7:43	7:44	7:42	7:42
	2016	7:44	7:43	7:44	7:42	7:42
退勤時間	2010	19:30	19:40	19:10	18:57	18:31
	2016	20:00	19:54	19:26	19:05	18:46
学校にいる時間	2010	11 時間 46 分	11 時間 57 分	11 時間 26 分	11 時間 15 分	10 時間 49 分
	2016	12 時間 26 分	12 時間 18 分	11 時間 46 分	11 時間 26 分	11 時間 06 分

表 1.1 出勤時刻・退勤時刻・学校にいる時間（平均時間、経年比較（教員年齢別（公立全体）））

表 1.1 によると，教員の労働時間は 2010 年に比べて 2016 年の方が各年次とも増加しており，教員のやるが増えている一方で，主であるはずの教材研究や教務準備に時間が避けていないことを示している。この状況では先生が生徒一人一人に時間をさき，指導することは難しい現状がつづいている。

この打開策として，IT 技術駆使した個人別最適化学習に注目が集まっている，しかし，教育の情報は，生徒の情報と結びついている個人情報なためオープン化できず，現在でているサービスでは各サービス利用者の利用状況からデータを取得し，その運用に利用しているため一部の大手企業が情報を独占している。そこで個人の統計データではなく，解く数式の方に着目し，計算式自体の特徴を抽出し，間違えた問題と同様の特徴を持つ問題が復習する類題として最適なのではないかという仮定のもと，本論文では数式の特徴を掘むために自然言語処理の分野で使われる分散表現を適用し，さらに再起ニューラルネットワークを用いて数式ベクトルを作り出すことを目標とし，そ

のベクトルを用いて実際に復習問題生成を行った.

第 2 章

分散表現

自然言語処理ではコンピュータで演算するために各単語を判別するために単語一つ一つを onehot ベクトルというものに置き換える．onehot ベクトルとはある語彙数 V の文章の中の単語 w_i がある時，次元数 V のベクトルに対し i 番目の要素のみ 1 で残りが全て 0 になっているベクトルのことである．(2.1)

$$\mathbf{A} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad (i = 3 \text{ の時}) \quad (2.1)$$

これにより単語一つ一つを別々のベクトルとして区別して表記することができる．しかし onehot ベクトルには問題点があり，一つ目として，ある単語の語彙数 V が増加すると比例して onehot ベクトルの次元 V も大きくなり，処理に時間がかかる点がある．また二つ目として onehot ベクトルでは情報が 0 か 1 しかないので疎なベクトルができる．疎なベクトルとは次元数が大きくてもそのベクトルの持つ意味が薄いベクトルをさす，このような問題を解決しようと考えられたのが分散表現である．

分散表現とは疎な onehot ベクトルを密な密な実数値をの集合をベクトルそして扱い，その実数値で単語の意味を表そうとする手法である．これにより疎なベクトルであった単語のベクトルを密な表現ができ，また，密度が上がることでベクトルの次元数を削減することができる．以下，3 手法はその分散表現を得るにあたって機械学習を応用した推論をベースとして編み出された手法である．

2.1 Continuous Bag-of-Words Model

Continuous Bag-of-Words Model(以降, CBOW と略記) は [1] で提唱された手法で, ある単語数 v の単語列 $w_0, w_1, w_2, \dots, w_{v-2}, w_{v-1}, w_v$ がある時, 単語 w_t が文脈中で前後 m 単語をコンテキスト C とする時の前後 m 単語が共に共起する事前確率が増大するように学習するモデルである。(図 2.2)

図 2.1 より w_t をターゲットにした時の周辺単語 m 個を対象としたときの事後確率は式 2.2 のように書くことができる. CBOW モデルでは式 2.2 の最大化問題と見ることができる. そこで CBOW の損失関数は, 単に式 2.2 の確率に対数を取りマイナスをつけ, 最小化問題として解く. 式 2.3 はコーパス全体に拡張した形を示す.

$$w_1, w_2, \dots, w_{t-1}, \mathbf{w}_t, w_{t+1}, \dots, w_{T-1}, w_T,$$

図 2.1 単語の列からターゲットとなる単語を推測する ($m = 1$)

$$P(w_t | w_{t-m}, \dots, w_{t+m}) \quad (2.2)$$

$$L = -\frac{1}{T} \sum_{t=0}^T \log P(w_t | w_{t-m}, \dots, w_{t+m}) \quad (2.3)$$

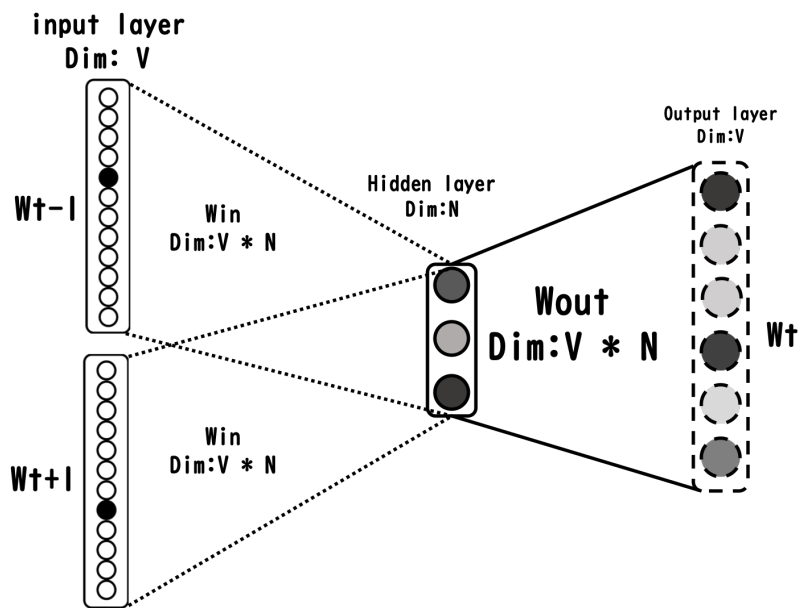


図 2.2 CBOW のネットワーク構成モデル模式図 ($m = 1$)

2.2 Skip Gram

Skip Gram は [1] で紹介されている章 2.1 で述べた CBOW とは別の手法である。CBOW とは逆に中心の単語 w_t から m 個の周辺単語を推測するモデルである。ネットワーク構成は図 2.4 のようになる。



図 2.3 単語の列からターゲットとなる単語を推測する ($m = 1$)

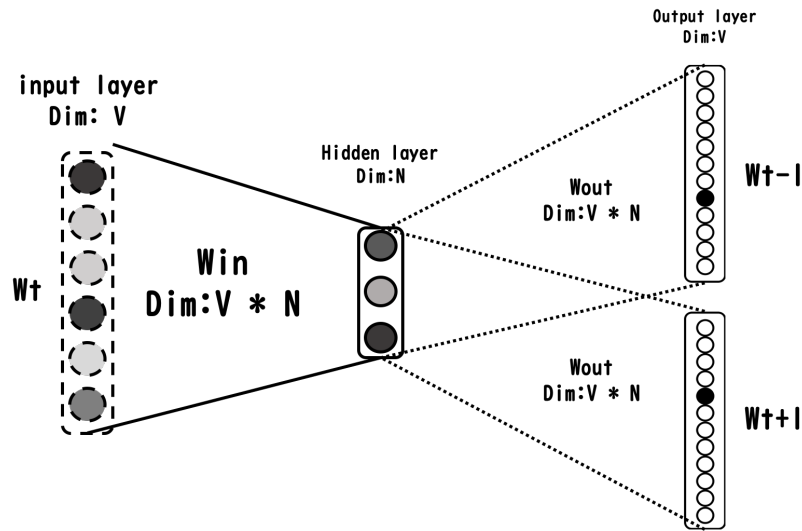


図 2.4 SkipGram のネットワーク構成モデル模式図 ($m = 1$)

ある単語 w_t が入力層に入力され、周辺単語の w_{t-m}, \dots, w_{t+m} を推測する時、その全てが同時に起こる確率は式 2.4 となる。

$$P(w_{t-m}, \dots, w_{t+m} | w_t) \quad (2.4)$$

ここで SkipGram モデルは w_{t-m}, \dots, w_{t+m} のそれぞれのあいだに関係性がないと仮定し、交差エントロピーを用いて損失関数 (式 2.5) を定義する。

$$\begin{aligned} \log P(w_{t-m}, \dots, w_{t+m} | w_t) &= -\log \prod_{k=0}^m P(w_{t-k} | w_t) \\ &= -\sum_{k=0}^m \log P(w_{t-k} | w_t) \end{aligned} \quad (2.5)$$

そして式 2.5 をコンテキスト C をコーパス全体に拡張すると式 2.6 となり、この式を学習によって最小化していく。

$$\begin{aligned} \sum_C \log P(w_{t-m}, \dots, w_{t+m} | w_t) &= \sum_C -\log \prod_{k=0}^m P(w_{t-k} | w_t) \\ &= -\sum_C \sum_{k=0}^m \log P(w_{t-k} | w_t) \end{aligned} \quad (2.6)$$

第 3 章

LSTM(Long short-term memory)

順伝播ニューラルネットワークでは前の情報はつかわないため文章や音声など、一つ前の情報に影響を受けるデータに対しては有用ではない。そこで、ある時刻 t の出力の際、過去の情報も扱う再起ニューラルネットワーク（以降 RNN と省略）というものが発案された。これは時刻 t の入力を x_t とする時、各ネットワークは過去の情報を記憶しているようにみなすことができる。しかしながら通常の RNN の場合、時間が経つほどに魔の情報は薄れていく勾配消失が起こることがあった。これを解決しようとしたのが記憶情報ごとにメモリーの役割を果たすネットワークを分けた Long short-term memory（以降、LSTM と省略）というモデルである。

第 4 章

提案手法（章題は変える）

4.1 システム全体の流れ

＜図をいれながら＞

4.2 計算式の特徴量抽出

4.2.1 概要

＜idea＞（内容充実させる）数式を分布化する際、そのベクトルの中に数式の特徴を入れ込んだベクトルを生成する手法が確立していない．そこで本論文では数式の各文字，記号を単語のようにみなし，onehot ベクトルを作成し，それを埋め込み層で特徴を踏まえた低次元ベクトルに変換したのち，系列変換モデルで読み込むことで低次元で数式の特徴を掴んだベクトルを生成できないかと考えた．

＜手法＞この考えを実現するために数式は我々が目にする $2x + 3 = 5$, $\frac{3x-1}{2} + 4 = \frac{2}{5}$ ではなく，テキスト化かつその特徴を強く受けた形に変換する必要がある．そこで本論文では数式をある一定のルールの中でテキスト化されている TeX 形式の数式を用いる．上記の計算式なら $2x+3=5, \frac{3x-1}{2}+4=\frac{2}{5}$ とし，このテキストデータを用いて文字単位の埋め込んだベクトルを作成する．

実験を行った手法は以下の三種法で行い，それぞれ分布を python を用いて確認した．

- CBOW
- SkipGram
- ...

onehot ベクトルの置き方は

- $[0, 1, 2, \dots, 9, +, -, =, x, \dots]$ のように各数字，各記号に割り当てる方法
- 出てきた数字，数式の塊を onehot を置く方法 ($[0, 1, 2, \dots, 3.6, 0.11, \dots, .+, -]$)
- 3 桁までの数字，数式に現れる記号

4.2.2 文字分布の入手

予備実験として文字の分布を入れる CBOW, SkipGram, (できれば grove も)

できればベクトルの足し引きとかできれば word2vec の論文にもそうなので結果を確かめたい．

＜おまけだからさらっと＞

4.3 判定機

今回提案するシステムではといたプリントを読み込みその結果を判別して間違っただ算問題を見つけ、その類題を選出する。

4.3.1 概要

『文章で説明』

→ あと誤差も（データ取り直しています）

第 5 章

結果とその検討

ポイント

自分の提案する方法が序論で提起した問題を解決できているかを評価・分析する。

- 目的. 何を確認するためのものか
- 方法. そのためにどういう実験を行ったか？ 実験環境・用いたデータとその選定理由・手順を示し，評価の適切性を論証すること。
- 結果. その結果はどうだったか？ 表やグラフを用いてまとめる。表は TeX，グラフは excel でなく python を用いて作成すること。
- 分析. その結果から何が言えるか？ 達成できた点・不足している点を理由と共に述べ，原因を考察する。

5.1 実験

計算式の特徴量抽出

- 目的. 数式のベクトル表現は可能なかどうか
- 方法. 埋め込み層，ネットワーク構成を変えながら Encoder-Decoder で復元を試みるその Encoder の出力値を数式ベクトルとしてみなし，pca,T-sne で二次元ベクトルに圧縮し図示する
- 結果. これから
- 分析. これから

計算式の特徴量からの類題選出

- 目的. 求めた計算式のベクトルから特徴を捉えた数式を選出できるか
- 方法. 特徴量ベクトルから k 近法で選出，ある生徒の間違った一問から類題を選出し，実際にその問題を間違えていたかを確認，また，逆にあっていた問題からも同様な手法で確認
- 結果. これから
- 分析. これから

5.2 評価

5.3 分析

5.4 検討

第 6 章

関連研究

ポイント

この研究に関連する他の研究を紹介し，この研究との違いを明確にする．

<https://code.google.com/archive/p/word2vec/>

<https://techblog.asahi-net.co.jp/entry/2018/10/05/180310>

第 7 章

結論と今後の課題

謝辭

[3]

参考文献

- [1] JeffreyDeanEfficient Estimation of Word Representations in Vector Space (2013).
- [2] year 第 6 回学习指导基本调查 DATA BOOK (高校版) (2016).
- [3] yearO.REILLY (2018).