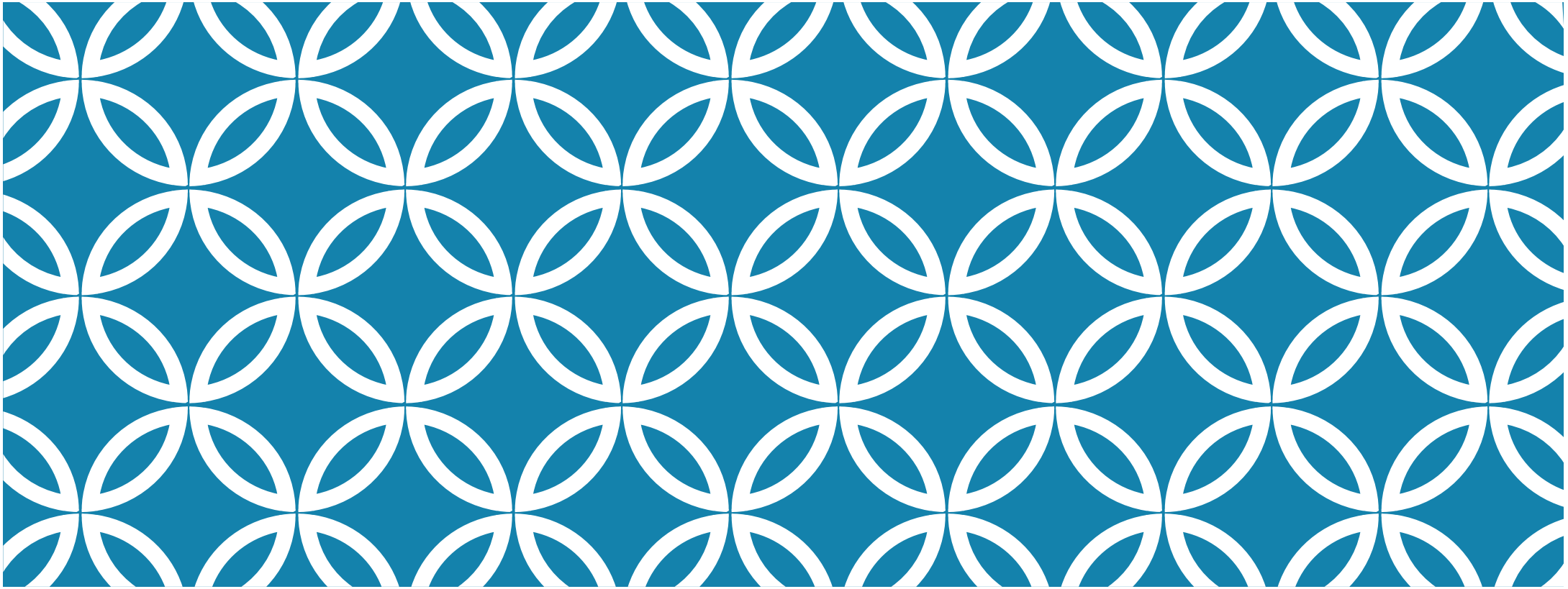
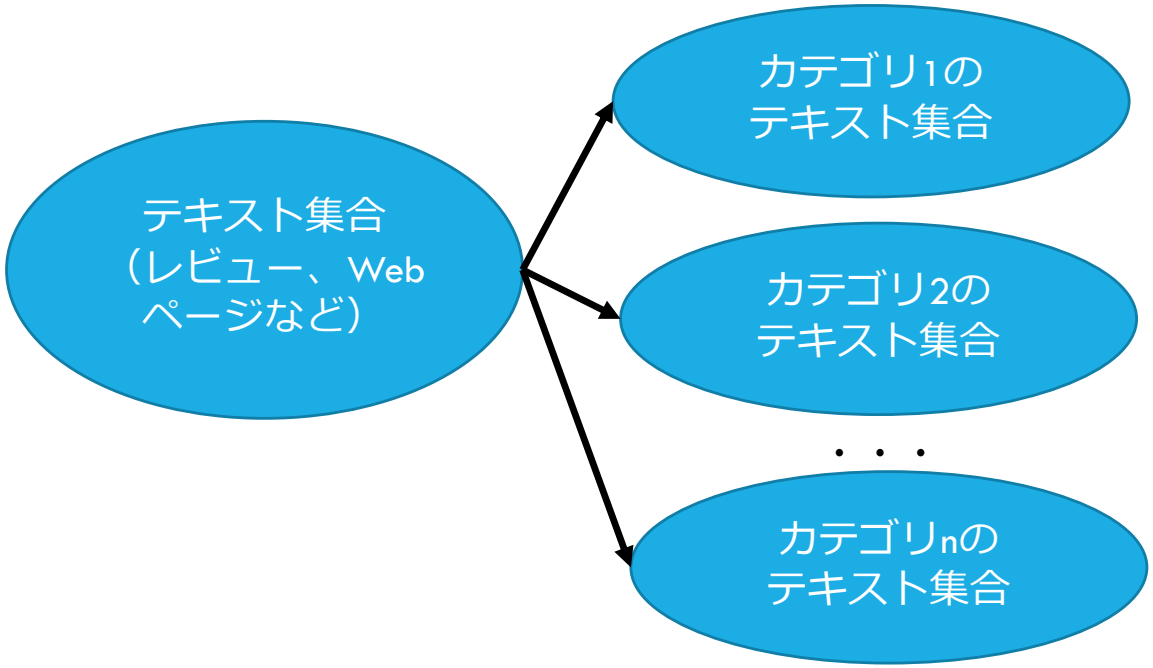


情報アクセス論 第13回 テキストマイニング(2)

西原陽子

文書分類

テキストの集合体から情報抽出をする際に、テキストをカテゴリ別に分類、あるいはクラスタリングすることが多い



カテゴリ別に分類された中で特徴的な単語や、共起する単語を調べることで詳細な分析が可能となる

文書分類の手法の例

教師あり学習

- ナイーブベイズ（単純ベイズ分類器）（第8章で紹介）
- サポートベクタマシン
- ニューラルネットワーク（1層のものと、多層（Deep）もの）
- 決定木
- その他

教師なし学習（正確には分類ではなく、クラスタリング）

- K-means（K平均法）（第8章で紹介）
- ニューラルネットワーク（説明が複雑になるため、ここでは教師なしは紹介しない）
- その他

サポートベクタマシン

教師あり学習を用いるパターン認識モデルの一つ

- 分類や回帰問題に適用可能
- 1963年にVapnikらが線形サポートベクタマシンを発表し、1992年に非線形にも対応できるよう拡張された

現在知られている中でも認識性能が最も優れた学習モデルの一つ

マージン最大化の考えに基づき、未学習のデータに対しても高い識別性能を持つ

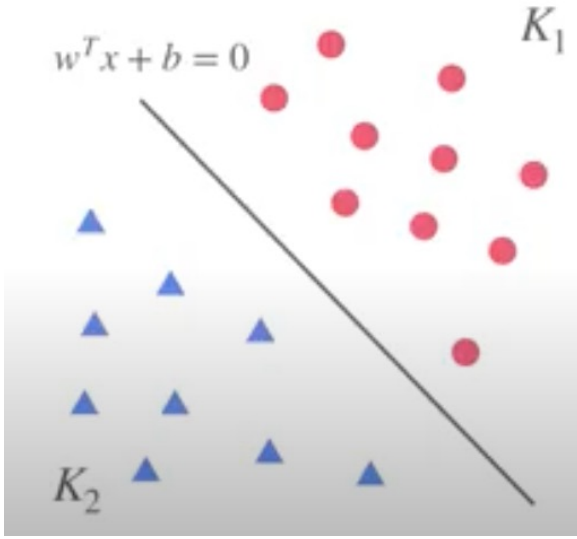
主に2クラス分類問題の解法に使われるが、多クラス分類への拡張も可能

データの線形分離可能性

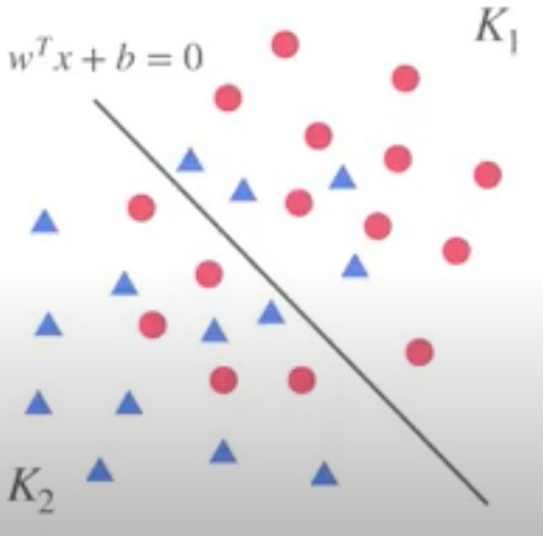
線形分離可能：与えられたデータを線形の平面で分離することができる

- 3次元空間に分布するデータを2次元の平面で分離できる

線形分離可能



線形分離不可能

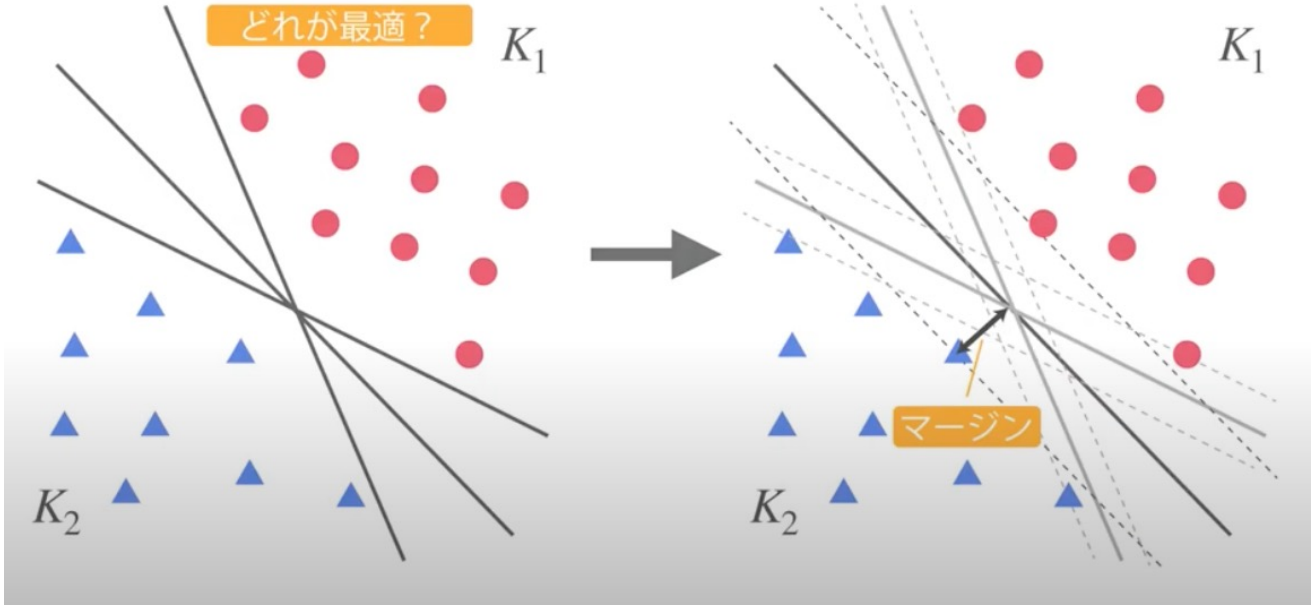


https://www.youtube.com/watch?v=cNEhKEb9-JUから図を引用

マージン最大化

線形分離可能な時に、どの平面を分離超平面として選択するかが問題となる

このときに、マージン最大化の概念を用いる

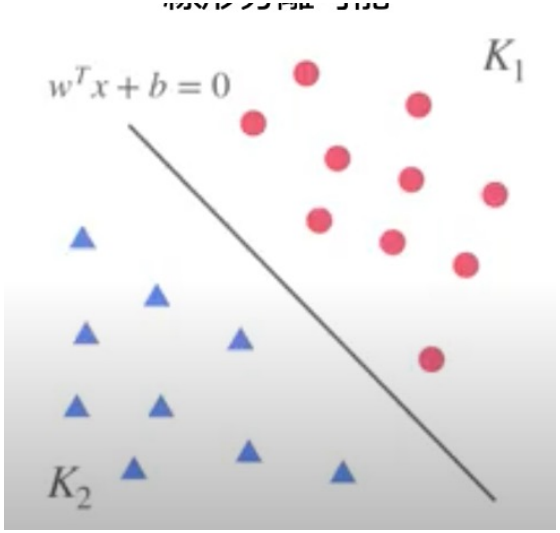


マージン：
分離超平面から最も近い場所にあるデータまでの距離

https://www.youtube.com/watch?v=cNEhKEb9-JUから図を引用

超平面の計算方法

データを2つのクラスK1とK2に分類するときに以下の数式が成り立つ



K1: $W^T x_i + b > 0$

K2 : $W^T x_i + b < 0$

クラスを識別するラベル変数tを導入し、
K1クラスならt=1、K2クラスならばt=-1とする

$t_i(w^T x_i + b) > 0 \ (i = 1,2, \dots, n)$

次元を2次元からp次元に拡張すると、データから超平面までの距離は

$d = \frac{|w_1 x_{1+} w_2 x_{2+} \dots + w_p x_p + b|}{\sqrt{w1^2 + w2^2 + \cdots + wp^2}} = \frac{|w^T x + b|}{\|w\|}$ ←この最小値を最大にすることでマージンが最大となる超平面が得られる

https://www.youtube.com/watch?v=cNEhKEb9-JUから図を引用

ニューラルネットワーク(NN)

教師あり学習を用いるパターン認識モデルの一つ

- 教師あり学習と、教師なし学習のいずれにも対応可能だが、本講義で詳細は紹介しない

人間の脳内にある神経細胞（ニューロン）とそのつながり、つまり神経回路網を人工ニューロンという数式的なモデルで表現したもの

多次元量のデータで線形分離不可能な問題に対しても適用可能

近年人工知能、AIと呼ばれるものの一部に、機械学習器として深層学習器（ディープニューラルネットワーク, DNN）が用いられているが、DNNはNNを多層にしたもの

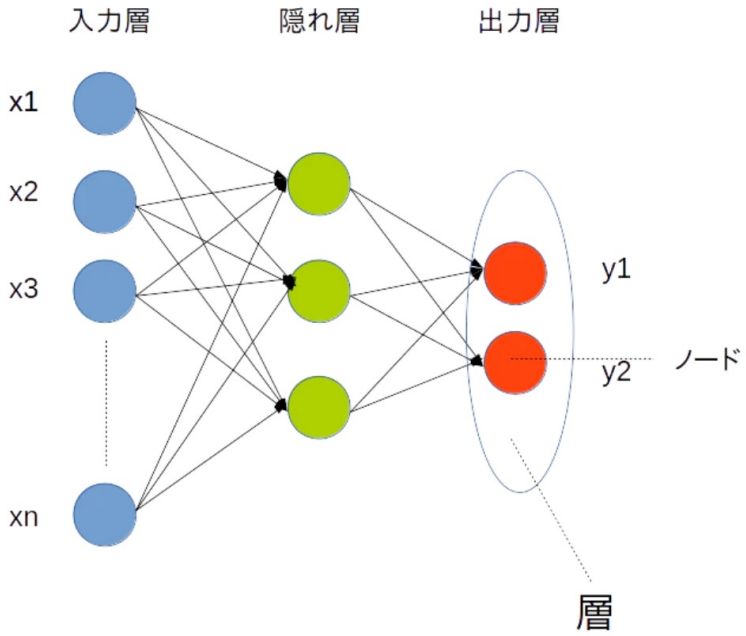
ニューラルネットワークの構成

入力層、隠れ層、出力層から構成される
- 入力層と出力層は1層
- 隠れ層の数は任意

各層は複数のノードにより構成される

ノードに与えられた信号が入力層->隠れ層->出力層と順に辿る

入力で与えられたn次元のベクトルを、
出力でm次元のベクトルに落とす
関数fを推定することができる



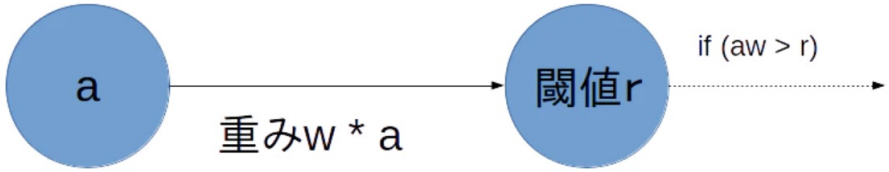
https://qiita.com/hatt0519/items/3188382ad85a8654cc42から図を引用

ノード間の信号のやりとり

あるノードから別のノードへ信号が伝えられる際、
ノードから出された信号の値aに重みwをかけ、
waの値が閾値rを超えるかを判定する

閾値を超えるならば次のノードへ信号を伝え、
超えないならば信号を伝えない

次のノードへ伝達するしないを判定する関数を活性化関数と呼ぶ



https://qiita.com/hatt0519/items/3188382ad85a8654cc42から図を引用

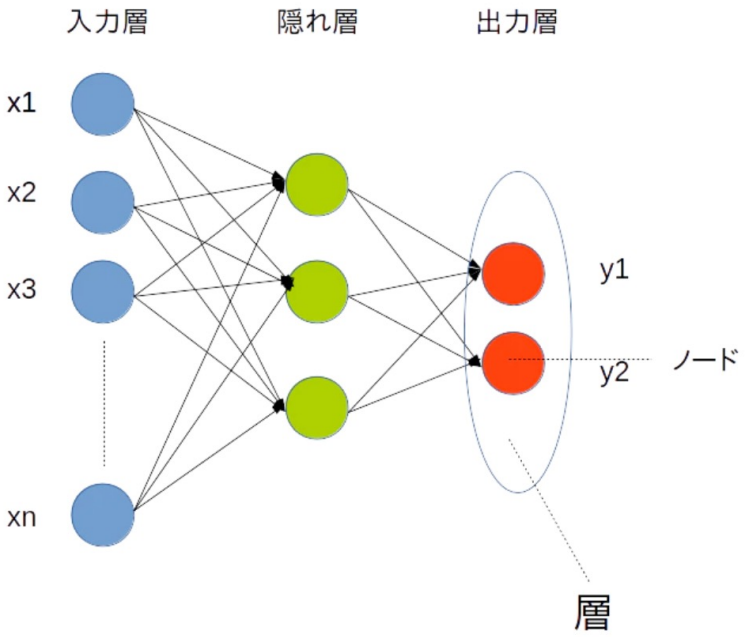
誤差逆伝播法

ニューラルネットワークでの重みは最初適当に与えられ、
訓練データの学習を通じて最適な重みへと更新していく

学習データで分類ミスが起きた場合、
隠れ層内の重みにミスがあると考える

出力結果と教師データの誤差を算出し、後ろのノードへと更新を伝えていく

具体的なアルゴリズムとしては、勾配降下法がある



y1: リンゴ 正解

y2: 非リンゴ 間違い！

https://qiita.com/hatt0519/items/3188382ad85a8654cc42から図を引用

文書をベクトル表現する際の問題点

文書を分類する際には、文書をベクトルで表現する必要がある。

5章で紹介したBag of Wordsモデルでは、文書に出現する単語をベクトルの要素とし、頻度や重要度をベクトルの値とした

- d = {1, 0, 0, 1, …, 0, 1}：Bag of Wordsモデルで文書ベクトルdの作成例

問題点

- (1) ベクトルの各要素の値に0が多くなる
- (2) ベクトルの次元数が単語の種類数となり、膨大な次元数となる
- (3) 類似した単語であっても、ベクトルの別の要素となる
  - さくら、桜、サクラ、花：これらの類似単語は別の単語として扱われる

ベクトル表現が適切に行われないと、文書分類に影響が生じる

複数の単語から構成されるベクトルの要素を作る方法が提案された

単語分散表現

ある文書dに「プログラマー」「打つ」という単語が含まれているとき、
これまでの文書ベクトル表現では以下のようになった

イチロー	プログラマー	パリ	道頓堀	打つ
0	1	0	0	1

d = {0,1,0,0,1}

単語分散表現では、使われ方が似ている単語をまとめ次元を作り、
新しいベクトル空間を定義する

怠惰さに関する次元	スポーツに関する次元	土地に関する次元	…
0.01	0.99	0.04	

イメージ図であり、
実際に次元に新しい名前が付くことはない

d = {0.01, 0.99, 0.04, …} ← 文書ベクトルから0が消え、次元数も圧縮される

単語分散表現を獲得するモデル

Word2Vec

- Continuous Bag of Wordsモデル
- Skip-gram モデル（本講義で紹介するモデル）

fastText

いずれのモデルも単語の分布仮説に基づき、
ニューラルネットワークを用いて単語分散表現を獲得する

- 単語の意味は周囲の単語により形成される
  - 「桜が咲いた」：花を表す
  - 「桜が居た」：人を表す
  - 「ひまわりが咲いた」：ひまわりと桜は似ている
  - 周囲の単語をみて、使われ方が似ている単語は類似する意味を持つとする

SKIP-GRAMのWORD2VEC

Skip-gramはニューラルネットワークのモデルの一つ

- 入力層、出力層、1層の隠れ層により構成される

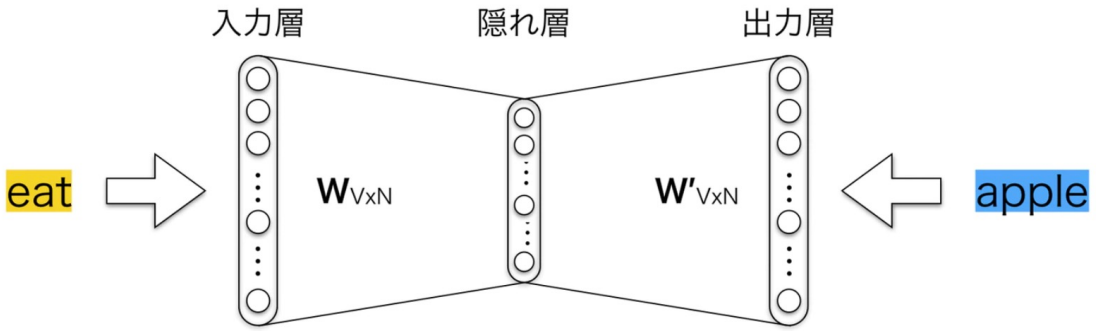
単語分散表現を得るためのモデル

- 直接単語分散表現を獲得するのではなく、
  別のタスクをさせ、その副産物として単語分散表現を獲得する

SKIP-GRAMで行うタスク(1)

ある単語を入力したとき、その周辺にどの単語が現れやすいかを予測する

- 例文：I want to eat an apple everyday.
- ある単語をeatとする。周辺に現れやすい単語はappleや、orangeなどが考えられる。
- tankやnetworkは現れにくいと考えられる。
- 現れやすい単語に高い予測確率を付与するように教師あり学習をする



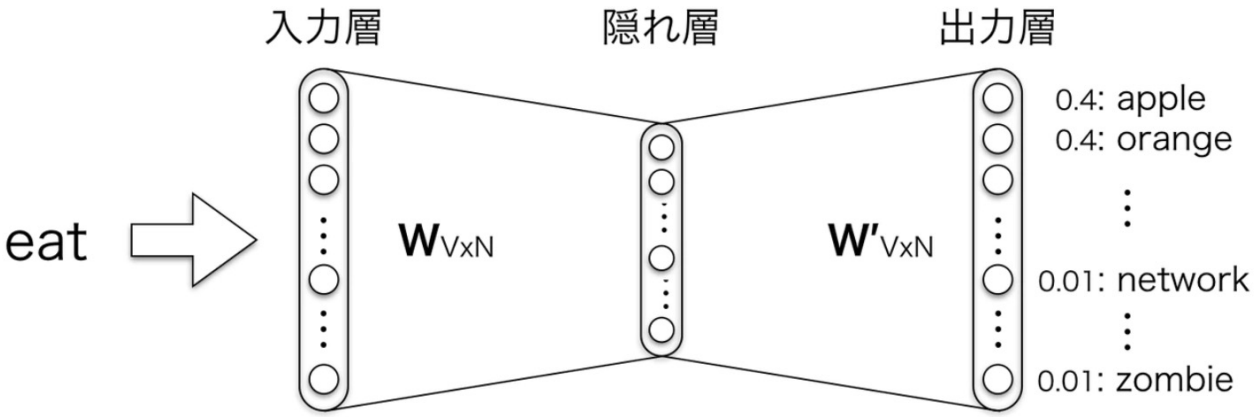
Eatを入れたら、appleが出力されるように隠れ槽の重みを学習する

https://qiita.com/Hironsan/items/11b388575a058dc8a46aから図を引用

SKIP-GRAMで行うタスク(2)

複数の文の全ての単語に対し
周辺に現れやすい単語の学習を行わせることで、
以下のような出力が可能となる

- I want to eat an apple. I want to eat an orange. ,,,など



eatを入れたら、関連する単語の
確率が高く出力される。

https://qiita.com/Hironsan/items/11b388575a058dc8a46aから図を引用

SKIP-GRAMの入力層

ニューラルネットワークの入力層は
固定長のベクトル

単語を入力するときにベクトルで表現するため、
文書集合から重複のない単語集合を作る
（語彙集合）

- 例文：I want to eat an apple. I like apple.
- 語彙集合：{apple, eat, I, like, to, want, .}；全部で7つの単語からなる
- eatをベクトルで表現すると右のようになる

$$\begin{bmatrix} 1 \cdots apple \\ 0 \cdots eat \\ 0 \cdots I \\ 0 \cdots like \\ 0 \cdots to \\ 0 \cdots want \\ 0 \cdots . \end{bmatrix}$$

https://qiita.com/Hironsan/items/11b388575a058dc8a46aから図を引用

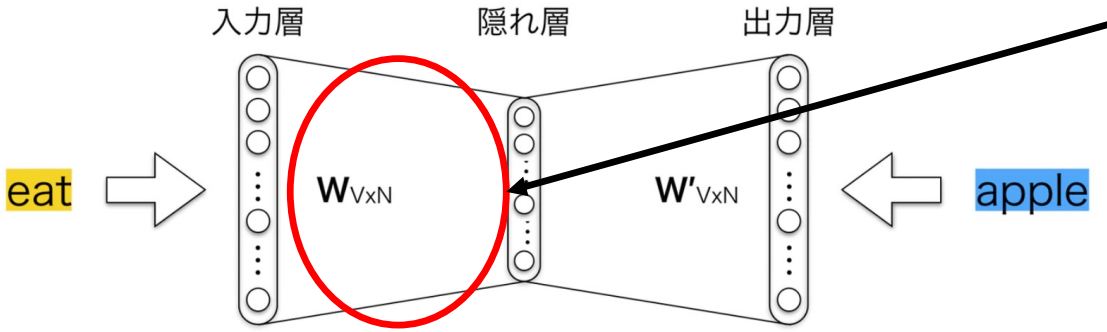
SKIP-GRAMの隠れ層

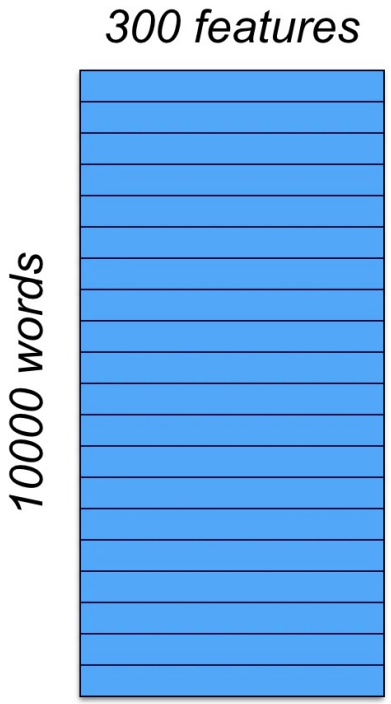
語彙数を1万、隠れ層のノード数を300とし、
入力層から隠れ層への重みの学習を行う。

隠れ層への重みは1万*300の重み行列により表現される。

重み行列の各行が単語ベクトルとなる

- つまり、単語分散表現





https://qiita.com/Hironsan/items/11b388575a058dc8a46aから図を引用

入力層と隠れ層の計算

語彙数を7、隠れ層のノード数を3とした場合の例

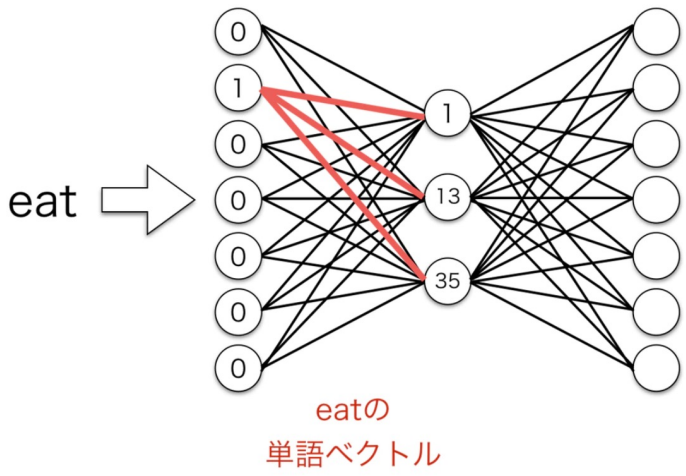
1*7のone-hot vectorを7*3の行列にかける

- one-hot vectorとは、1箇所だけ1で、あとは全て0のベクトル

入力層から隠れ層への重み行列
（実は単語分散表現）

eat

$$\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \times \begin{bmatrix}74 & 29 & 98\\ 1 & 13 & 35\\ 65 & 31 & 37\\ 96 & 88 & 84\\ 45 & 94 & 96\\ 21 & 88 & 9\\ 6 & 78 & 94\end{bmatrix} = \begin{bmatrix}1 & 13 & 35\end{bmatrix}$$

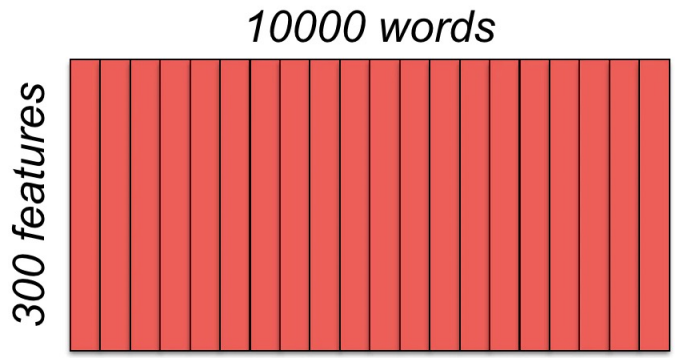


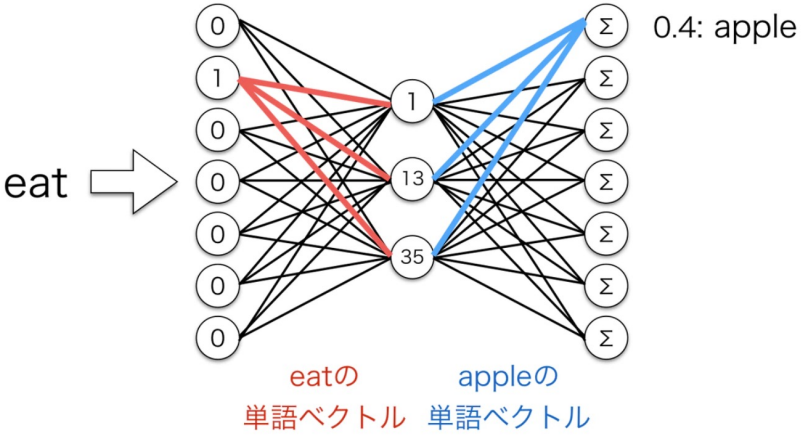
https://qiita.com/Hironsan/items/11b388575a058dc8a46aから図を引用

SKIP-GRAMの出力層

隠れ層から出力された入力語に対応する1*300の単語ベクトルは、隠れ層から出力層へ至る重みがかけられた後、出力層に入力される

重み行列は入力層から隠れ層へ至る重み行列の転置したものと同じになる





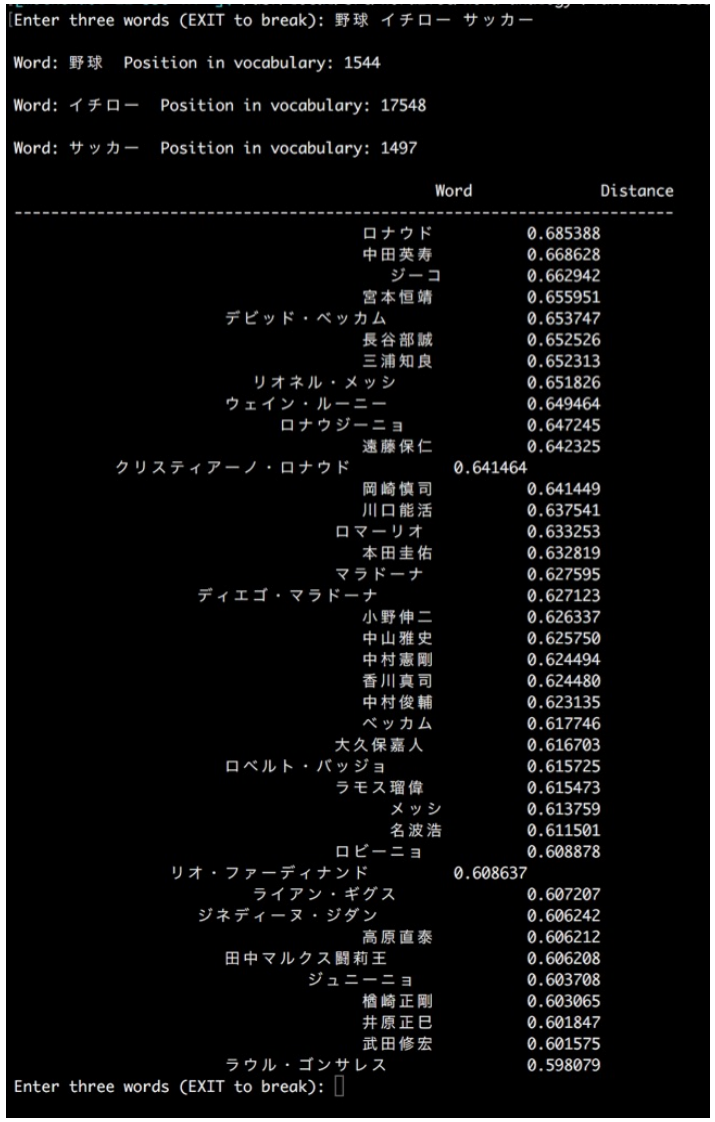
https://qiita.com/Hironsan/items/11b388575a058dc8a46aから図を引用

WORD2VECを用いてできる演算

単語をベクトル表現できるようになり、
ベクトルの足し算引き算により意味の足し算引き算が可能となった

（例）

- king – man + woman = queen
- king + woman = princess
- イチロー – 野球 + サッカー = ロナウド
- 東京 – 日本 + アメリカ = ?



https://marunouchi-tech.i-studio.co.jp/3705/から図を引用

まとめ

テキストマイニングで使われる技術を紹介した

文書分類手法

- サポートベクタマシン
- ニューラルネットワーク

単語分散表現

- Skip-gramのword2vec