自然言語処理
―単語ベクトル―

https://satoyoshiharu.github.io/nlp/

単語ベクトルと、100本ノックの第7章の位置づけ

- ニューラルネットを使って、単語ベクトルを作成します。どういう分かち書きを利用し、どういう単語ベクトルが欲しいかは、応用目的によるので、結局自作できるようにしていないとアプリが組めません。自作できるだけの予備知識を以下で説明します。

- 一方、100本ノックの第7章は、単語ベクトルがすでにあるとして、それで何ができるかを知る的な課題です。第7章の課題にはクラスタリングや次元圧縮といった機械学習の使用も含まれます。

自然言語処理:
なぜ単語ベクトル？

解説動画

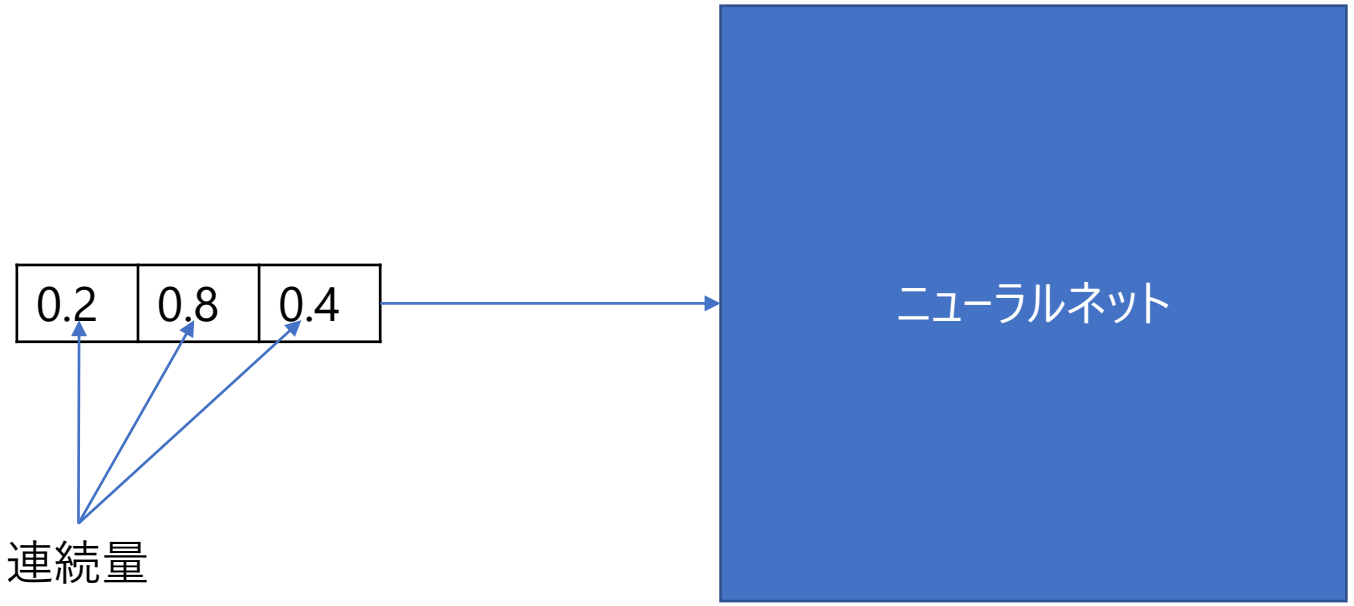
連続量と離散量

- 連続量
  - 例
    - 身長：170cmと170.11cmとの間に無限に取りうる値がある。
    - 体重
    - カメラから写っている物体までの距離
    - 画像のあるピクセルのRedチャネルの量(floatの場合)
    - 加速度センサーのX軸の反応量
    - スピーカーの音量
    - ・・・
- 離散量
  - 例
    - さいころの目： 1 から 6 の間の値を飛び飛びでとり、それらの間の値を取りえない。
    - 何月か
    - 学生番号
    - トランプのマーク
    - 人の性別
    - コロナ検査で陽性か陰性か
    - ・・・

自然言語の処理単位は、離散的

- 文字：あ、い、山、川、…
- 単語：今日、明日、太郎、花子、…
- 文：今日は晴天です。明日は木曜日です。週末が楽しみです。…

ニューラルネットの入力は通常連続量のベクトル



One-hot-Vector

離散的なデータを、連続量（たまたま０か１）のベクトルに変換する

• 今日	1	0	0	0	0	0	…
• 明日	0	1	0	0	0	0	…
• 太郎	0	0	1	0	0	0	…
• 花子	0	0	0	1	0	0	…
• …							

⇒　ニューラルネットで取り扱いが可能になる。

One-hot-Vector の問題点

- 文字数は、*常用漢字に限定しても*　2136 文字。
- 単語数は、通常の日常用語は 5 万語。Wikipediaなどは、数百万語～。
- 文の数は、構文的な自由度があり、再帰も可能なため、実質、無限大。

- 文字、単語、文のどれも、これらの個数だけ長いベクトルを、多数、利用するのは、メモリ効率が悪い。しかも、0 である領域が多すぎて、無駄。

埋め込み（**Embedding**）
別名 **Word2Vec**
一般的には **単語の分散表現**

固定長の実数値ベクトルで、それぞれの単語などを表現する。

実数値は連続量なので、無限に多くの値をとりえて、さらにそのベクトルは実数を並べるので、表現力も十分にリッチ。

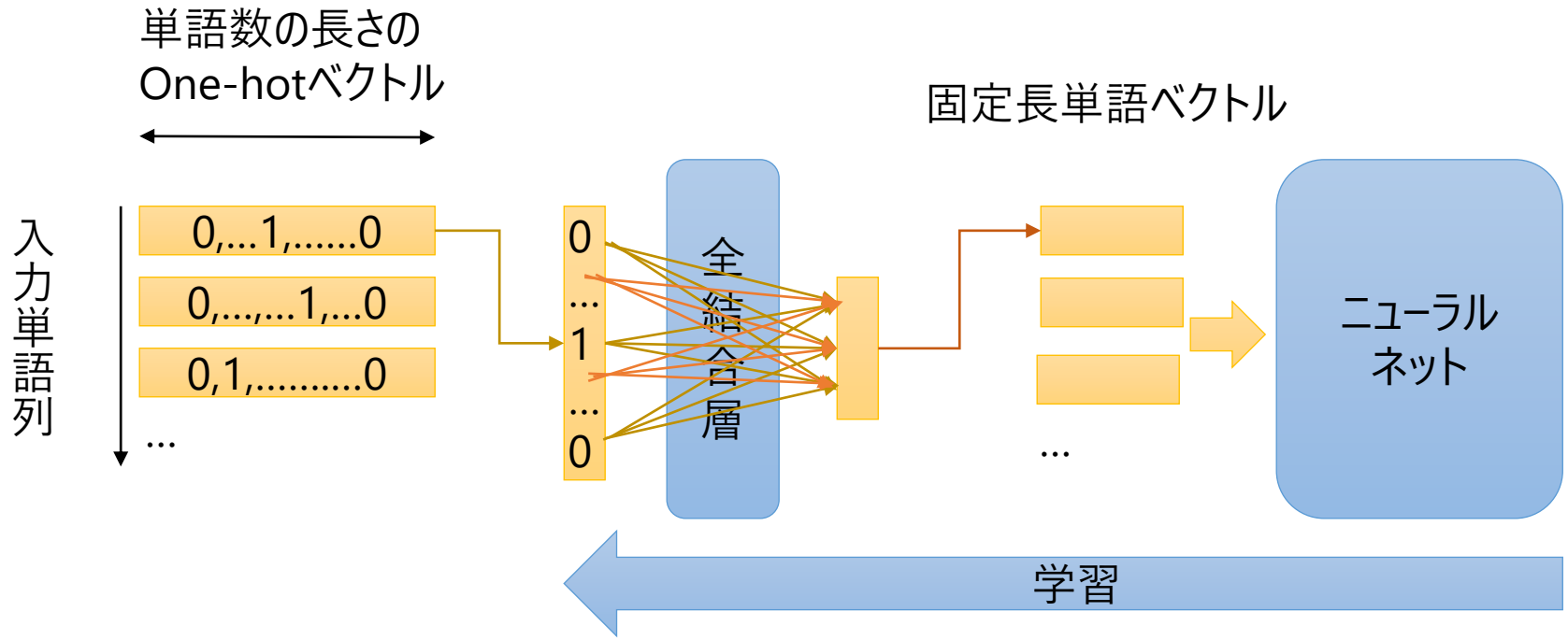
単語の分散表現

ニューラルネットが処理できるように、単語を固定長の実数値ベクトルで表現したいが、その表現こそ、ニューラルネットを使って学習させる。この発想が妙、考えた人偉い。

自然言語処理：
単語ベクトルの導出ー埋め込み表、CBOW、Skipgram

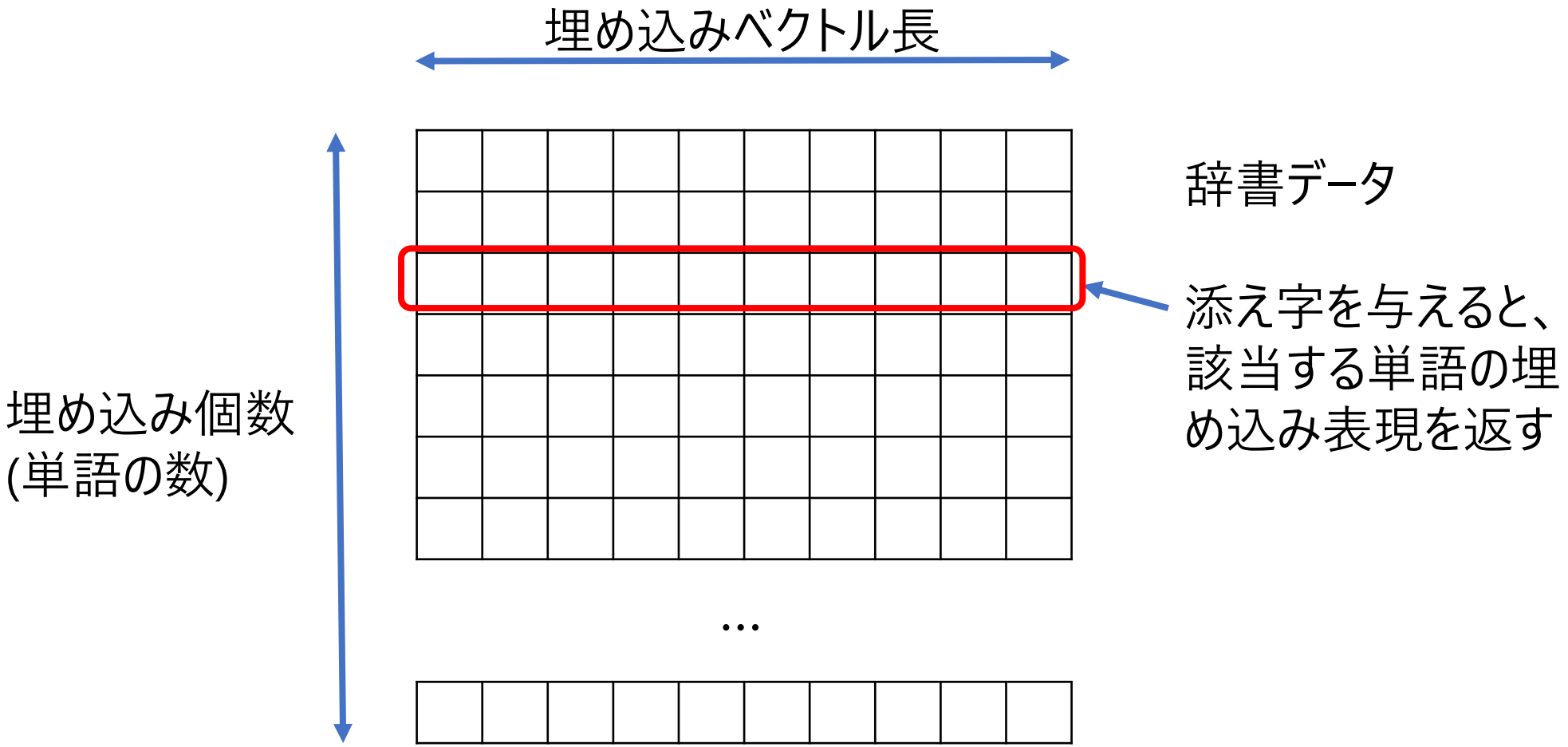
解説動画

単語を固定長ベクトル(埋め込み)として学習する



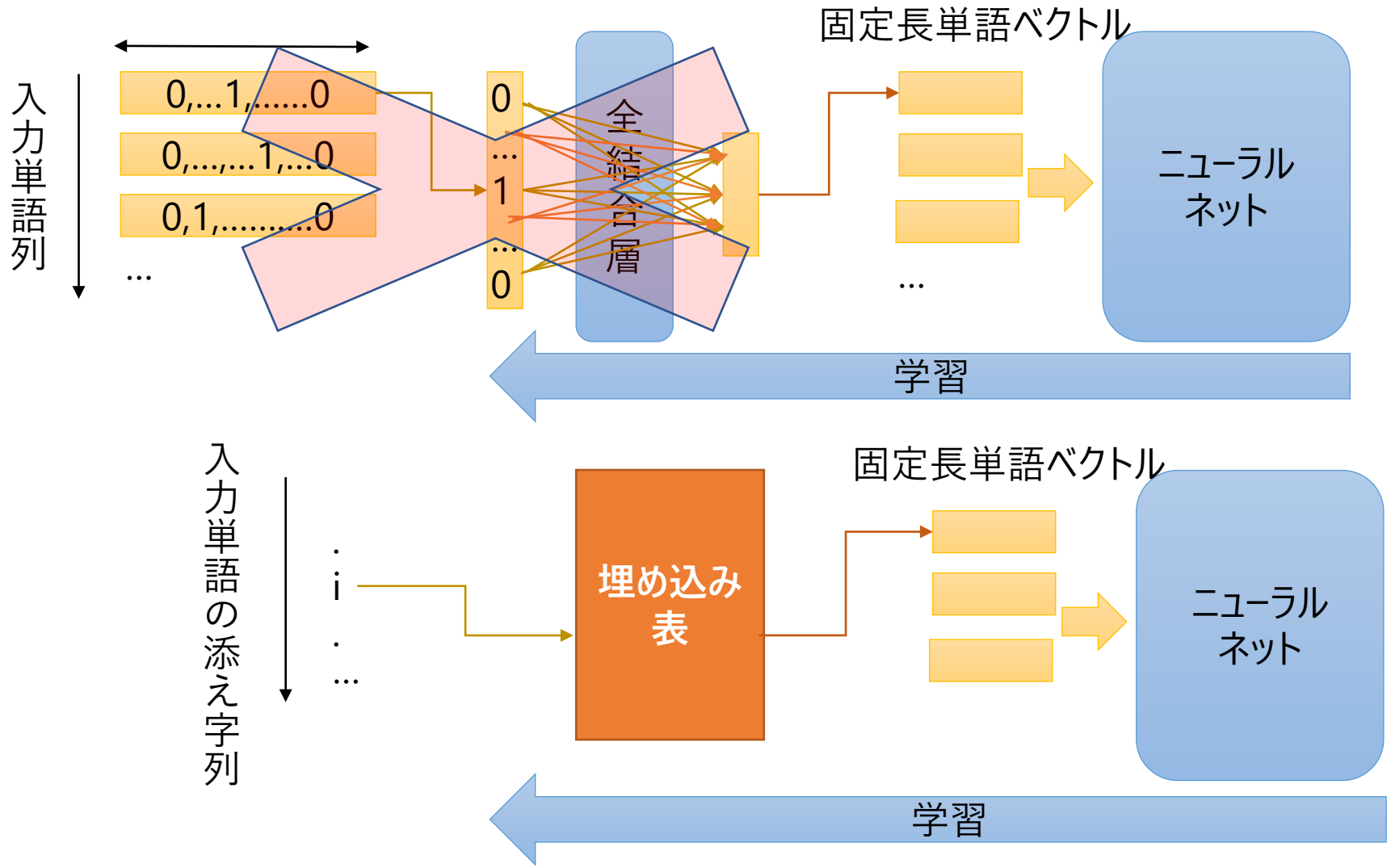
結合線の組み合わせが大きい
（単語数 x 埋め込みサイズ）

埋め込み表

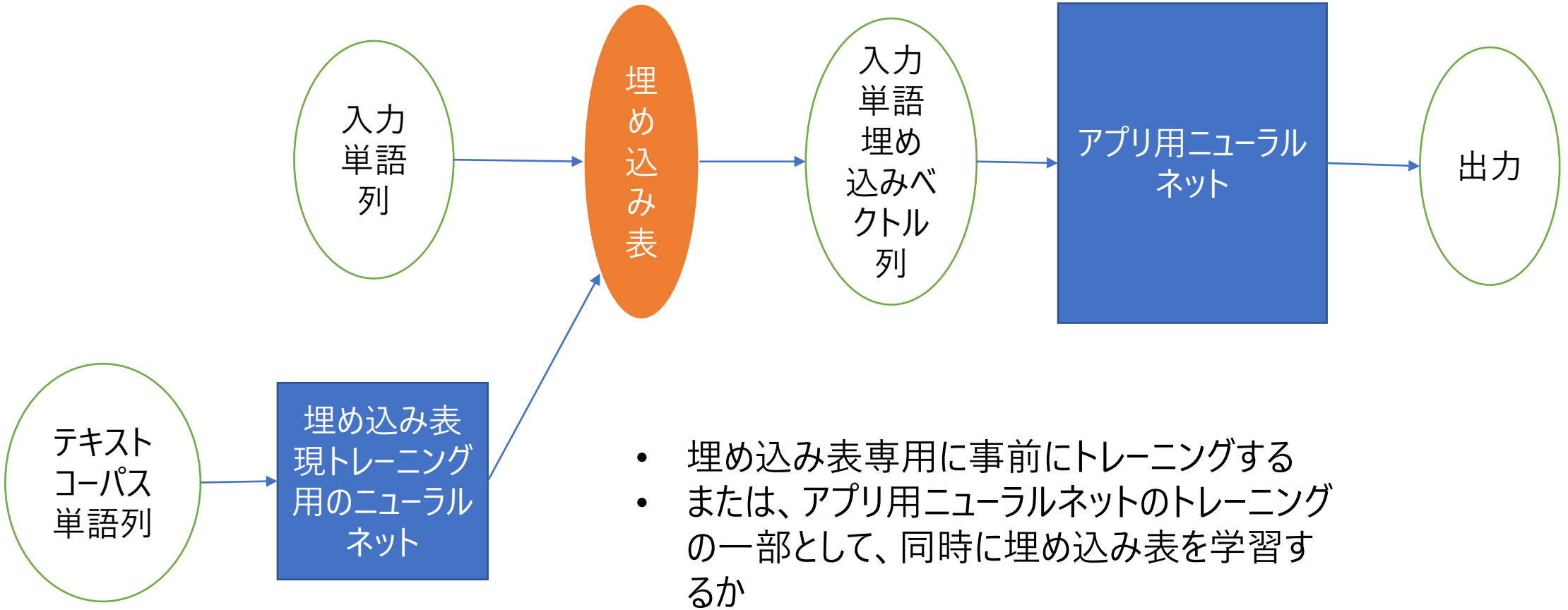


PyTorch nn.Embedding（埋め込み個数、埋め込みベクトル長）

埋め込み表を学習すれば、単語数 x 埋め込みサイズの全結合は不要



埋め込みベクトルの位置づけ



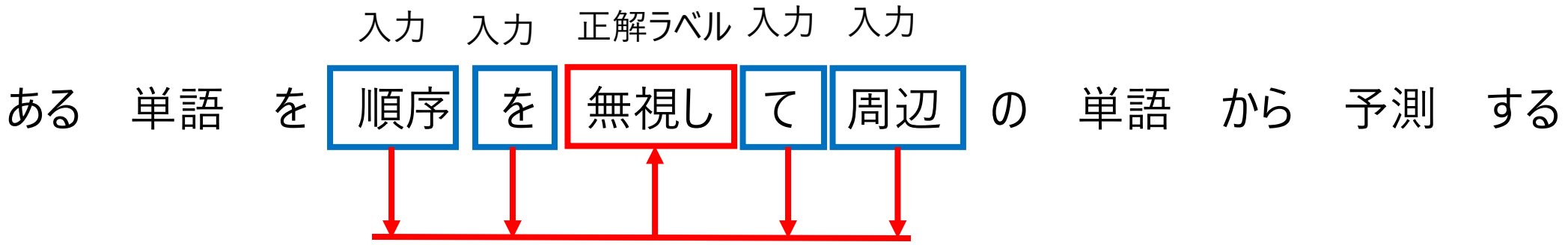
- 埋め込み表専用に事前にトレーニングする
- または、アプリ用ニューラルネットのトレーニングの一部として、同時に埋め込み表を学習するか

分布仮説

- 同じような文脈で使われる単語は、それらの意味が近い。
- 単語の意味の違いはその用法（周囲の単語との関係）で決められる。

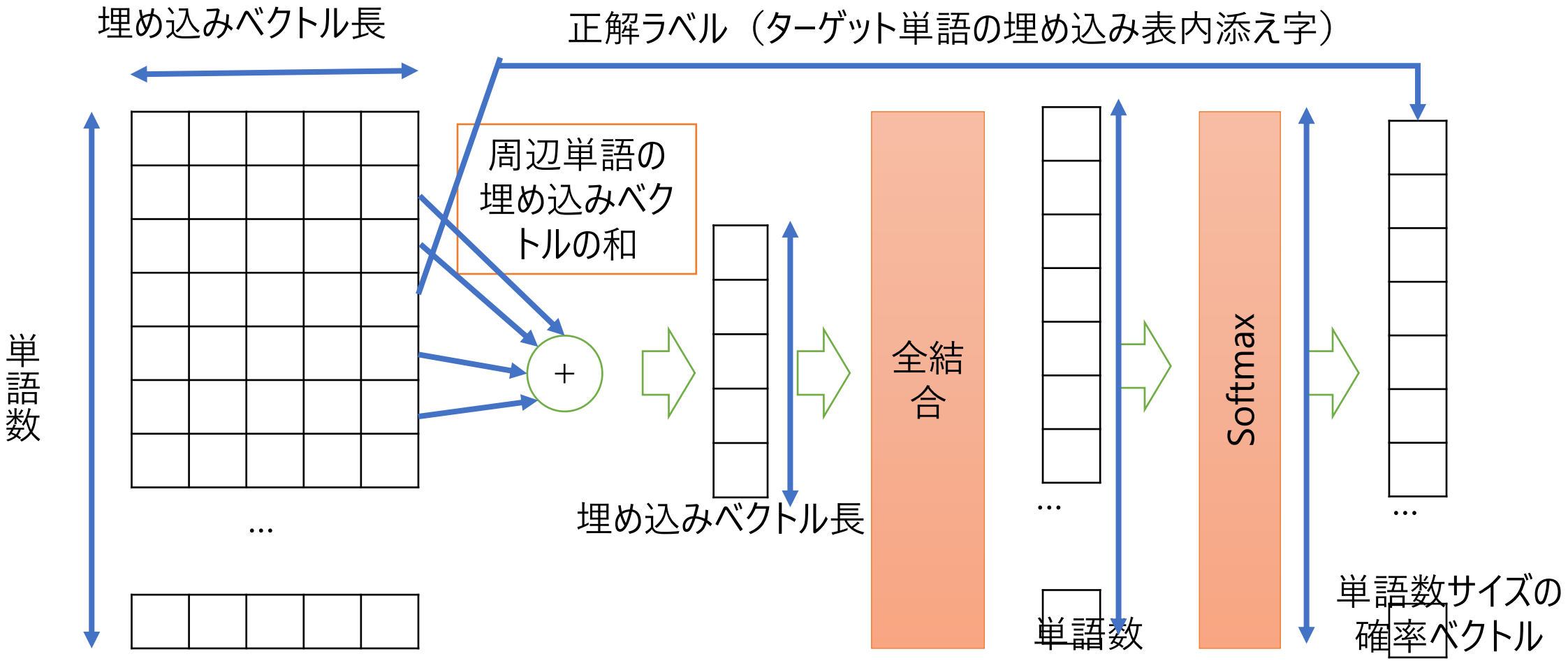
  - 「日」
    - まだ日(day)が浅い
    - 日(sun)が照っている
  - 「甘・い」
    - 君はまだ甘い(naive)
    - このリンゴは甘い(sweet)よ

CBOW (Continuous Bag-Of-Words)



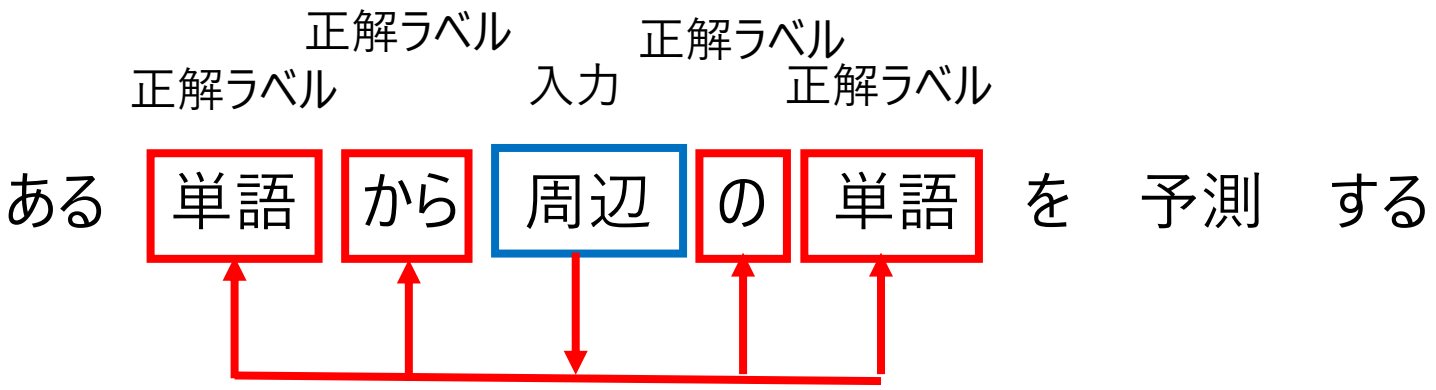
その学習は、教師なし学習っぽく、正解ラベルを
文データ自体から自動的に持ってきます。

CBOWネットワーク構成

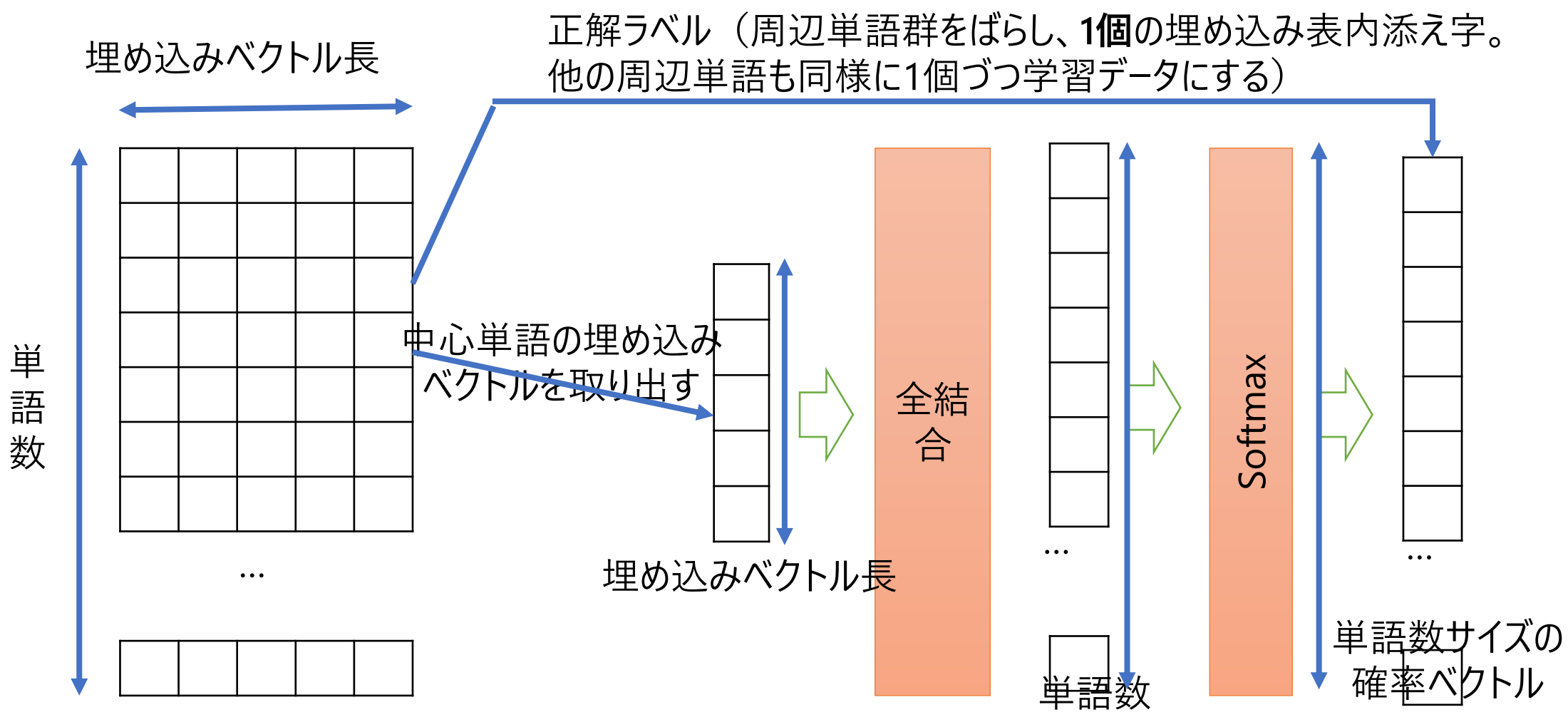


周辺単語の情報からターゲット単語への対応付けによって、埋め込み表内の単語表現を、学習させる。

SkipGram



SkipGramネットワーク構成



中心単語から周辺単語への対応付けによって、埋め込み表内の単語表現を、学習させる。

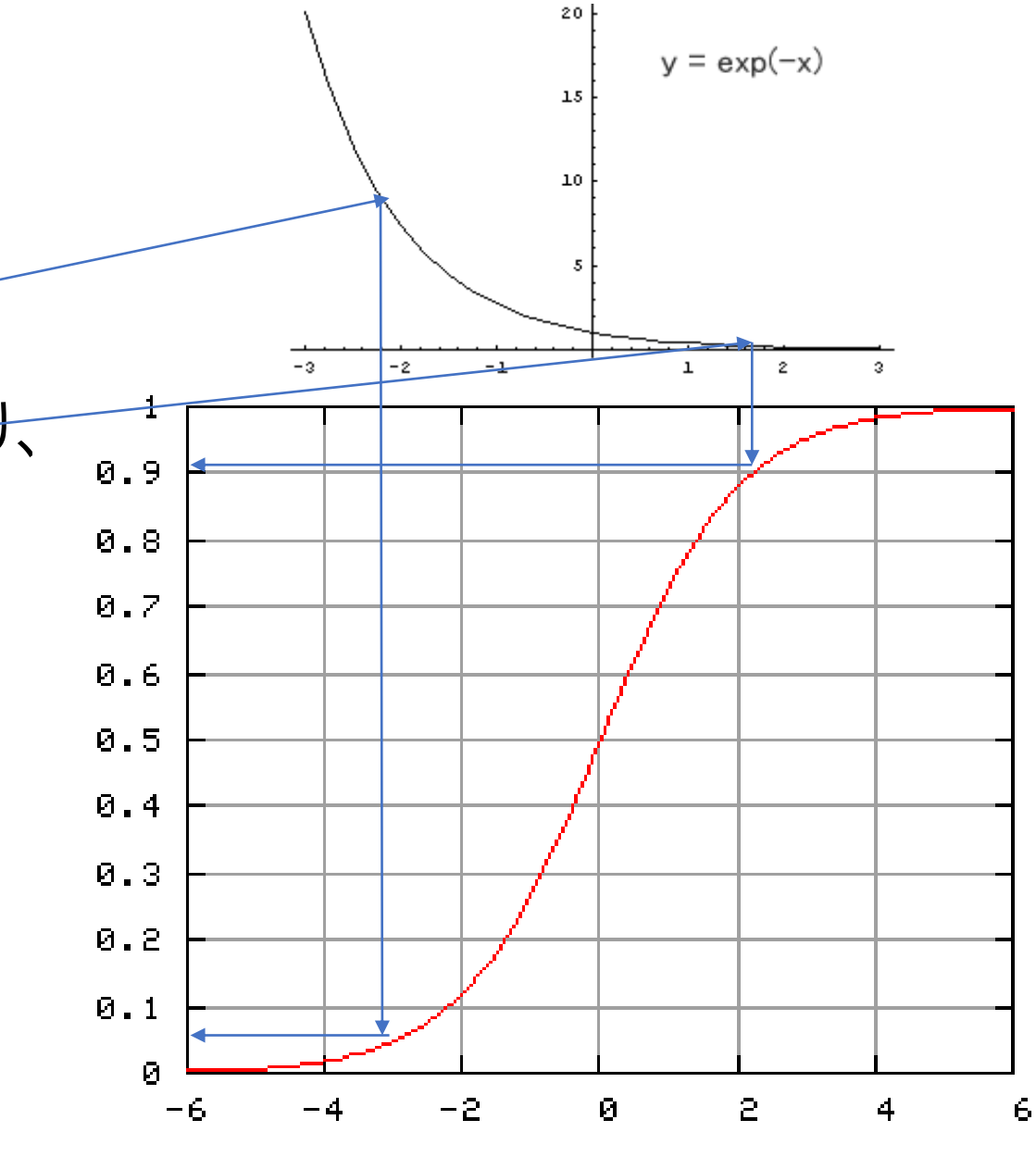
自然言語処理:
ニューラルネット、Sigmoid関数

解説動画

Sigmoidグラフ

- 入力xが負数だとEXP(-x)は大きくなり、Sigmoidは0に近くなる。
- 入力xが正数だと、EXP(-x)は小さくなり、Sigmoidは 1 に近くなる。
- つまりSigmoidは、任意の実数値を、0..1の範囲の値に変換する。

$$S = \frac{1}{1 + e^{-x}}$$

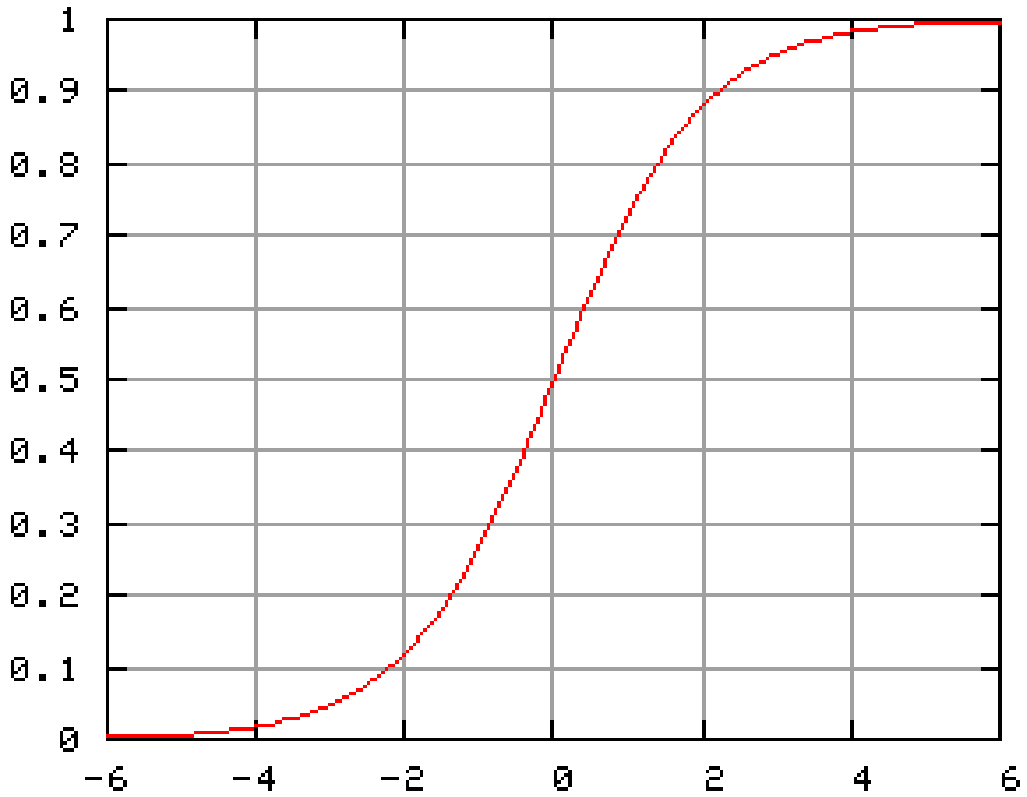


Sigmoid
機能

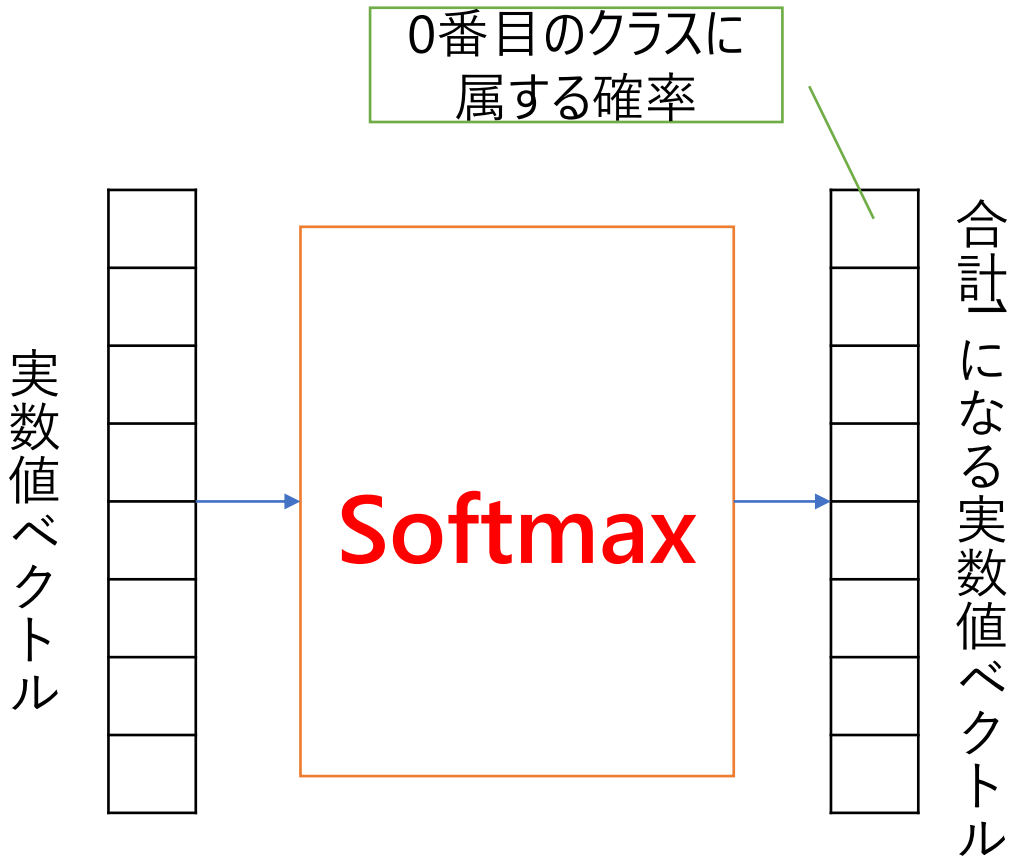
0..1の範囲へ変換する

＝

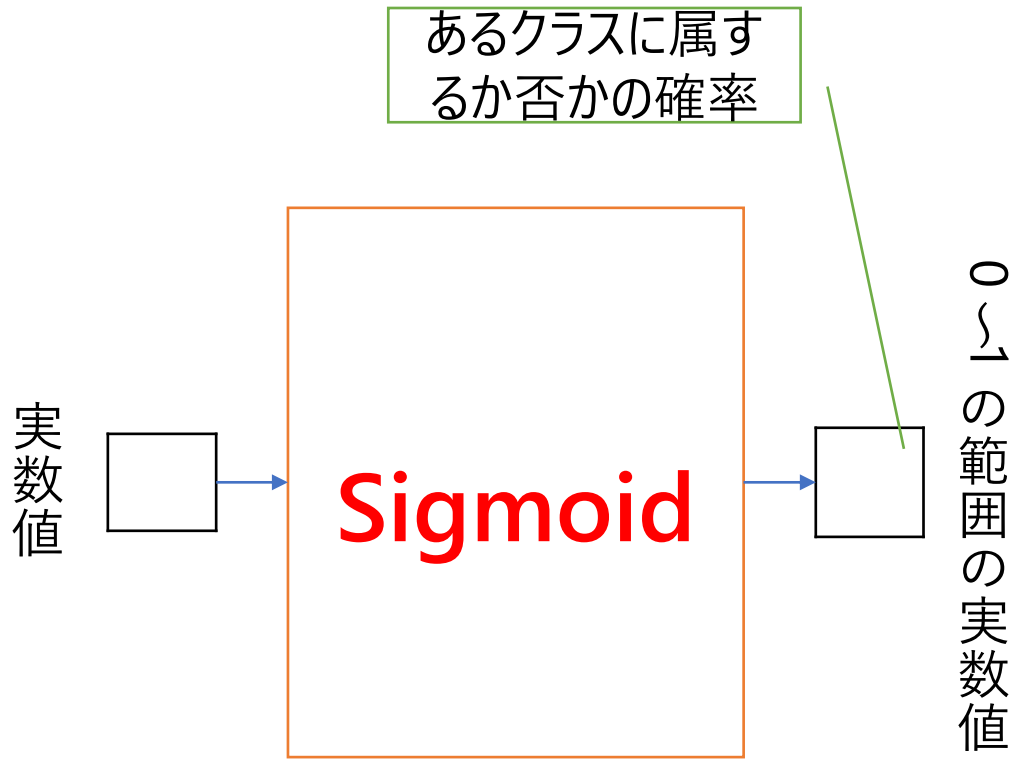
YESである確率を得る。



多クラス分類



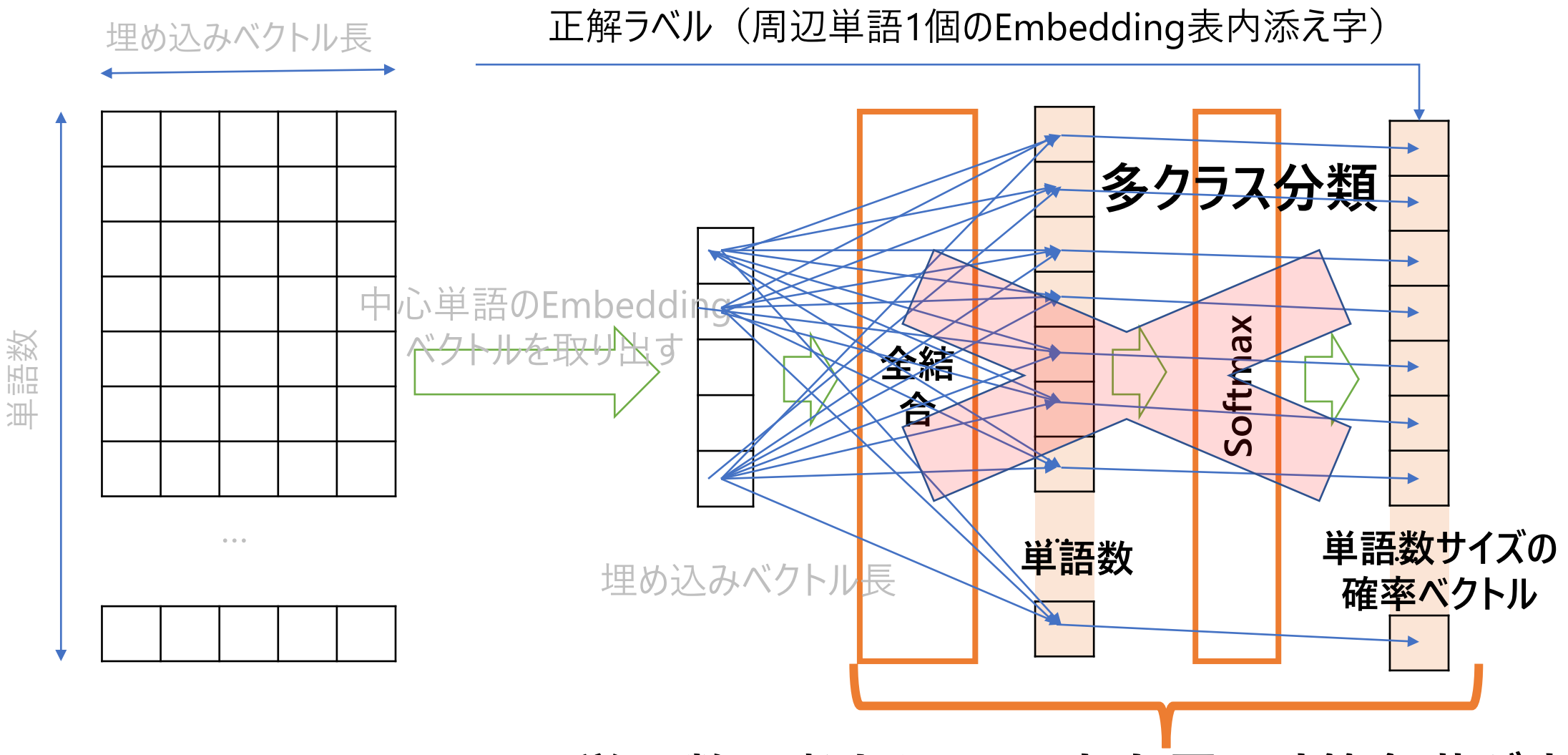
2クラス分類



自然言語処理:
単語ベクトルの導出ー
Negative Sampling

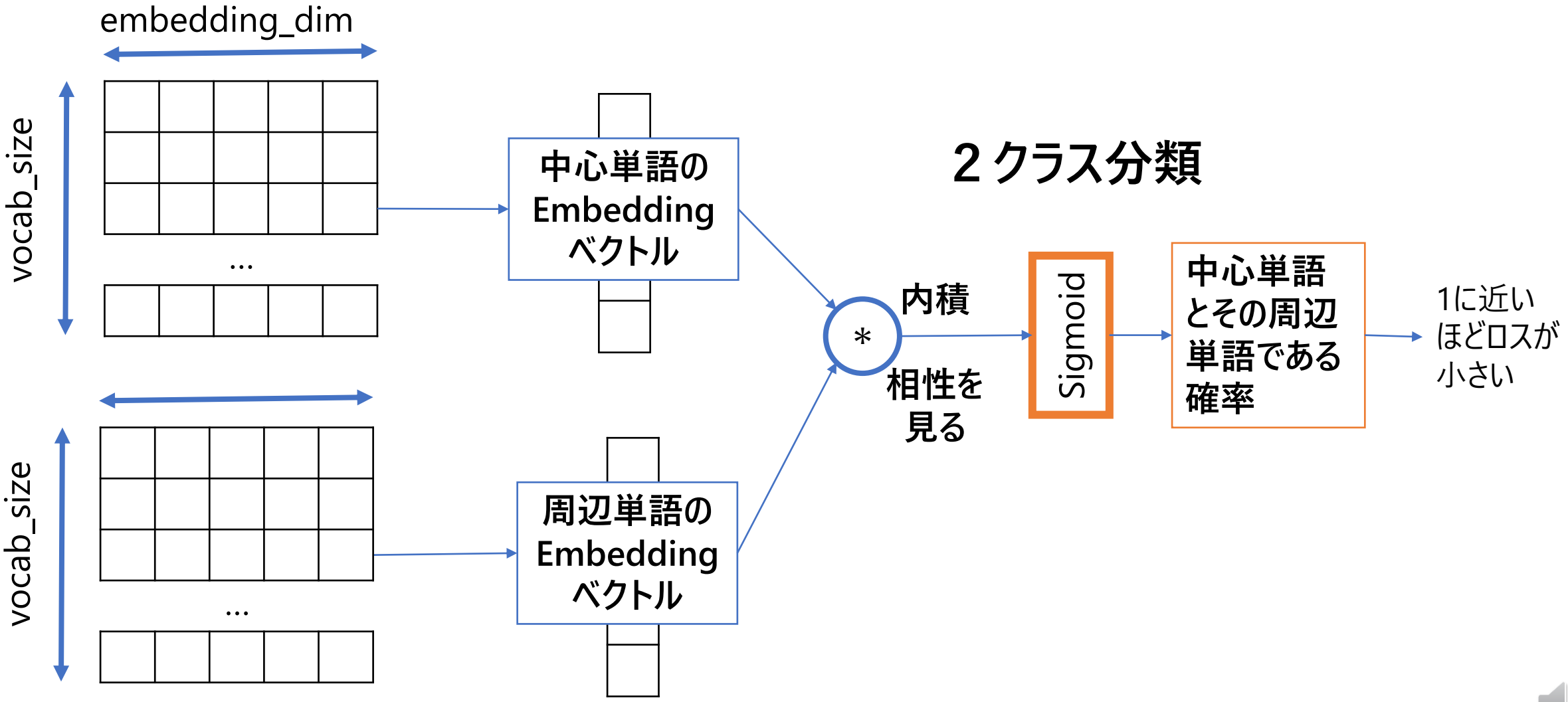
解説動画

CBOWやSkipGramの問題点



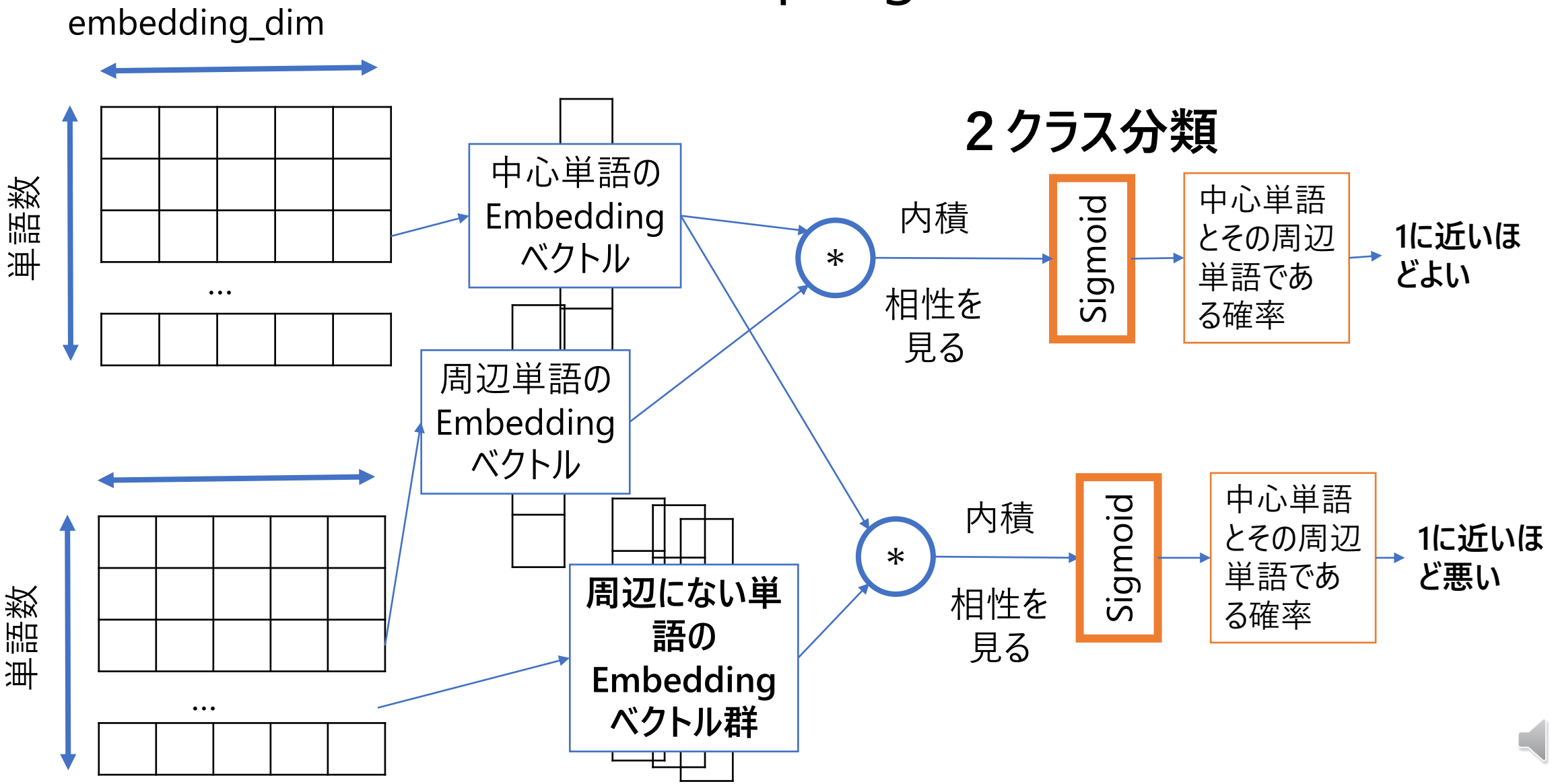
単語数は膨大なので、出力層の計算負荷が高い

Softmaxによる語彙数分のクラス分類問題を、Sigmoidで2クラス分類問題に変える



Negative Sampling

正解でない単語もSamplingして学習に加える



gensimパッケージ

gensim パッケージ

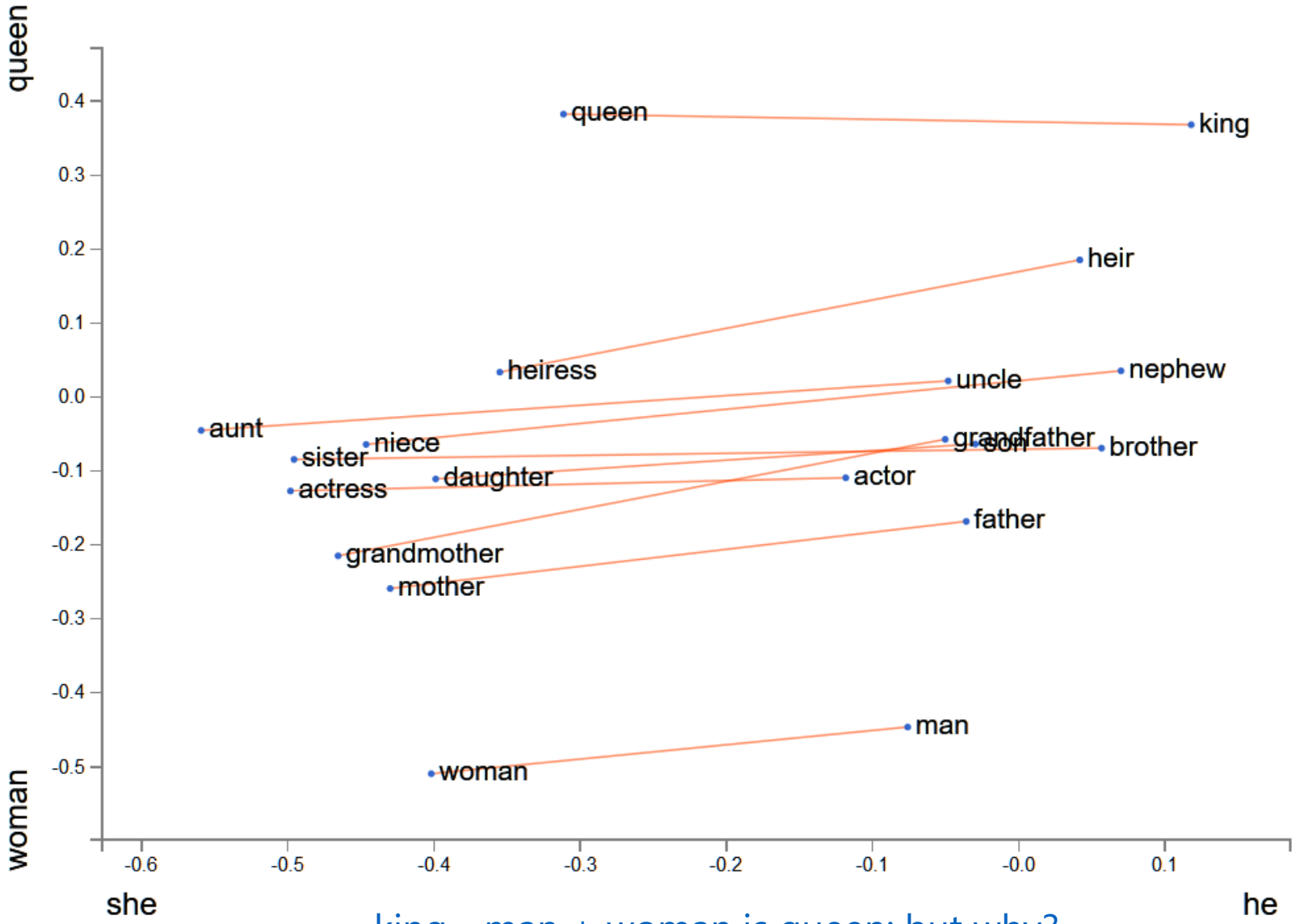
- 単語から文書まで、ベクトル化して、モデル作成できる Python Package。
  - ホームページ　https://radimrehurek.com/gensim/
  - 単語ベクトルを作ってくれる　https://techacademy.jp/magazine/30591
  - 100本ノックの課題では、出来合いの単語ベクトルをロードして、いろいろ操作するメソッドを利用します。

単語ベクトルの効果

単語ベクトルの性質

- 単語埋め込み表現は、ある単語を、周辺単語の何らかの特徴群を抽出することで、表現する。それら特徴は、その単語の形態的、構文的、意味的な属性をいろいろ含む。
- 100本ノックの課題の中に、単語ベクトルの足し引きをして、単語ベクトルが単語の意味を表現していることを体験するものがあります。Enjoy。

king - man + woman = queen



king - man + woman is queen; but why?

参考資料

- オリジナルペーパ:　“Efficient Estimation of Word Representations in Vector Space”
- 解説動画
  - Deep Learningで行う自然言語処理入門
  - 【レクチャー: word2vecの概要】自然言語処理とチャットボット: AIによる文章生成と会話エンジン開発
- king - man + woman is queen; but why?
- Python 実装
  - word2vec(Skip-Gram with Negative Sampling)の理論と実装2
  - https://github.com/theeluwin/pytorch-sgns/blob/master/model.py
- genism の Word2Vec パッケージ
  - gensimのWord2Vecを使ってみる。
  - gensimを使ってWikipediaの全日本語記事からWord2Vecを作る

課題に即した補足

スピアマン相関係数

- 順位の相関を求める

$$\rho = 1 - \frac{6 \sum D^2}{N^3 - N}$$

で定義される。ただし

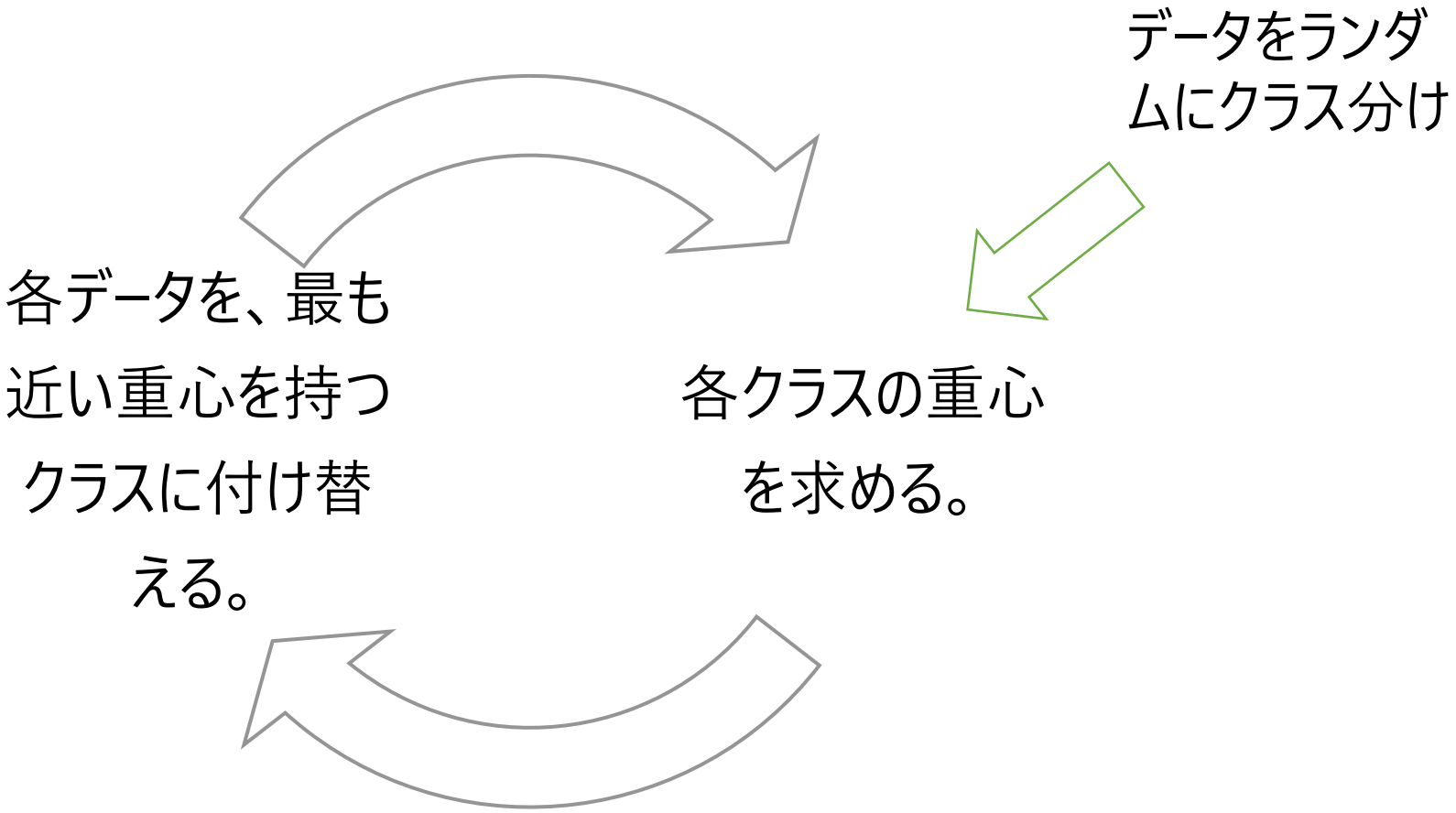
D = 対応するXとYの値の順位の差

N = 値のペアの数

自然言語処理：
機械学習、クラスタリング

解説動画

k-means クラスタリング

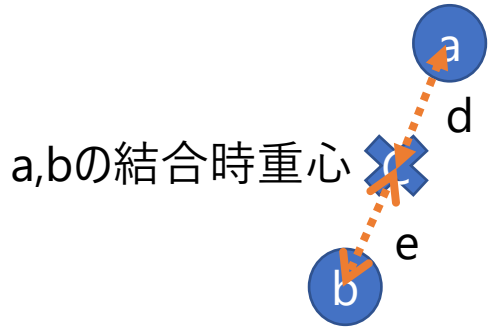


ward クラスタリング

それぞれのデータを
それぞれ１クラスに
する。

全ての２クラス組み合わせに関し、結合した時の重心と結合する各クラスの重心のまとまり具合を求める

結合する前と比べて、まとまり具合が最も変化しないペアを一つのクラスとしてまとめる。



まとまり具合＝ $d^2 + e^2$
これが大きいほど離れていて、
小さいほどまとまっている

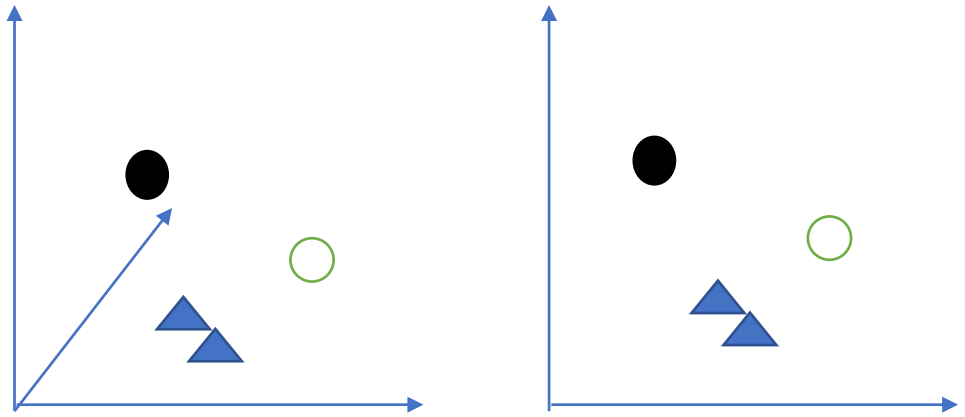
https://www.albert2005.co.jp/knowledge/data_mining/cluster/hierarchical_clustering
https://mathwords.net/wardmethod

自然言語処理:
機械学習、t-SNE

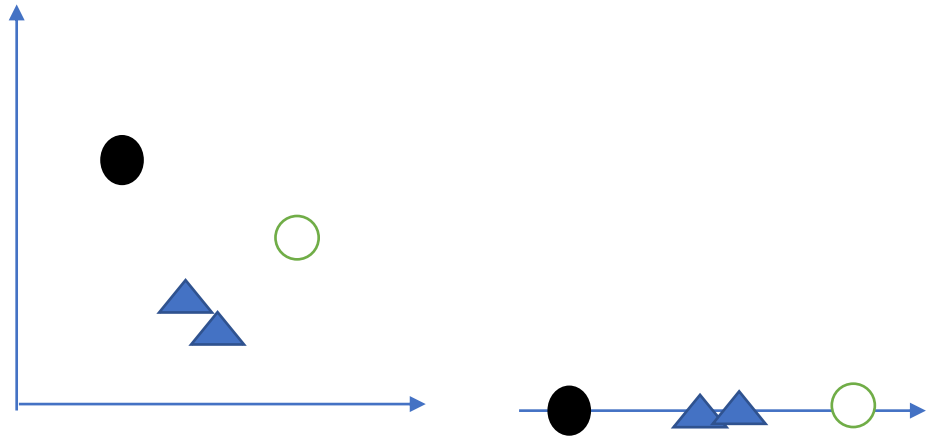
解説動画

t-SNE (t-Distributed Stochastic Neighbor Embedding)：次元圧縮

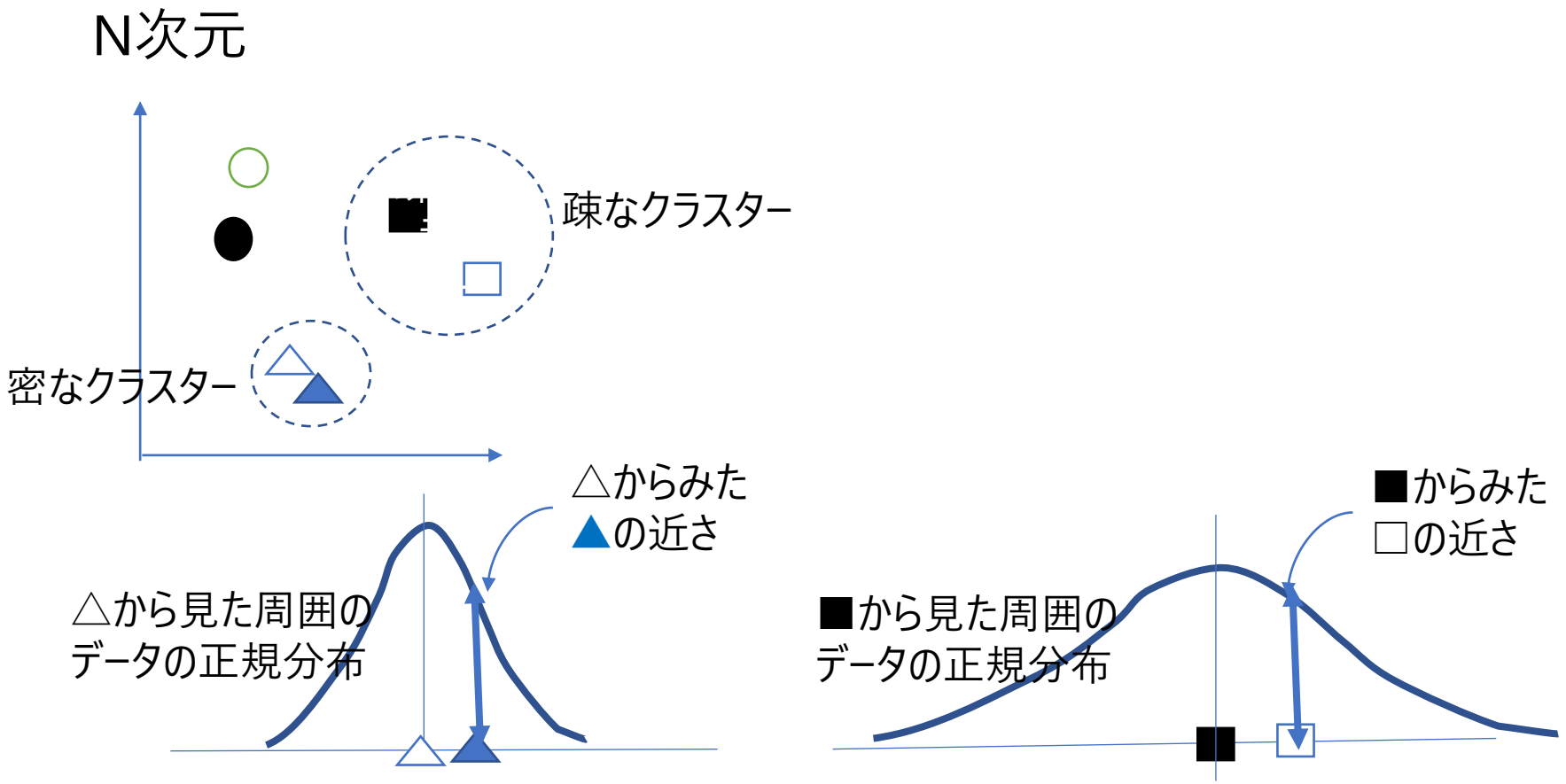
3次元　－＞　2次元



2次元　－＞　1次元

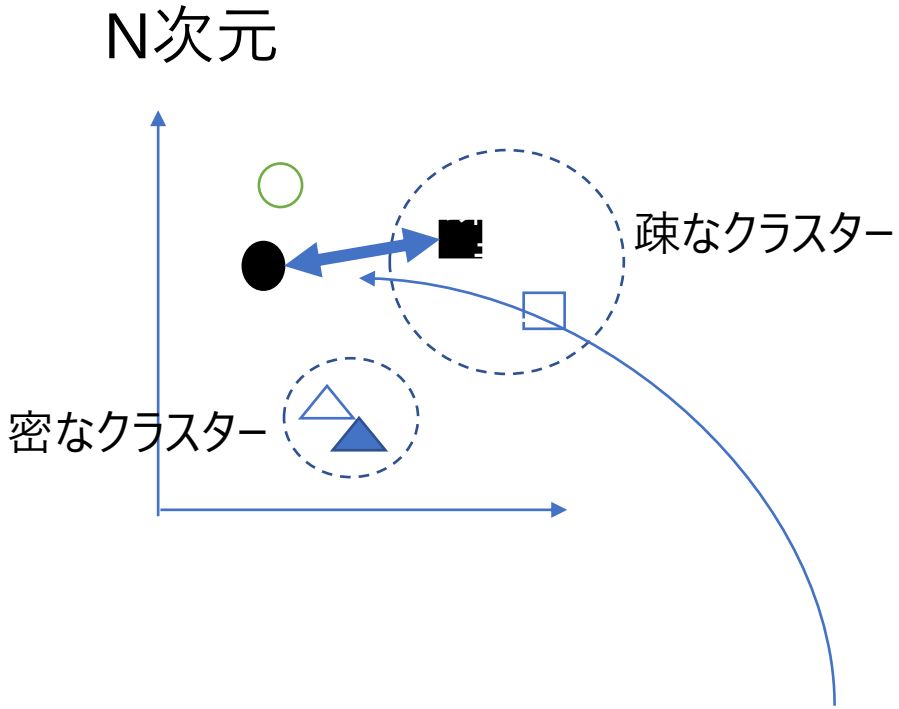


t-SNE：近さの正規化



正規分布の形状は、密なクラスターでは急に、疎なクラスターはなだらかにし、どちらもまとまっているなら近いとなるように、注目点に応じて変更する。

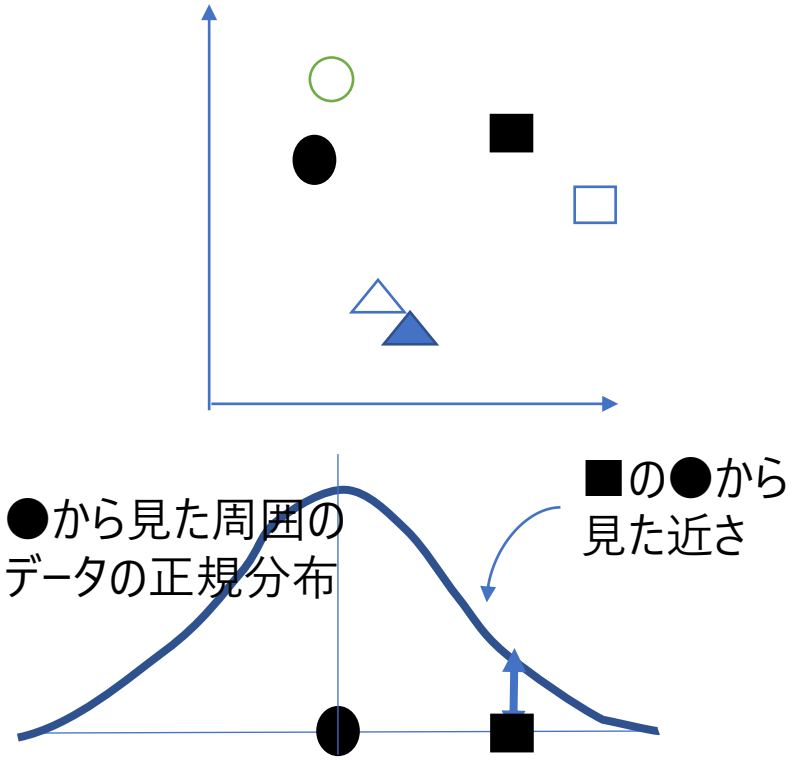
t-SNE：平均をとる



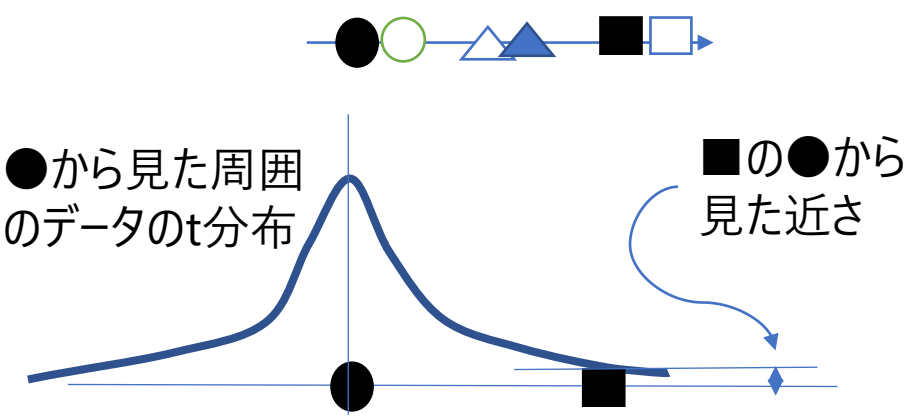
方向によって、●からみた■と、■から見た●の近さは異なるので、平均をとる。

t-SNE：t-分布の役割

N次元



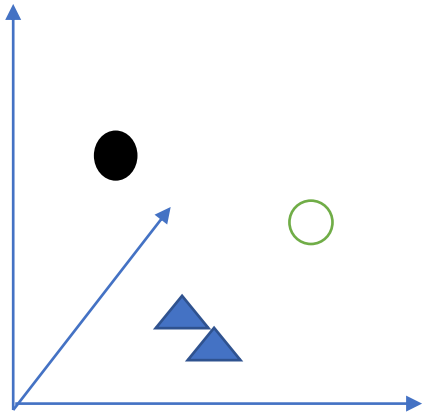
圧縮後の次元



N次元で遠いものは、次元圧縮後に、より遠い（近さが小さい）ものと扱う

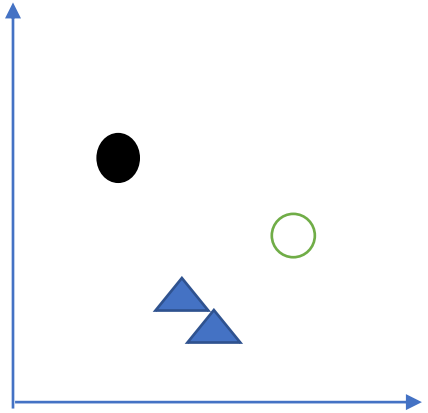
t-SNE：処理ステップ

N次元



元のN次元で、各データポイントの周囲の点の正規分布を得て、それを使って各データポイント間の近さを求め、ＳＮ(ij)とする

2次元



データをランダムにプロットする

2次元空間で各データポイント間の近さ S2(ij) を求める

データポイント1個づつ取り上げて、S2(ij)とSN(ij)の違いが小さくなる方向へ、データを動かす。収束したら終了。

t-SNE 参考資料

- 英語だがわかりやすい
- PCAとの対照データが印象的

100本ノック第７章課題：単語ベクトルの利用60～63、機械学習66～69

- この章の課題は、単語ベクトルがすでにあるものとして、使ってみるものです。前半の60～63で、単語ベクトルがどういうものか、その効果が興味深く示されます。後半の66～69は、単語ベクトルを利用してできること、面白いネタが並んでいます。「NLP、単語ベクトル.ipynb」というノートをコピーし、サンプルコードを完成させ、実行してください。ノートには解説やコメントを入れています。

オプション課題：CBOW, SkipGram, NegativeSampling コード読解

- cbow_skipgram_negativesampling.ipynb に、cbow, skipgram, negative sampling で、単語ベクトルを学習するコードのひな型を入れてあります。学習のため、ミニバッチを使わず、おまけ処理を極力省いています。予備知識で説明したことのコード版です。

オプション課題：gensimで単語ベクトルを作るコード読解

- 自然言語処理アプリを組むときは結局、単語ベクトルを自作する必要があります。自然言語アプリを組む人は、まずgensimを利用すると思います。
- gensim_word2vec.ipynbに、小規模ながら、gensimパッケージで、単語ベクトルを作成するコードを入れました。読解し、実行してください。
- 参考：データが大きく時間がかかりすぎますが、wikipedia_corpus.ipynbにもgensimを使った単語ベクトル作成コードがあります。

確認クイズ

- 確認クイズをやってください。