

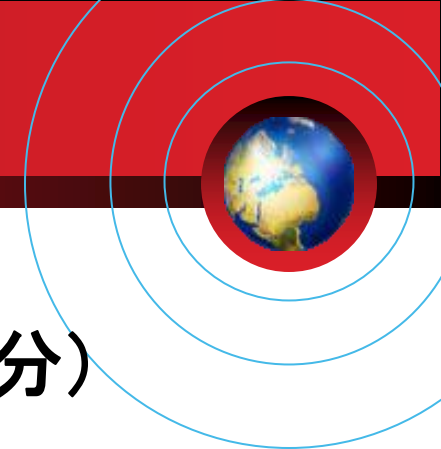
深層学習入門：深層学習とはなにか？



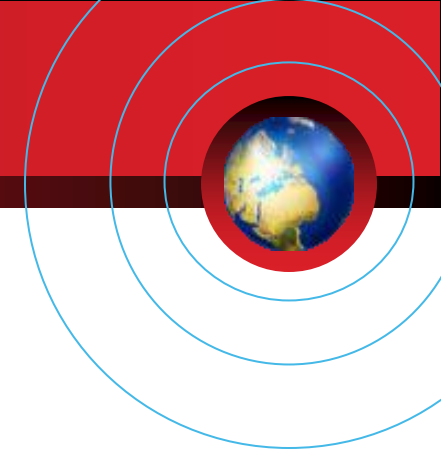
永田亮(甲南大学／理研)

川崎義史(東京大学)

内田諭(九州大学)



- **導入：深層学習とはなにか？（15分）**
 - － 入門編：ある観点から**大胆に要約**
- **基礎：深層学習を利用した言語分析（30分）**
 - － 深層学習ベースの自然言語処理
 - － そこから得られる**副産物が重要**
 - － 単語の意味用法を考慮した検索



■ 分かり易さ > 厳密性

- 分かり易さを重視
- 厳密性を少し欠く部分があります
- 詳細は関連文献などを参照してください

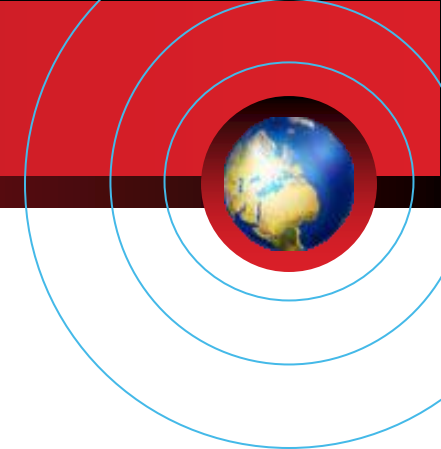
■ 質疑

- なんでも質問してください
- 初歩的な質問也大歓迎

導入: What is 深層学習?



深層学習とは



AI=人工知能

機械学習

深層学習=ニューラルネット

自然言語処理(NLP)

利用

深層学習 \div 単なる数値(列)変換機構

ようこそ深層学習(数値列変換)の 世界へ



画像認識の場合(画素値→カテゴリID)



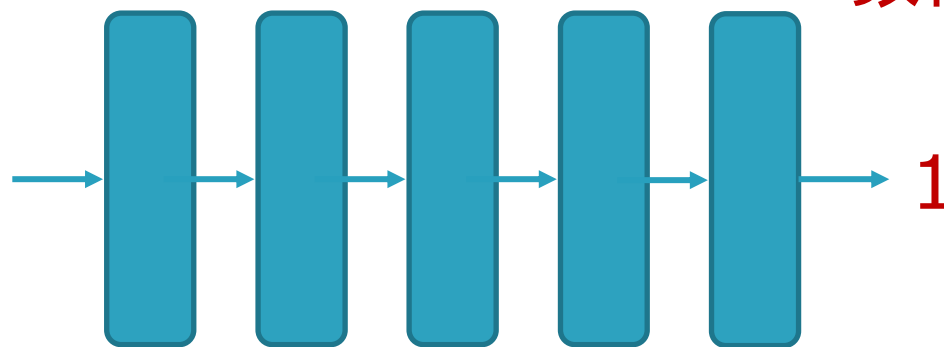
画像



カテゴリID

数値

数値列

$$\begin{pmatrix} 0.7 \\ 1.7 \\ -1.6 \end{pmatrix}$$


1

- 1. 犬
- 2. 車
- ⋮
- N. 猫

深層学習(=ニューラルネット)
(足し算, 掛け算など)

画像認識(分類)ニューラルネット

画像認識の場合（画素値→カテゴリ確率）



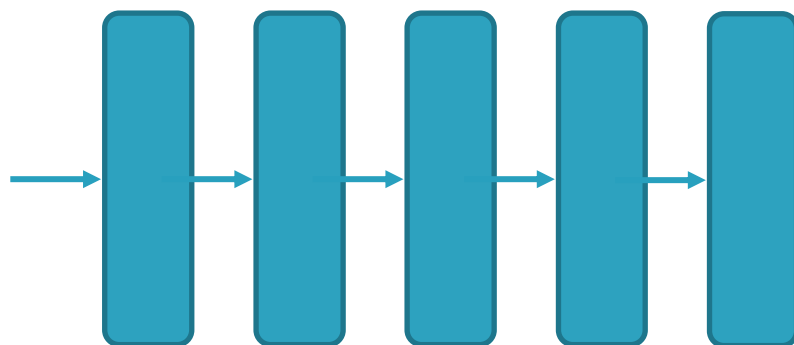
画像



カテゴリID

各カテゴリの確率
数値列

数値列 $\begin{bmatrix} 0.7 \\ 1.7 \\ -1.6 \end{bmatrix}$



$\begin{bmatrix} 0.8 \\ 0.0 \\ \vdots \\ 0.2 \end{bmatrix}$ 1. 犬
2. 車
:
N. 猫

各カテゴリの確率も予想可能！

それでもやっぱり数値列→数値列

数値列変換の中身(ひとつの例)



画像



カテゴリID

$$\begin{pmatrix} 0.7 \\ 1.7 \\ -1.6 \end{pmatrix} \rightarrow \begin{matrix} 0.7 * \textcolor{red}{1} + 1.7 * \textcolor{red}{1} - 1.6 * \textcolor{red}{1} \\ \vdots \\ 0.7 * \textcolor{red}{1/7} + 1.7 * \textcolor{red}{1} - 1.6 * \textcolor{red}{1} \end{matrix} \begin{pmatrix} 0.8 \\ 0 \\ \vdots \\ 0.2 \end{pmatrix} \begin{matrix} \textcolor{red}{1}. \text{ 犬} \\ 2. \text{ 車} \\ \vdots \\ N. \text{ 猫} \end{matrix}$$

訓練(=学習): 赤い数値&数式の種類の決定
パラメータ数 : 赤い数値の数

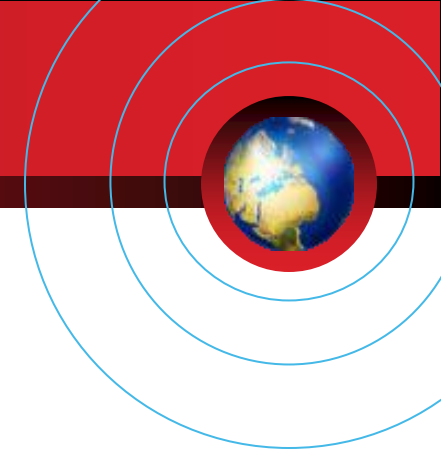


■ 深層学習 (= ニューラルネット)

- 結局のところ **数値列変換機構**

i.e., 数値列 → 数値列

- 訓練: 入力数値が所望の数値になるよう調整



■ 深層学習

- 総称的(理論, 手法全体, . . .)

■ ニューラルネットワーク

- ネットワークそのもの(ネットワーク図)
- ニューラルネット, ニューラルモデルとも

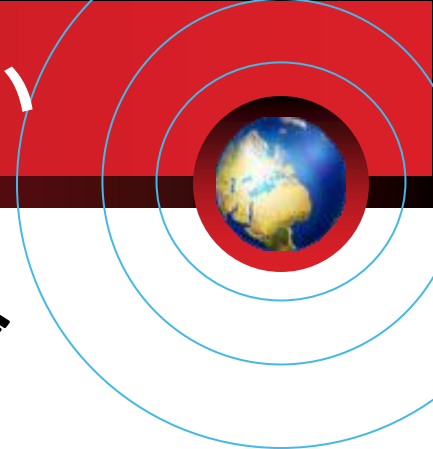
■ ベクトル

- 数値列(数値一個も含む)

基礎：深層学習の言語データへの適用



大きな壁：言語データは数値でない



- 単語，文，文書，色々あるけれど
 - 数値でない
 - カテゴリカルなデータ
- 深層学習
 - 数値列変換機構

言語データは深層学習で扱えない？

結論からいうと: 単語の数値列化で対応







■ 単語を数値列 (=ベクトル) に変換!

- 従来: 単語は記号 (例: Japan \neq France)
- 現在: 単語を数値列 (単語ベクトル) に



似た単語は似た数値列に

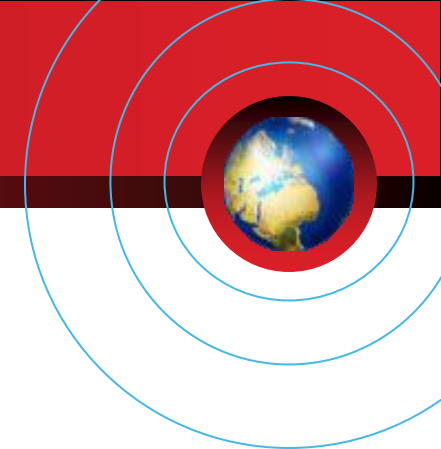
実際は数百～数千の数値の列

Tokyo  Japan  France  Paris 

$$\begin{bmatrix} 1.0 \\ -1.2 \end{bmatrix} - \begin{bmatrix} -0.3 \\ 1.0 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0.7 \end{bmatrix} = \begin{bmatrix} 0.8 \\ -1.5 \end{bmatrix}$$

打開策: Language Model(LM)





■ NLPでは**単語予測器**のこと

- 次に来る単語は？

In Valencia, I enjoyed ...

- 複数回適応することで文章の生成も可！

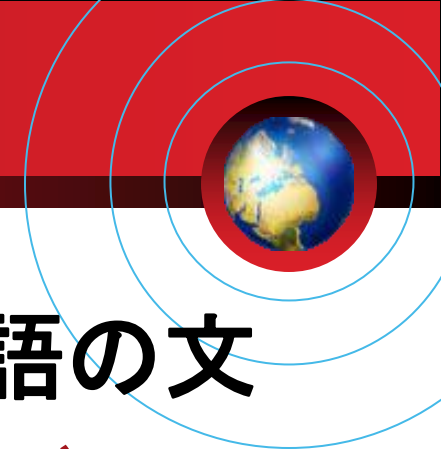
In Valencia, I enjoyed some ...

In Valencia, I enjoyed some paella...

■ 複数種類のLM

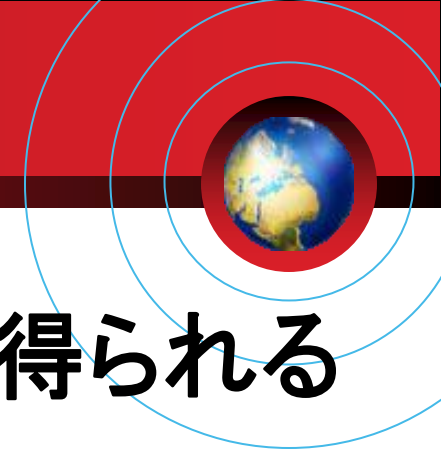
- 次の単語を予測：GPT, ChatGPTなど
- 文中のある単語を予測（穴埋め）：**BERT**

LMでなんでも解けてしまう！？



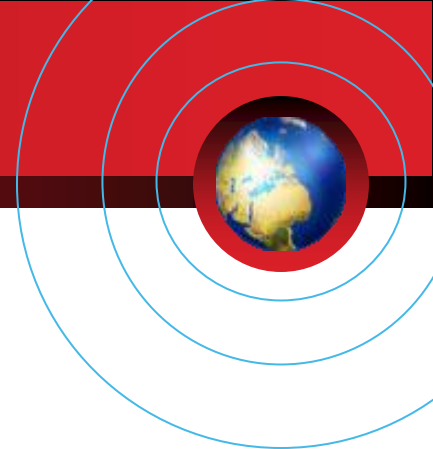
- 翻訳：ある言語の文から別の言語の文
 - *In Valencia, I enjoyed paella* 私はバレンシアでパエリアを楽しんだ.
- 質問応答：質問文から解答文
 - パエリアの発祥の地はどこですか？ パエリアはスペインのバレンシアで生まれました.

ChatGPTは数値変換機構を用いて
単語予測をしているに過ぎない！



- 処理過程で数値列(ベクトル)が得られる
 - 単語ベクトル
 - 文ベクトル
- 記号から数値列へ
 - 各種演算が可能(前述)
 - 単語や文の類似度が計算可能

代表的なLMの構造と処理の流れ

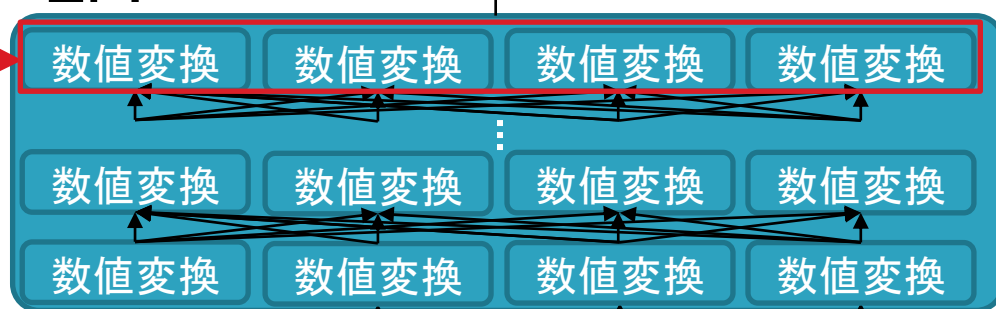


出力：次に来る単語の確率

steak 0.5
stew 0.3
⋮

単語ベクトルとして利用

LM



③ベクトル化(LMへ入力)

②ID化:

101, 146, 1108, 8739

①分割(tokenize): BOS (文頭) | I | was | cooking

入力: I was cooking

今回利用する二種類の単語ベクトル



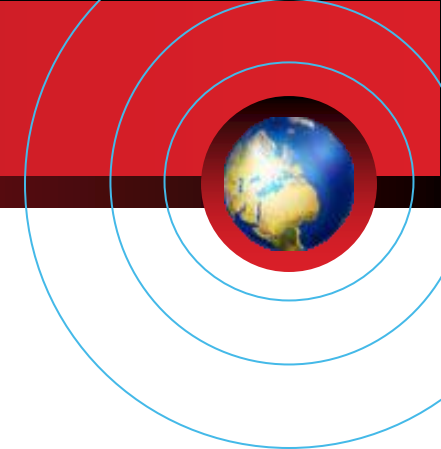
■ Word2vec: 単語タイプベース

- 単語タイプに対して一**種類**の単語ベクトル
- 単語のタイプとしての性質や特徴を反映
例: bankのベクトルは「銀行」も「土手」も反映

■ BERT: 単語トークン(事例)ベース

- 各事例に応じた複数の単語ベクトル
- 文脈を考慮した単語ベクトル
例: *bank* clerk と *river bank* では異なるベクトル

言語分析で有益なのは



■ 単語ベクトル

- 言語モデルの副産物
- (単語)分散表現とも

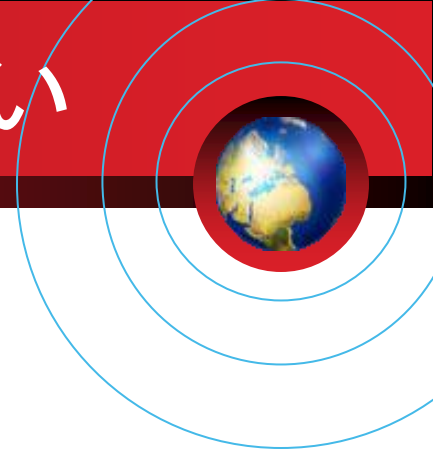
■ 文ベクトル, 文書ベクトル

- 単語ベクトルと同様な方法で

■ 言語モデルそのもの

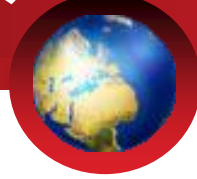
- 言語データ, 言語現象のモデルとして
- 人間の言語能力のモデルとして

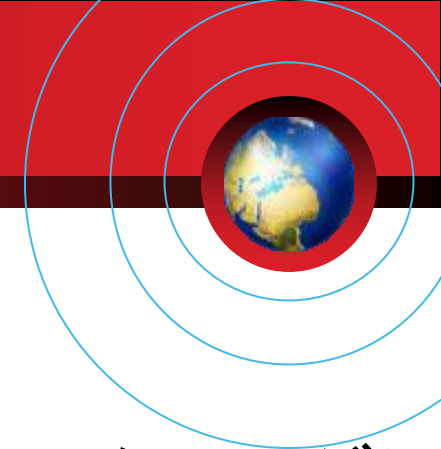
ここまでの重要なポイントのおさらい



- 深層学習(=ニューラルネット)
 - 結局のところ数値列変換機構
- 言語データへの適用
 - 単語も数値列に変換(by LM)
- 言語モデル
 - 単語予測器
 - 副産物の単語ベクトル

基礎：深層学習の言語データへの適用 強力なツール：単語ベクトル





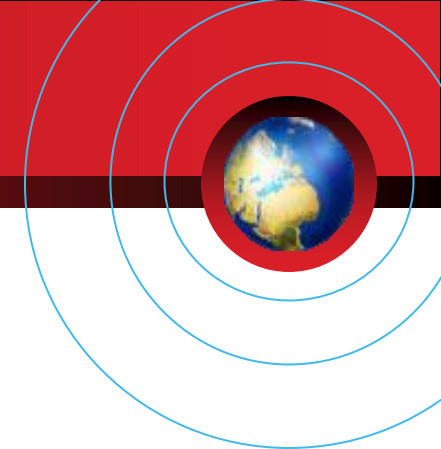
■ 分析ツール

- 曖昧検索, 自動グルーピング(クラスタリング)

■ 言語データ／言語現象のモデル化

- 各種現象の定量化(e.g., 意味変化の度合い)

単語や文が数値列になっているので
なんとなくイメージできる？



- **単語タイプに一種類のベクトル**
 - コーパス中の全事例に応じたベクトル
 - 例：bankのベクトルは銀行も岸も含む
- **様々な呼び名があるので注意が必要**
 - 例：単語分散表現, word embeddings
- **代表的な実装例**
 - Word2vec
 - Glove

応用例(文脈なしベクトル) : 類似単語検索



コーパス

① 単語タイプベースのベクトル 検索対象単語: bank

$\begin{pmatrix} 2.5 \\ \vdots \\ 1.7 \end{pmatrix}$ $\begin{pmatrix} 2.5 \\ \vdots \\ 1.3 \end{pmatrix}$... $\begin{pmatrix} 0.7 \\ \vdots \\ 0.3 \end{pmatrix}$
apple book river

② 類似度計算

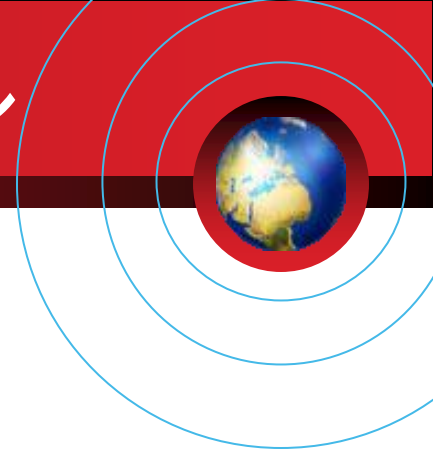


③ 高類似単語出力

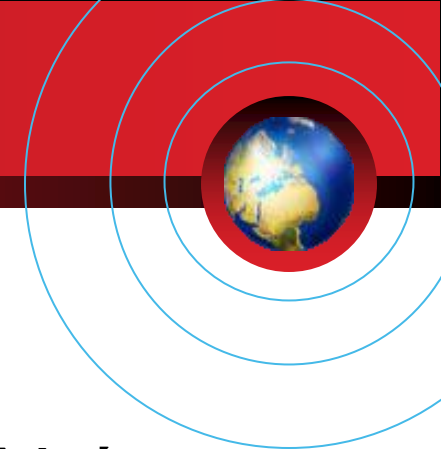
$\begin{pmatrix} 0.6 \\ \vdots \\ 0.3 \end{pmatrix}$
bank

$\begin{pmatrix} 0.5 \\ \vdots \\ 0.2 \end{pmatrix}$ $\begin{pmatrix} 0.7 \\ \vdots \\ 0.3 \end{pmatrix}$
money river

コーパス内で
類似した使われ方をしている単語の発見



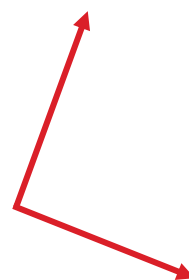
- 単語トークンごとにベクトルを
 - コーパス中の各事例に応じたベクトル
a river *bank* vs. a *bank* account
- 様々な呼び名があります
 - 文献を読む際には注意
- 代表的な実装例
 - BERT (から得られるベクトル)
 - GPT (から得られるベクトル)



- ベクトルは空間(平面)上の矢印
 - 余弦類似度: ベクトル間の角度に対応



こっちのが似ている

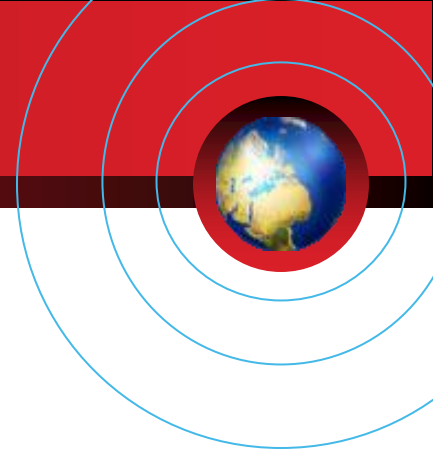


余弦類似度

$$\cos(v_1, v_2) = \frac{v_1 \text{と} v_2 \text{の内積}}{v_1 \text{のノルム} \times v_2 \text{のノルム}}$$

- 深層学習により単語, 句, 文, 文書は
 - ベクトルに!

応用例：曖昧用例検索 (e.g., bank)



コーパス

①対象単語の用例抽出

I have a bank account in Japan
They were sitting on the bank.
⋮

LM

②文脈付きベクトル化

$$\begin{bmatrix} 0.5 \\ \vdots \\ 0.2 \end{bmatrix} \begin{bmatrix} 2.5 \\ \vdots \\ 1.3 \end{bmatrix} \dots \begin{bmatrix} 0.7 \\ \vdots \\ 0.3 \end{bmatrix}$$

検索対象単語

He drew some bucks from the bank.

LM

③文脈付きベクトル化

④類似度計算



⑤高類似事例出力

$$\begin{bmatrix} 0.6 \\ \vdots \\ 0.3 \end{bmatrix}$$