



여보세요

CIAO

HOLA

你好

HALLO

HELLO

สวัสดิ

こんにちは

BONJOUR

イントロダクション



講師紹介



我妻 幸長

Yukinaga Azuma

@yuky_az

SAI-Lab株式会社 代表取締役

AI関連の教育、研究開発に従事

理学博士（物理学）

Udemyで数万人を指導 / 有名企業でAI研修を担当

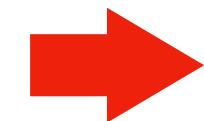
著書に「はじめてのディープラーニング」など



コースの特徴

- **BERTの基礎を学ぶ**
 - Transformer、BERTなどの自然言語処理技術を学びます
- **PyTorchによる実装**
 - Google Colaboratory環境でBERTを動作させます
- **講座の対象**
 - 一步進んだ自然言語処理技術を身に付けたい方
 - ディープラーニングに関する基礎的な知識のある方
 - プログラミング、英語に抵抗感が小さい方

講座の内容



Section 1. 講座とBERTの概要

Section 2. シンプルなBERTの実装

Section 3. BERTの仕組み

Section 4. ファインチューニングの活用

Section 5. BERTの応用

今回の内容

- 
1. イントロダクション
 2. コースの概要
 3. 自然言語処理の概要
 4. Transformerの概要
 5. BERTの概要
 6. Google Colaboratoryの使い方

教材の紹介

- Pythonの基礎

コースの概要

สวัสดิ์

HALLO

HELLO

여보세요

CIAO

HOLA

你好

BONJOUR

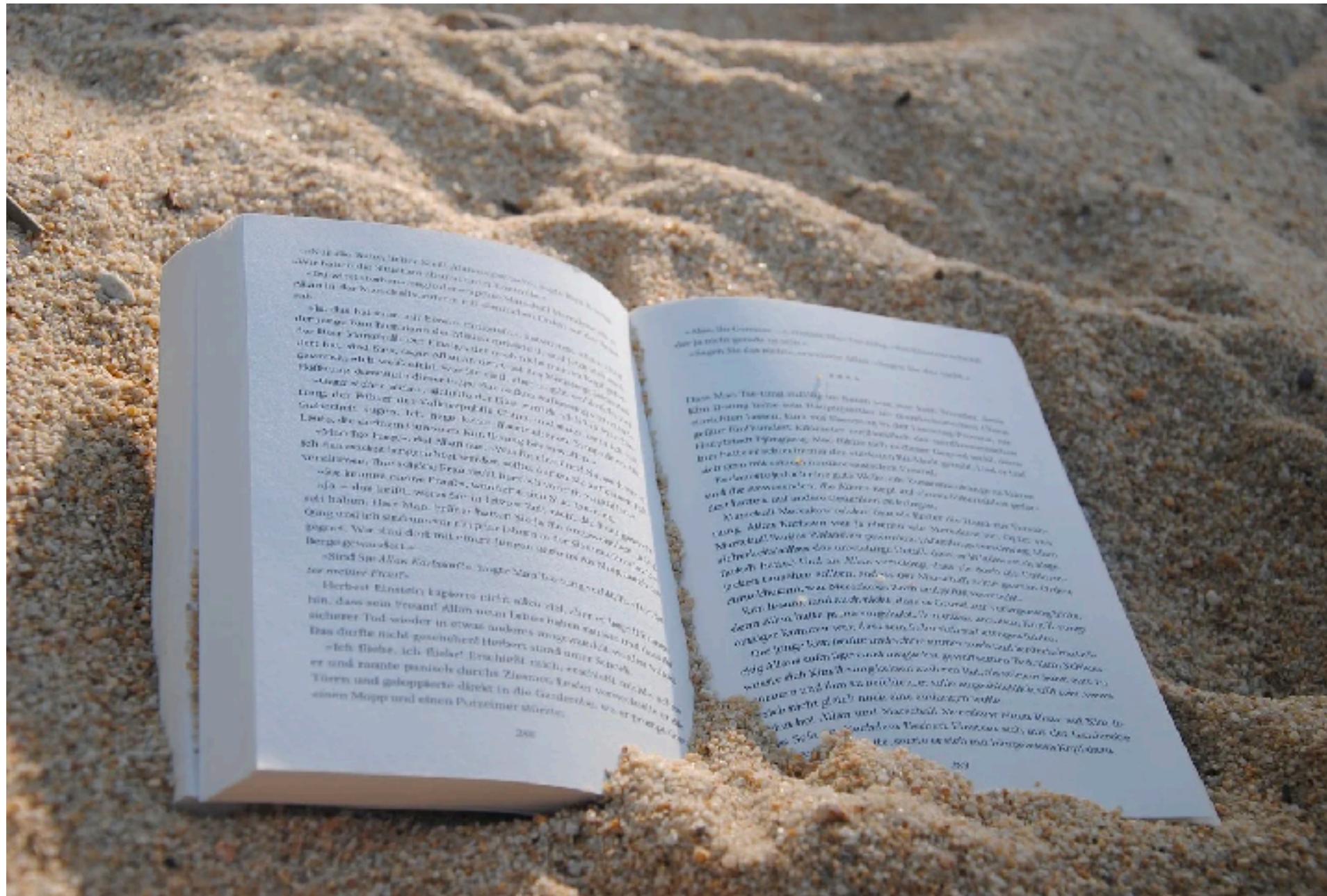
こんにちは

Section 1. 講座とBERTの概要



自然言語処理、Transformer、BERTについて概要を学びます

Section 2. シンプルなBERTの実装



最小限のPythonのコードでBERTを実装します

Section 3. BERTの仕組み



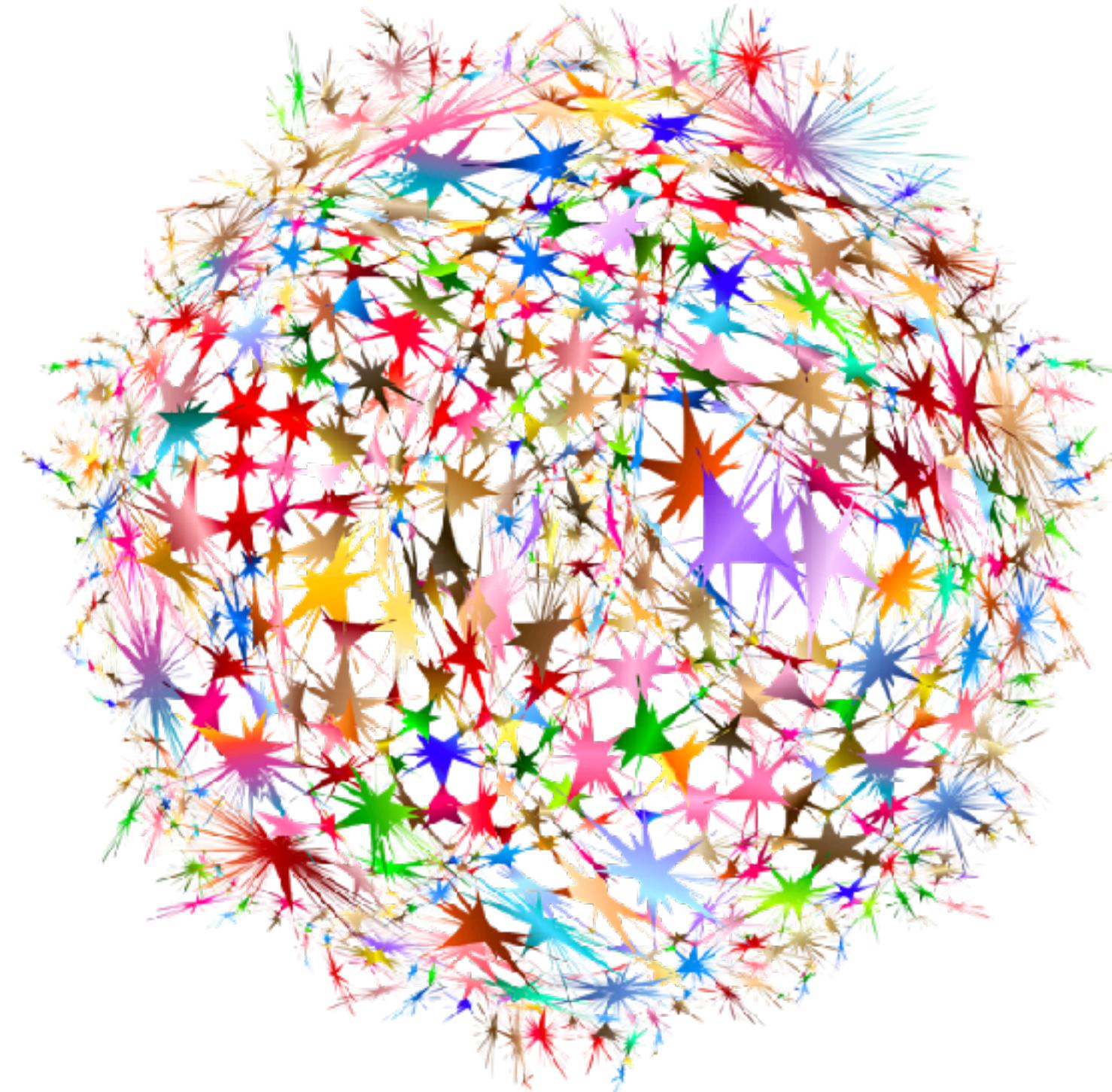
Transformer、BERTなどについて仕組みを詳しく学びます

Section 4. ファインチューニングの活用



ファインチューニングの概要、そしてBERTにおける活用について解説します

Section 5. BERTの応用



BERTを使って、自然言語処理のタスクに取り組みます

自然言語処理の概要



自然言語処理とは

- ・ 自然言語とは、日本語や英語などの我々が普段使う言語のこと
- ・ 自然言語処理（Natural Language Processing、NLP）とは、
自然言語をコンピュータで処理する技術のこと

自然言語処理の応用

- 検索エンジン
- 機械翻訳
- 予測変換
- スパムフィルタ
- 音声アシスタント
- 小説の執筆
- 対話システム
- etc...

自然言語処理技術の要素

- 形態素解析
 - 文書を単語に分割する技術
- 単語の分散表現
 - 文書内での関係性を踏まえて、単語をベクトル化する技術
- 再帰型ニューラルネットワーク (**RNN**)
 - 時系列を扱うのが得意なニューラルネットワークの一種
- Seq2Seq
 - RNNをベースにした、文章などを生成可能なモデル
- etc...

日本語の形態素解析

- 形態素とは、言葉が意味を持つまとまりの単語の最小単位のこと
- 形態素解析とは、自然言語を形態素にまで分割すること
- 日本語や中国語、タイ語は単語間にスペースが無いので、
形態素解析が必要
- 以下は代表的な日本語の形態素解析ライブラリ
 - MeCab → 知名度が高く、高速、高精度
 - Janome → 速度はMeCabに劣るが、導入が簡単
 - etc...

one-hot表現

す も も も も も も も の うち

	すもも	も	もも	の	うち
ID	0	1	2	3	4

「すもも」のone-hot表現: [1 0 0 0 0]

「も」のone-hot表現: [0 1 0 0 0]

分散表現

- 单語間の関連性や類似度に基づくベクトルで、単語を表現する

The diagram illustrates word embeddings as vectors. A horizontal arrow points from the text '200要素程度' (approximately 200 dimensions) down to a rectangular grid representing a vector. The grid has three rows labeled on the left: '男性' (Male), 'ロンドン' (London), and 'Python'. The grid contains numerical values in each cell, with ellipses indicating continuation. The first row has values [0.01, 0.58, 0.24, ...]. The second row has values [0.34, 0.93, 0.02, ...]. The third row has values [0.97, 0.08, 0.41, ...].

男性	0.01	0.58	0.24	...
ロンドン	0.34	0.93	0.02	...
Python	0.97	0.08	0.41	...

- word2vecなどを使えば、足し算や引き算が可能なベクトルを作ることができる

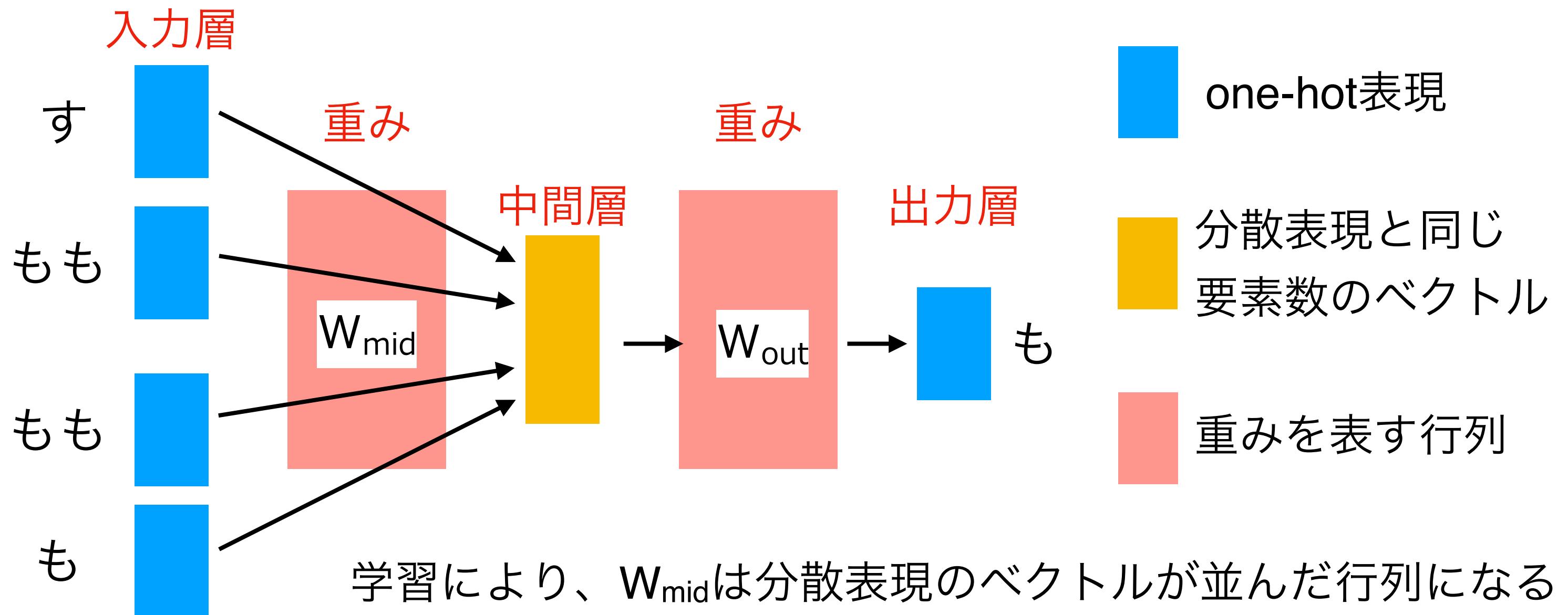
例: 「王」 - 「男」 + 「女」 = 「女王」

word2vec

- word2vecは、分散表現を作成するための技術
- word2vecでは、CBOW (continuous bag-of-words) もしくは、skip-gramというニューラルネットワークが用いられる

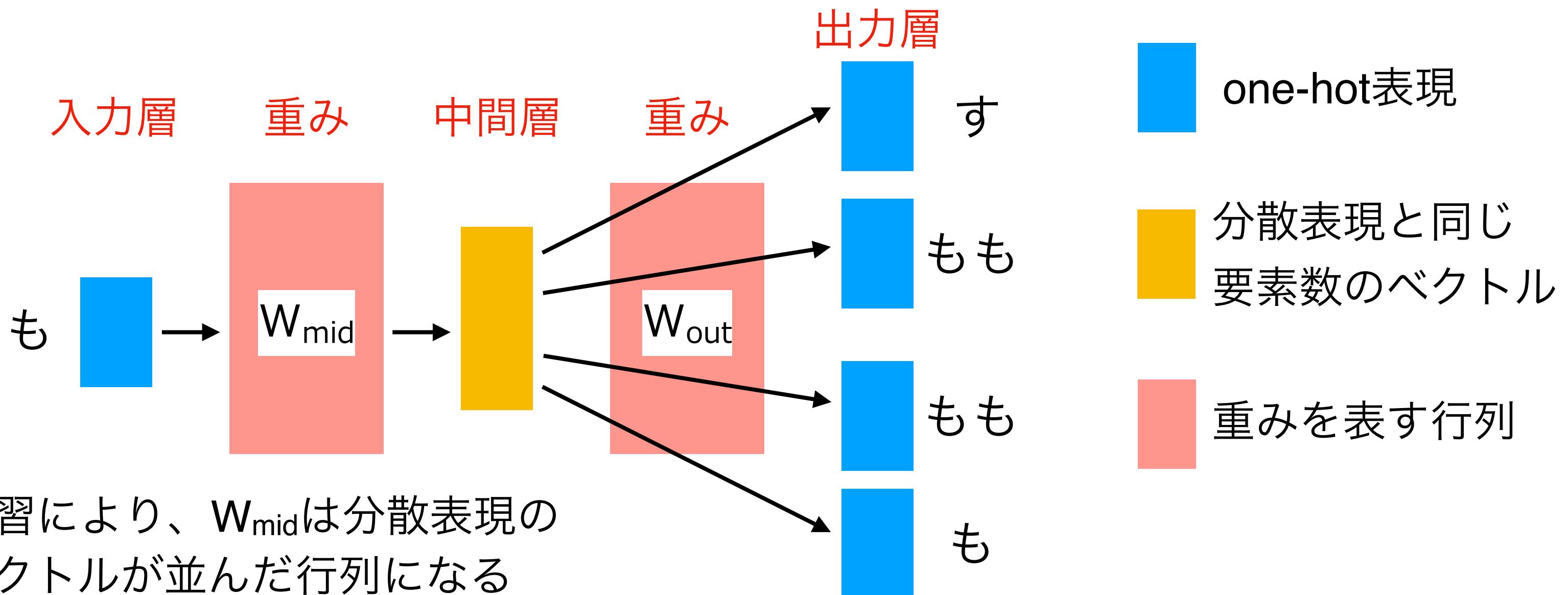
CBOW (continuous bag-of-words)

- 前後の単語から対象の単語を予測するニューラルネットワーク
- 学習に要する時間がskip-gramよりも短い



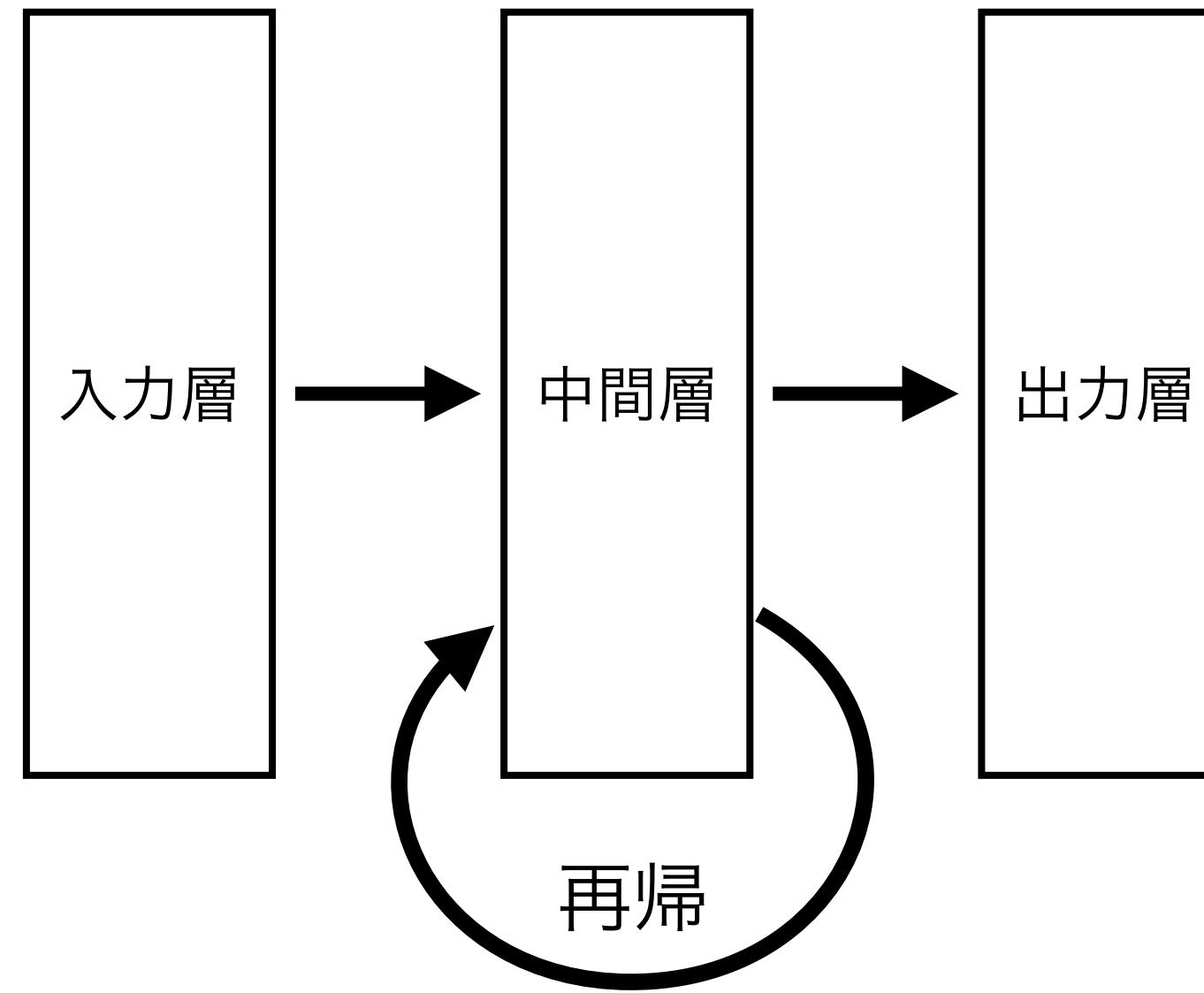
skip-gram

- ある単語から、前後の単語を予測するニューラルネットワーク
- CBOWよりも学習に時間がかかるが、精度がよい

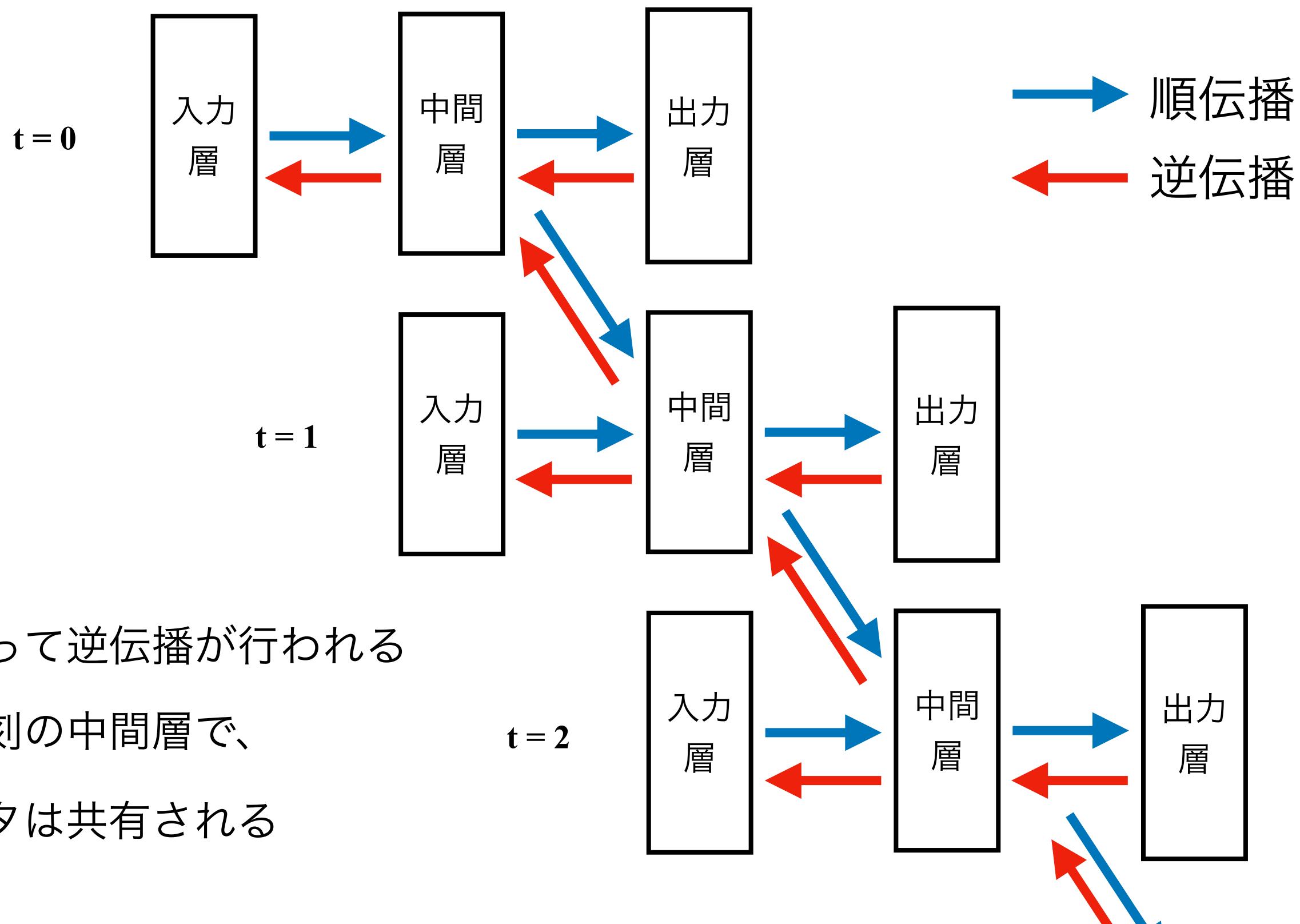


再帰型ニューラルネットワーク (RNN) とは？

- **再帰型ニューラルネットワーク (Reccurent Neural Network)**
 - 入力と正解が「時系列データ」となる
 - 中間層が「再帰」の構造を持ち、前後の時刻の中間層とつながる



RNNの順伝播と逆伝播



時系列データの例

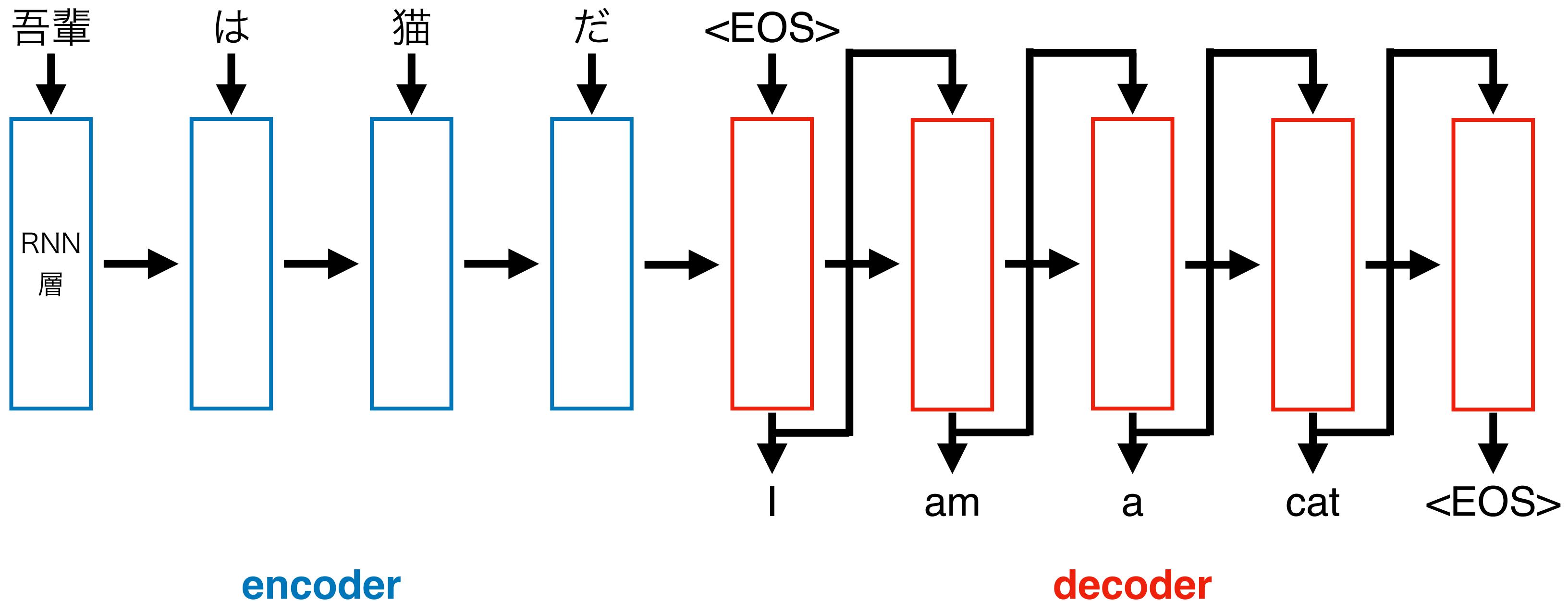
- 文書
- 音声データ
- 動画
- 株価
- 産業用機器の状態
- etc...

S_{eq}2S_{eq}とは

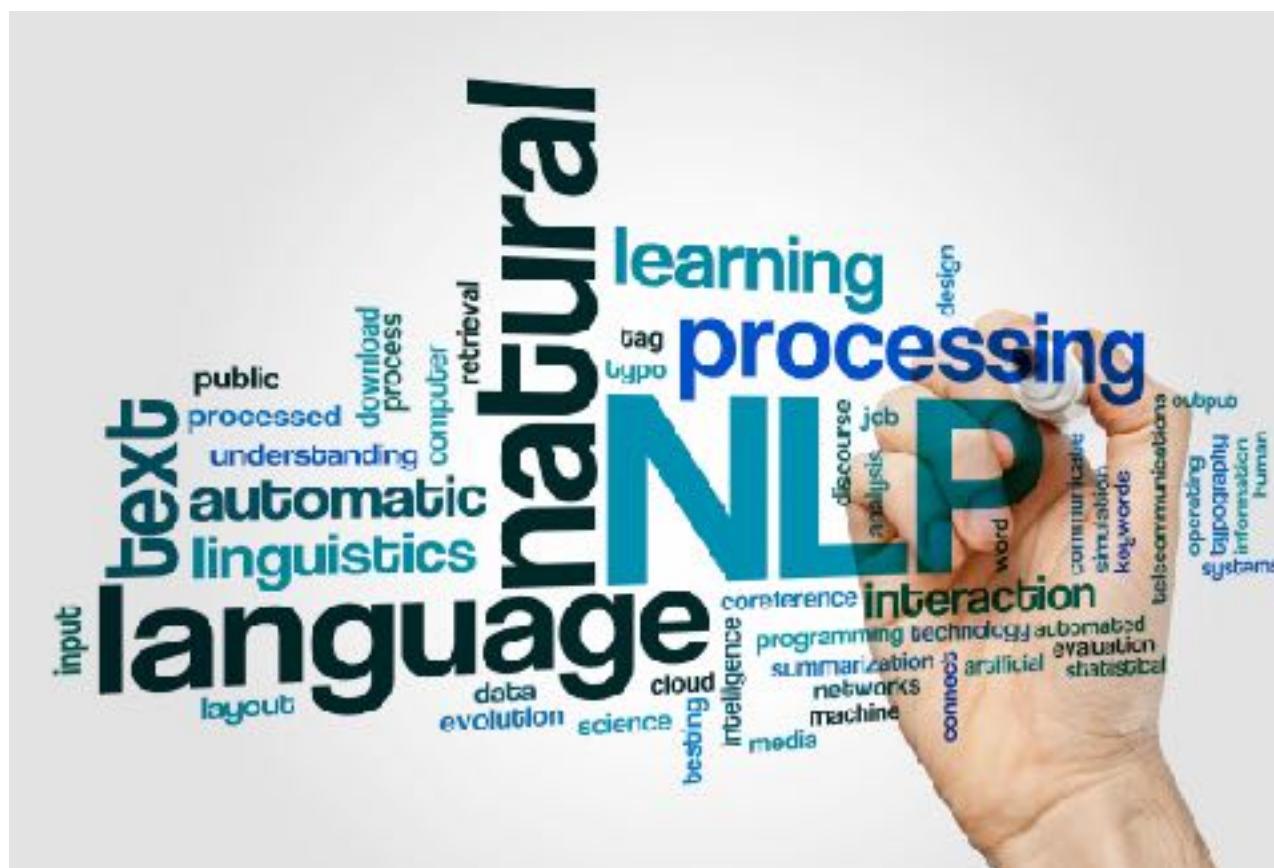
- S_{eq}2S_{eq}は、系列(sequence)を 受け取り、別の系列へ変換するモデル
- 自然言語処理でよく利用される
- 文章などの入力を圧縮するencoderと、出力を展開するdecoderからなる
- 以下は活用例
 - 機械翻訳 (例: 英語の文章 → フランス語の文章)
 - 文章要約 (元の文章 → 要約文)
 - 対話 (自分の発言 → 相手の発言)
 - etc...

S_{eq}2S_{eq}の構造

S_{eq}2S_{eq}による翻訳の例



さらに詳しく学びたい方へ...



Udemyコース

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



Udemyコース

人工知能（AI）を搭載したTwitterボットを作ろう 【Seq2Seq+Attention+Colab】

Transformerの概要



RNNによる自然言語処理の問題点

- **学習時間が長い**
 - データを並列で処理できないため、学習には長い時間がかかる
- **文脈をとらえるのが難しい**
 - 長時間の関係性をとらえるのが苦手

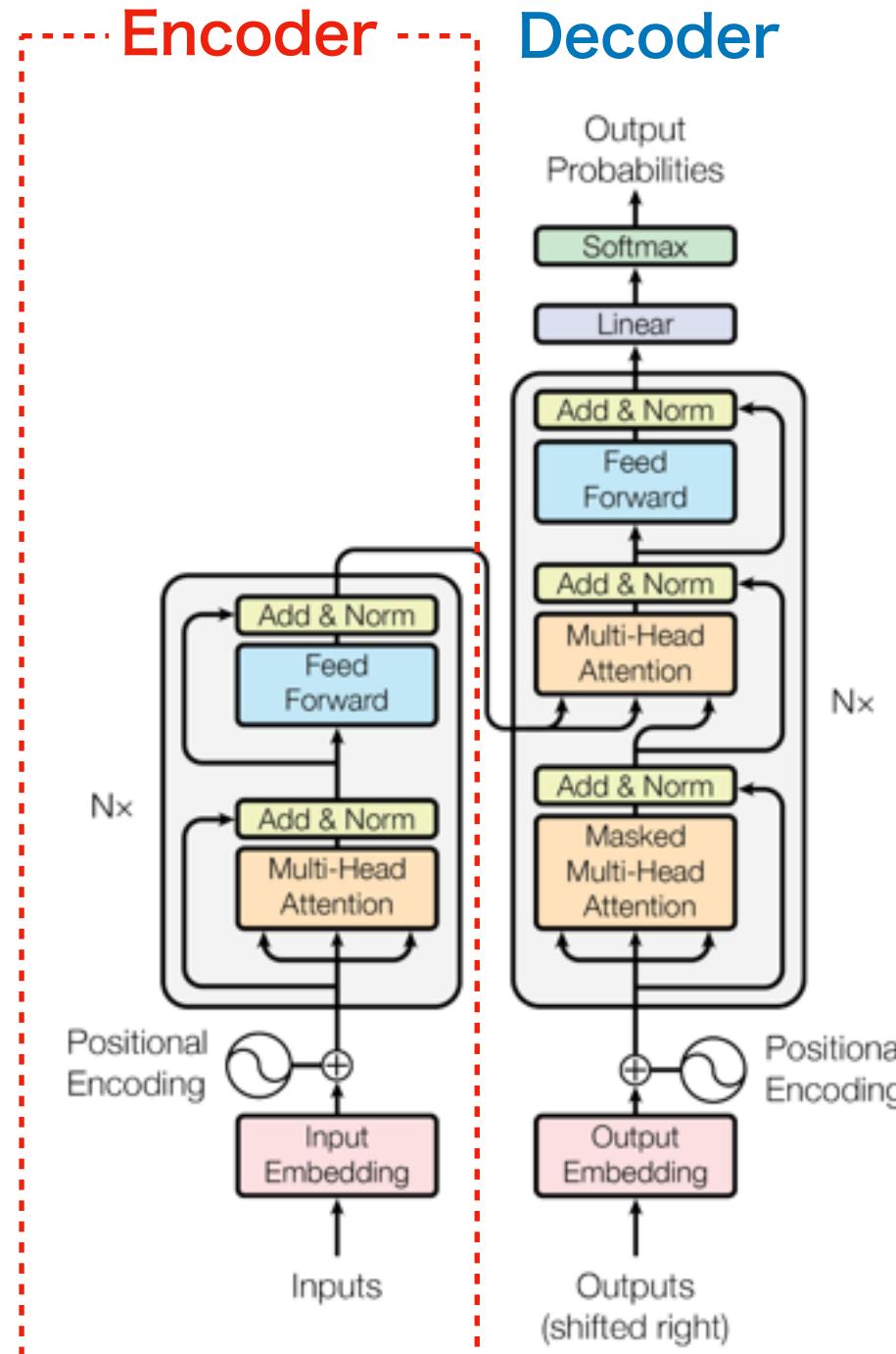
Transformerの概要

- **Transformerとは？**
 - 2017年に導入されたディープラーニングモデルで、
主に自然言語処理の分野で使用される
 - RNNと同様に、自然言語などの時系列データを処理するように設計
されているが、RNNで用いる再帰、CNNで用いる畳み込みは使わない
 - Attention層のみで構築される
 - 翻訳やテキストの要約など、様々なタスクで利用可能
 - 並列化が容易であり、訓練時間を大きく削減できる

Transformerの論文

- **Attention Is All You Need**
→ <https://arxiv.org/abs/1706.03762>
- 「Attention」は時系列データの特定の部分に
注意を向けるように学習させていく方法

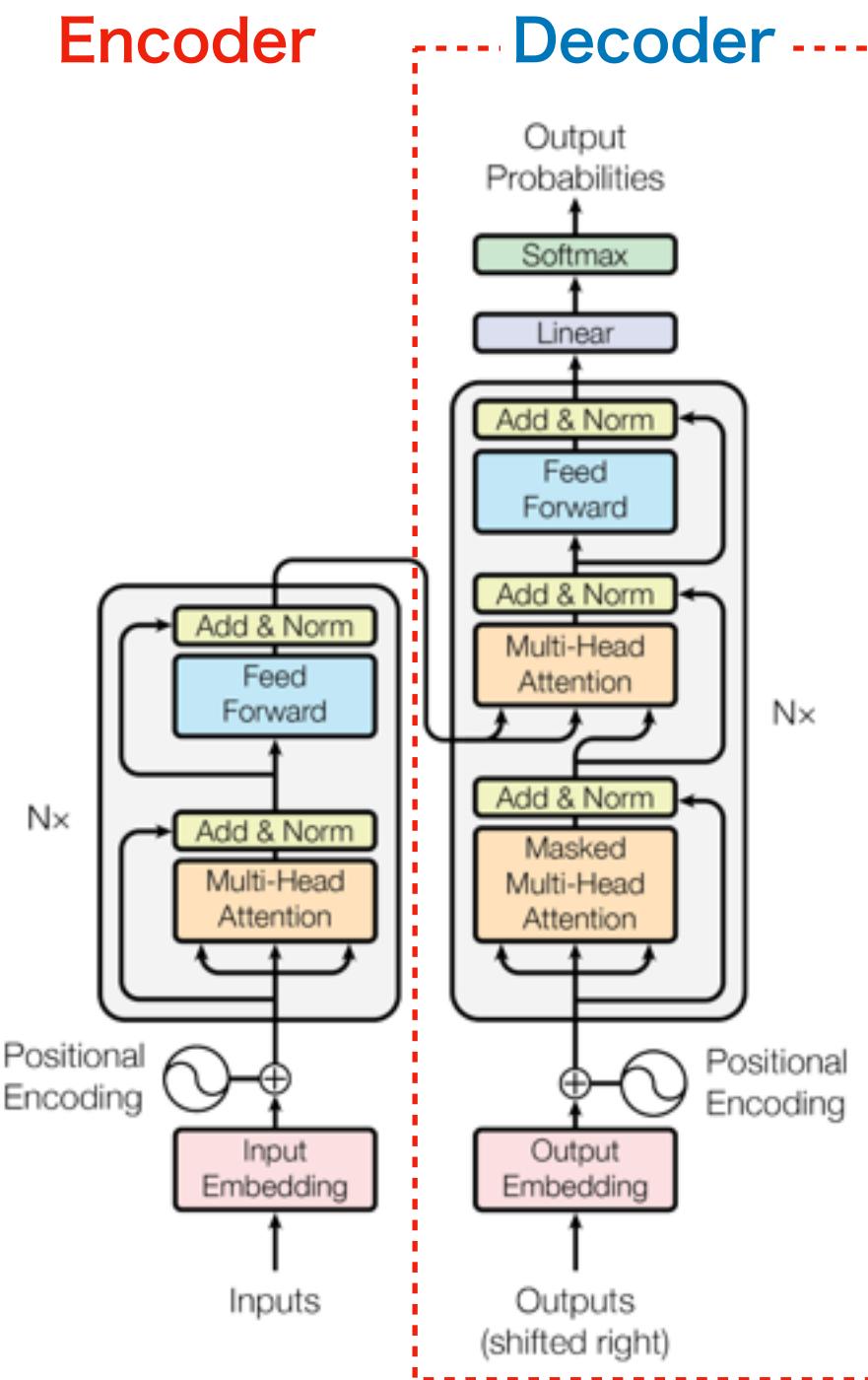
Transformerのモデル



Encoderの構造

1. Embedding層により入力文章をベクトルに圧縮
 2. Positional Encoder層によって位置情報を加える
 3. Multi-Head Attention層
 4. normalization (正規化) など
 5. Positionwise fully connected feed-forward network
 6. normalization (正規化) など
- 3-6を6回繰り返す

Transformerのモデル



Decoderの構造

1. Embedding層により入力文章をベクトルに圧縮
 2. Positional Encoder層によって位置情報を加える
 3. Multi-Head Attention層
 4. normalization (正規化) など
 5. Multi-Head Attention層 (Encoderの入力を使用)
 6. normalization (正規化) など
 7. Positionwise fully connected feed-forward network
 8. normalization (正規化) など
- 3-8を6回繰り返す

BERTの概要

สวัสดิ์

こんにちは

여보세요

CIAO

HOLA

你好

HALLO

HELLO

BONJOUR

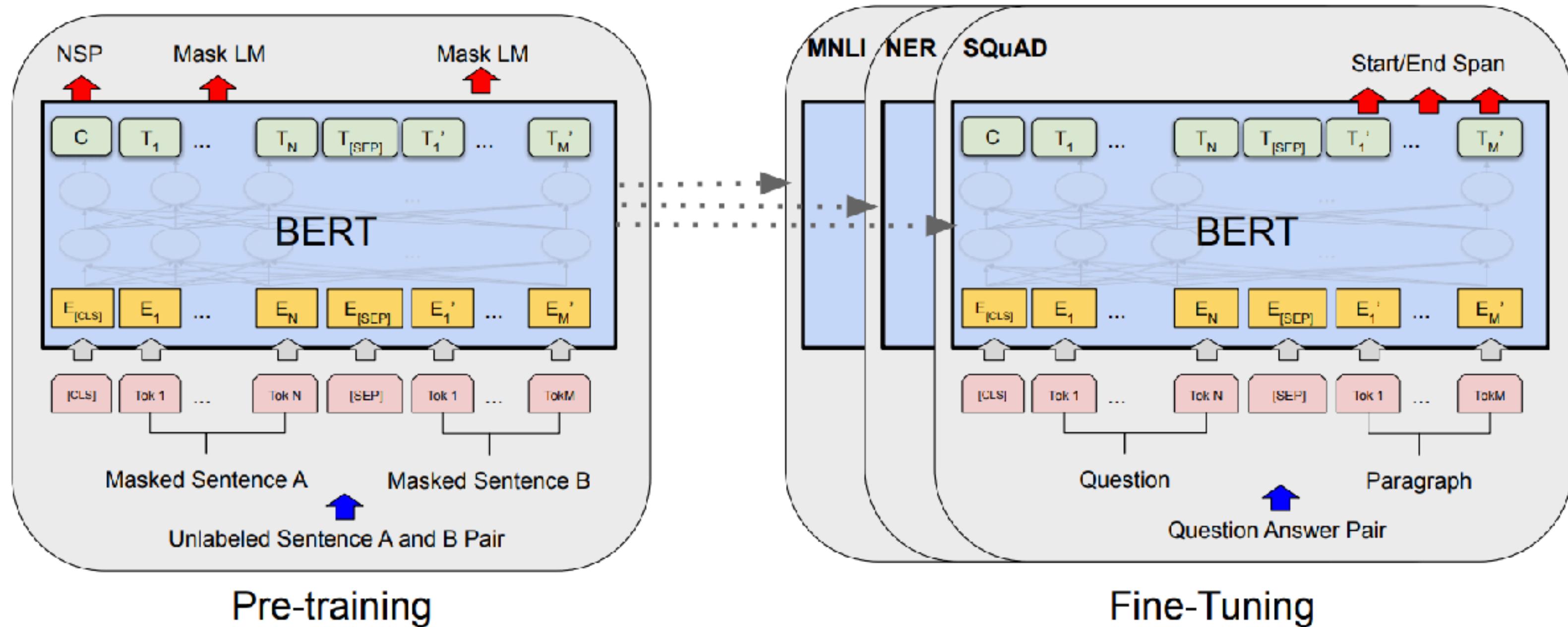
BERTの概要

- **BERT (Bidirectional Encoder Representation from Transformers) とは？**
 - 2018年の後半にGoogleから発表された、
自然言語処理のための新たなディープラーニングのモデル
 - Transformerがベースとなっている
 - 様々な自然言語処理タスクでファインチューニングが可能
 - 従来の自然言語処理タスクと比較して、高い汎用性

BERTの論文

- **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
→ <https://arxiv.org/abs/1810.04805>

BERTの学習



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin, J. et al. (2018) より引用

BERTの学習

- 事前学習
 - Transformerが、文章から文脈を双方向 (Bidirectional) に学習する
 - Masked Language ModelおよびNext Sentence Predictionによる双方向学習
- フайнチューニング
 - 事前学習により得られたパラメータを初期値として、ラベル付きのデータで
ファインチューニングを行う

Masked Language Model

- **Masked Language Model**
 - 文章から特定の単語を15%ランダムに選び、[MASK]トークンに置き換える
 - 例: my dog is hairy → my dog is [MASK]
 - [MASK]の単語を、前後の文脈から予測する

Next Sentence Prediction

- **Next Sentence Prediction**
 - 2つの文章に関係があるかどうかを判定する
 - 後ろの文章を50%の確率で無関係な文章に置き換える
 - 後ろの文章が意味的に適切であればIsNext、そうでなければNotNextの判定
- [CLS] the man went to [MASK] store [SEP] / he bought a gallon [MASK] milk [SEP]
判定 : IsNext

[CLS] the man went to [MASK] store [SEP] / penguin [MASK] are flight #less birds
[SEP]

判定 : NotNext

BERTの性能

- **SQuAD**
 - 「Stanford Question Answering Dataset」の略
 - スタンフォード大学が一般公開している言語処理の精度を測るベンチマーク
 - データは約10万個の質問応答のペアを含む

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
BERT _{BASE} (Single)	80.8	88.5	-	-
BERT _{LARGE} (Single)	84.1	90.9	-	-
BERT _{LARGE} (Ensemble)	85.8	91.8	-	-
BERT _{LARGE} (Sgl.+TriviaQA)	84.2	91.1	85.1	91.8
BERT _{LARGE} (Ens.+TriviaQA)	86.2	92.2	87.4	93.2

BERTの性能

- **GLUE**

→ 自然言語処理のための9種類の学習データを含むデータセット

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

日本語の訓練済みモデル

- 京都大学 黒橋・褚・村脇研究室
→ <http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT日本語Pretrainedモデル>

Google Colaboratoryの使い方



Google Colaboratoryとは？

- **Google Colaboratory**
 - Googleが提供する、ブラウザでPythonを実行できる環境
 - Googleアカウントで利用可能
 - 基本的に無料
 - 環境構築が簡単
 - 共有が簡単
 - etc...

<https://colab.research.google.com/>

コードセルとテキストセル

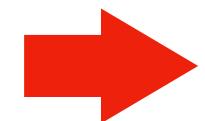
- コードセル
 - Pythonのコードを記述し、実行する
- テキストセル
 - 文章や数式を記述する

Google Colaboratoryの様々な機能

- スクラッチコードセル
- コードスニペット
- フォーム
- etc...

次回の内容

Section 1. 講座とBERTの概要



Section 2. シンプルなBERTの実装

Section 3. BERTの仕組み

Section 4. ファインチューニングの活用

Section 5. BERTの応用