

サクッとRNN・ LSTM実装講座

Ver. 1.0.0

自己紹介



システムアーキテクト
機械学習エンジニア

井上 秀樹

Study-AI登録講師/機械学習エンジニア

略歴

- Slerを中心にPM、インフラエンジニア、バックエンド・フロントエンドエンジニア、システムアーキテクト、ITコンサルタントを担当。
- 近年は、主にヘルスケア企業で機械学習エンジニアとして自然言語処理を利用した業務システムの企画・開発・運用を進めている。

目的

- フレームワークを使ってRNN/LSTMを構築できる
- 様々なフレームワークを使える
- 自然言語処理での応用方法を知る

目次

- ニューラルネットワークの基本
- RNNの概要
- LSTMの概要
- ハンズオン
- 自然言語処理での応用

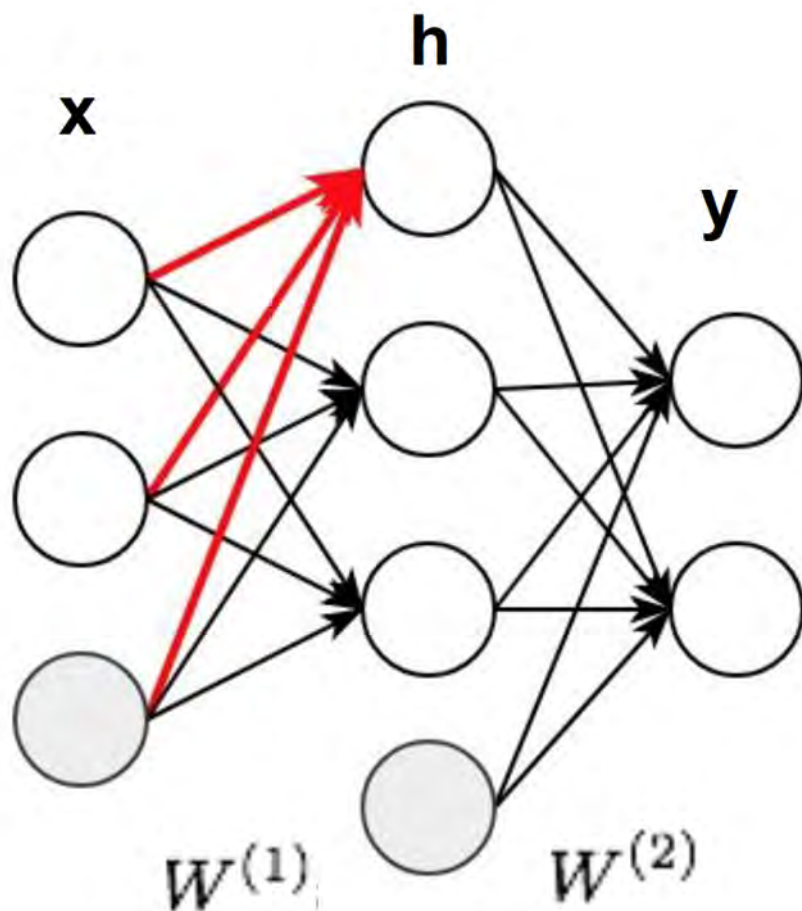
ニューラル ネットワーク の基本

ニューラルネットワーク

- 機械学習モデルの一種
 - 自動特徴量設計
- 様々なモデル
 - CNN(コンピュータビジョン)
 - RNN/LSTM(自然言語処理、時系列予測)
 - GAN(生成モデル)

順伝播型ニューラルネットワーク

- 順伝播



$$h^{\ell} = f^{\ell}(W^{\ell}h^{\ell-1} + b^{\ell})$$

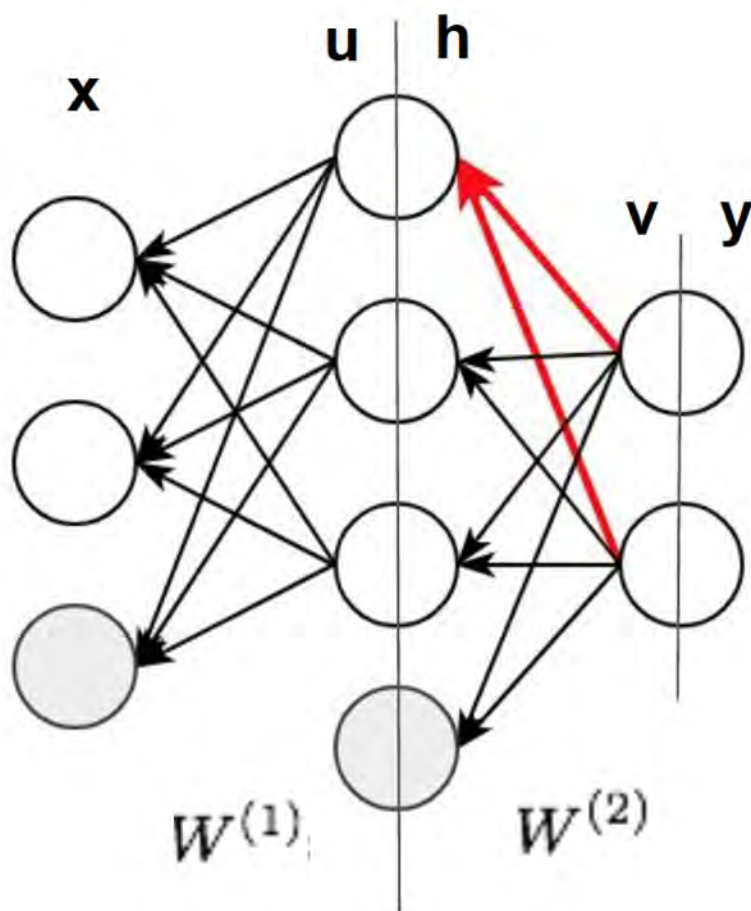


$$h = f(W^{(1)}x + b^{(1)})$$

$$y = g(W^{(2)}h + b^{(2)})$$

順伝播型ニューラルネットワーク

- 逆伝播(誤差逆伝搬法)



勾配降下法

$$\theta \leftarrow \theta - \nabla_{\theta} L$$

順伝播の別記法

$$u = W^{(1)}x + b^{(1)}$$

$$h = f(u)$$

$$v = W^{(2)}h + b^{(2)}$$

$$y = g(v)$$

誤差の勾配

$$\delta^{(v)} = \frac{\partial L}{\partial v} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial v} = \frac{\partial L}{\partial y} g'(v)$$

$$\delta^{(u)} = \frac{\partial L}{\partial u} = \frac{\partial L}{\partial v} \frac{\partial v}{\partial h} \frac{\partial h}{\partial u} = (W^{(2)})^T \delta^{(v)} f'(u)$$

順伝播型ニューラルネットワーク

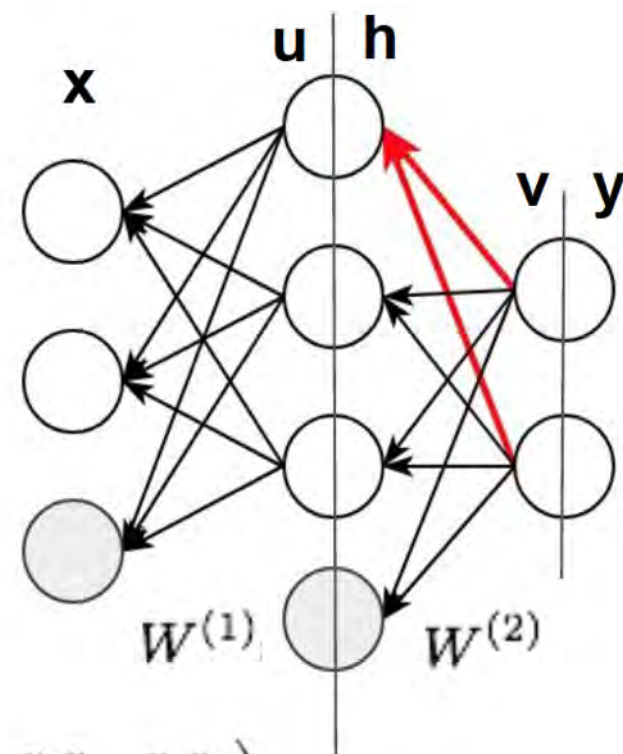
- 各パラメータについての勾配計算

$$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial v} \frac{\partial v}{\partial W^{(2)}} = \delta^{(v)} \otimes h$$

$$\frac{\partial L}{\partial b^{(2)}} = \frac{\partial L}{\partial v} \frac{\partial v}{\partial b^{(2)}} = \delta^{(v)}$$

$$\frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial u} \frac{\partial u}{\partial W^{(1)}} = \delta^{(u)} \otimes x$$

$$\frac{\partial L}{\partial b^{(1)}} = \frac{\partial L}{\partial u} \frac{\partial u}{\partial b^{(1)}} = \delta^{(u)}$$



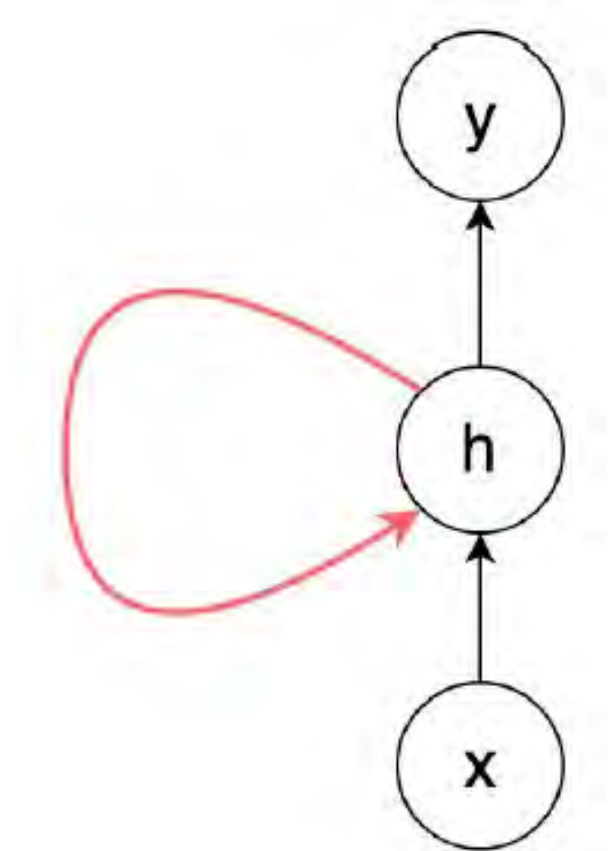
テンソル積(直積)

$$u \otimes v = uv^{\top} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} (v_1 \quad v_2 \quad v_3) = \begin{pmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \\ u_4 v_1 & u_4 v_2 & u_4 v_3 \end{pmatrix}.$$

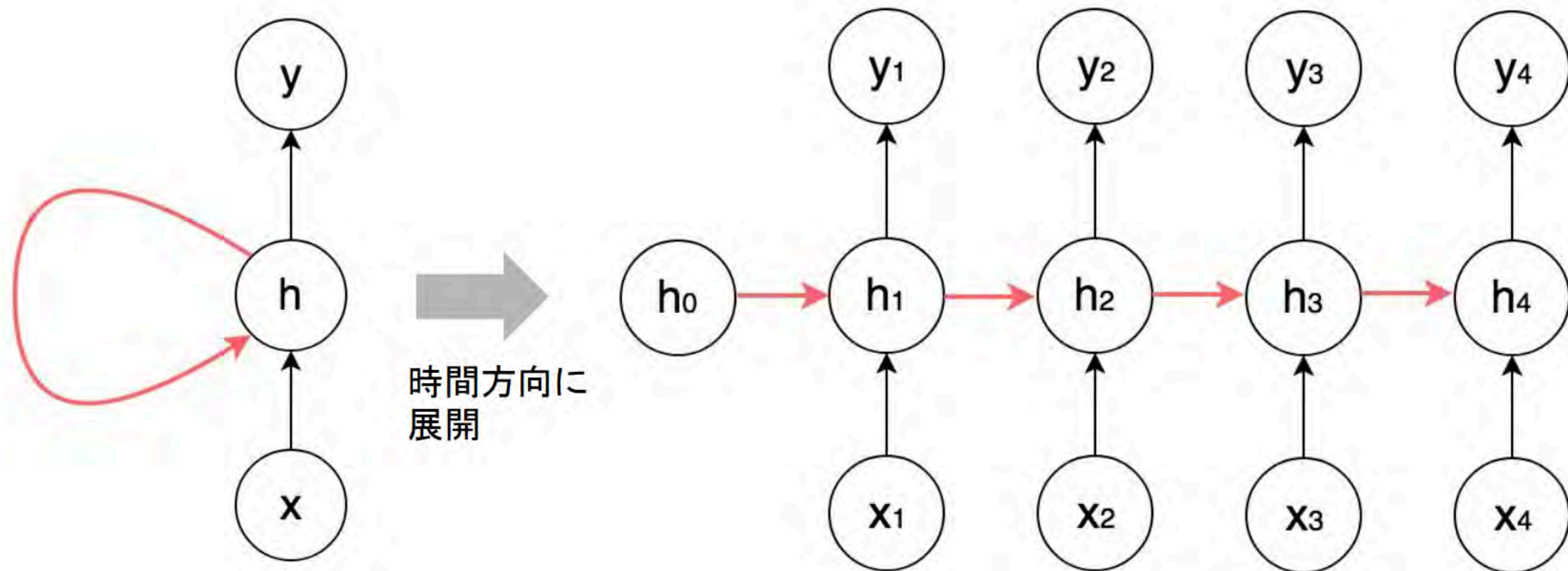
RNNの概要

RNN(Recurrent Neural Network)とは

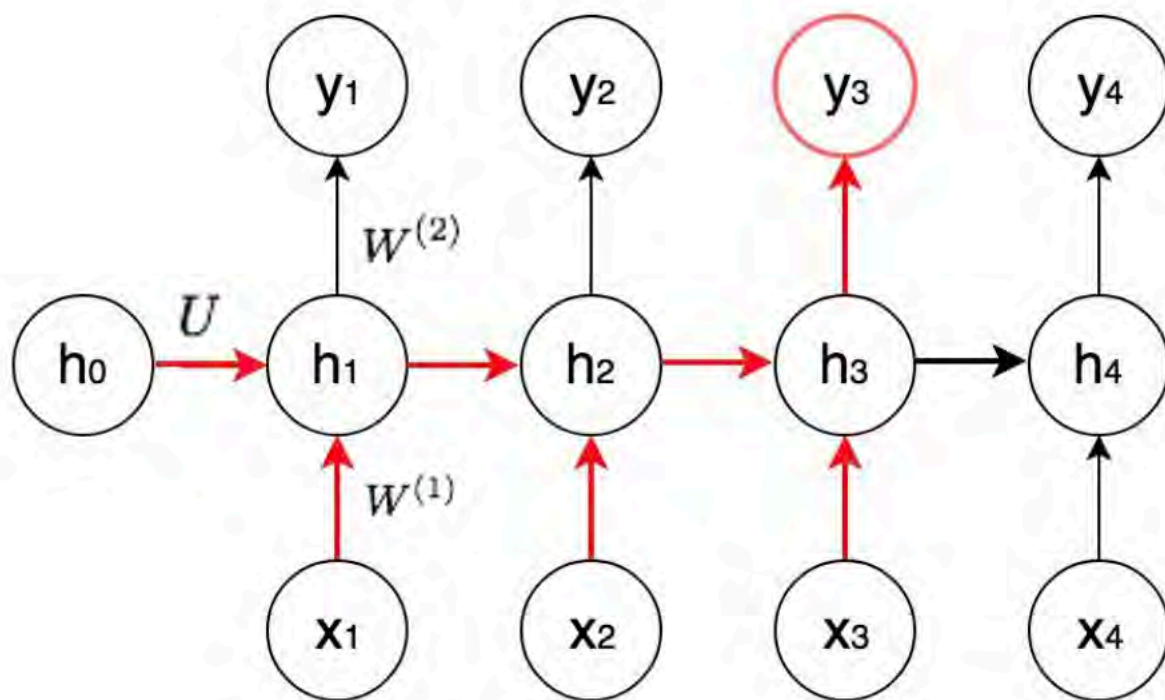
- 過去の隠れ層の状態も入力に
- 系列データを扱うのに適したニューラルネットワーク
- 過去の情報を保持
- 可変長入力
 - 例) 「I like sports.」 「I got up early this morning.」



RNNユニット



RNNの順伝播



$$h_t = f(W^{(1)}x_t + \underline{Uh_{t-1}} + b^{(1)})$$
$$y_t = g(W^{(2)}h_t + b^{(2)})$$

RNNの誤差逆伝播

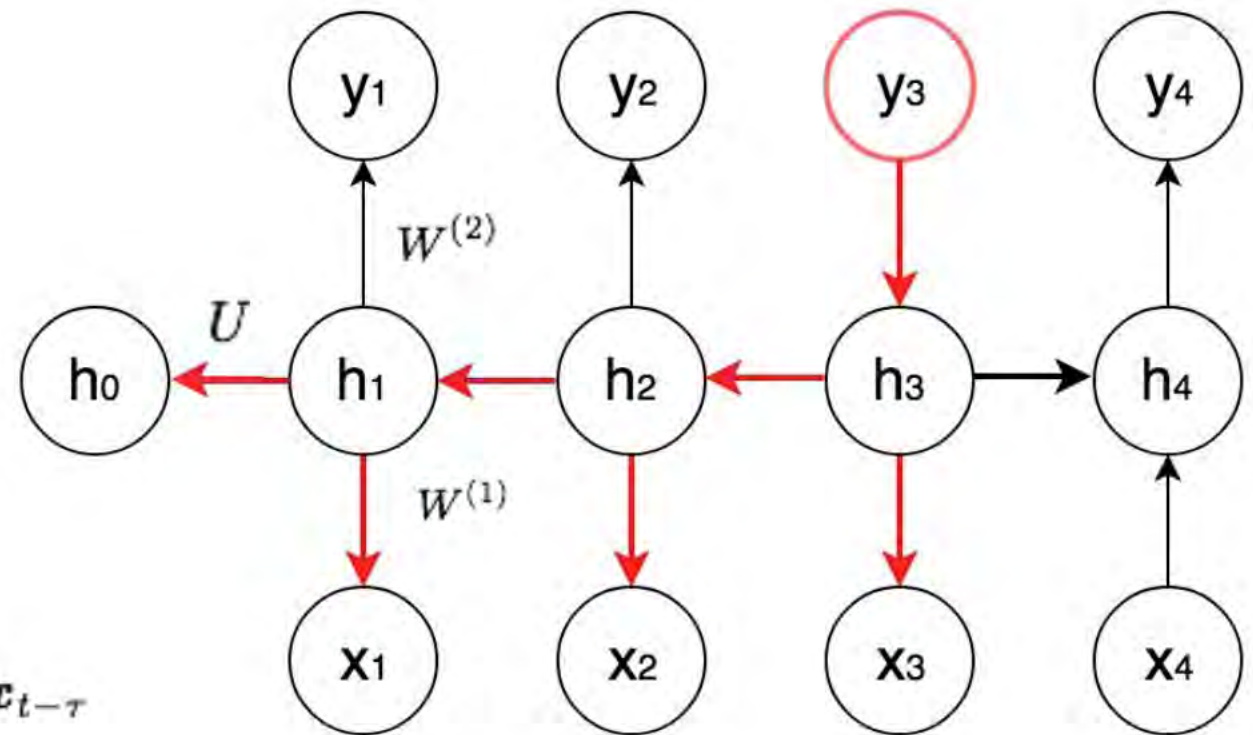
- BPTT(Back Propagation Through Time)
 - 時間展開した上での誤差逆伝播
- Truncated BPTT
 - 遡るステップ数を限定したBPTT

誤差 $L = \sum_{t=1}^T L_t(\mathbf{y}_t, \mathbf{t}_t)$

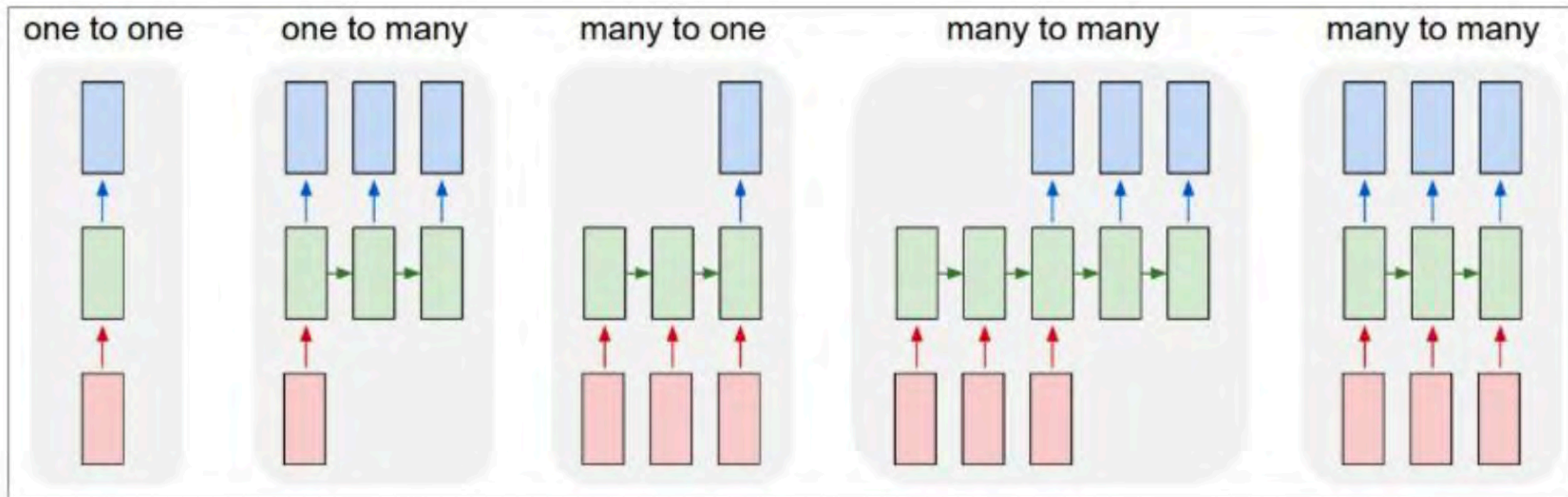
例)

$$\frac{\partial L_t}{\partial W^{(2)}} = \frac{\partial L_t}{\partial \mathbf{v}_t} \frac{\partial \mathbf{v}_t}{\partial W^{(2)}} = \delta_t^{(v)} \otimes \mathbf{h}_t$$

$$\frac{\partial L_t}{\partial W^{(1)}} = \sum_{\tau=0}^{t-1} \frac{\partial L_t}{\partial \mathbf{u}_{t-\tau}} \frac{\partial \mathbf{u}_{t-\tau}}{\partial W^{(1)}_{(t-\tau)}} = \sum_{\tau=0}^{t-1} \delta_{t-\tau}^{(u)} \otimes \mathbf{x}_{t-\tau}$$



RNNモデルの応用例



画像認識

画像キャプション
生成

感情分析

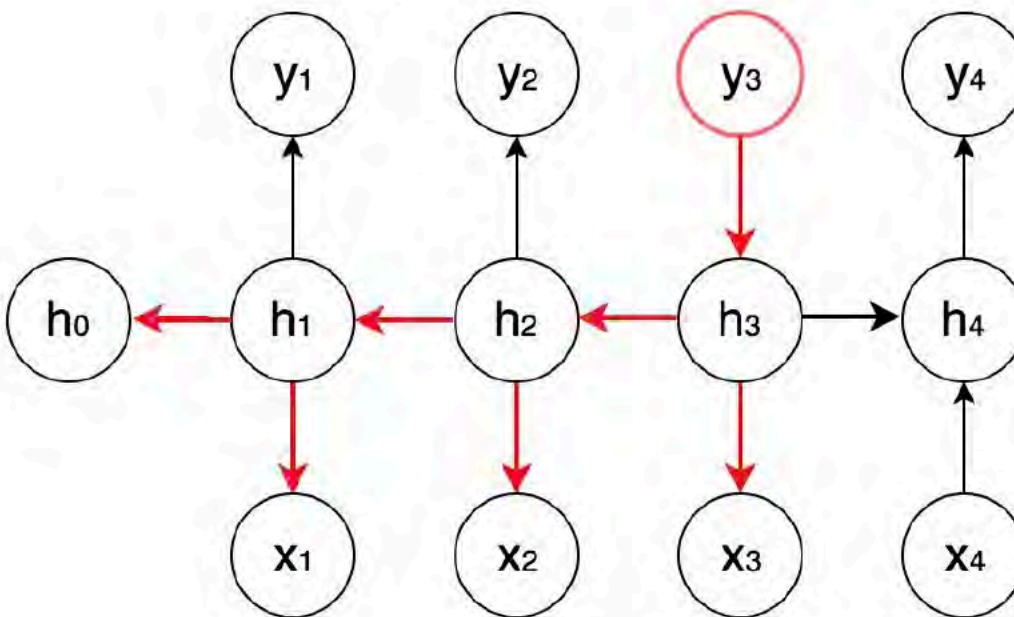
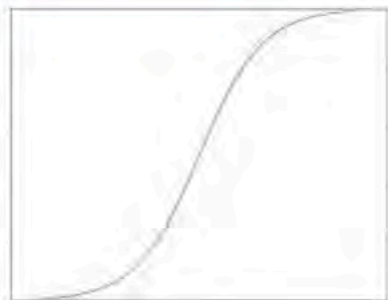
機械翻訳

動画分類

RNNの問題

- 長期的な依存関係を学習できない。
- 勾配消失(層を遡るごとに指数関数的に勾配が小さくなる)

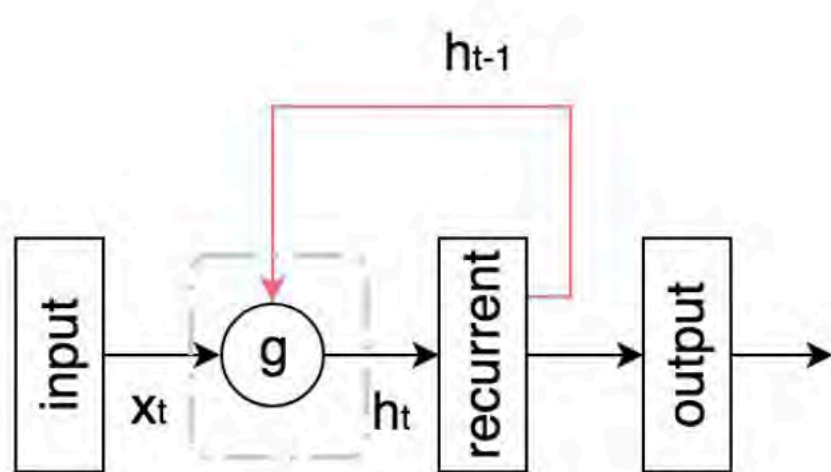
$$\begin{aligned}\delta_{t-1}^{(u)} &= \frac{\partial L_t}{\partial \mathbf{u}_{t-1}} = \frac{\partial L_t}{\partial \mathbf{u}_t} \frac{\partial \mathbf{u}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{u}_{t-1}} \\ &= U^T \delta_t^{(u)} \underline{f'(\mathbf{h}_{t-1})}\end{aligned}$$



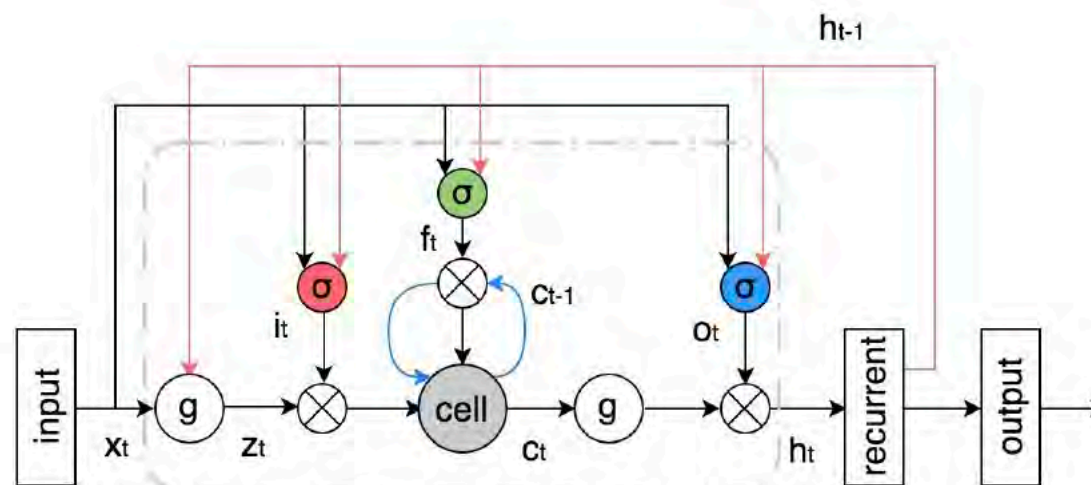
LSTMの概要

LSTM(Long Short-Term Memory)とは

- RNNの派生モデル
 - 長期的な依存関係を学習可能(勾配消失を避ける)
 - セルとゲートを導入



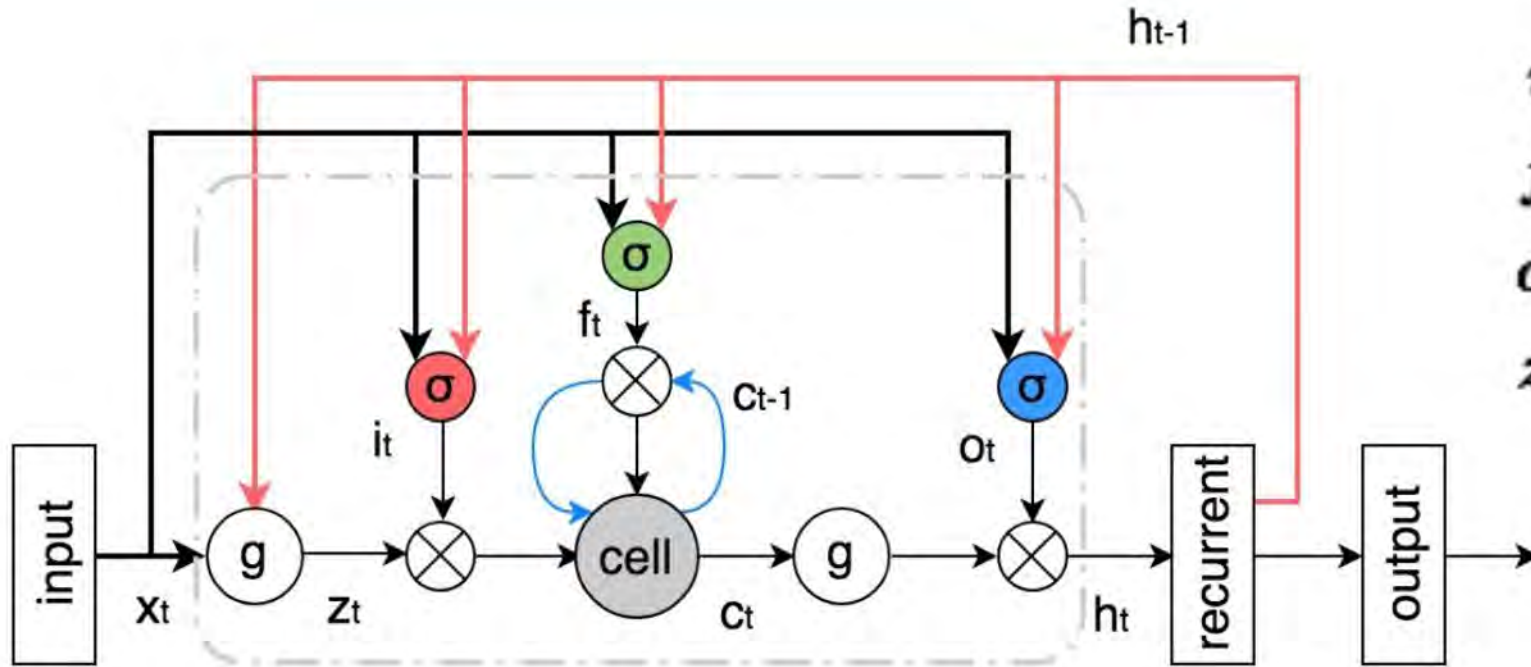
RNN



LSTM

LSTMの順伝播

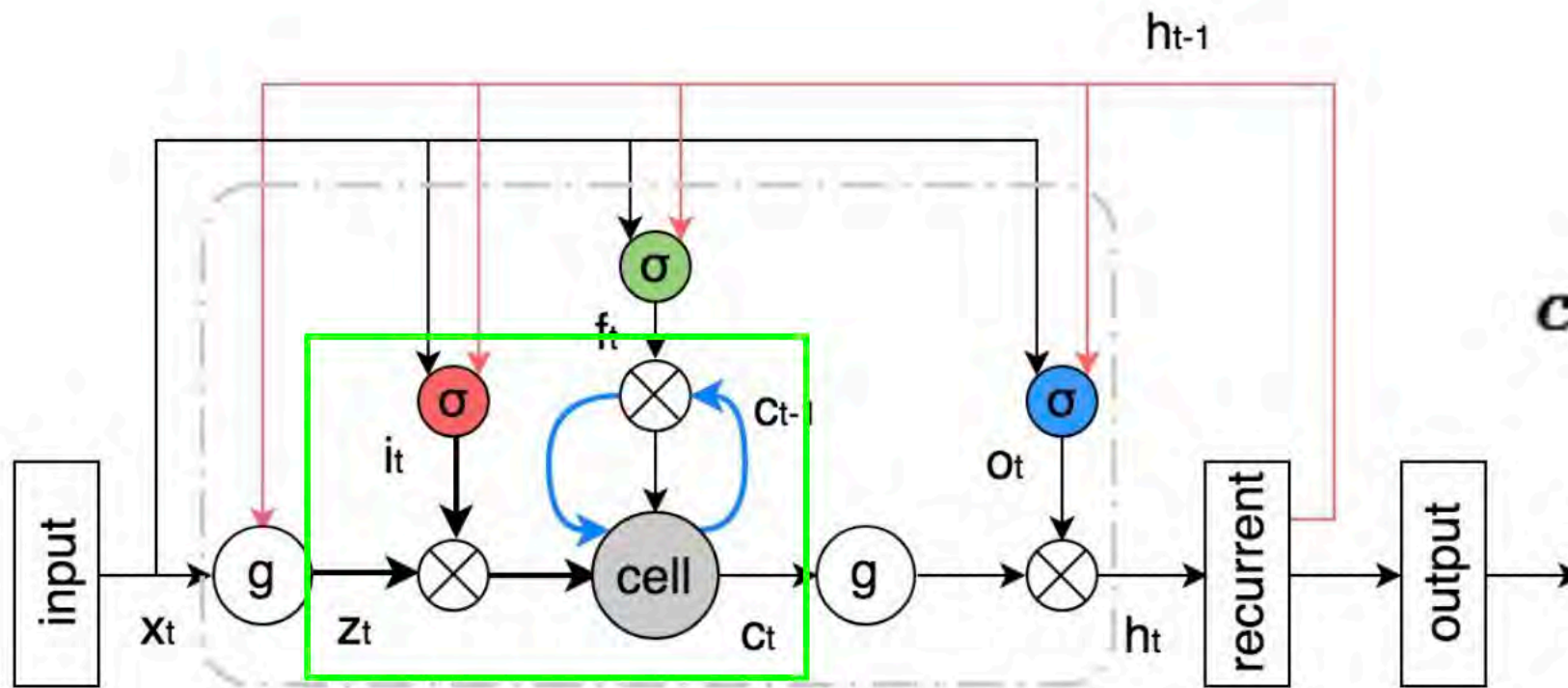
- ゲート(入力・忘却・出力)とセルへの入力の計算



$$\begin{aligned} i_t &= \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) \\ f_t &= \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) \\ o_t &= \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) \\ z_t &= g(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1} + \mathbf{b}_z) \end{aligned}$$

LSTMの順伝播

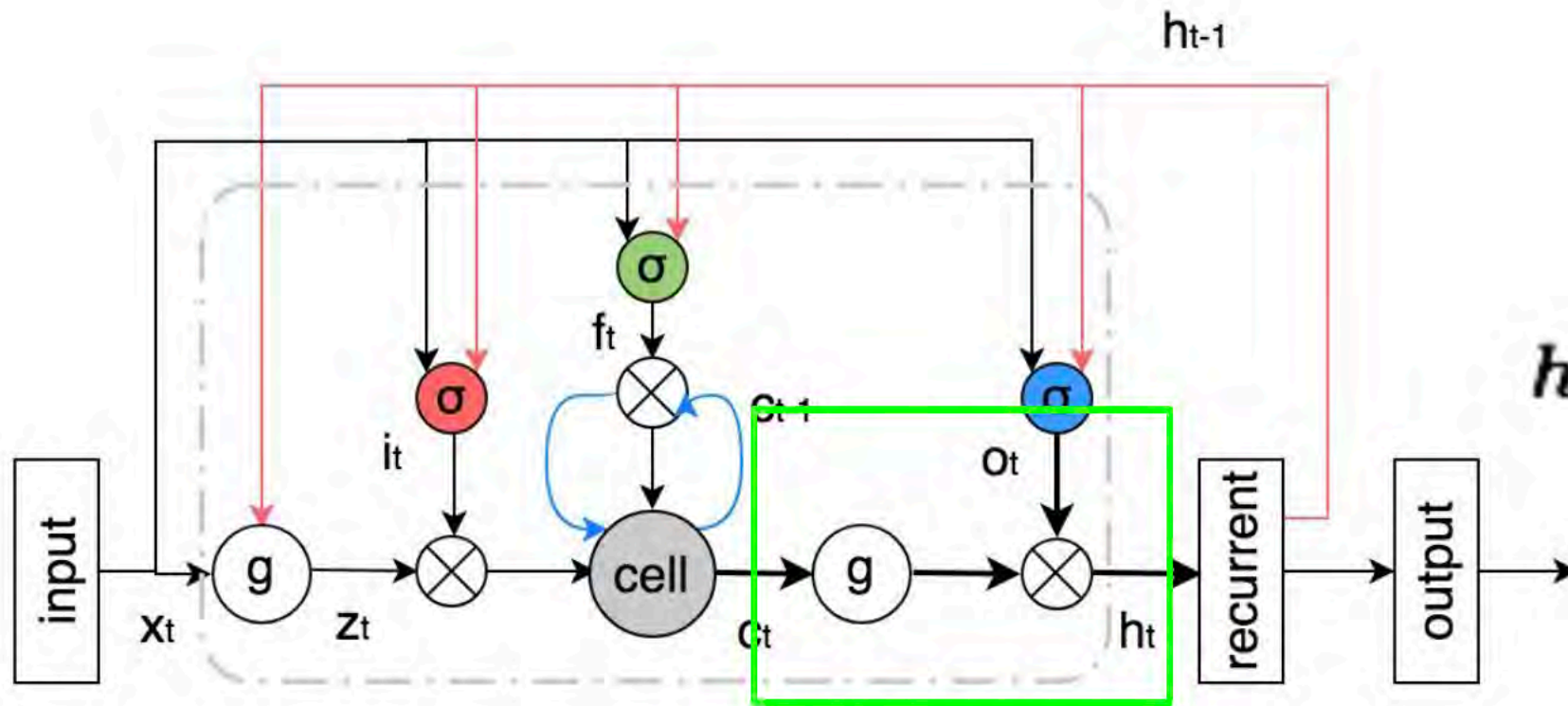
- セルの値を計算



$$c_t = f_t \odot c_{t-1} + i_t \odot z_t$$

LSTMの順伝播

- 出力値を計算

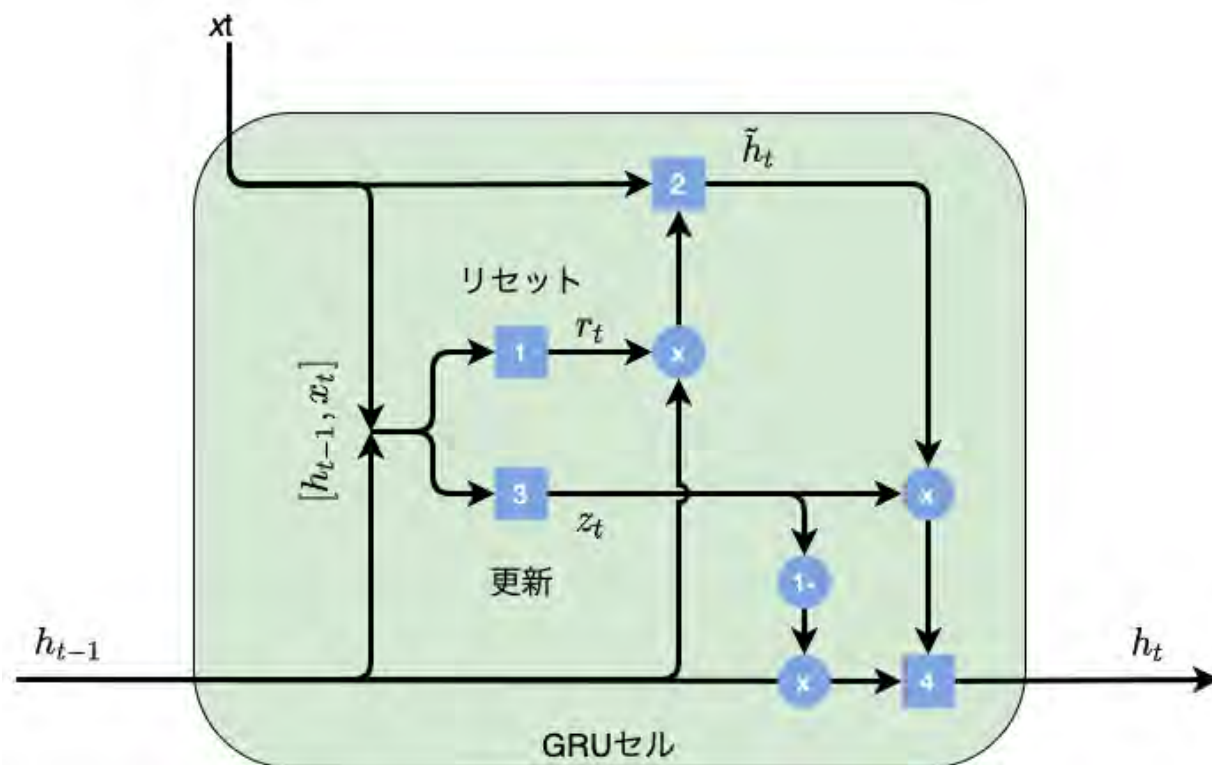


LSTMの誤差逆伝播

- RNNと同じくBPTT
- RNNに比べてパラメータ数が多い(4倍)
- 勾配爆発を避けるために勾配クリッピングを使う事がある
 - 大きい勾配に対してノルムを閾値以下に正規化

GRU(Gated Recurrent Unit)

- RNNの派生モデル
 - LSTMの計算コストを抑えたもの



- 1 $r_t = \sigma(W_r) \cdot [h_{t-1}, x_t]$
- 2 $\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$
- 3 $z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$
- 4 $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$

まとめ

- RNN
 - 系列データを扱うのに適したニューラルネットワーク
- LSTM
 - RNNの派生
 - 長期的な依存関係を学習できる
 - 単純なRNNではなくLSTMやGRUを利用

ハンズオン

RNN with Keras

RNN with PyTorch

為替の予想プログラム実装例

自然言語処理 での応用

自然言語処理の概要

人の話す言葉をコンピュータに学習・理解させ、人の役に立つようにコンピュータに処理させること。

要素技術

単語分割

構文解析

意味解析

辞書構築

文書分類

自動要約

情報抽出

著者推定

機械翻訳

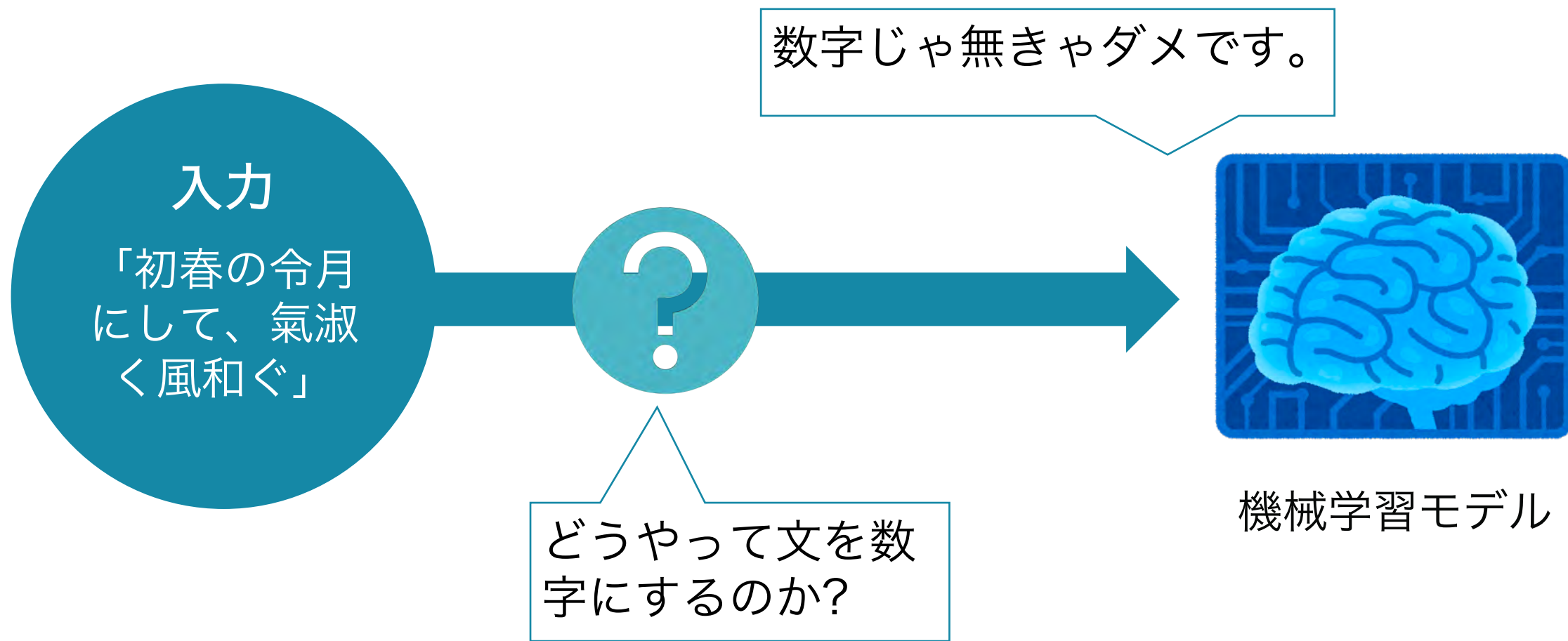
音声認識

情報検索

質問応答

複合技術

どうやって自然言語を入力するのか？



単語の意味は周辺の単語で決まる？

私は __ をよく食べる

脂っこい __ が好きだ

私は __ 屋の常連だ

今日は __ を食べたい

“__”に当てはまる単語を考える。

自然言語を数字にする方法

- カウントベース
 - TF-IDF、共起行列など
- 推論ベース
 - Word2Vec、GloVe、BERTなど

単語ベクトルの例

- spaCy(word2vec)で取得した単語ベクトルの例。

	token	vector
0	私	[-0.13697676, -0.23937745, 0.045566633, -0.20059128, -0.08979624, 0.11869049, -0.03709601, -0.14...
1	は	[-0.05035316, -0.15731327, -0.08336552, -0.15989235, -0.12370043, -0.0015842685, -0.015121695, -...
2	焼肉	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
3	を	[-0.19509408, -0.13202968, -0.018012488, -0.12985665, -0.11748332, 0.16473995, -0.08152997, -0.0...
4	よく	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
5	食べる	[-0.13252193, -0.40151066, 0.04001627, -0.08204308, -0.013501916, 0.23513791, 0.2380793, -0.3093...



300次元

単語ベクトルは意味を定量化している

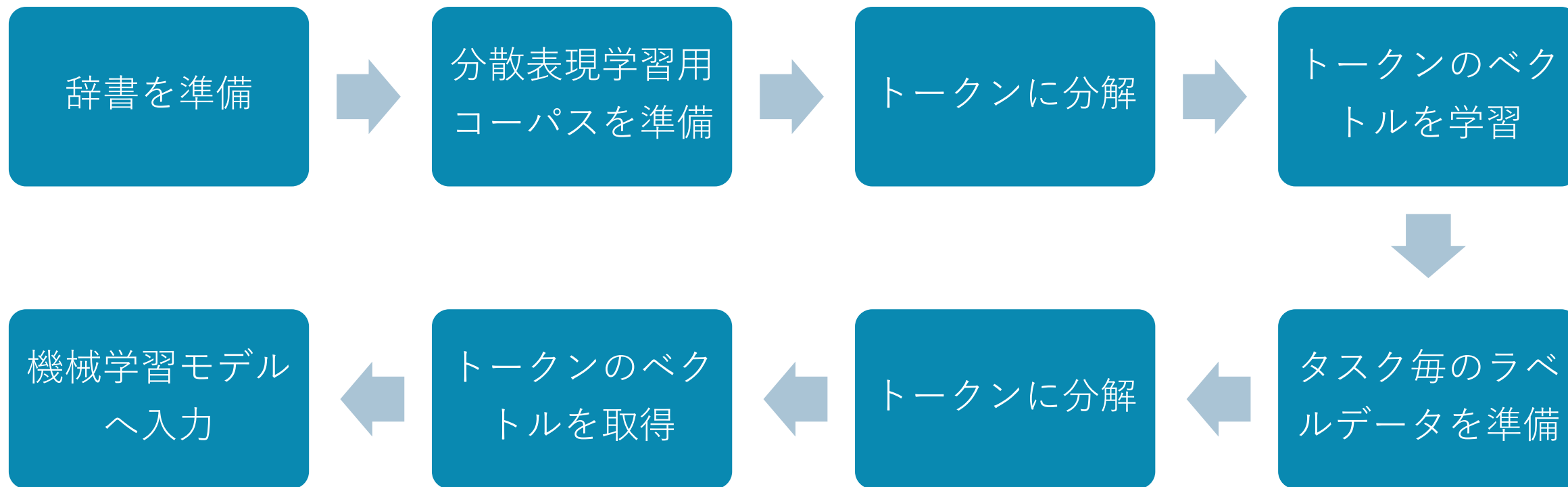
“テスト”に近い単語は？

Word ◆	Similarity ◆
試験	0.830351
デモンストレーション	0.713327
トライアル	0.689451
トレーニング	0.686056
Acid3	0.653619
シェイクダウン	0.646347
追試	0.641687
宇宙遊泳	0.634437
フライト	0.633194
プレゼンテーション	0.629233

“薬”に近い文字は？

Char ◆	Similarity ◆
剂	0.912732
毒	0.731842
医	0.710113
療	0.689038
液	0.676660
胃	0.670220
廠	0.655366
尿	0.651356
染	0.648082
肝	0.645307

分散表現を利用した自然言語処理の流れ



ハンズオン

spaCy

spaCyはExplosion AI 社の開発するオープンソースの自然言語処理ライブラリ。

リクルート社の研究機関が国立国語研究所と共同で「GiNZA」を公開。日本語Wikipediaに加え、国立国語研究所が保守しているコーパスで学習したモデルを実装することで日本語でも広く利用が可能になった。

京都大学情報学研究科解析済みブログコーパス

京都大学情報学研究科とNTTコミュニケーション科学基礎研究所の共同研究ユニットが纏めたブログに関するコーパス。

4テーマ(京都観光、携帯電話、スポーツ、グルメ)、249記事、4,186文の解析済みブログコーパス。形態素、構文、格・省略・照応、評判情報がアノテーションされている。

LSTMによる分類タスク

LSTMによる自然言語生成

今どきの自然言語処理

- Attention
- Transformer
- BERT
- GPT-3
- ELECTRA