自然言語処理
—RNN —

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

自然言語処理
RNN

解説動画

100本ノック第9章とRNNの位置づけ

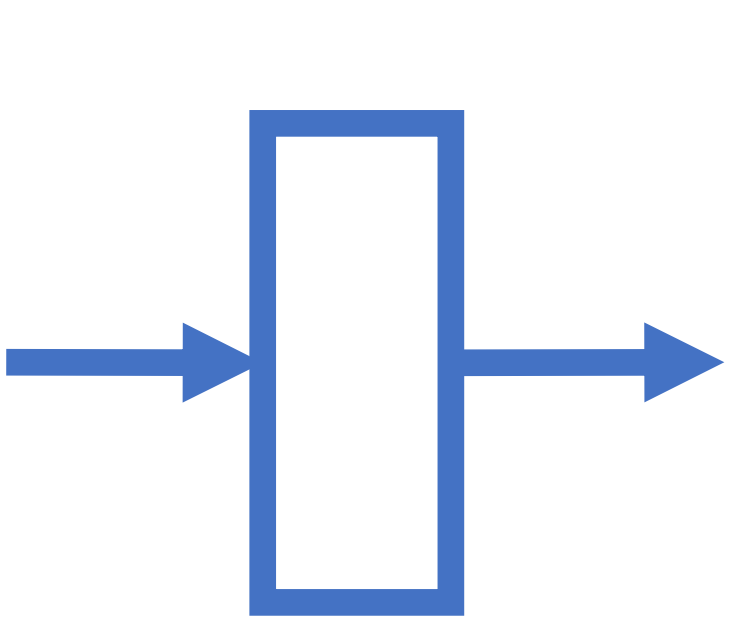
- １００本ノック課題集第9章は、RNN、CNN、Transformerを扱っている。
- RNNの扱いが大きいが、RNN、そしてその改善版のゲート付きRNNは、時系列データ（自然言語は単語の時系列）の基幹技術だった。
- しかし、音声認識（信号の時系列）は別として、文章テキストデータの処理に関しては、その位置をTransformerに譲ってしまった。

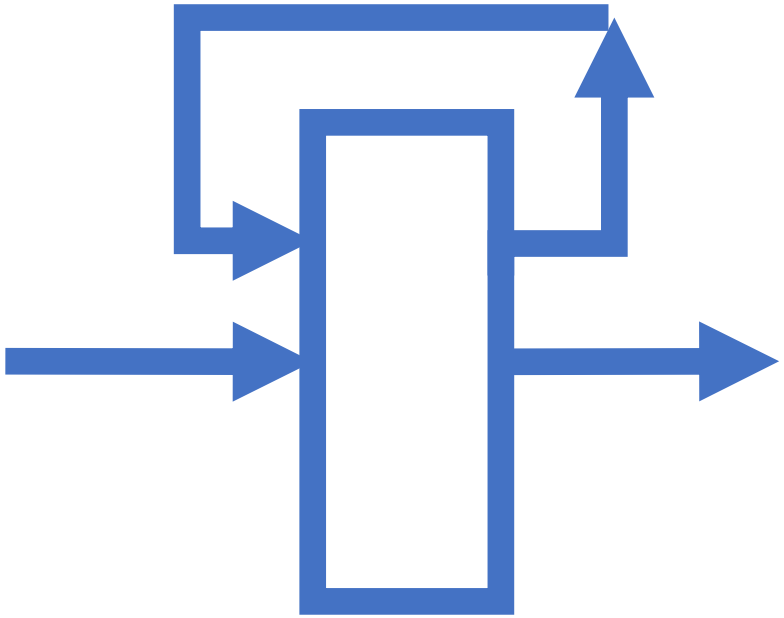
項目と系列

- 年代区別：子供、大人、老人
- 訪問場所：東京、富士山、京都
- コマンドの種類：ls,cd,pwd
- 音素　ji,n,koo,ti,noo
- 文字　人,工,知,能
- 単語　人工,知能,は
- …

- 年齢変化：子供->大人->老人
- 訪問順：東京->富士山->京都
- スクリプト、プログラム: pwd->cd->ls
- 音素列　ji->n->koo->ti->nou
- 文字列　人->工->知->能
- 単語列　人工->知能->は
- …

Feed Forward（前向き推論）とRecurrent（再帰推論）





Feedforward	Recurrent
固定長ベクトル	可変長のベクトル
ベクトル、空間情報	順序情報
内部状態を持たない関数	内部状態を持つオブジェクト

RNNの応用

言語モデル（文字レベル、単語レベル）

テキスト分類

テキスト生成

音声認識

機械翻訳

株価予測

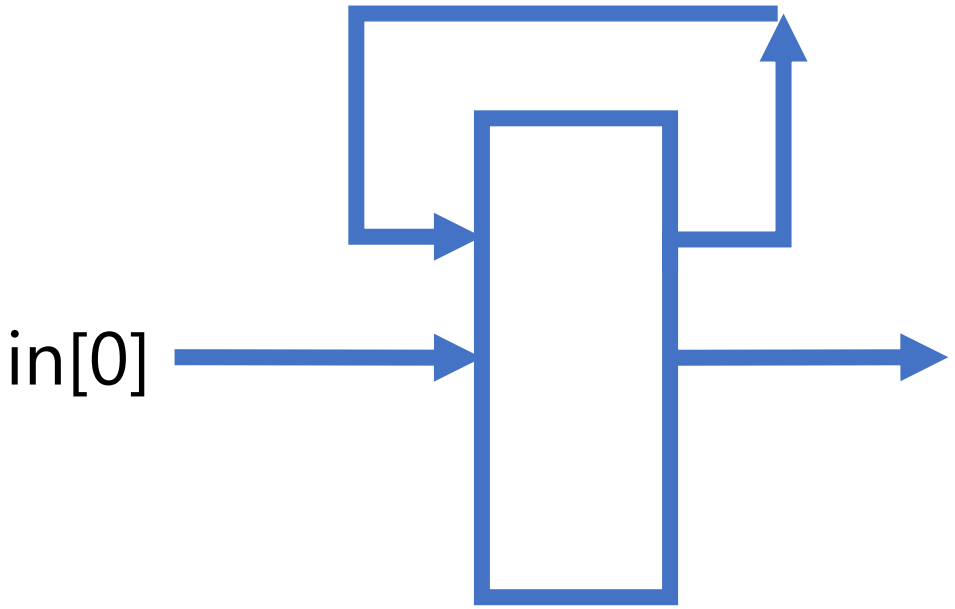
ロボット制御

自動応答

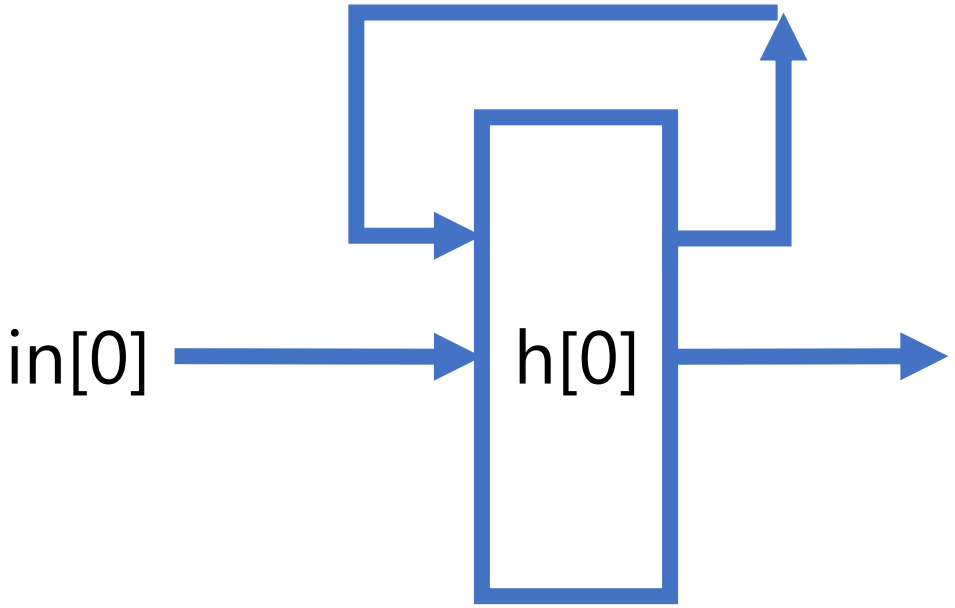
チャットボット

動画解析（自動キャプション）

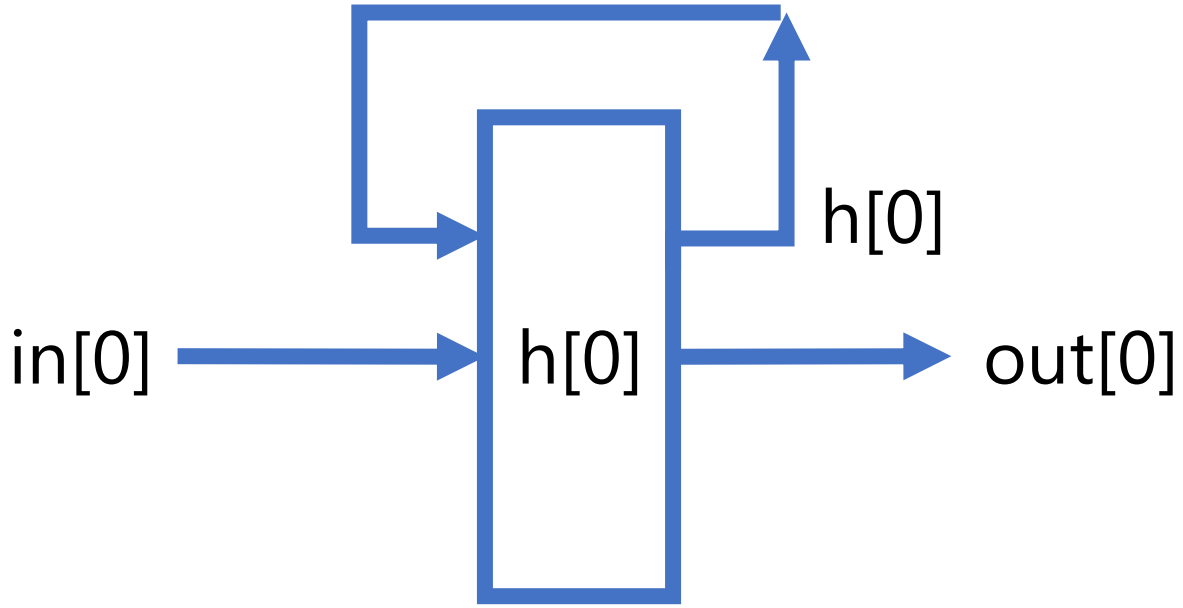
RNNの動作



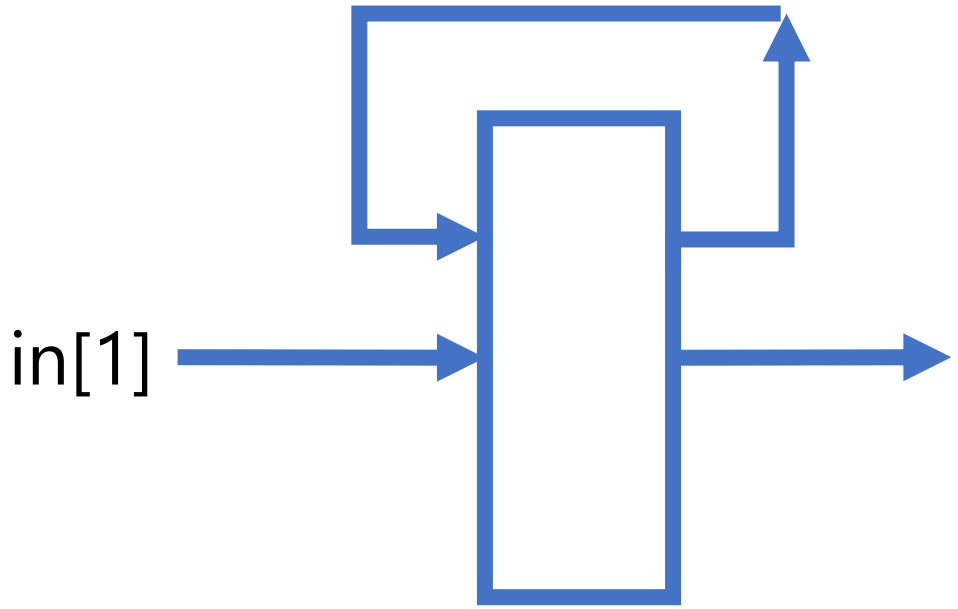
RNNの動作



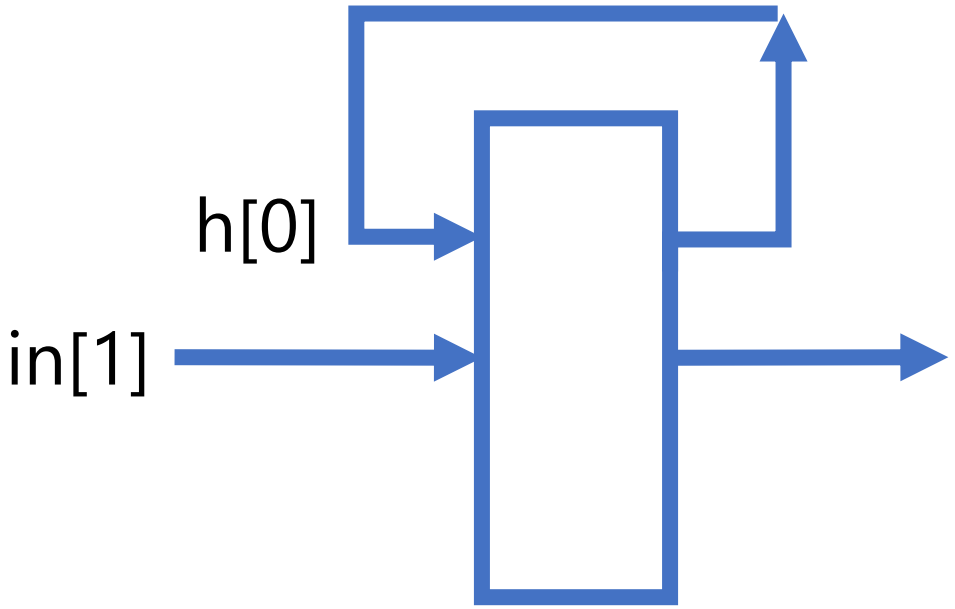
RNNの動作



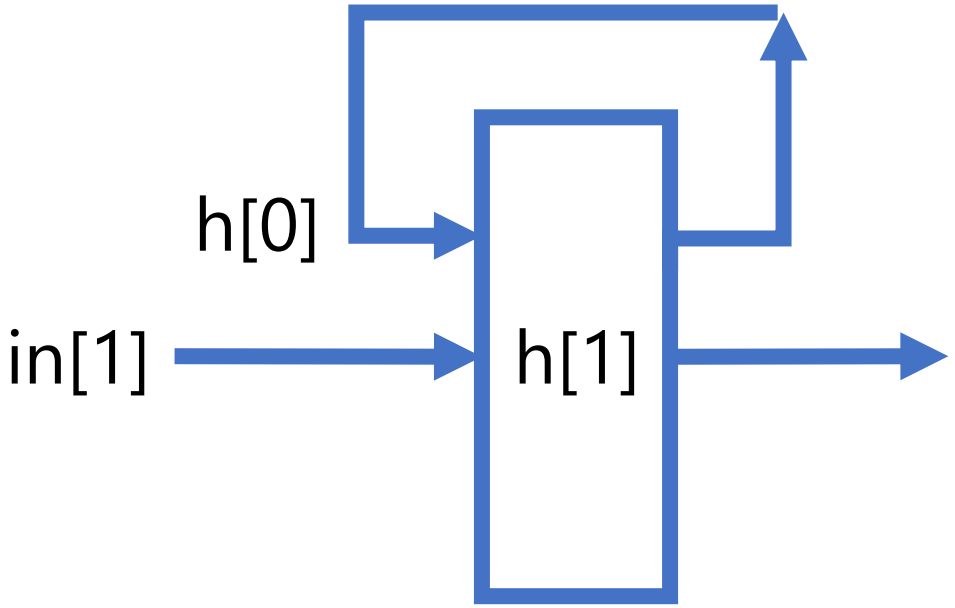
RNNの動作



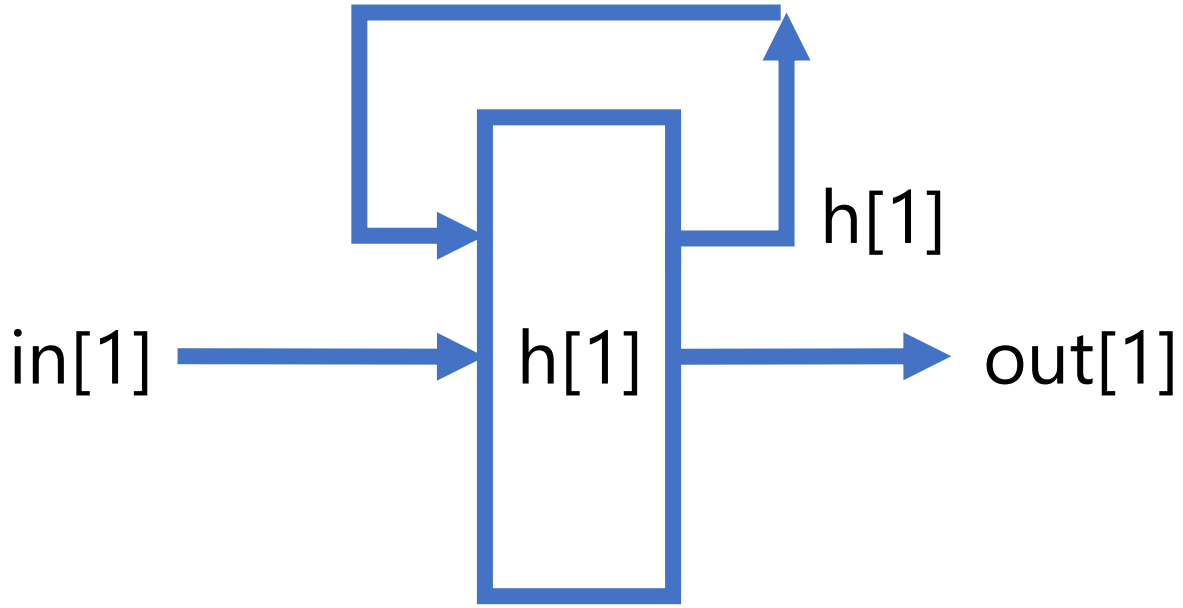
RNNの動作



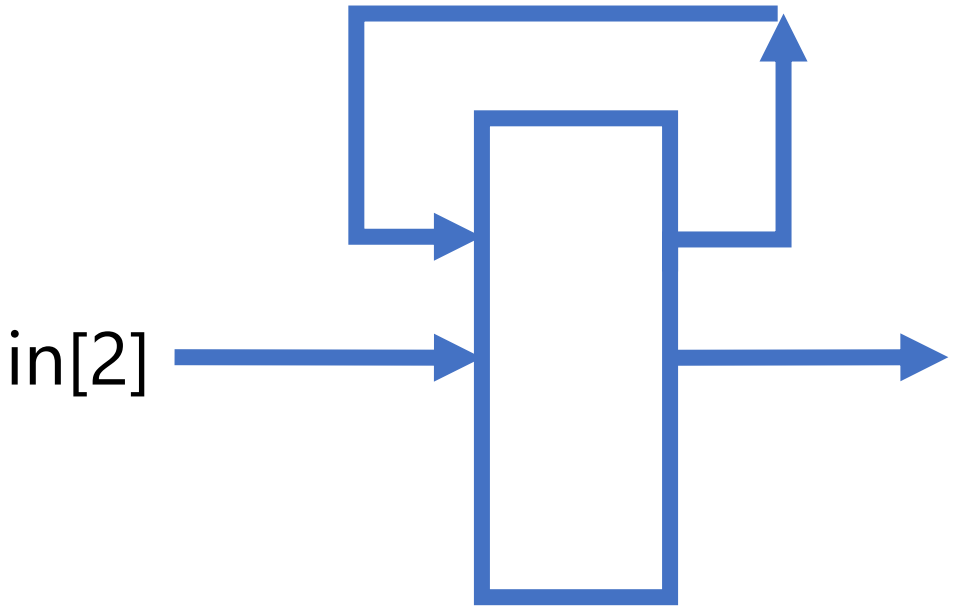
RNNの動作



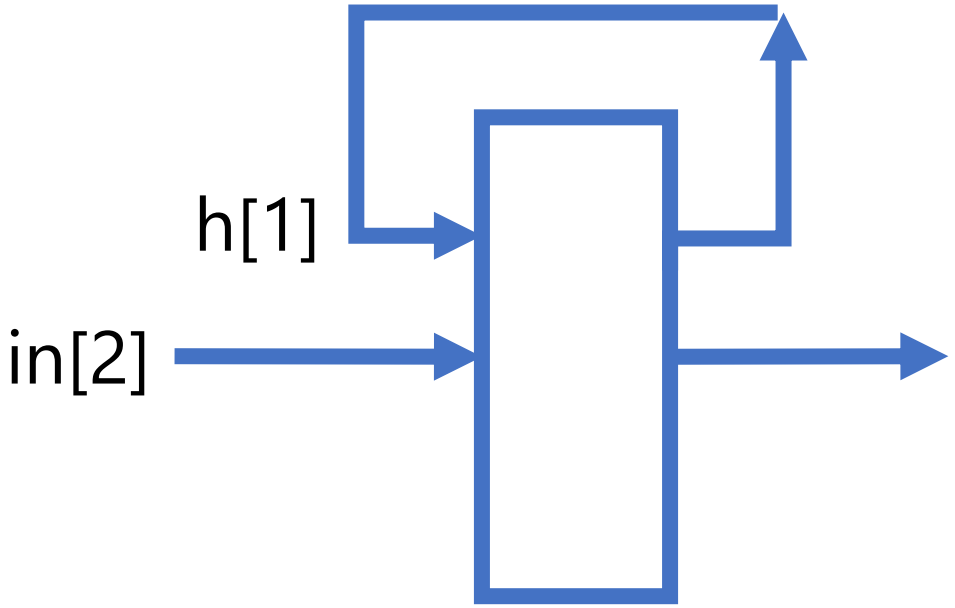
RNNの動作



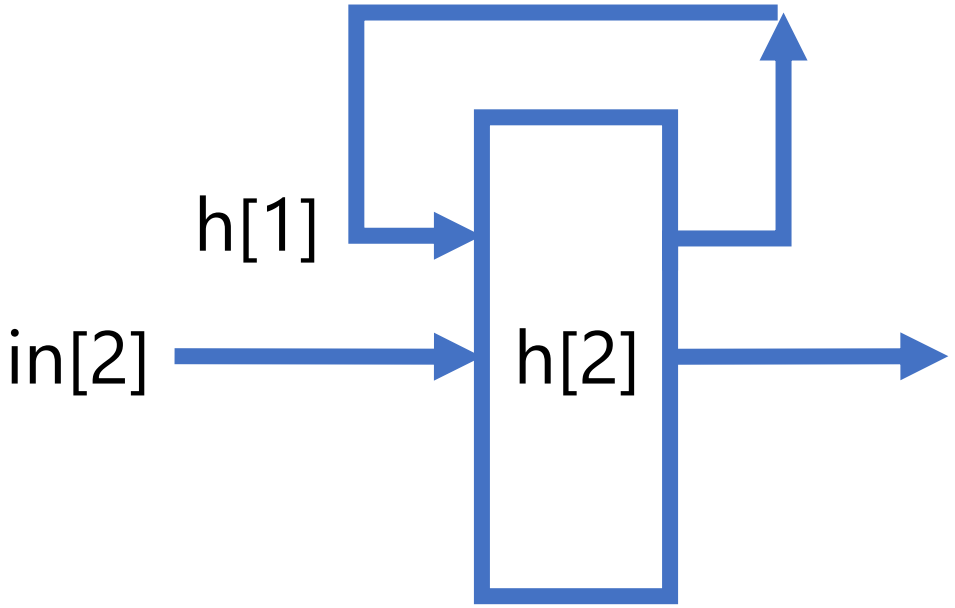
RNNの動作



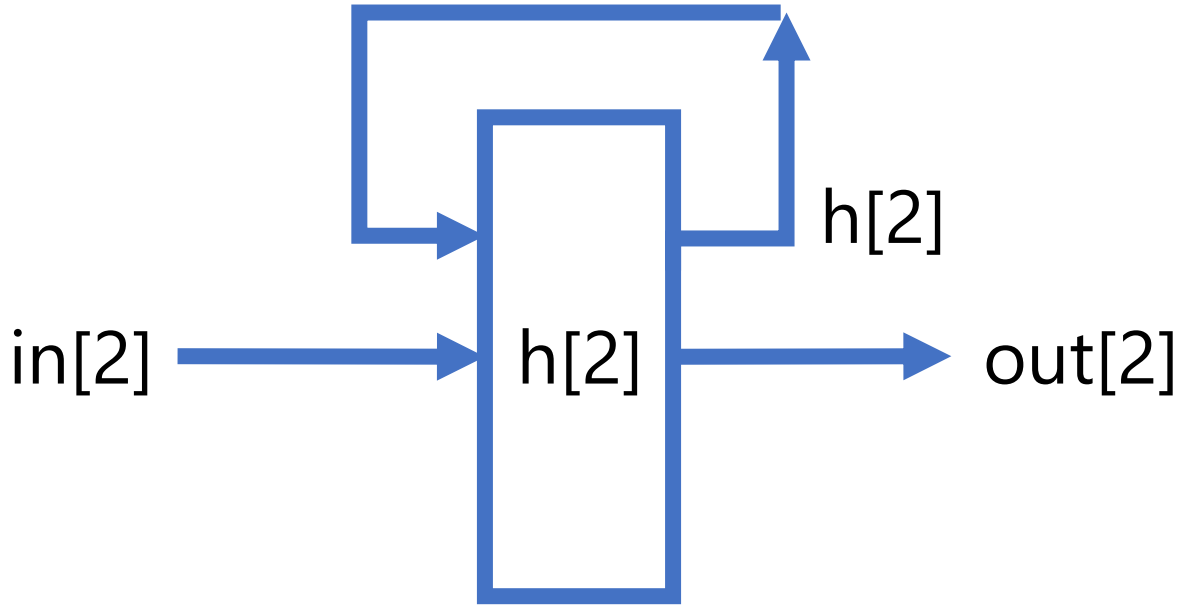
RNNの動作



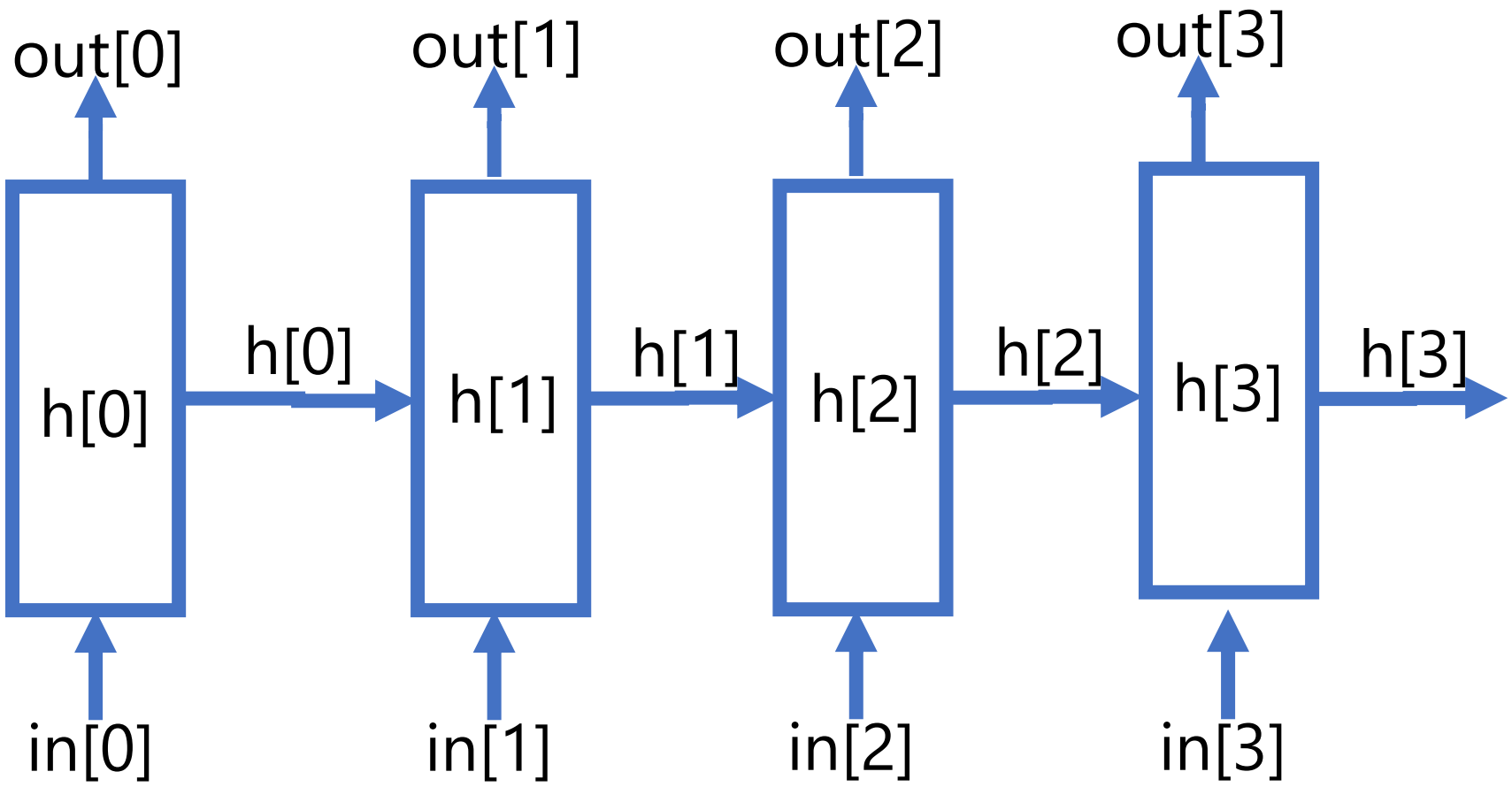
RNNの動作



RNNの動作



並べると（箱は同一のもの）

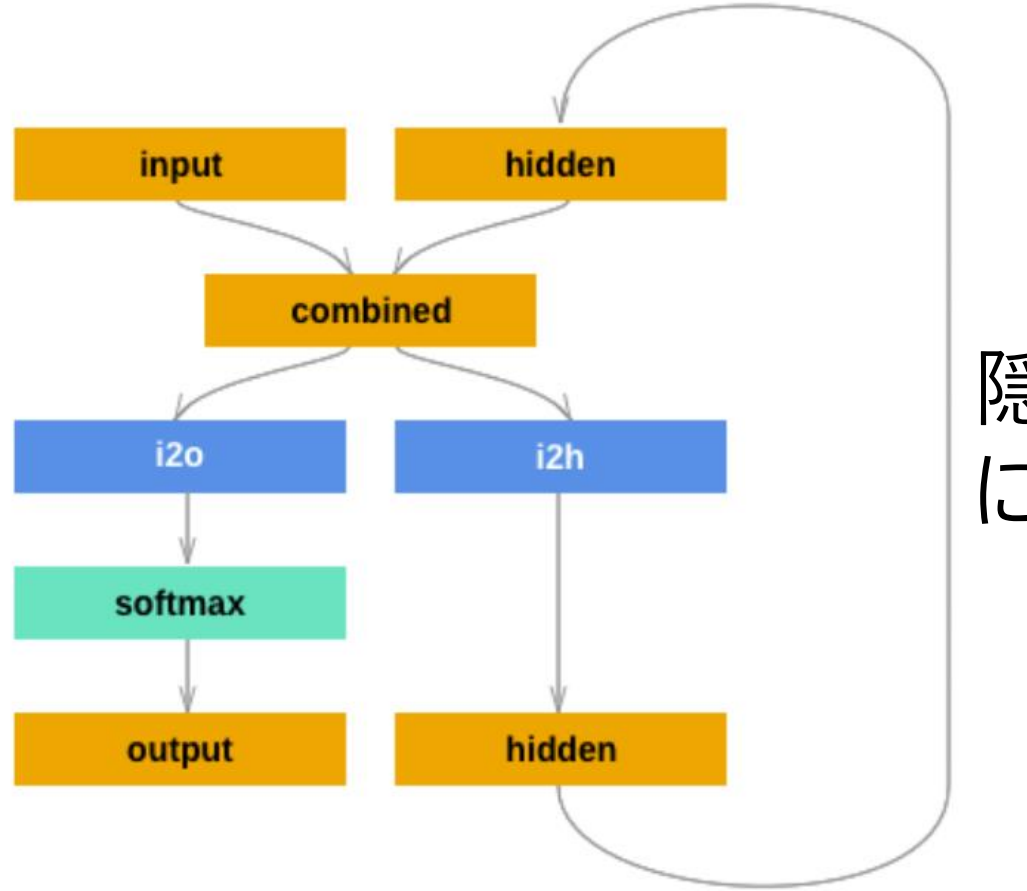


RNN課題１

- char_rnn.ipynb に、文字列を読んでクラス判定するサンプル（最後の出力からクラスに所属する確率をとる）と名前文字列を生成するサンプル（RNNの各時刻の出力を順番に取り出して並べる）があります。
- PyTorchサイトにあるチュートリアルのサンプルをさらに単純化し、日本語データを扱うように変更したものです。
- 実用的には、PyTorchにrnnクラスがすでにあるのでそれを使うだけでよいのですが、これらサンプルは教育目的で、rnnクラスを使わず、rnnのロジックを生で書き、原理を説明しています。
- PythonやPyTorchに慣れるのにいいサンプルです。読解しましょう。

サンプル：文字レベルRNNで名前分類

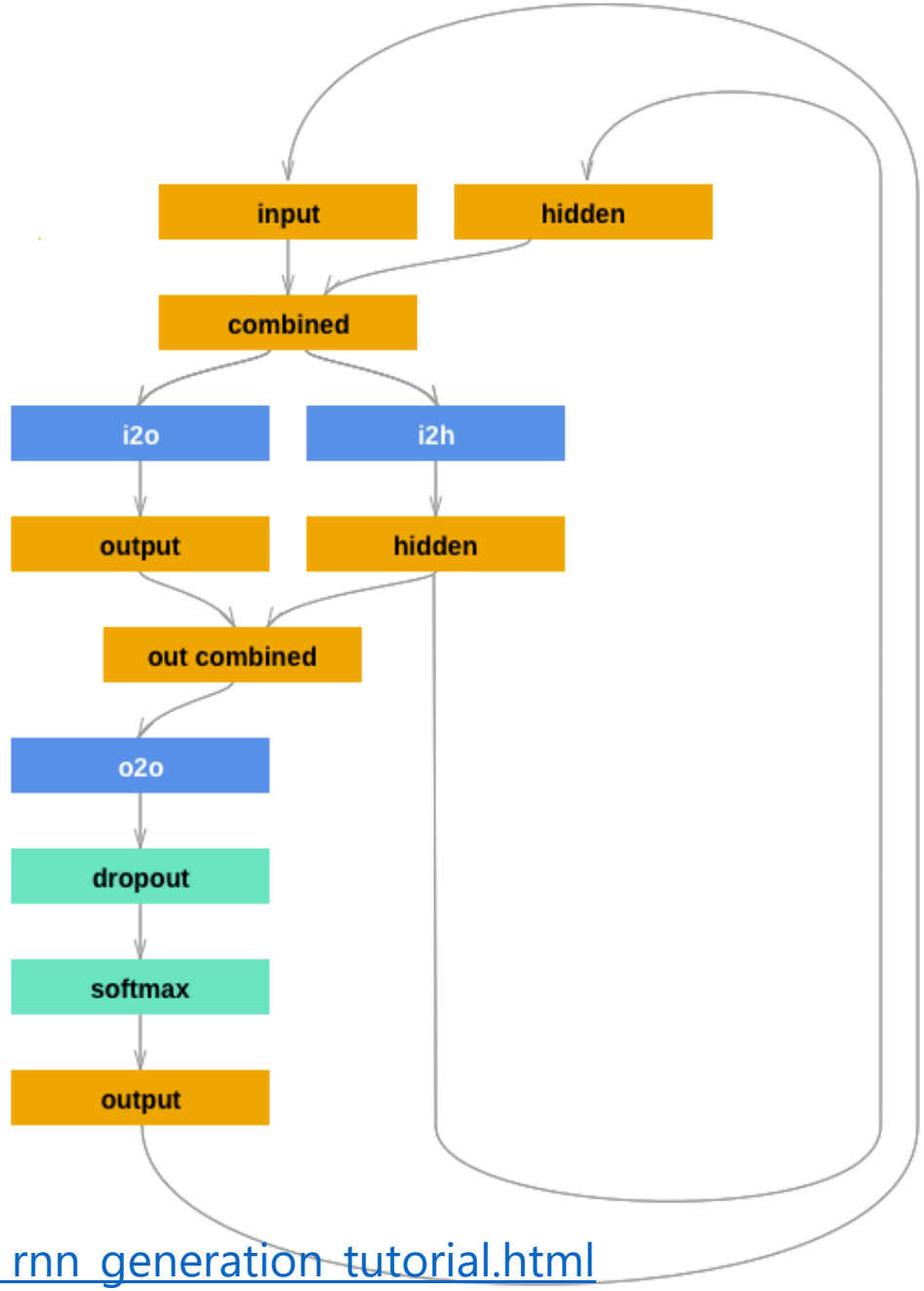
最後の出力は、入力系列の圧縮表現となる



隠れ状態を再帰的に入力へ

https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html

サンプル：文字レベルRNNで名前生成



出力を順次取り出すと、系列が生成できる

生成問題ではある時刻の出力を次の時刻の入力とする

https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html

参考資料

- ゼロから作るDeep Learning ② 第5章
- The Unreasonable Effectiveness of Recurrent Neural Networks
- 教育動画
  - Deep Learning入門：Recurrent Neural Networksとは？
  - RNN：時系列データを扱うRecurrent Neural Networksとは

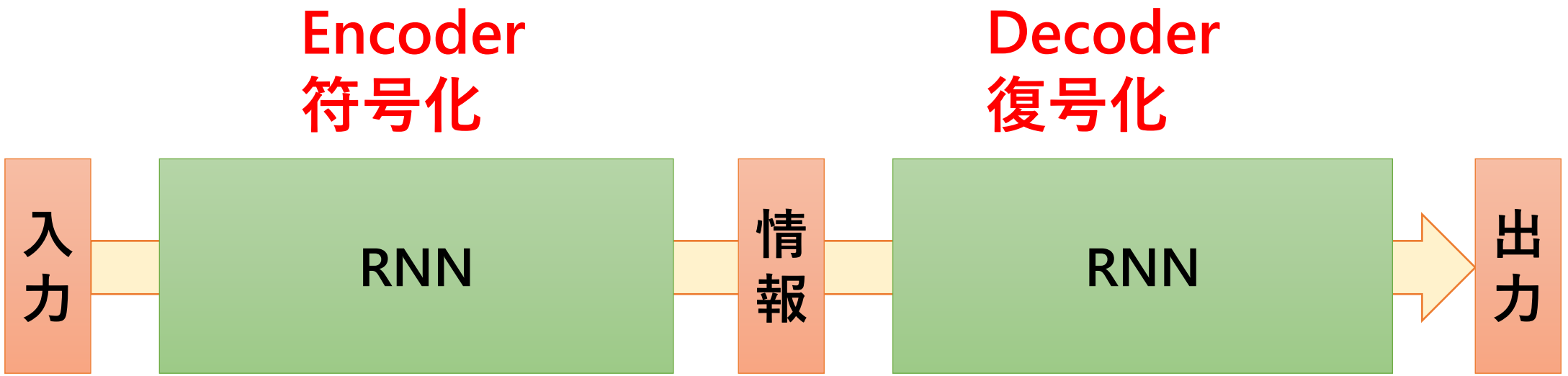
自然言語処理
Seq2Seq（Encoder-Decoder）

解説動画

RNNの二つの顔

- 最後の隠れ状態が系列データの情報集約（符号化）
- ある時刻の出力を次の時刻の入力とすれば、系列生成（復号化）

Seq 2 Seq



英文
質問文
…

和文
回答文
…

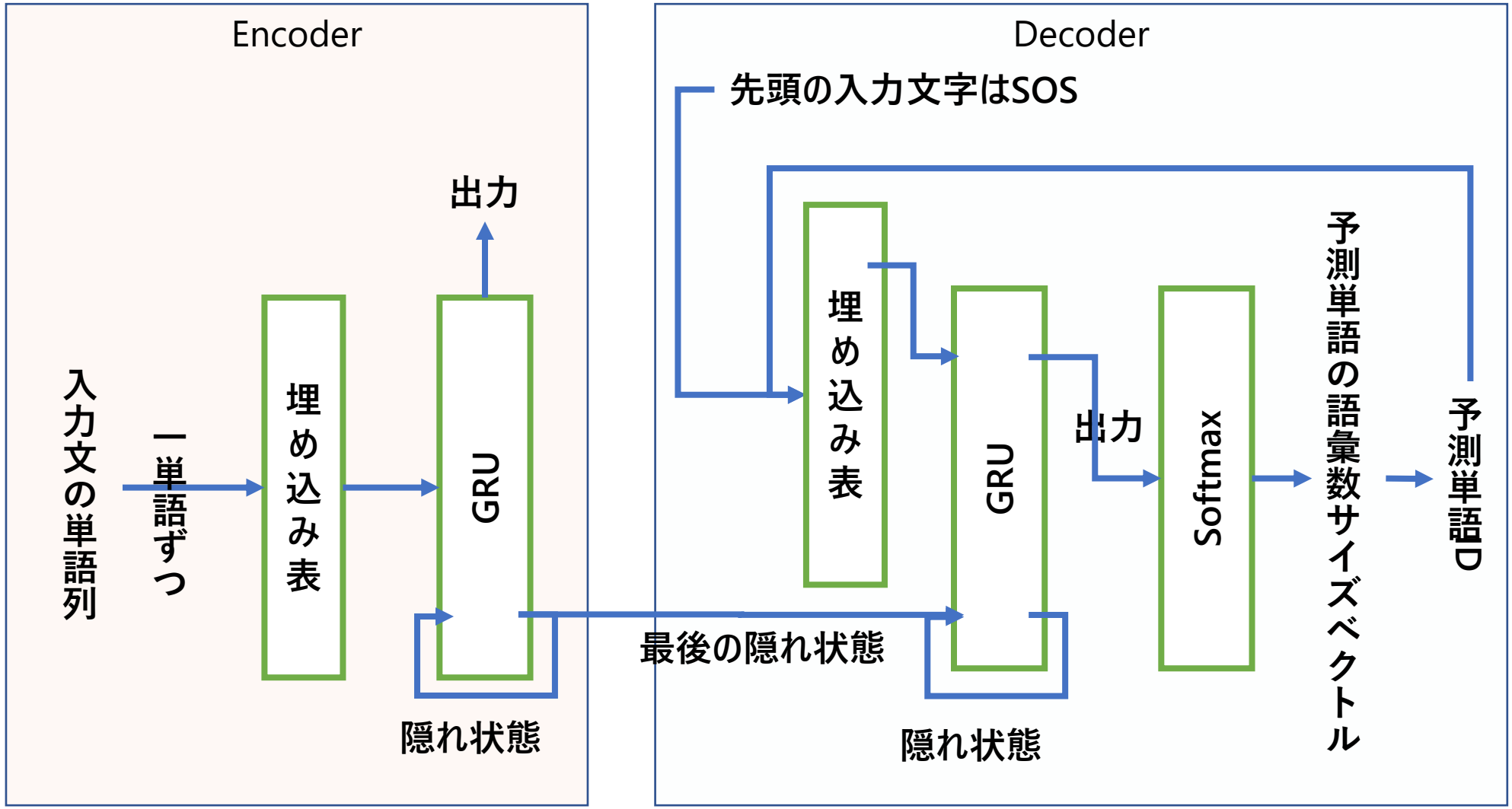
Seq2Seq あるいは Encoder-Decoderの応用

- 機械翻訳：　ソース言語->ターゲット言語
- 音声認識：　音響特徴列->単語列
- 要約：　記事->要約
- Ｑ＆Ａ：　質問文->回答文
- チャット：　つぶやき->返答つぶやき
- 映像自動キャプション：　画像フレーム列->説明文

RNN課題２

- 日英翻訳データ jpn.txt をdownloadし、グーグルのマイドライブの下、Colab Notebooksフォルダーに、uploadしてください。
- スタログにrnn_translation.ipynb があります。それをjpn.txtと同じ場所においてください。これもPyTorchサンプルを単純化し、日本語データを扱うように変更したものです。
- GRUというのは、ゲート付きRNNの一つで、この後で解説します。
- 埋め込み表に関しては、単語ベクトルのスライドを見てください。
- 読解します。

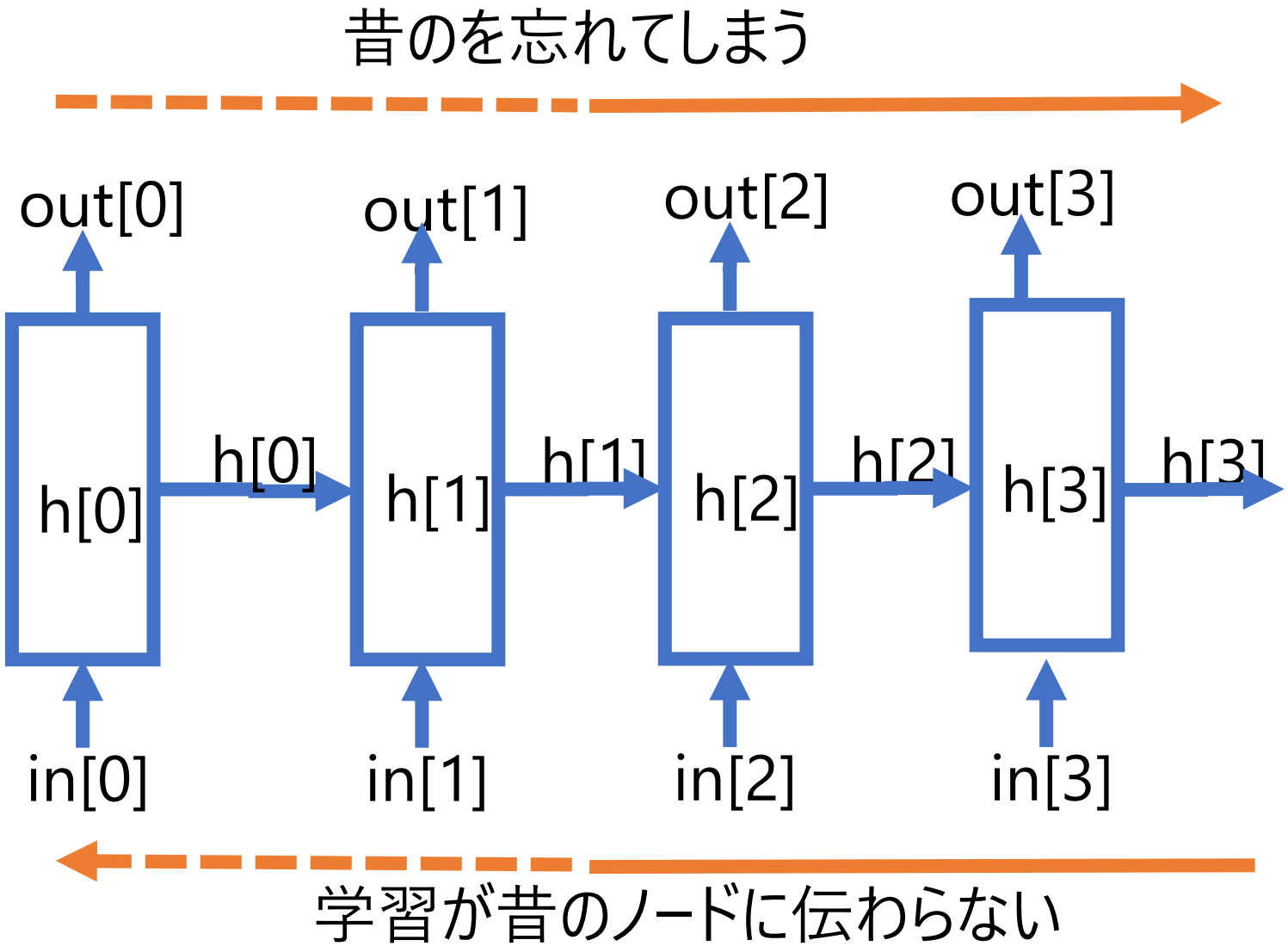
ネットワーク構成


Encoder
Decoder
先頭の入力文字はSOS
出力
入力文の単語列
一単語ずつ
埋め込み表
GRU
隠れ状態
最後の隠れ状態
埋め込み表
GRU
出力
Softmax
予測単語の語彙数サイズベクトル
予測単語ID
隠れ状態

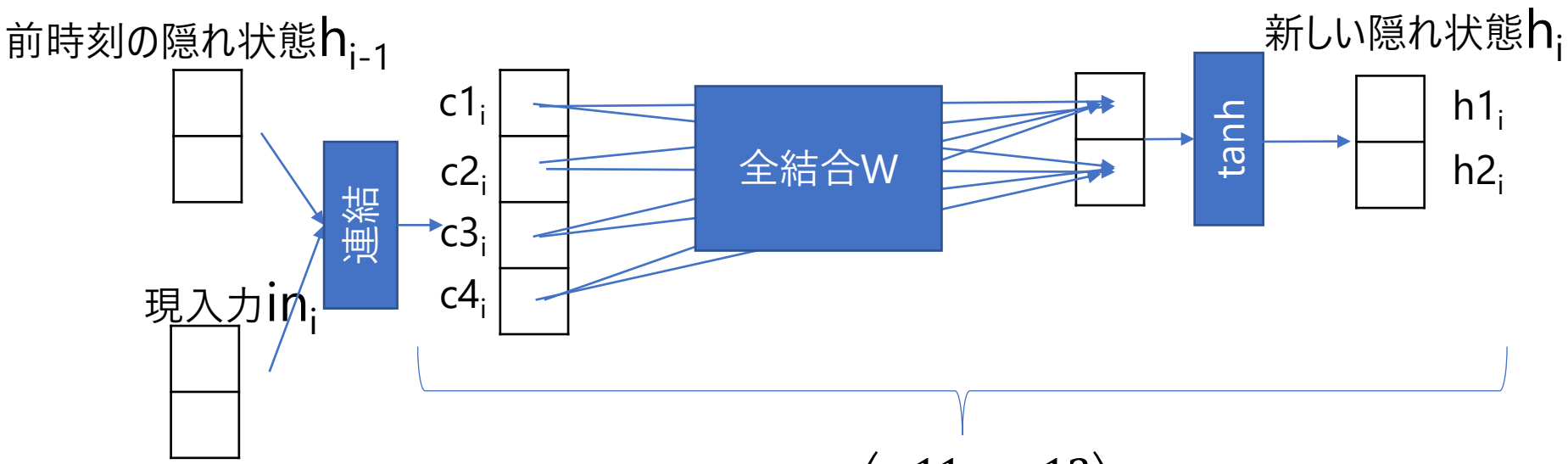
自然言語処理
ゲート付きRNN（LSTM, GRU）

解説動画

素のRNN(Elman-netと呼ばれる)の問題点



Elman Netは、忘れやすい



$$\tanh\{(c1_i, c2_i, c3_i, c4_i) \begin{pmatrix} w11 & w12 \\ w21 & w22 \\ w31 & w32 \\ w41 & w42 \end{pmatrix}\} = (h1_i, h2_i)$$

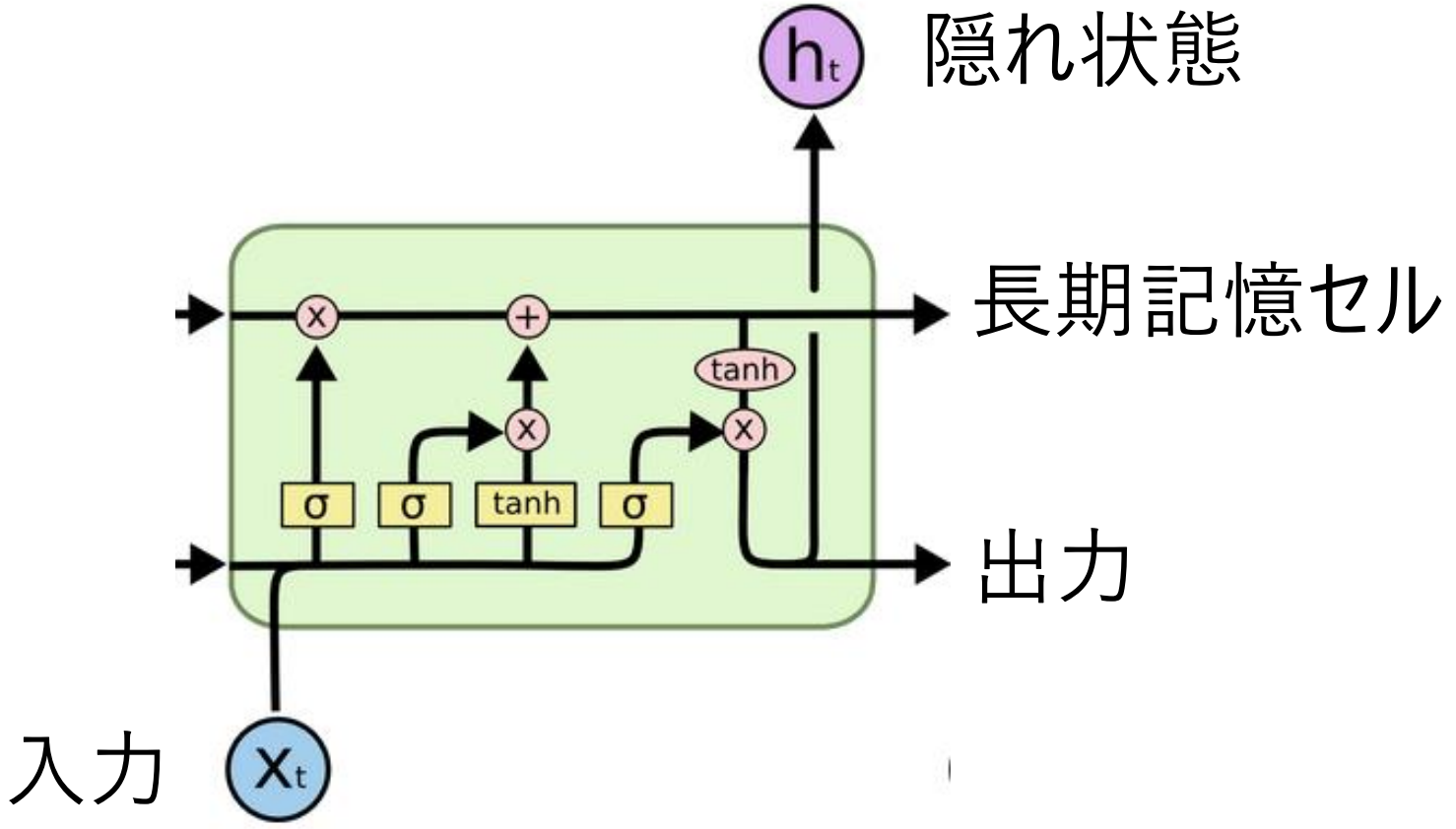
毎時刻ごと上書きされるので、最終入力の影響を最も受けやすく、最初の頃の入力の影響は薄れてしまう。

ゲート付きRNN (LSTM,GRU)

加算によって長期記憶を実現

情報の経路上にゲートを配置し、記憶する情報を学習で最適化

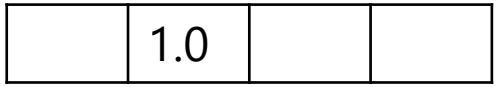
LSTM (Long Short-term Memory)



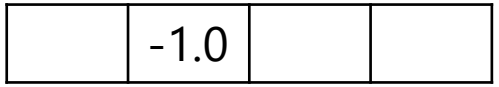
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

足し算は、記憶の効果がある

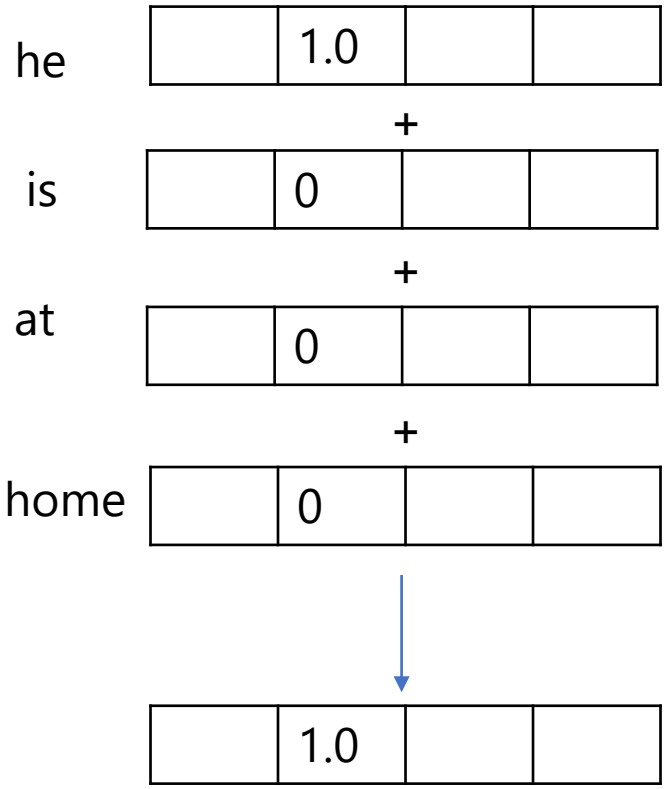
heの埋め込み表現



sheの埋め込み表現

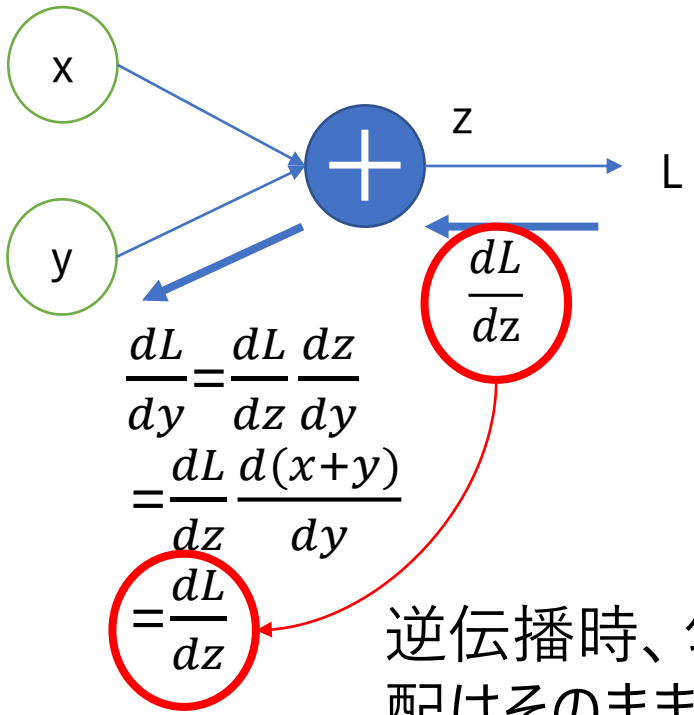


学習の結果、たまたま、状態ベクトルの第 2 要素が性別を表現したとする

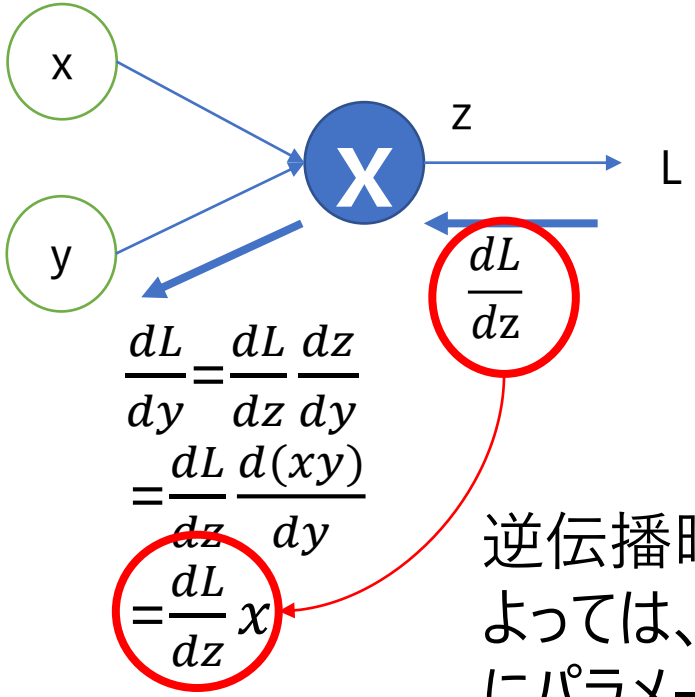


ベクトルの加算の結果、最終状態であっても、性別情報が記憶されている

足し算は逆伝播時に勾配をまま伝えるので先祖状態にも学習効果が届く

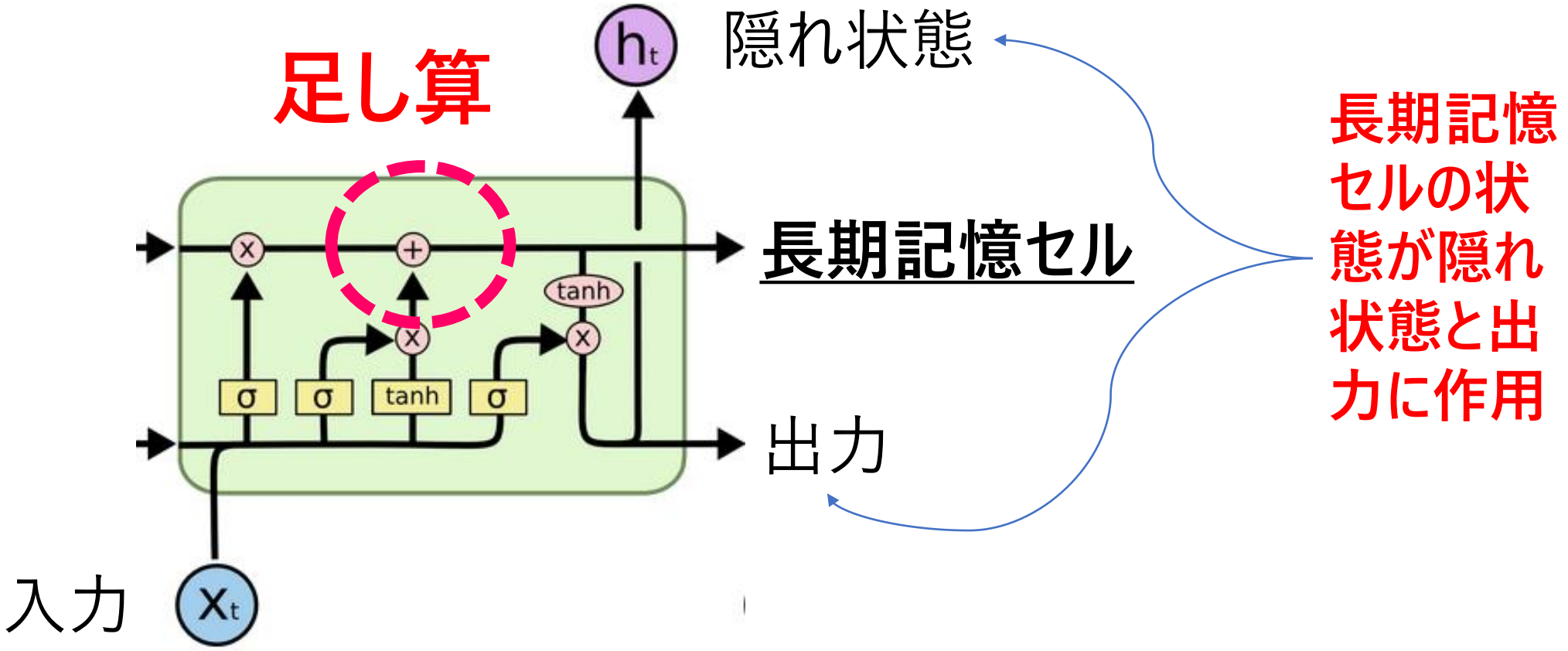


逆伝播時、勾配はそのまま伝わる



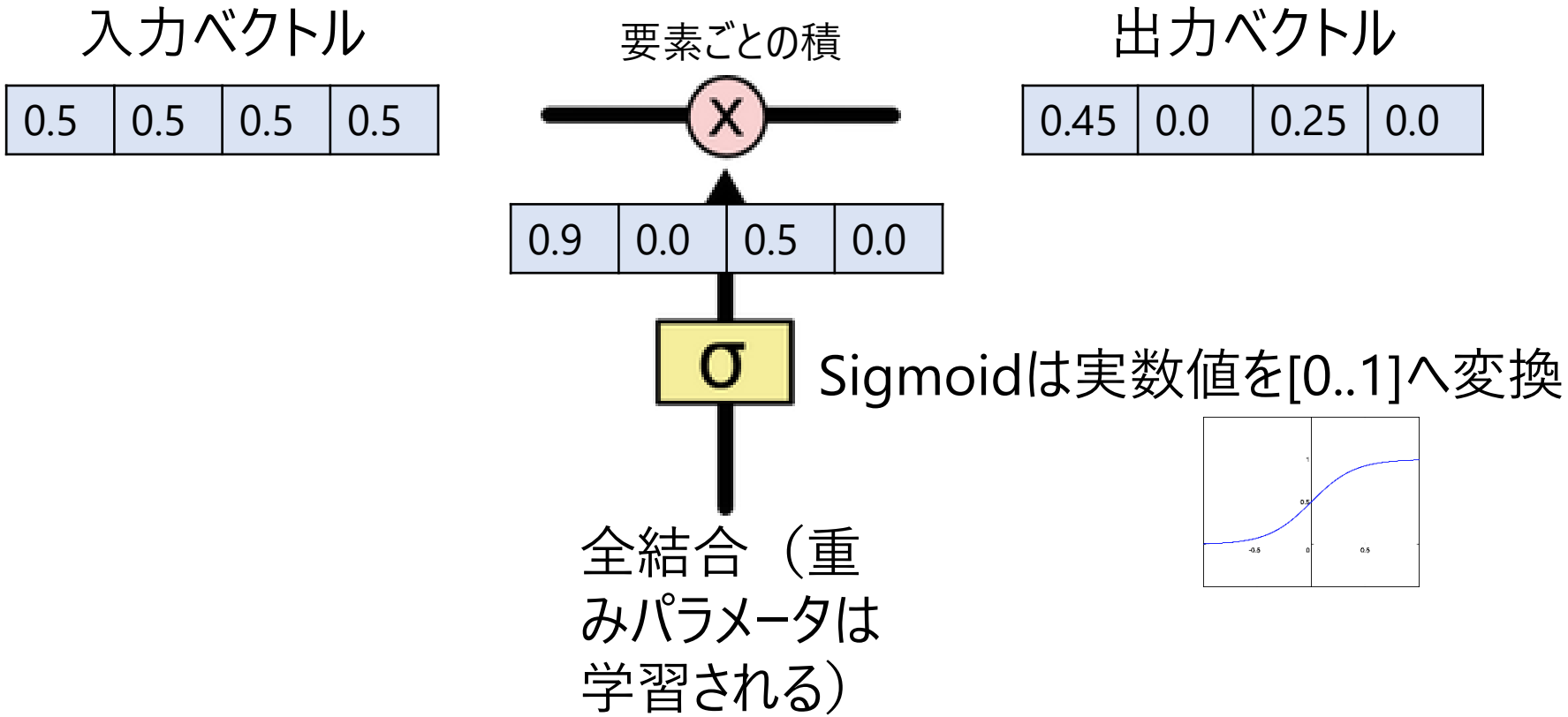
逆伝播時、相棒の値によっては、勾配（学習時にパラメータ更新量を決める）は急速にしぼんだり、爆発したリする

LSTMの長期記憶

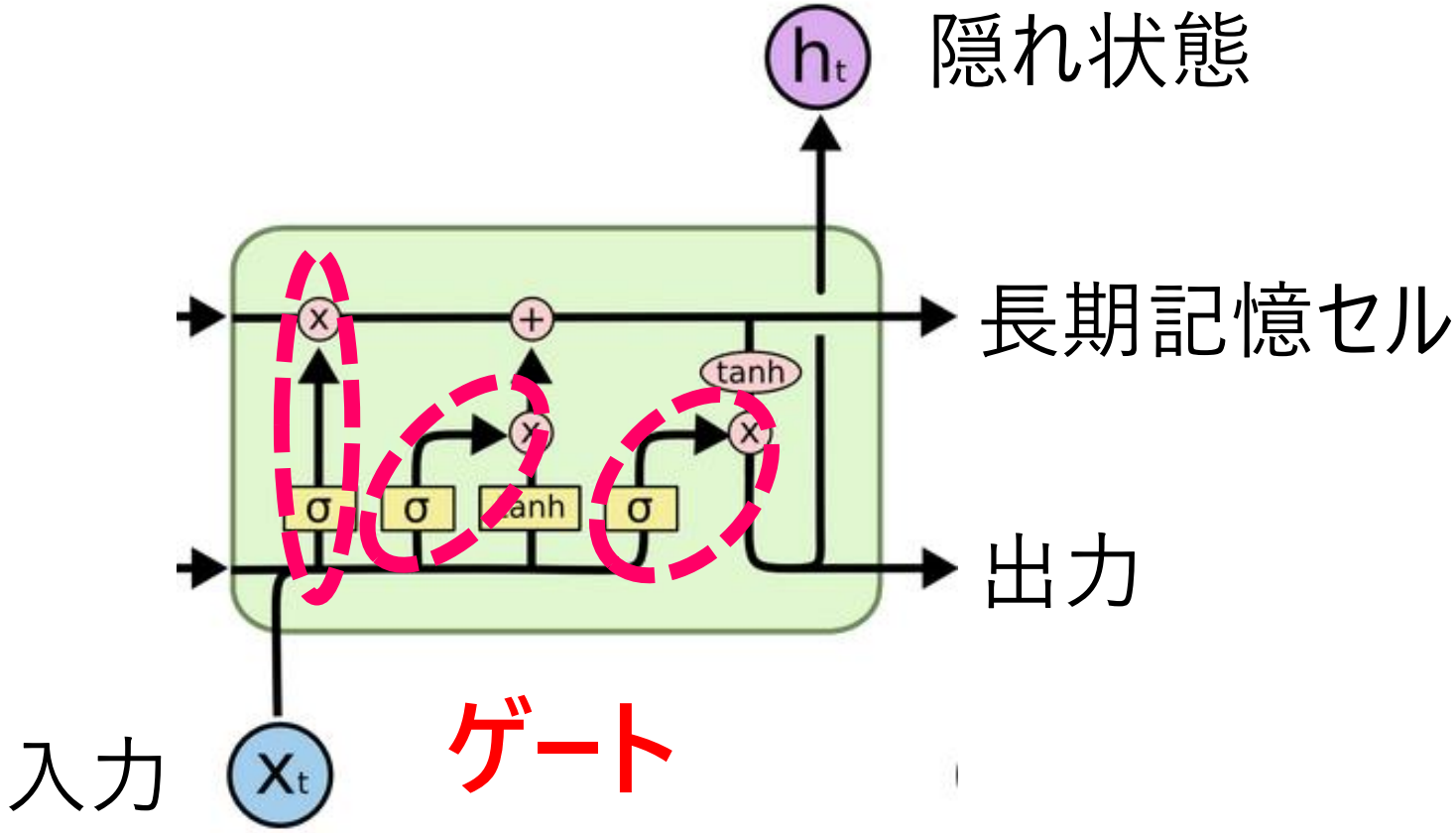


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

ゲート：ベクトルの各位置に対して[0..1]の重みをかけて、通過する値を制御する



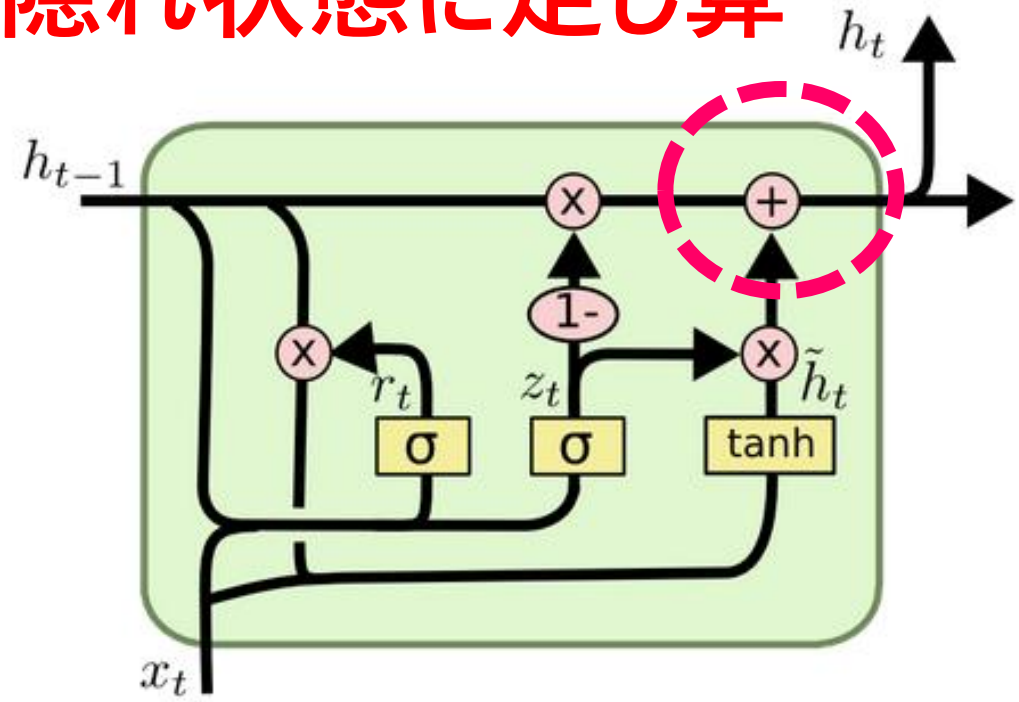
LSTMのゲート



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

GRU (Gated Recurrent Unit)

隠れ状態に足し算

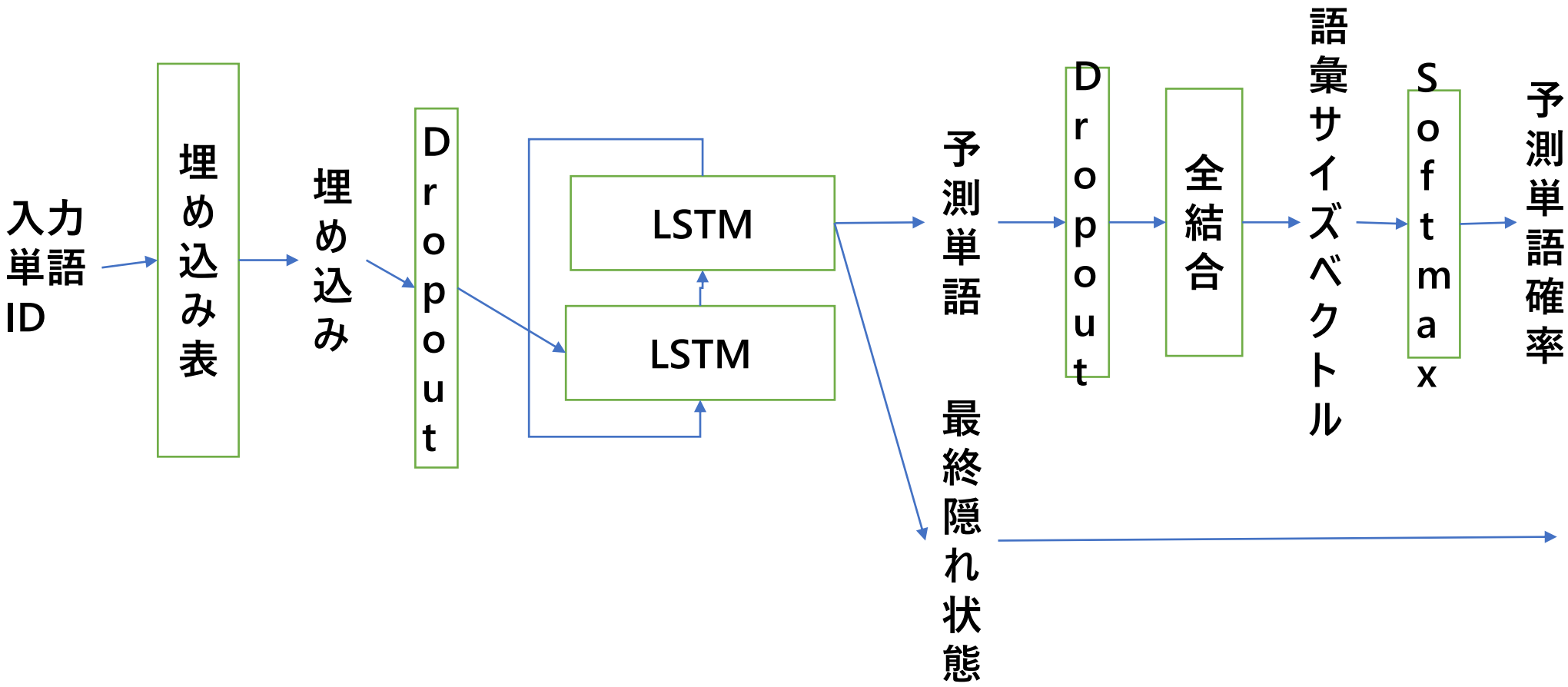


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN課題3（Option）

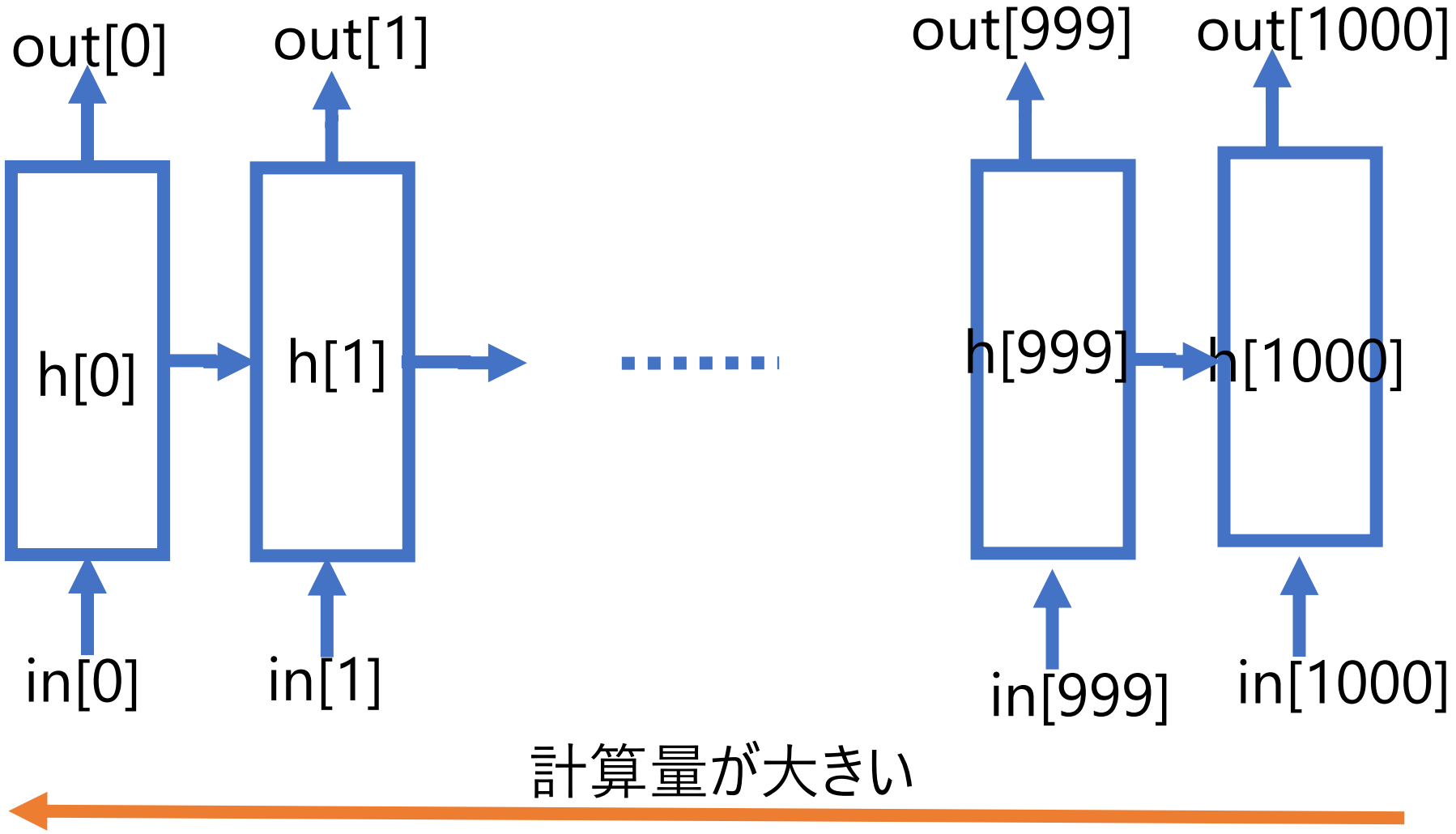
- rnn_languagemodel.ipynb があります。これもPyTorchのサンプルを単純化したものです。
- 単語列を学習させて、その後、適当な先頭単語を与えて作文させるものです。
- 読解しましょう。コードが複雑なので、オプションです。
- フレームワークを使うと、GRUやLSTMはクラスを呼ぶだけですし、パーツの組み合わせも数行で済みます。が、メソッドの引数の意味を理解しないといけないこと、ニューラルネット開発の実態は、実は多次元データを準備するところに時間がとられること、位を見ておいてください。

ネット構成

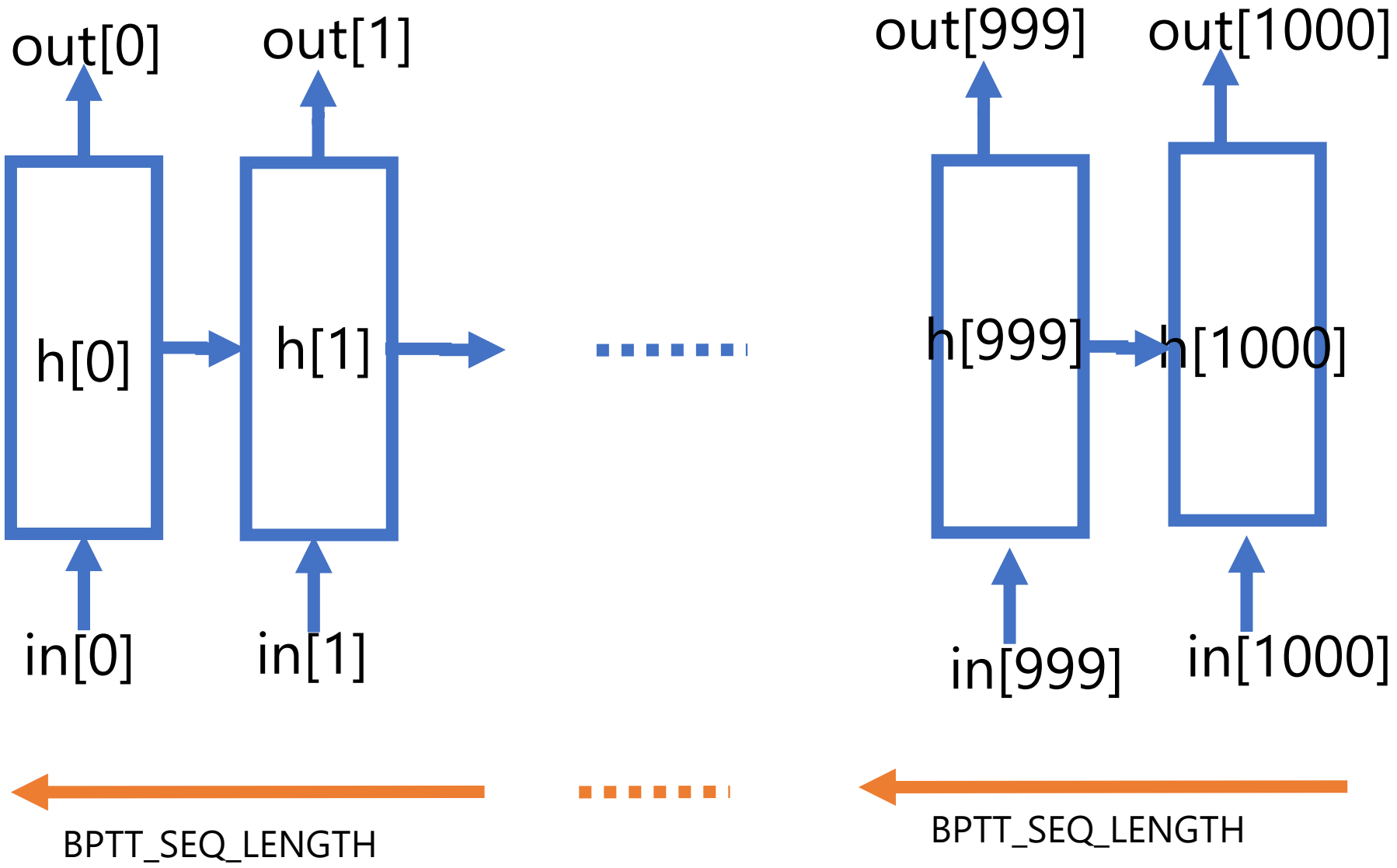

入力単語ID
埋め込み表
埋め込み
Dropout
LSTM
LSTM
予測単語
最終隠れ状態
Dropout
全結合
語彙サイズベクトル
Softmax
予測単語確率

コードを読むときの追加参考情報

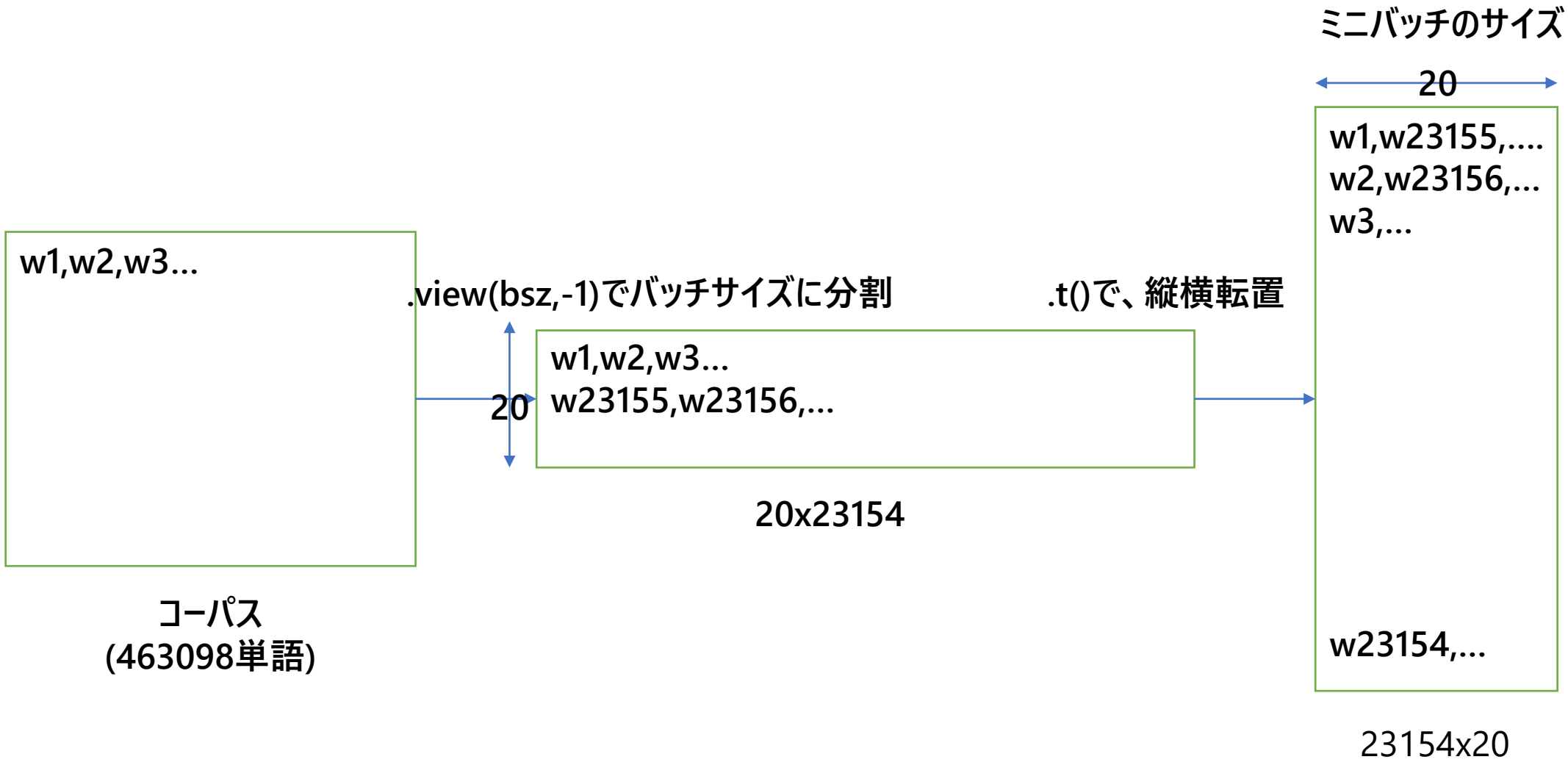
入力が長いと？


out[0]
out[1]
out[999]
out[1000]
h[0]
h[1]
h[999]
h[1000]
in[0]
in[1]
in[999]
in[1000]
計算量が大きい

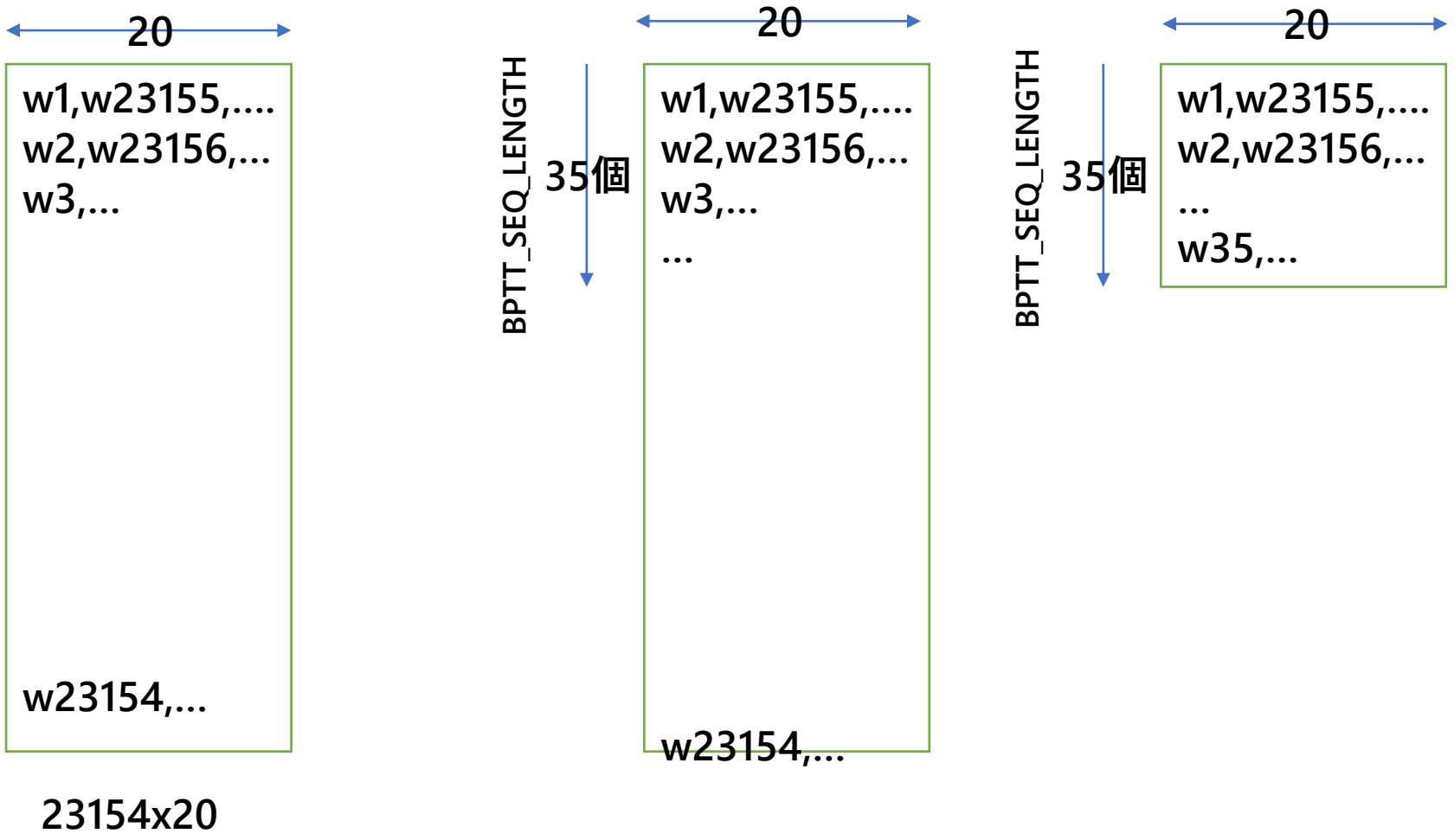
ある固定長で系列処理を分断する



batchify(コーパス、バッチサイズ=20)



get_batch(バッチ行列、開始添え字)



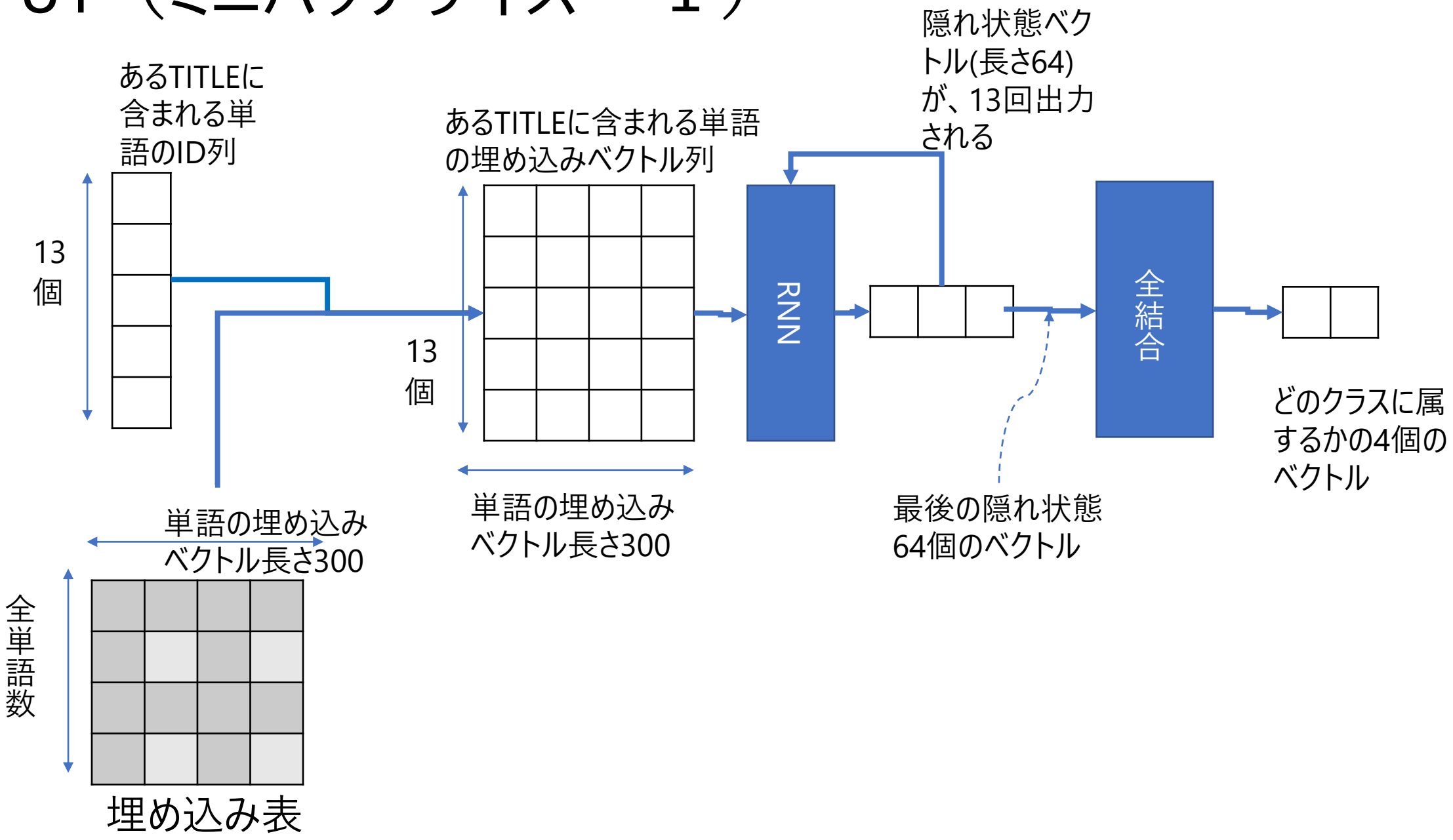
参考資料

- ゼロから作るDeep Learning ② 第6章
- Tutorial記事
  - Animated RNN, LSTM and GRU
  - Understanding LSTM Networks
- 教育動画
  - Deep Learning入門：数式なしで理解するLSTM (Long short-term memory)

100本ノック第9章課題80,81,83

- 「100本ノック」の9章の課題 の80(データ準備）、81(RNN)、83(RNN)に取り組みましょう。
- 「NLP、CNNRNNTransformer.ipynb」というノートをコピーしてください。80、81、83のポイントとなる部分を??????にしています。完成させなさい。実行ログを残してください。

81（ミニバッチサイズ＝１）



確認クイズ

- 確認クイズをやってください。