RNN, LSTM

目的

- フレームワークを使ってネットワーク（RNN）を構築できる
- 様々なフレームワークを使える

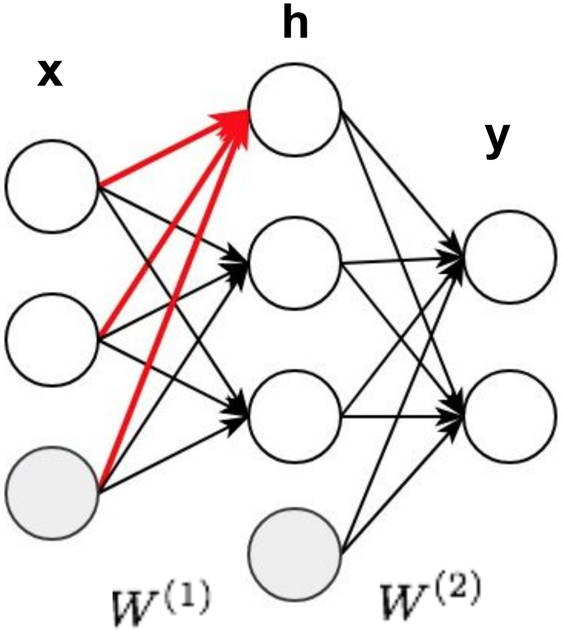
目次

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（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

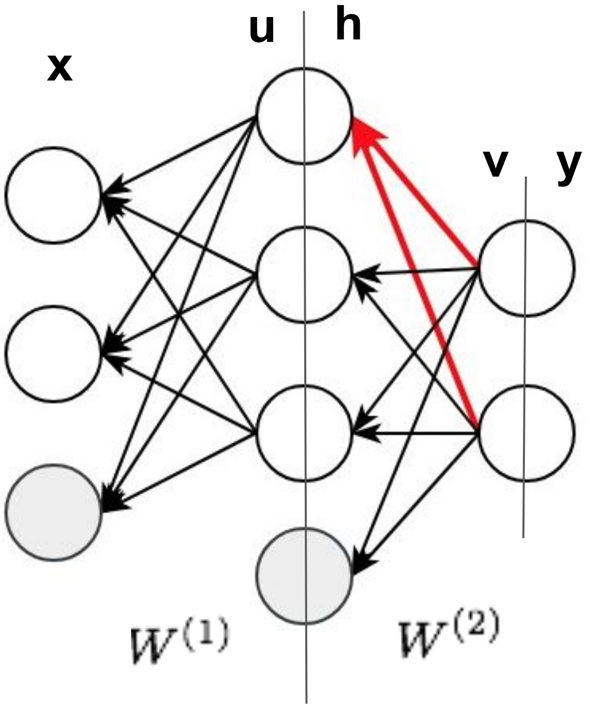


$$\boldsymbol{h}^{\ell} = f^{\ell}(W^{\ell}\boldsymbol{h}^{\ell-1} + \boldsymbol{b}^{\ell})$$

$$\boldsymbol{h} = f(W^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$$
$$\boldsymbol{y} = g(W^{(2)}\boldsymbol{h} + \boldsymbol{b}^{(2)})$$

順伝播型ニューラルネットワーク

- 逆伝播



勾配降下法

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

順伝播の別記法

$$\boldsymbol{u} = W^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}$$
$$\boldsymbol{h} = f(\boldsymbol{u})$$
$$\boldsymbol{v} = W^{(2)}\boldsymbol{h} + \boldsymbol{b}^{(2)}$$
$$\boldsymbol{y} = g(\boldsymbol{v})$$

誤差の勾配

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

順伝播型ニューラルネットワーク

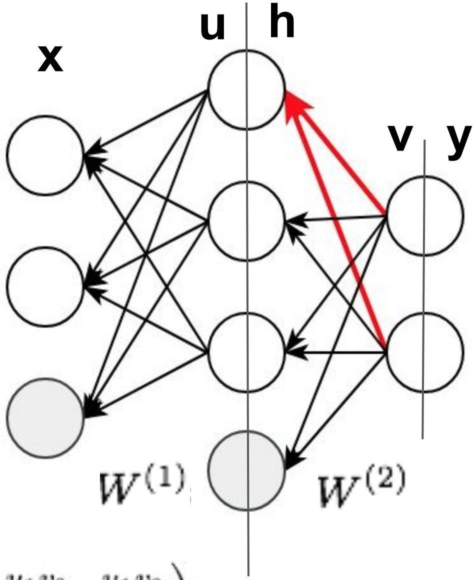
- 各パラメータについての勾配

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

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

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

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

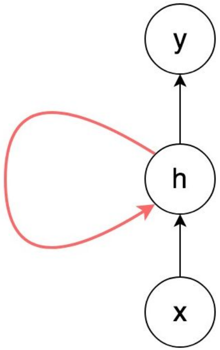


テンソル積（直積）

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

RNN(Recurrent Neural Network)とは

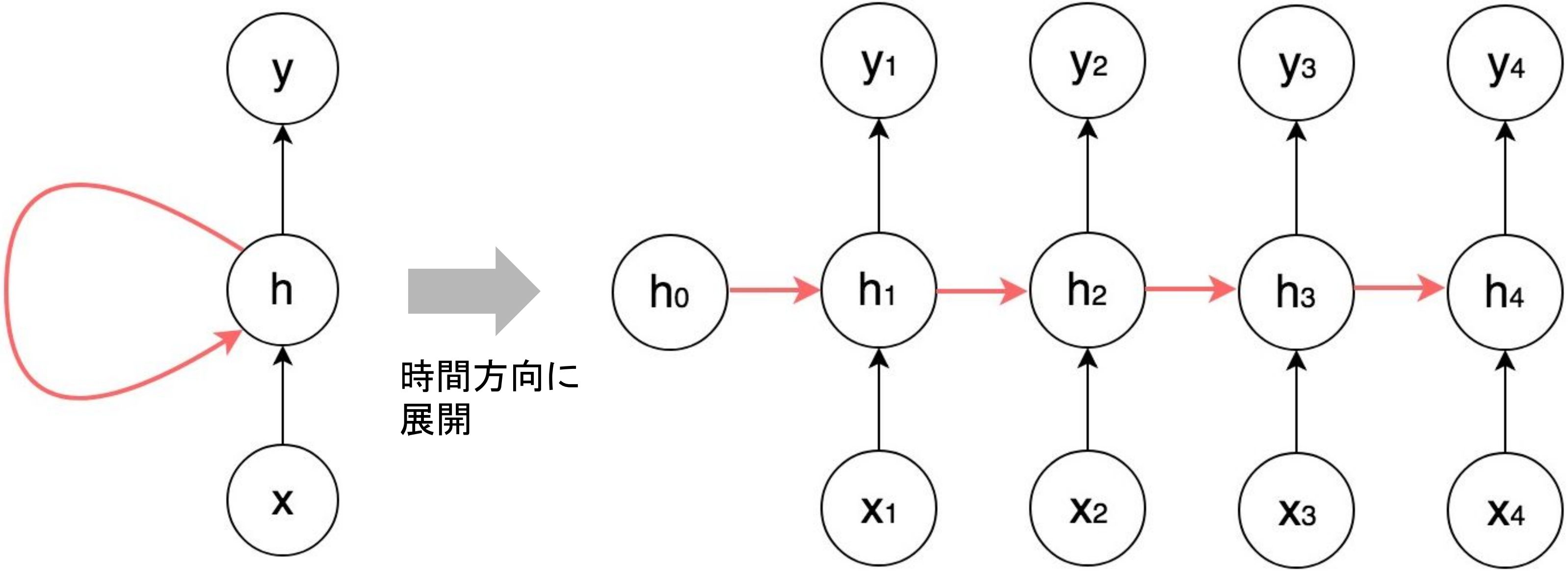
・過去の隠れ層の状態も入力に



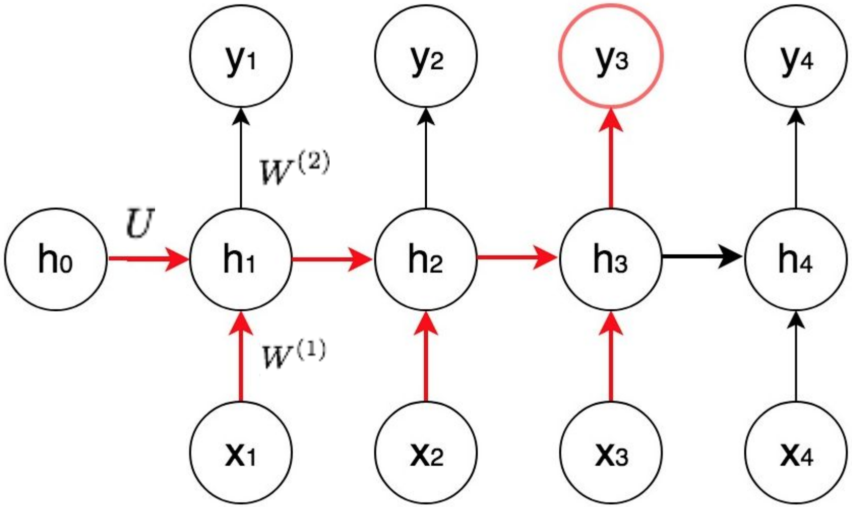
・系列データを扱うのに適したニューラルネットワーク

- ・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



$$\boldsymbol{h}_t = f(W^{(1)}\boldsymbol{x}_t + \underline{U\boldsymbol{h}_{t-1}} + \boldsymbol{b}^{(1)})$$
$$\boldsymbol{y}_t = g(W^{(2)}\boldsymbol{h}_t + \boldsymbol{b}^{(2)})$$

RNNの誤差逆伝播

・BPTT(BackPropagation Through Time)

　・時間展開したうえでの誤差逆伝播

・Truncated BPTT

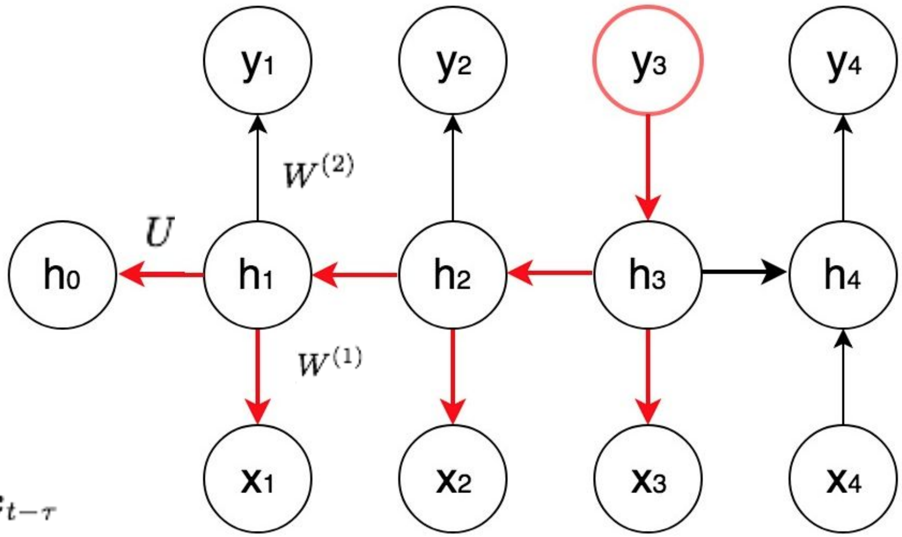
　・遡るステップ数を限定したBPTT

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

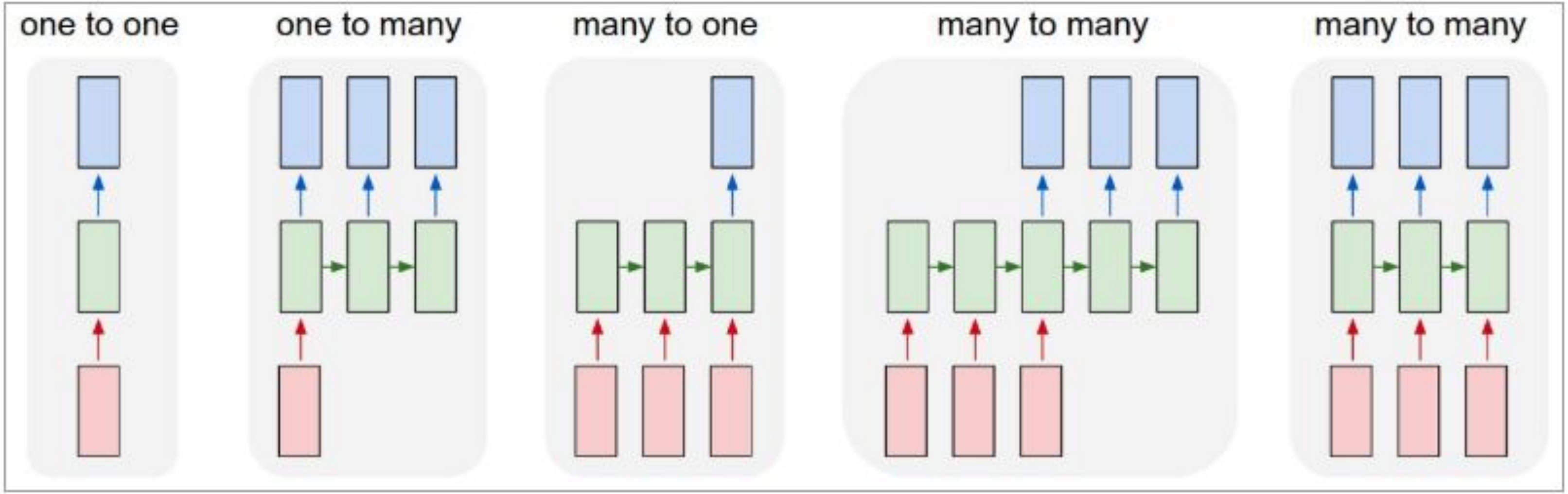
例)

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

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



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**

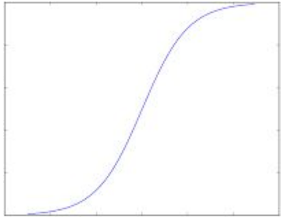
Image by Andrej Karpathy blog

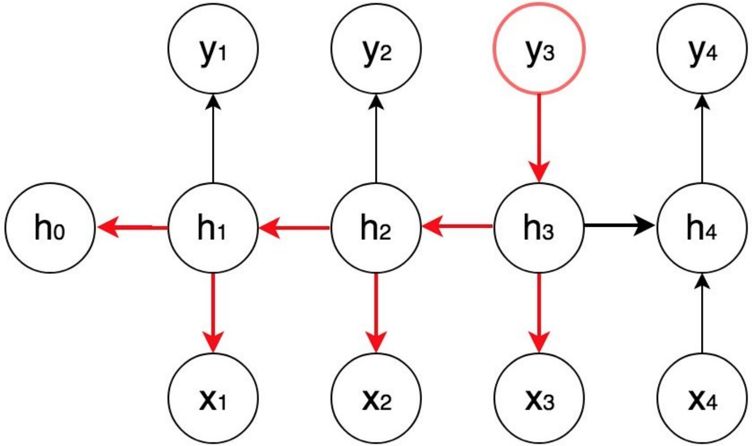
RNNの問題

・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）

$$\boldsymbol{\delta}_{t-1}^{(u)} = \frac{\partial L_t}{\partial \boldsymbol{u}_{t-1}} = \frac{\partial L_t}{\partial \boldsymbol{u}_t}\frac{\partial \boldsymbol{u}_t}{\partial \boldsymbol{h}_{t-1}}\frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{u}_{t-1}}$$
$$= U^T \boldsymbol{\delta}_t^{(u)} \underline{f'(\boldsymbol{h}_{t-1})}$$

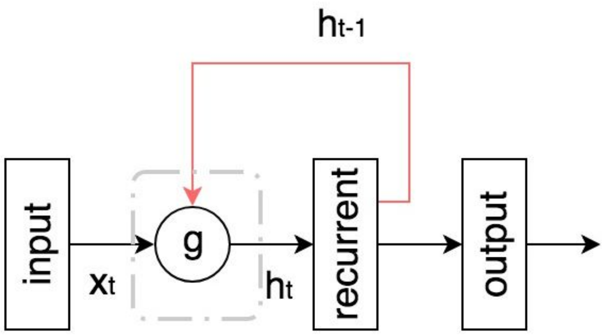




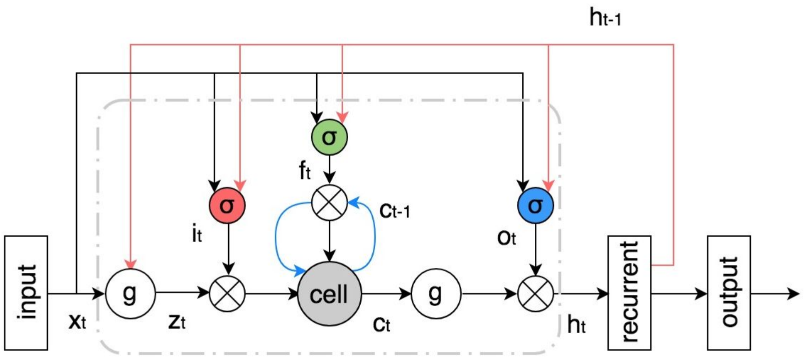
LSTM(Long Short Term Memory)とは

・RNNの派生モデル

・長期的な依存関係を学習できる（勾配消失を避ける）
・セルとゲートを導入



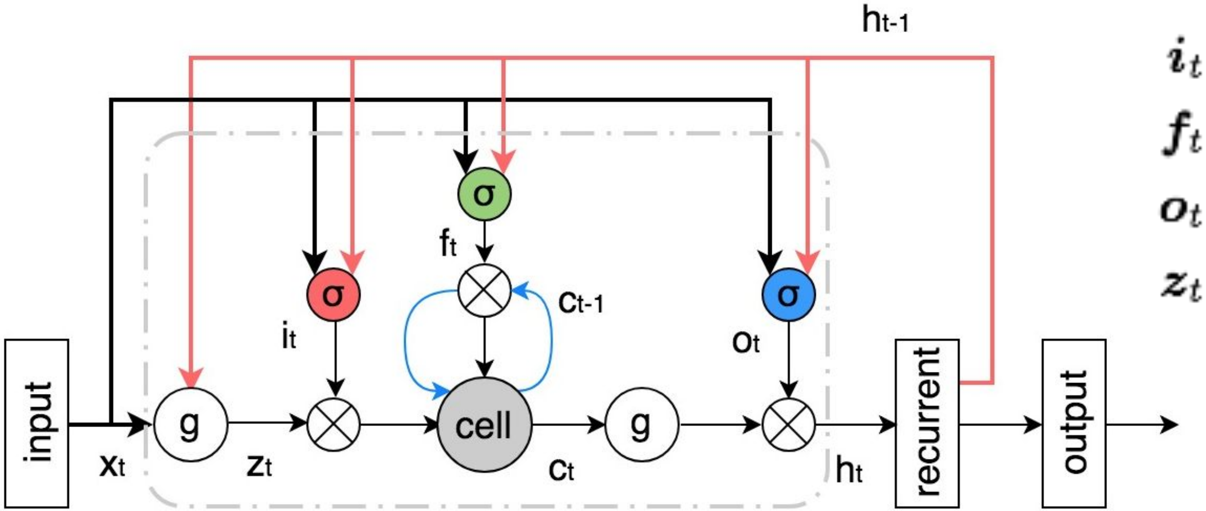
RNN



LSTM

LSTMの順伝播①

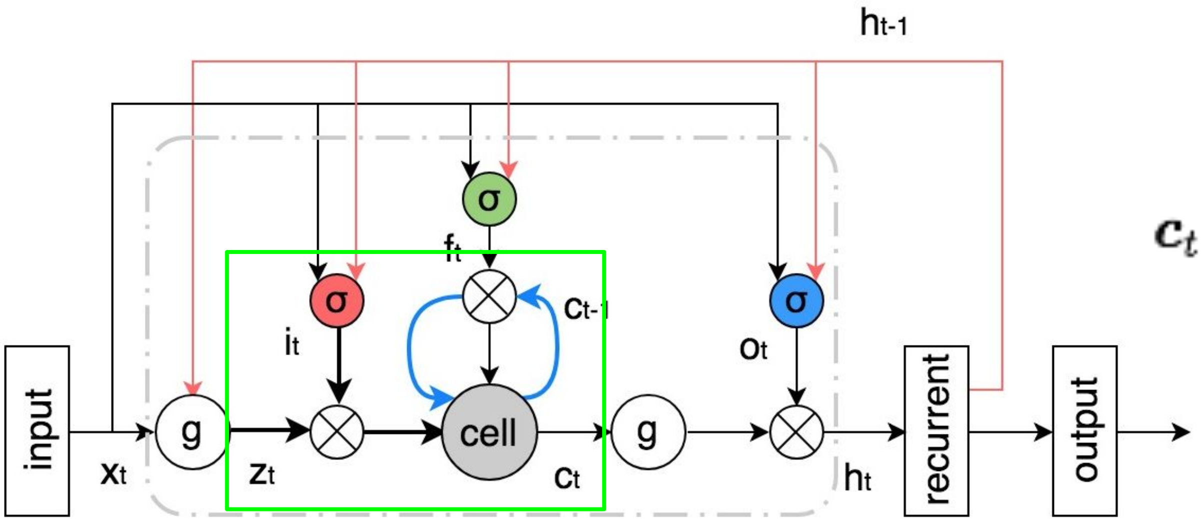
・ゲート（入力ゲート・忘却ゲート・出力ゲート）と
　セルへの入力の計算



$$\boldsymbol{i}_t = \sigma(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i)$$
$$\boldsymbol{f}_t = \sigma(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f)$$
$$\boldsymbol{o}_t = \sigma(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o)$$
$$\boldsymbol{z}_t = g(W_z \boldsymbol{x}_t + U_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z)$$

LSTMの順伝播②

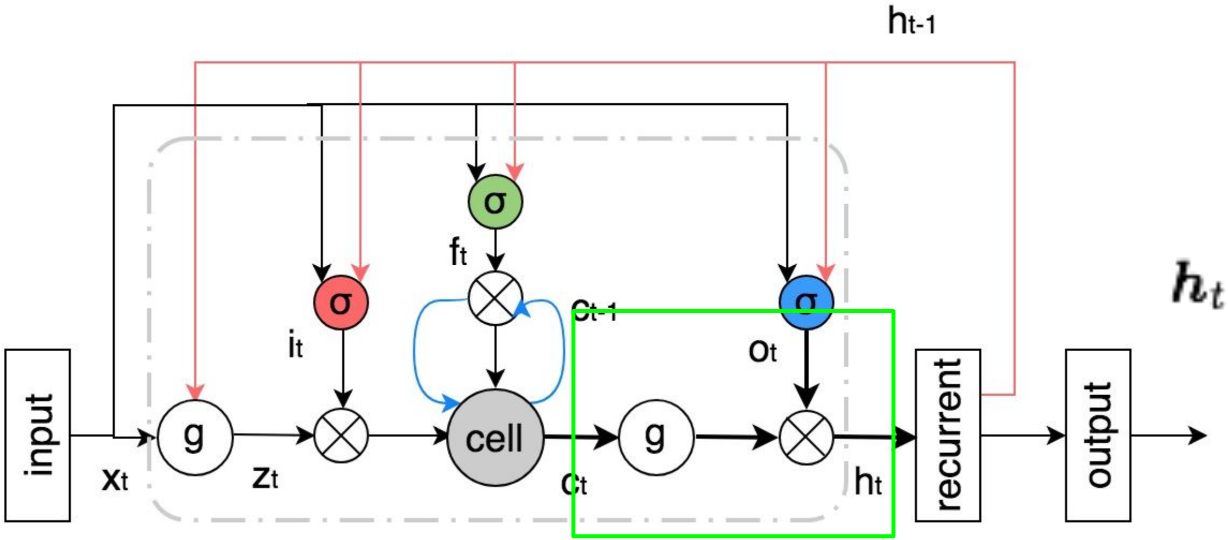
・セルの値を計算



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

LSTMの順伝播③

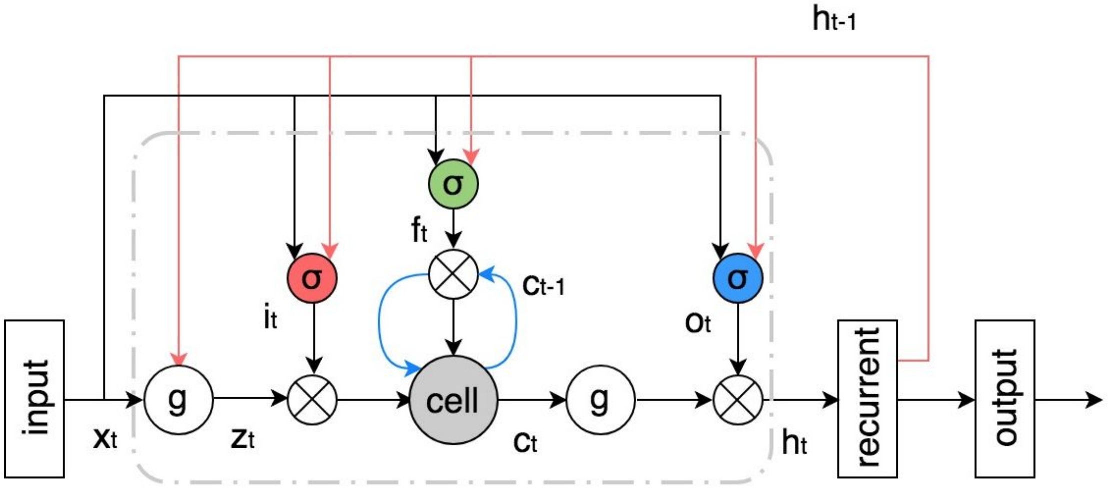
・出力値を計算



$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot g(\boldsymbol{c}_t)$$

LSTM例

- 空欄補充



「映画おもしろかったね。ところで、とてもお腹が空いたから、何か＿＿＿＿。」

食べよう
見に行こう
話そう

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ことがある

　・大きい勾配に対してノルムをしきい値以下に正規化

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

　・ほとんどのRNNモデルはLSTMを使用

Appendix

RNNのAPI(TensorFlow)

・tf.contrib.rnn.BasicRNNCell

(num_units, activation=tanh)

・RNNセルを生成

・tf.contrib.rnn.BasicLSTMCell

(num_units, forget_bias=1.0, activation=tanh)

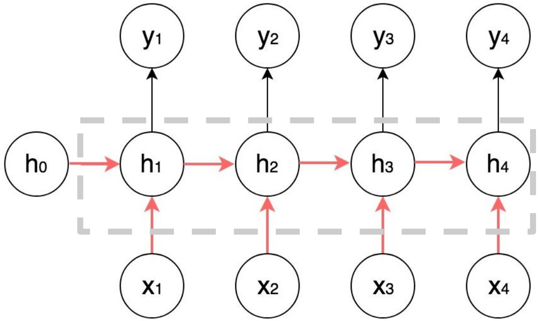
・LSTMセルを生成

・tf.contrib.rnn.static_rnn

(cell, inputs, initial_state=None)

・時間方向に展開

・出力と最後の内部状態を返す



LSTMのAPI(Chainer)

・chainer.links.LSTM

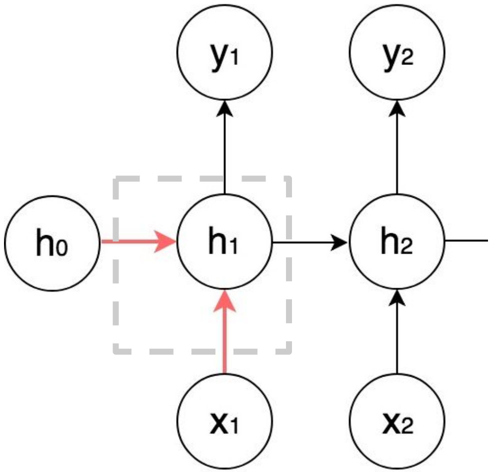
(in_size, out_size, forget_bias_init=1.0)

・LSTM層を生成

・__call__(x)

・内部状態を更新

・LSTMの出力を返す



順伝播型ニューラルネットワークの誤差勾配

・テンソル積・転置となる過程

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

$$\frac{\partial L}{\partial W_{ij}^{(2)}} = \frac{\partial L}{\partial v_i} \frac{\partial v_i}{\partial W_{ij}^{(2)}} = \delta_i^{(v)} h_j$$

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

$$\begin{aligned}\delta_i^{(u)} &= \frac{\partial L}{\partial u_i} = \sum_j \left(\frac{\partial L}{\partial v_j} \frac{\partial v_j}{\partial h_i}\right) \frac{\partial h_i}{\partial u_i} \\ &= \sum_j (\delta_j^{(v)} W_{ji}^{(2)}) f'(u_i)\end{aligned}$$

RNNのパラメータについての誤差勾配

順伝播

$$\boldsymbol{u}_t = W^{(1)}\boldsymbol{x}_t + U\boldsymbol{h}_{t-1} + \boldsymbol{b}^{(1)}$$

$$\boldsymbol{h}_t = f(\boldsymbol{u}_t)$$

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

$$\boldsymbol{y}_t = g(\boldsymbol{v}_t)$$

u, vについての
誤差勾配

$$\boldsymbol{\delta}_t^{(v)} = \frac{\partial L_t}{\partial \boldsymbol{v}_t} = \frac{\partial L_t}{\partial \boldsymbol{y}_t}\frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{v}_t} = \frac{\partial L_t}{\partial \boldsymbol{y}_t} g'(\boldsymbol{v}_t)$$

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

全体誤差は各時
刻での誤差の和

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

1ステップ前のｕ
の誤差勾配

$$\boldsymbol{\delta}_{t-1}^{(u)} = \frac{\partial L_t}{\partial \boldsymbol{u}_{t-1}} = \frac{\partial L_t}{\partial \boldsymbol{u}_t}\frac{\partial \boldsymbol{u}_t}{\partial \boldsymbol{h}_{t-1}}\frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{u}_{t-1}}$$
$$= U^T \boldsymbol{\delta}_t^{(u)} f'(\boldsymbol{h}_{t-1})$$

RNNの各パラメータについての誤差勾配

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

$$\frac{\partial L_t}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial L_t}{\partial \boldsymbol{v}_t}\frac{\partial \boldsymbol{v}_t}{\partial \boldsymbol{b}^{(2)}} = \boldsymbol{\delta}_t^{(v)}$$

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

$$\frac{\partial L_t}{\partial U^{(1)}} = \sum_{\tau=0}^{t-1}\frac{\partial L_t}{\partial \boldsymbol{u}_{t-\tau}}\frac{\partial \boldsymbol{u}_{t-\tau}}{\partial U^{(1)}_{(t-\tau)}} = \sum_{\tau=0}^{t-1}\boldsymbol{\delta}_{t-\tau}^{(u)} \otimes \boldsymbol{h}_{t-\tau-1}$$

$$\frac{\partial L_t}{\partial \boldsymbol{b}^{(1)}} = \sum_{\tau=0}^{t-1}\frac{\partial L_t}{\partial \boldsymbol{u}_{t-\tau}}\frac{\partial \boldsymbol{u}_{t-\tau}}{\partial \boldsymbol{b}^{(1)}_{(t-\tau)}} = \sum_{\tau=0}^{t-1}\boldsymbol{\delta}_{t-\tau}^{(u)}$$

LSTMが勾配消失を避ける理由

cについての勾配は、1ステップ前においても、活性化関数の微分を通さず、忘却ゲートの値倍となる(忘却ゲートの値が 1に近ければ誤差は消失しない)

$$\boldsymbol{\delta}_t^{(c)} = \frac{\partial L_t}{\partial \boldsymbol{c}_t}$$

$$\boldsymbol{\delta}_{t-1}^{(c)} = \frac{\partial L_t}{\partial \boldsymbol{c}_{t-1}} = \frac{\partial L_t}{\partial \boldsymbol{c}_t}\frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_{t-1}} = \boldsymbol{\delta}_t^{(c)} \boldsymbol{f}_t$$

各パラメータについての勾配は、cについての勾配を元に計算できるため、cについての勾配が消失しないなら、より以前の情報も考慮できる。(以下は例として WiとWoについて具体的に計算)

$$\begin{aligned}\frac{\partial L_t}{\partial W_i} &= \sum_{\tau=0}^{t-1} \frac{\partial L_t}{\partial \boldsymbol{c}_{t-\tau}} \frac{\partial \boldsymbol{c}_{t-\tau}}{\partial \boldsymbol{i}_{t-\tau}} \frac{\partial \boldsymbol{i}_{t-\tau}}{\partial W_{i,(t-\tau)}} \\ &= \sum_{\tau=0}^{t-1} (\boldsymbol{\delta}_{t-\tau}^{(c)} \boldsymbol{z}_{t-\tau}) \otimes \boldsymbol{x}_{t-\tau}\end{aligned}$$

$$\boldsymbol{\delta}_t^{(h)} = \frac{\partial L_t}{\partial \boldsymbol{h}_t} = \frac{\partial L_t}{\partial \boldsymbol{c}_t}\frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{h}_t} = \boldsymbol{\delta}_t^{(c)} \boldsymbol{o}_t \frac{1}{\cosh^2(\boldsymbol{c}_t)}$$

$$\begin{aligned}\frac{\partial L_t}{\partial W_o} &= \sum_{\tau=0}^{t-1} \frac{\partial L_t}{\partial \boldsymbol{h}_{t-\tau}} \frac{\partial \boldsymbol{h}_{t-\tau}}{\partial \boldsymbol{o}_{t-\tau}} \frac{\partial \boldsymbol{o}_{t-\tau}}{\partial W_{o,(t-\tau)}} \\ &= \sum_{\tau=0}^{t-1} (\boldsymbol{\delta}_{t-\tau}^{(h)} \tanh(\boldsymbol{c}_{t-\tau})) \otimes \boldsymbol{x}_{t-\tau}\end{aligned}$$