自然言語処理の深層学習
ーLSTM編

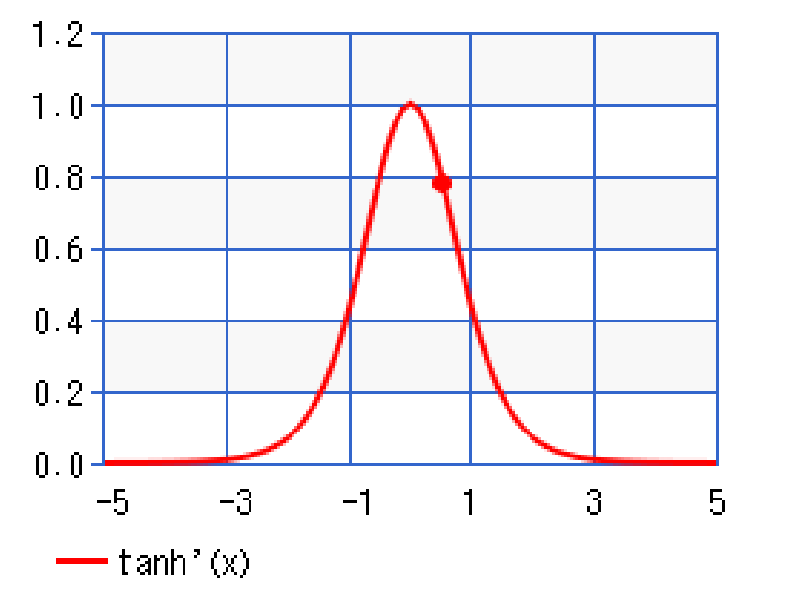
aotszk66

RNNの問題点

- 長い時系列データの関係をうまく学習できない
  - 勾配消失・勾配爆発が原因
  - 勾配消失
    - 時間が遡るにつれ勾配が小さくなり,重みを更新できない
  - 勾配爆発
    - 時間とともに勾配が大きくなり,オーバーフローが発生
- データが多いと計算量が膨大になる

勾配消失の原因1

- RNNの隠れ層は次の式で計算
  - $h_t = \tanh(h_{t-1} * W_h + x_t * W_x + b)$
- y = tanh(x)の微分は$y' = 1 - y^2$となり，
  y'のプロットは下図のようになる



・学習を繰り返すと勾配が小さくなる

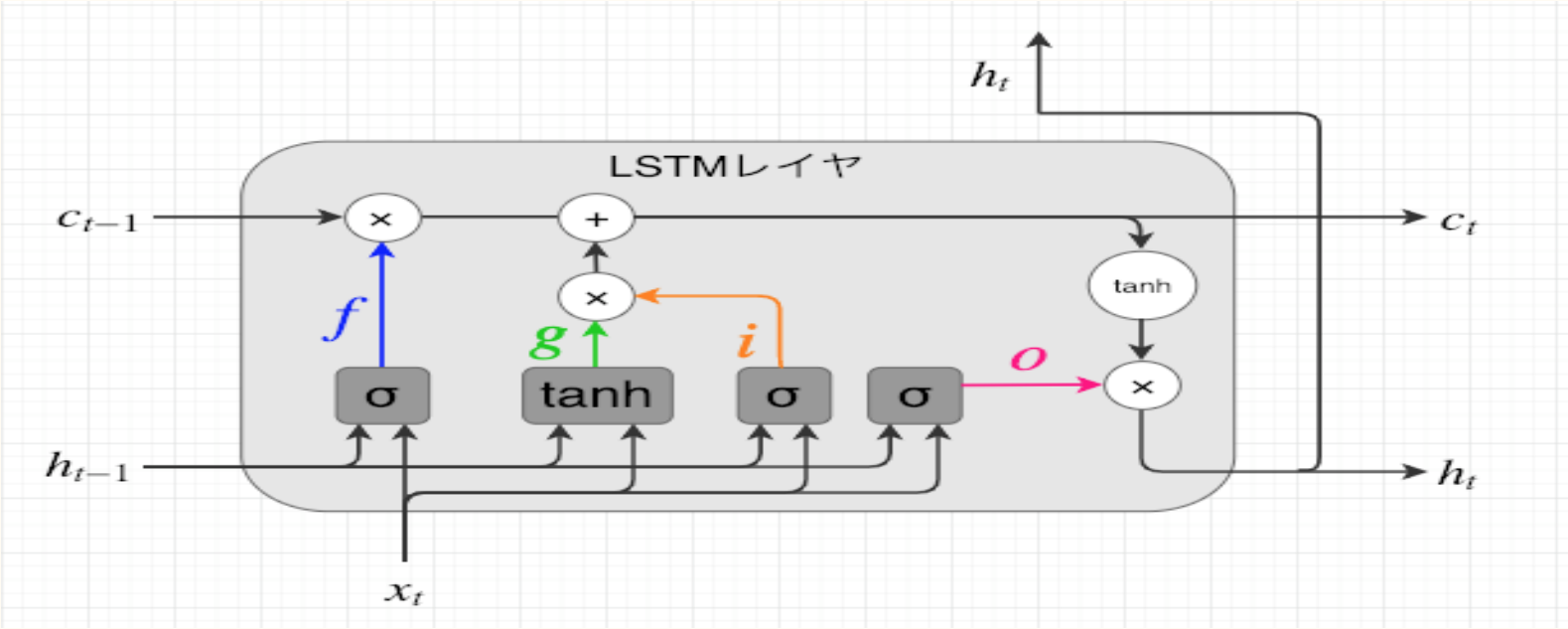
勾配消失が発生

勾配消失・爆発の原因2

- 隠れ層hの逆伝播は$h * W_h^T$で計算される
- この計算がデータのサイズ分だけ行われる
  - 隠れ層hの重みW_hの値が大きい場合
    - 重みW_hの値が増加していく
      →勾配爆発が発生
  - 隠れ層hの重みW_hの値が小さい場合
    - 重みW_hの値が減少していく
      →勾配消失が発生
- 勾配クリッピング
  - 勾配が設定した閾値を超えると,次式のように勾配を修正
    - $\mathrm{g} = \frac{\text{閾値}}{||g||} * g$
    - 勾配爆発の対策

LSTMの構成

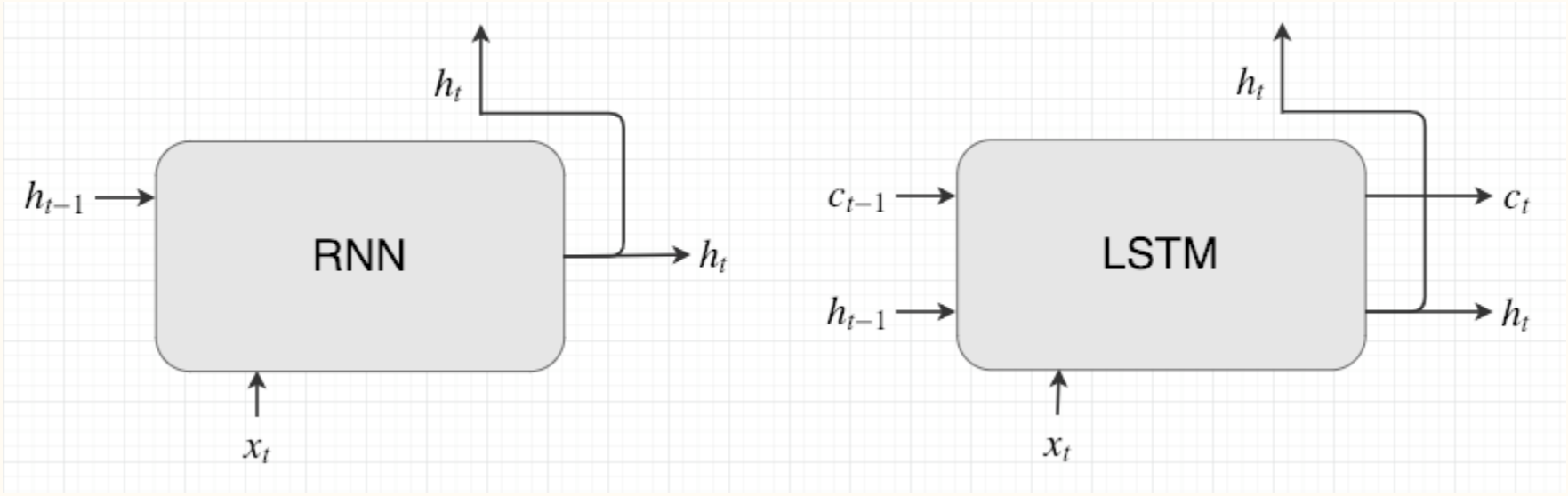
- LSTMの構成を以下に示す



- RNNに比べ,記憶セルC,output(o),forget(f),input(i)ゲート,新たな記憶セルgが追加されている

記憶セルC

- 記憶セルCはLSTM専用の記憶部



- 記憶セルはLSTMレイヤだけでデータの受け渡しを行うため,他のレイヤへの出力はない.

outputゲート

- h_tが次の層にどの程度,重要かを調整するゲート
- $o = \sigma(x_t * W_x^{(o)} + h_{t-1} * W_h^{(o)} + b^{(o)})$で求める
  - 入力はx_t,入力に対する重み$W_x^{(o)}$,前時刻h_{t-1}に対する重み$W_h^{(o)}$, $b^{(o)}$がバイアス,σはシグモイド
- $h_t = o \circledcirc \tanh(C_t)$で求められる

※◉はアダマール積

forgetゲート

- 前の記憶セルC_{t-1}から不要な情報を取り除くゲート
- $f = \sigma(x_t * W_x^{(f)} + h_{t-1} * W_h^{(f)} + b^{(f)})$で求める
  - 入力はx_t,入力に対する重み$W_x^{(f)}$,前時刻h_{t-1}に対する重み$W_h^{(f)}$, $b^{(f)}$がバイアス, σはシグモイドである
- $c_t = f \circledcirc C_t$で求められる

新しい記憶セルg

- 前時刻の隠れ層h_{t-1}と入力x_tを受け取る
- $g = tanh(x_t * W_x^{(g)} + h_{t-1} * W_h^{(g)} + b^{(g)})$
- gが前時刻の記憶セルC_{t-1}に加算され新しい記憶が生まれる

inputゲート

- 新たな記憶セルがどのくらい重要なのかを調整
- $i = \sigma(x_t * W_x^{(i)} + h_{t-1} * W_h^{(i)} + b^{(i)})$
  - 入力はx_t,入力に対する重み$W_x^{(i)}$,前時刻h_{t-1}に対する重み$W_h^{(i)}$, $b^{(i)}$がバイアス, σはシグモイドである
- C_{t-1} にg◉iを加算する

- 全体の最終的な計算はP4の図を参照

LSTMのさらなる改善

- LSTMレイヤの多層化
  - LSTMレイヤを多層化することで精度が上昇

- Dropout
  - LSTMの多層化することで,複雑な依存関係が学習可能
    - 過学習を起こしてしまう
      - Dropoutを行うことで過学習を抑制