Recurrent Batch Normalization

12056 外山洋太

目次

- 概要
- 導入
  - LSTM
  - バッチ正規化
- 提案手法
- 実験
- 結論

概要

- 題名
  - Recurrent Batch Normalization
- 著者
  - Tim Cooijmans, Nicolas Ballas, Cesar Laurent, Caglar Gulcehre
- LSTMにバッチ正規化を導入
  - 学習の高速化
  - 高い汎化性能
- 以前の研究では導入に失敗
  - パラメータの初期化が悪かった

LSTM

- Long Short-Term Memory
  - x: 入力
  - h: 出力
  - c: メモリーセル
  - σ: シグモイド関数（ゲートに対応）
  - forget, input, outputゲートがある

$$\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}$$

$$\begin{aligned} \mathbf{c}_t &= \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t) \\ \mathbf{h}_t &= \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\mathbf{c}_t), \end{aligned}$$

導入

バッチ正規化

- バッチ毎に各インスタンスに対応する値（今回はベクトル）を正規化
- 共変量シフトを抑える
- 共変量シフト
  - 訓練データx_{train}の確率分布$p_{train}(x_{train})$とテストデータx_{test}の確率分布$p_{test}(x_{test})$が異なる問題
    → 入力となる値を正規化して対処
- 深いニューラルネットワークでは、隠れ層に対する内部共変量シフトも発生

導入

バッチ正規化

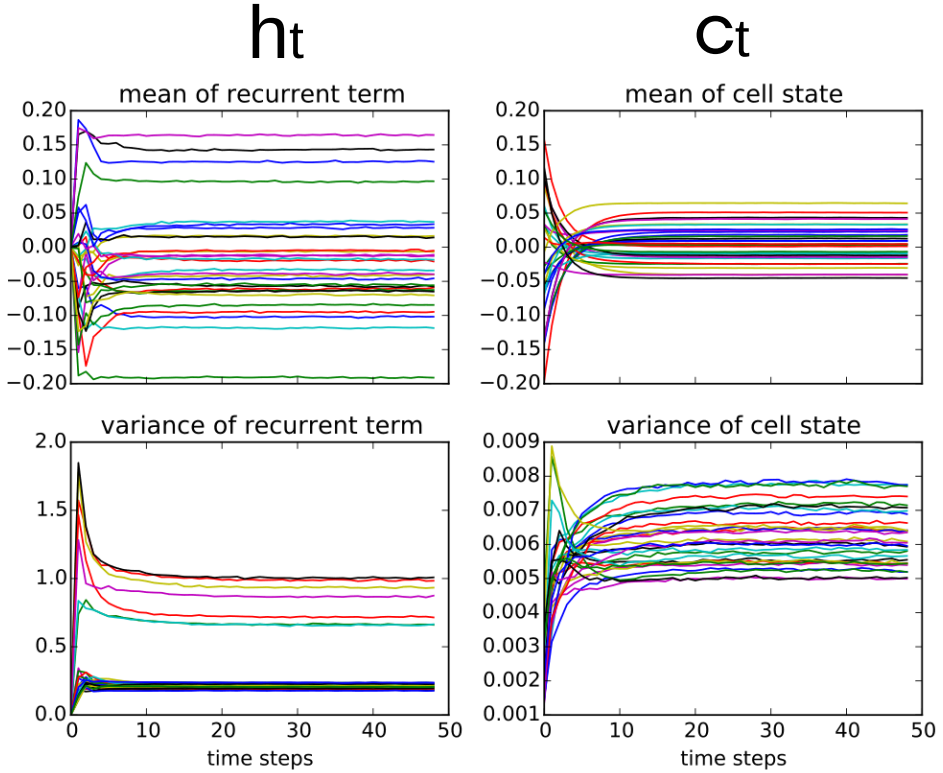
- はじめに入力値hをバッチ単位で正規化（平均0、分散1）
- 正規化後に最適な平均と分散に確率分布を直す
- β: 平均値を示すパラメータ
- γ: 標準偏差を示すパラメータ
- ε: 正則化のためのハイパーパラメータ

$$\mathrm{BN}(\mathbf{h};\gamma,\beta) = \boxed{\beta} + \boxed{\gamma}\frac{\mathbf{h} - \widehat{\mathbb{E}}(\mathbf{h})}{\sqrt{\widehat{\mathrm{Var}}(\mathbf{h}) + \epsilon}}$$

提案手法

β、γの時間依存性

- β:ある隠れ層の最適な平均値
- γ:ある隠れ層の最適な標準偏差



∴ β、γは時間に依存させたほうが良い → β_t、γ_t

提案手法

- BN-LSTM（Batch Normalization - LSTM）
- LSTMにバッチ正規化を導入し、パラメタを改変
- 昔の自分の出力と入力、メモリーセルの出力に対してバッチ正規化

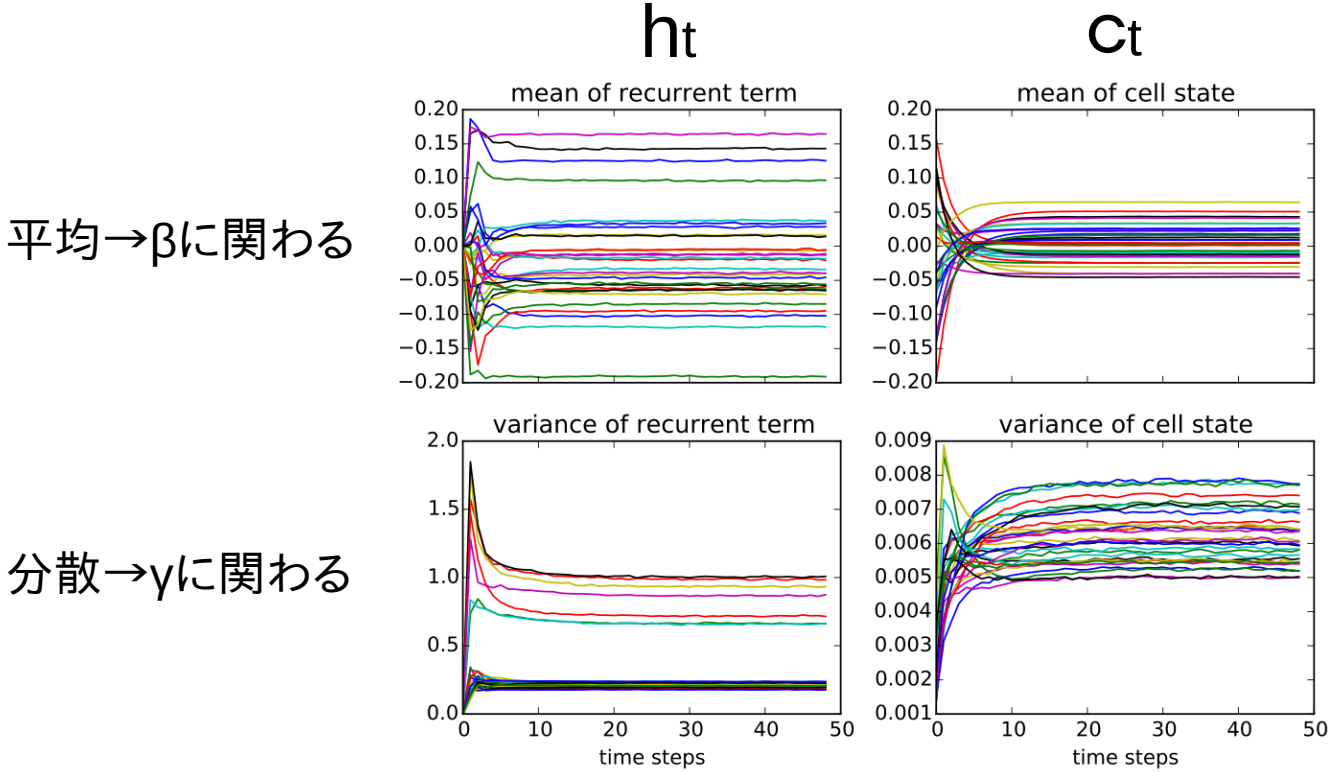
$$
\begin{pmatrix} \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \boxed{\mathrm{BN}(\mathbf{W}_h \mathbf{h}_{t-1}; \gamma_h, \beta_h)} + \boxed{\mathrm{BN}(\mathbf{W}_x \mathbf{x}_t; \gamma_x, \beta_x)} + \mathbf{b}
$$

$$
\begin{aligned}
\mathbf{c}_t &= \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} + \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t) \\
\mathbf{h}_t &= \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\boxed{\mathrm{BN}(\mathbf{c}_t; \gamma_c, \beta_c)})
\end{aligned}
$$

提案手法

β、γの時間依存性

- β:ある隠れ層の最適な平均値
- γ:ある隠れ層の最適な標準偏差



∴ β、γは時間に依存させたほうが良い → β_t、γ_t

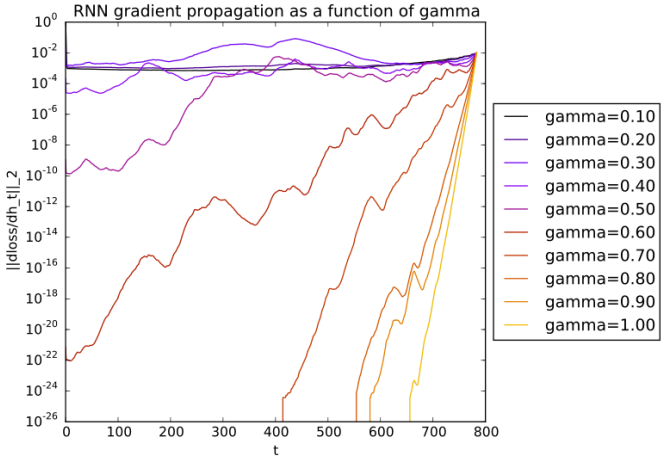
提案手法

γの初期化と勾配の伝播

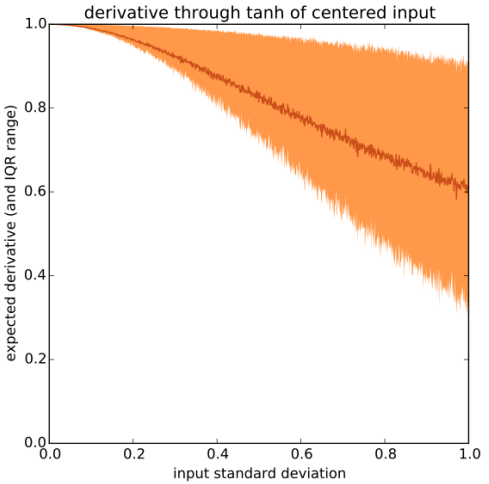
- γが大きすぎると勾配が伝播中に0に収束
- γが小さすぎると訓練が不安定に

  → γは0.1程度で初期化

- 従来研究の失敗はこれができていなかったため



(a) Gradient flow through a batch-normalized tanh RNN as a function of γ. High variance causes vanishing gradient.



(b) Empirical expected derivative of tanh nonlinearity as a function of input variance. High variance causes saturation, which decreases the expected derivative.

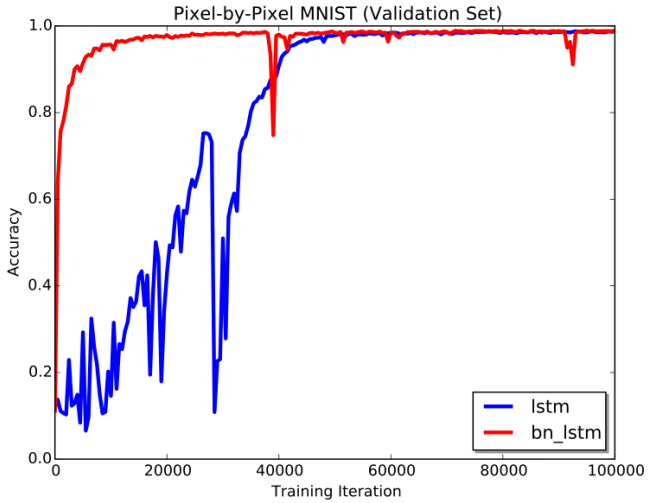
実験

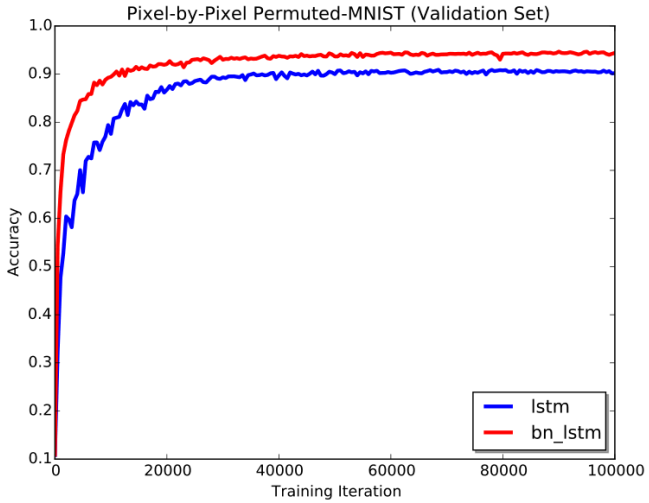
- Sequential MNIST
- 文字レベル言語モデル
  - Character-level Penn TreeBank
  - Text8
- 質問応答
  - Attensive Reader Model

Sequential MNIST

- MNIST
  - 数字画像のピクセルを1つ1つ順に読んでいき、その数字を予測
- pMNIST
  - 画像のピクセルの配置をランダムにしたもの

Model	MNIST	pMNIST
TANH-RNN [15]	35.0	35.0
iRNN [15]	97.0	82.0
uRNN [2]	95.1	91.4
sTANH-RNN [28]	98.1	94.0
LSTM (ours)	98.9	90.2
BN-LSTM (ours)	**99.0**	**95.4**





文字レベル言語モデル

- Character-level Penn TreeBank
- Text8
  - Wikipediaを元にした英語コーパス
- bits-per-character：ave($-\log_2 \Pr(x_{t+1} \mid y_t)$)

Model	Penn Treebank
LSTM [7]	1.26[a]
HF-MRNN [19]	1.41
Norm-stabilized LSTM [13]	1.39
ME n-gram [19]	1.37
LSTM (ours)	1.38
BN-LSTM (ours)	1.32

Table 2: Bits-per-character on the Penn Treebank test sequence.

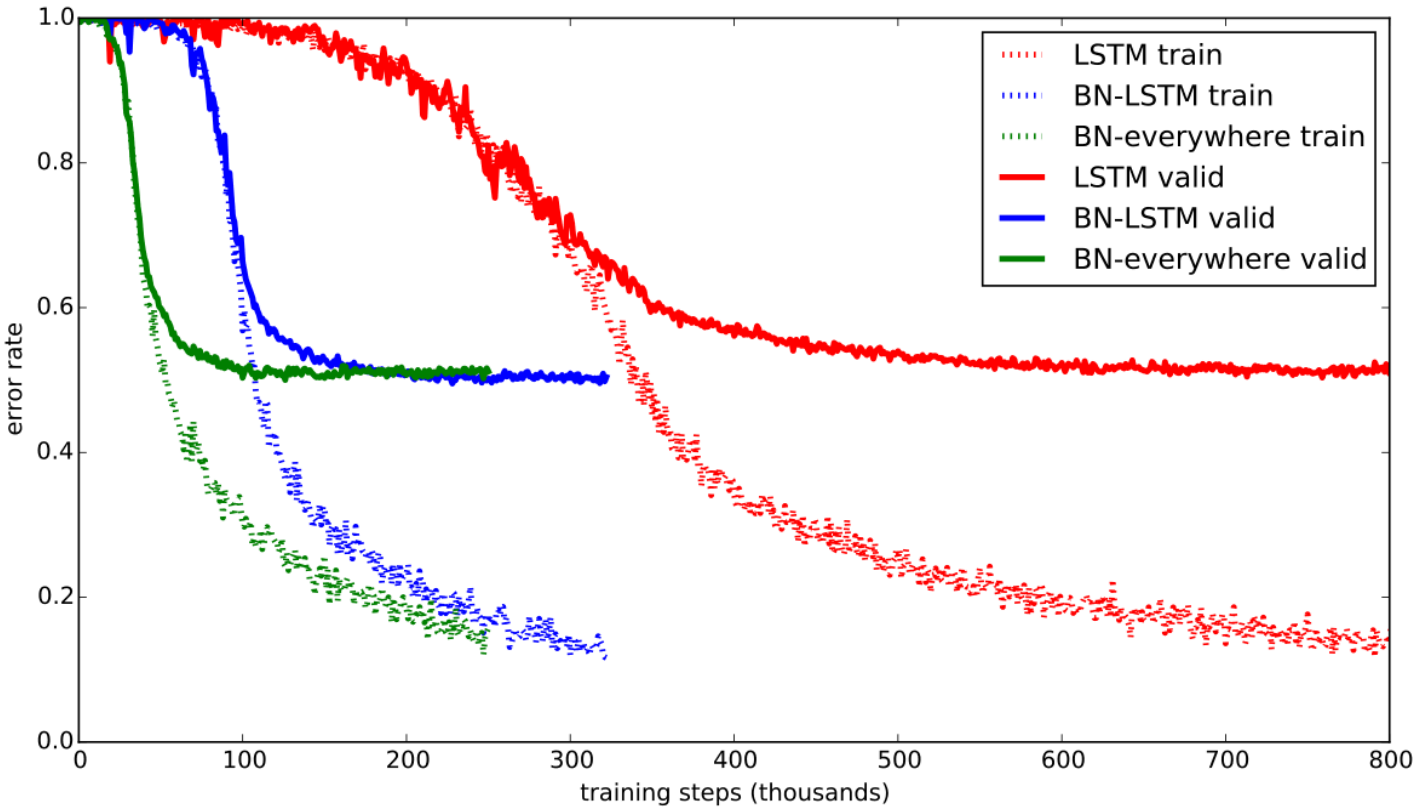
[a] Our performance does not directly compare against [7] as they use a different dataset split.

Model	text8
td-LSTM [28]	1.63
HF-MRNN [19]	1.54
skipping RNN [21]	1.48
BN-LSTM (ours)	**1.39**

Table 3: Bits-per-character on the text8 test sequence.

Attensive Reader Model

- RNNとAttentionを用いた質問応答のモデル
- BN-LSTM: LSTMをBN-LSTMに置き換え
- BN-everywhere: tanhの活性化関数に入る全ての項を正規化



結論

- LSTMにバッチ正規化を導入
  - 学習の高速化
  - 高い汎化性能
- 従来研究はパラメータの初期化がダメだっただけ