Normalization Helps Training of Quantized LSTM

Lu Hou[1], Jinhua Zhu[2], James T. Kwok[1], Fei Gao[3], Tao Qin[3], Tie-Yan Liu[3]

[1]Hong Kong University of Science and Technology, [2]University of Science and Technology of China, [3]Microsoft Research, Beijing, China

[1]{lhouab, jamesk}@cse.ust.hk, [2]teslazhu@mail.ustc.edu.cn, [2]{feiga, taoqin, tyliu}@microsoft.com

Backgroud and Motivation

Quantized LSTM

- BinaryConnect fails (Hou et al., 2017) on LSTM
- BWN, TWN, LAB, LAT perform much better, but sometimes inferior to the full-precision network on some tasks (Ardakani et al. 2019)
- SOTA performance when training binarized/ternarized LSTMs with batch normalization (Ardakani et al. 2019)
- Why does batch normalization work for quantized LSTM?
- Does weight/layer normalization also help?

Recurrence of a LSTM

$$\begin{bmatrix}\mathbf{i}_t\\ \mathbf{f}_t\\ \mathbf{a}_t\\ \mathbf{o}_t\end{bmatrix} = \begin{bmatrix}\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1}\\ \mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}\\ \mathbf{W}_{xa}\mathbf{x}_t + \mathbf{W}_{ha}\mathbf{h}_{t-1}\\ \mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}\end{bmatrix} + \begin{bmatrix}\mathbf{b}_i\\ \mathbf{b}_f\\ \mathbf{b}_a\\ \mathbf{b}_o\end{bmatrix},$$

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

- storage dominated by $\mathbf{W}_{x*}, \mathbf{W}_{h*}$
- computation dominated by $\mathbf{W}_{x*}\mathbf{x}_t + \mathbf{W}_{h*}\mathbf{h}_{t-1}$
- → **quantize** $\mathbf{W}_{x*}, \mathbf{W}_{h*}$

Our Findings

- Quantized LSTM is hard to train due to the exploding gradient problem
- popularly used weight/layer/batch normalization schemes can help stabilize the gradient magnitude in training quantized LSTMs

Exploding Gradient in LSTM

Proposition 1 $\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_{t-1}}\right\| \le \lambda_1\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_t}\right\| + \lambda_2\left\|\frac{\partial\xi_m}{\partial\mathbf{c}_{t+1}}\right\|.$

- $\lambda_1 = \frac{1}{4}\|\mathbf{W}_{hi}\|_2 + \frac{\gamma_1}{4}\|\mathbf{W}_{hf}\|_2 + \|\mathbf{W}_{ha}\|_2 + \frac{1}{4}\|\mathbf{W}_{ho}\|_2$
- $\lambda_2 = \frac{1}{4}\|\mathbf{W}_{hi}\|_2 + \frac{\gamma_1}{4}\|\mathbf{W}_{hf}\|_2 + \|\mathbf{W}_{ha}\|_2$

Corollary 1 *When $\lambda_2 = 0$, a necessary condition for exploding gradients in the LSTM is $\lambda_1 > 1$.*

- empirically, λ_2 is rarely zero
- upper bound of $\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_{t-1}}\right\|$ even larger, and gradient explode even more easily

Quantization → Gradient Easier to Explode

λ_1, λ_2 in the upper bound related to the spectral norm of weight matrix

- larger spectral norm → more easily to explode

Spectral norm of quantized matrix

- **For any $\mathbf{W}\in\{-1,+1\}^{d\times d}$, $\|\mathbf{W}\|_2 \ge \sqrt{d}$.**
- **For any $\mathbf{W}\in\{-B_k,\dots,-B_1,B_0,B_1,\dots,B_k\}^{d\times d}$ where $0=B_0<B_1<\cdots<B_k$, $\|\mathbf{W}\|_2 \ge (1-s)B_1\sqrt{d}$, where s is the sparsity of $\mathbf{W}$.**

the gradient is more easily to explode

- when d is large
- for binarization than ternarization

Normalized LSTM

- Apply normalization as $\mathcal{N}(\mathbf{W}_{x*}\mathbf{x}_t)$ and $\mathcal{N}(\mathbf{W}_{h*}\mathbf{h}_{t-1})$
- $\mathcal{N}$ can be weight, layer or batch normalization

$$\begin{bmatrix}\tilde{\mathbf{i}}_t\\ \tilde{\mathbf{f}}_t\\ \tilde{\mathbf{a}}_t\\ \tilde{\mathbf{o}}_t\end{bmatrix} = \begin{bmatrix}\mathcal{N}(\mathbf{W}_{xi}\mathbf{x}_t) + \mathcal{N}(\mathbf{W}_{hi}\mathbf{h}_{t-1})\\ \mathcal{N}(\mathbf{W}_{xf}\mathbf{x}_t) + \mathcal{N}(\mathbf{W}_{hf}\mathbf{h}_{t-1})\\ \mathcal{N}(\mathbf{W}_{xa}\mathbf{x}_t) + \mathcal{N}(\mathbf{W}_{ha}\mathbf{h}_{t-1})\\ \mathcal{N}(\mathbf{W}_{xo}\mathbf{x}_t) + \mathcal{N}(\mathbf{W}_{ho}\mathbf{h}_{t-1})\end{bmatrix} + \begin{bmatrix}\mathbf{b}_i\\ \mathbf{b}_f\\ \mathbf{b}_a\\ \mathbf{b}_o\end{bmatrix},$$

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

Weight Normalization:

decouples length and direction of the weight vector

- each row $\mathbf{W}_{j,:}$ of $\mathbf{W}$ ($\mathbf{W}_{h*}$ or $\mathbf{W}_{x*}$) is separately normalized as

$$\mathcal{WN}(\mathbf{W}_{j,:}\mathbf{x}) = g_j\frac{\mathbf{W}_{j,:}}{\|\mathbf{W}_{j,:}\|}\mathbf{x}$$

- $g_* = \max_{1\le j\le d} g_j$; $\mathbf{D}_* = \mathrm{diag}([\|(\mathbf{W}_{h*})_{1,:}\|, \|(\mathbf{W}_{h*})_{2,:}\|, \dots, \|(\mathbf{W}_{h*})_{d,:}\|]^\top)$

Proposition 2 *With weight normalization,*

$$\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_{t-1}}\right\| \le \left(\frac{g_i}{4}\|\mathbf{D}_i^{-1}\mathbf{W}_{hi}\|_2 + \frac{\gamma_1 g_f}{4}\|\mathbf{D}_f^{-1}\mathbf{W}_{hf}\|_2 + g_a\|\mathbf{D}_a^{-1}\mathbf{W}_{ha}\|_2 + \frac{g_o}{4}\|\mathbf{D}_o^{-1}\mathbf{W}_{ho}\|_2\right)\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_t}\right\| + \left(\frac{g_i}{4}\|\mathbf{D}_i^{-1}\mathbf{W}_{hi}\|_2 + \frac{\gamma_1 g_f}{4}\|\mathbf{D}_f^{-1}\mathbf{W}_{hf}\|_2 + g_a\|\mathbf{D}_a^{-1}\mathbf{W}_{ha}\|_2\right)\left\|\frac{\partial\xi_m}{\partial\mathbf{c}_{t+1}}\right\|.$$

- if $\mathbf{W}_{h*}$ is scaled by a factor α, $\mathbf{D}_*$ will also be scaled by α → $\mathbf{D}_*^{-1}\mathbf{W}_{h*}$ not affected.

Layer Normalization

normalizes activities in each layer

- input $\mathbf{x}\in\mathbb{R}^d$ ($\mathbf{W}_{x*}\mathbf{x}_t$ or $\mathbf{W}_{h*}\mathbf{h}_{t-1}$) with mean μ and standard deviation σ

$$\mathbf{y} = \mathcal{LN}(\mathbf{x}) = \mathbf{g}\odot\mathbf{z} + \mathbf{b}, \text{where } \mathbf{z} = (\mathbf{x}-\mu\mathbf{1})/\sigma$$

- for $\mathcal{LN}(\mathbf{W}_{h*}\mathbf{h}_{t-1})$: $g_* = g_k, \sigma_* = \sigma_k$, where $k = \arg\max_{1\le j\le d} g_j$

Proposition 3 *With layer normalization,*

$$\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_{t-1}}\right\| \le \left(\frac{1}{4}\frac{g_i}{\sigma_i}\|\mathbf{W}_{hi}\|_2 + \frac{\gamma_1}{4}\frac{g_f}{\sigma_f}\|\mathbf{W}_{hf}\|_2 + \frac{g_a}{\sigma_a}\|\mathbf{W}_{ha}\|_2 + \frac{1}{4}\frac{g_o}{\sigma_o}\|\mathbf{W}_{ho}\|_2\right)\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_t}\right\| + \left(\frac{1}{4}\frac{g_i}{\sigma_i}\|\mathbf{W}_{hi}\|_2 + \frac{\gamma_1}{4}\frac{g_f}{\sigma_f}\|\mathbf{W}_{hf}\|_2 + \frac{g_a}{\sigma_a}\|\mathbf{W}_{ha}\|_2\right)\left\|\frac{\partial\xi_m}{\partial\mathbf{c}_{t+1}}\right\|.$$

- if the elements of $\mathbf{W}_{h*}$ grow twice as large, the corresponding σ_* will be twice as large

Batch Normalization

operates on a minibatch (N samples in a batch)

- $\mathbf{H}_t = [\mathbf{h}_t^1,\dots,\mathbf{h}_t^N]^\top\in\mathbb{R}^{N\times d}$; $\mathbf{X}_t = [\mathbf{x}_t^1,\dots,\mathbf{x}_t^N]^\top\in\mathbb{R}^{N\times r}$
- input $\mathbf{X}\in\mathbb{R}^{N\times d}$ ($\mathbf{X}_t\mathbf{W}_{x*}^\top$ or $\mathbf{H}_{t-1}\mathbf{W}_{h*}^\top$), with mean μ_j and std σ_j for the jth column

$$\mathbf{y}_{:,j} = \mathcal{BN}(\mathbf{X}_{:,j}) = g_j\frac{\mathbf{X}_{:,j} - \mu_j\mathbf{1}}{\sigma_j} + b_j\mathbf{1}$$

- for $\mathcal{BN}(\mathbf{H}_{t-1}\mathbf{W}_{h*}^\top)$: $(\sigma_*, g_*) = \arg\max_{1\le j\le d}\frac{g_j}{\sigma_j}$

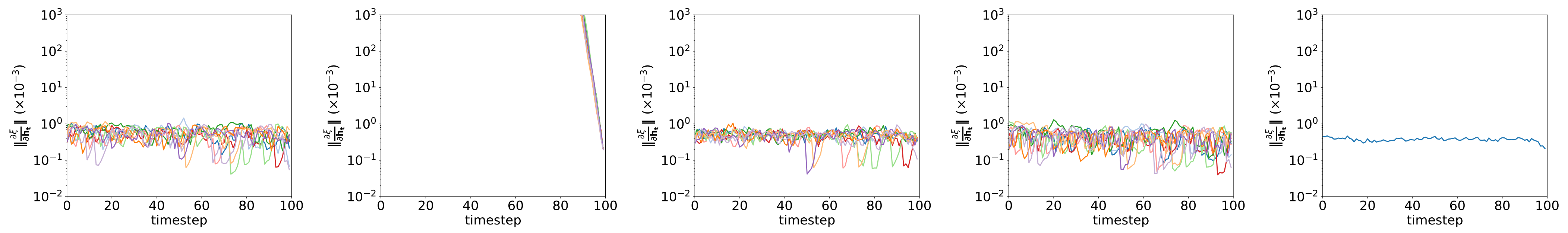
Proposition 4 *With batch normalization,*

$$\sum_{k=1}^N\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_{t-1}^k}\right\|^2 \le \left(\frac{1}{2}\frac{g_i^2}{\sigma_i^2}\|\mathbf{W}_{hi}\|_2^2 + \frac{\gamma_2^2}{2}\frac{g_f^2}{\sigma_f^2}\|\mathbf{W}_{hf}\|_2^2 + 8\frac{g_a^2}{\sigma_a^2}\|\mathbf{W}_{ha}\|_2^2 + \frac{1}{4}\frac{g_o^2}{\sigma_o^2}\|\mathbf{W}_{ho}\|_2^2\right)\sum_{k=1}^N\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_t^k}\right\|^2 + \left(\frac{1}{2}\frac{g_i^2}{\sigma_i^2}\|\mathbf{W}_{hi}\|_2^2 + \frac{\gamma_2^2}{2}\frac{g_f^2}{\sigma_f^2}\|\mathbf{W}_{hf}\|_2^2 + 8\frac{g_a^2}{\sigma_a^2}\|\mathbf{W}_{ha}\|_2^2\right)\sum_{k=1}^N\left\|\frac{\partial\xi_m}{\partial\mathbf{c}_{t+1}^k}\right\|^2.$$

- if the elements of $\mathbf{W}_{h*}$ grow twice as large, the corresponding σ_* will be twice as large

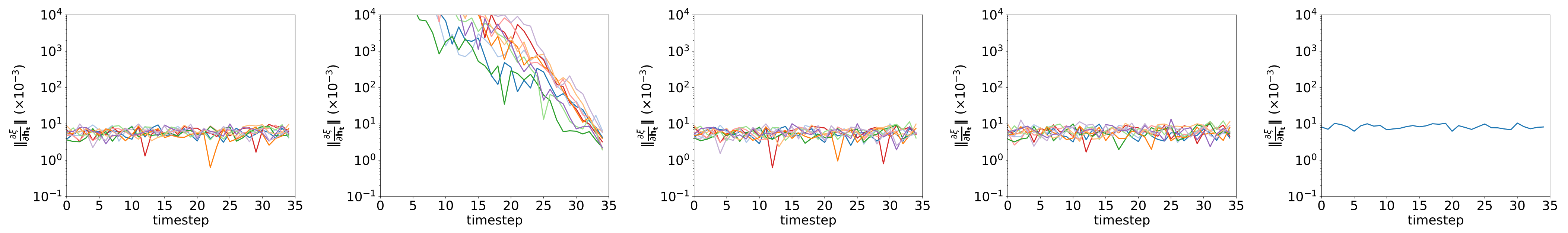
Observations

Figure 1: Gradient norms of character-level language modeling on *Penn Treebank* dataset.



(a) Full-precision. (b) BC. (c) BC+WN. (d) BC+LN. (e) BC+BN (shared).

Figure 2: Gradient norms of word-level language modeling on *Penn Treebank* dataset.



(a) Full-precision. (b) BC. (c) BC+WN. (d) BC+LN. (e) BC+BN (shared).

- **By normalization , $\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_t}\right\|$ not affected by the possibly large scaling of the weight caused by quantization → the exploding gradient problem can be alleviated**

Experiments

Character-level Language Modeling

- Bits Per Character (BPC) and size (in KB) of 1-layer LSTM.

precision	quantization	normalization	*War and Peace* BPC	*War and Peace* size	*Penn Treebank* BPC	*Penn Treebank* size	*Text8* BPC	*Text8* size
full	-	-	1.72	4800	1.45	4504	1.46	63375
		weight	1.73	4816	1.45	4520	1.48	63438
		layer	**1.69**	4832	**1.43**	4536	**1.45**	63500
		batch (shared)	1.72	4864	1.45	4568	1.46	63625
		batch (separate)	1.72	8032	1.45	7736	1.46	86000
1-bit	SBN	batch (separate)	1.78	3794	1.60	3785	1.54	27464
	BinaryConnect	-	4.24	158	2.51	149	N/A	2011
		weight	1.74	174	1.50	165	1.50	2073
		layer	**1.69**	**190**	**1.49**	**181**	**1.47**	**2136**
		batch (shared)	1.72	222	1.51	213	**1.47**	**2261**
		batch (separate)	1.72	3390	1.50	3381	1.48	24636
2-bit	STN	batch (separate)	1.72	3944	1.60	3521	1.51	15303
	TerConnect	-	6.35	308	5.84	289	N/A	3990
		weight	1.72	324	**1.42**	**305**	**1.42**	**4053**
		layer	**1.67**	**340**	1.43	321	1.44	4115
		batch (shared)	1.70	372	1.44	353	1.44	4240
		batch (separate)	1.71	3540	1.45	3521	1.44	26615

Word-level Language Modeling

- Test Perplexity and size (in KB) of 1-layer LSTM with d hidden units

precision	quantlization	normalization	$d=300$ PPL	$d=300$ size	$d=650$ PPL	$d=650$ size
full		-	91.5	2817	87.6	13213
1-bit	SBN	batch (separate)	92.2	852	87.2	2068
	BinaryConnect -	-	8247.4	93	1244.2	423
		weight	**87.6**	**102**	84.8	443
		layer	89.4	111	**82.3**	**463**
		batch (shared)	92.4	130	84.8	504
		batch(separate)	91.9	767	85.6	1885
	alternating LSTM	-	103.1	180	-	-
2-bit	STN	batch (separate)	90.7	940	86.1	2481
	TerConnect	-	113.8	180	113.8	835
		weight	**86.5**	**190**	84.9	856
		layer	88.2	199	**82.5**	**876**
		batch (shared)	90.6	218	85.8	917
		batch (separate)	91.6	855	86.5	2298

Observations

- Vanilla BinaryConnect and TerConnect fail, but normalized versions work
- Normalized quantized LSTM is comparable to the full-precision baseline
- Applying weight/layer/batch (shared) normalization perform similarly or better than SBN and STN (Ardakani et al., 2019), while being much smaller

More experiments in the paper !

Conclusion

- WHY: Quantization tends to increase spectral norm of weights in LSTM, making the exploding gradient problem much more severe than its full-precision counterpart.
- HOW: By using normalization, backpropagation of $\left\|\frac{\partial\xi_m}{\partial\mathbf{h}_t}\right\|$ in the quantized LSTM is not affected by the possibly large scaling of the weight matrix caused by quantization, and the exploding gradient problem can be alleviated.
- CODE: `https://github.com/houlu369/Normalized-Quantized-LSTM`