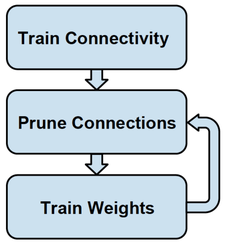
Learning Intrinsic Sparse Structures within Long Short-Term Memory

Wei Wen[1], Yuxiong He[2], Samyam Rajbhandari[2], Minjia Zhang[2], Wenhan Wang[2], Fang Liu[2], Bin Hu[2], Yiran Chen[1] & Hai Li[1]
Duke University[1], Microsoft[2]
{wei.wen,yiran.chen,hai.li}@duke.edu, {yuxhe,samyamr,minjiaz,wenhanw,fangliu,binhu}@microsoft.com

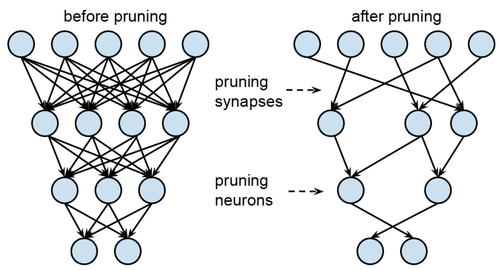
Background & Motivation

Goal of this work: Accelerate the inference of Recurrent Neural Networks (RNNs) by learning structurally sparse connections.

S. Han et. al. NIPS 2015

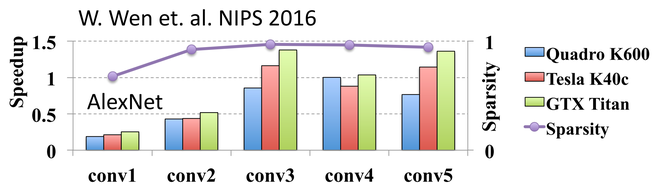


Good speedup in customized hardware [S. Han et. al. ISCA 2016]



Limited speedup in general purpose platforms (e.g., CPUs and GPUs)

W. Wen et. al. NIPS 2016

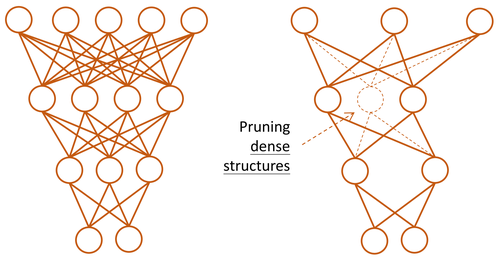


Reason: Hardware is designed for dense computation, but sparse weights break the regularity and locality in data

This work

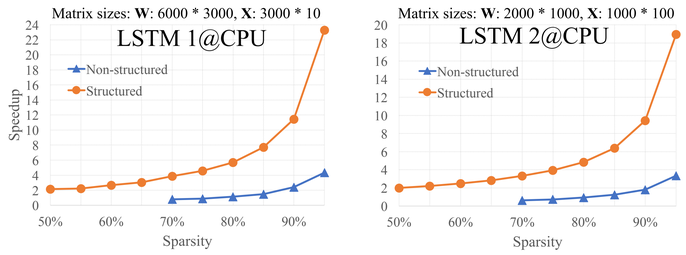
Train weights with Group Lasso Regularization

One-shot pruning from scratch (with the same epochs)



Pruning dense structures

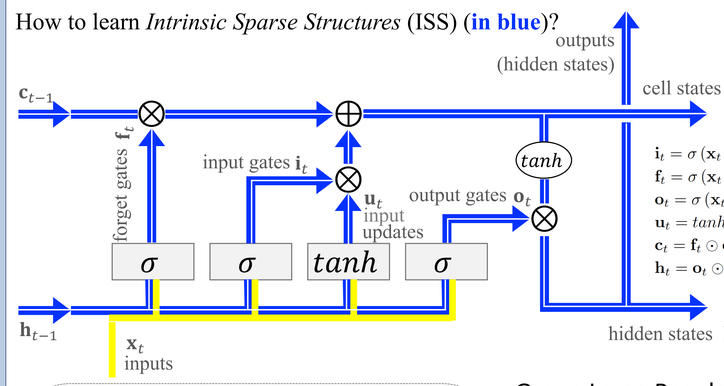
Structured Sparsity

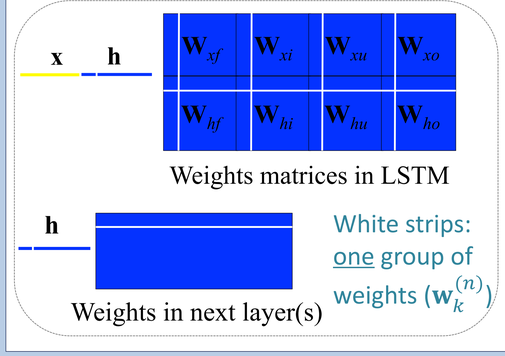


Structured: Columns and rows are removed in weight matrices

Learning Intrinsic Sparse (Hidden) Structures in RNNs

How to learn *Intrinsic Sparse Structures* (ISS) (**in blue**)?





Group Lasso Regularization:

$$R(\mathbf{w}) = \sum_{n=1}^{N} \sum_{k=1}^{K^{(n)}} \left\| \mathbf{w}_k^{(n)} \right\|_2$$

Additional Gradient in SGD :

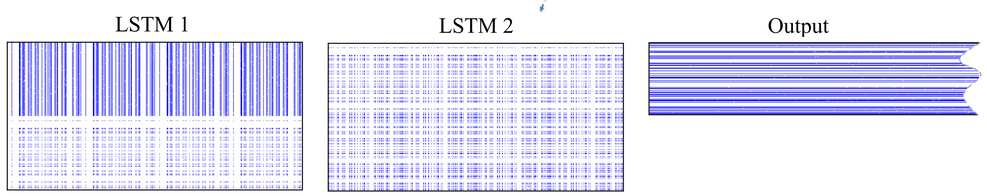
$$\mathbf{w}_k^{(n)} \leftarrow \mathbf{w}_k^{(n)} - \eta \cdot \left(\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}_k^{(n)}} + \lambda \cdot \frac{\mathbf{w}_k^{(n)}}{\left\| \mathbf{w}_k^{(n)} \right\|_2} \right)$$

Experiments

Baseline: two stacked LSTMs for Language modeling (Zaremba et al. (2014))

Method	Dropout keep ratio	Perplexity (validate, test)	ISS # in (1st , 2nd) LSTM	Weight #	Total time*	Speedup	Mult-add reduction†
baseline	0.35	(82.57, 78.57)	(1500, 1500)	66.0M	157.0ms	1.00×	1.00×
ISS	0.60	(82.59, 78.65)	(373, 315)	21.8M	14.82ms	10.59×	7.48×
		(80.24, 76.03)	(381, 535)	25.2M	22.11ms	7.10×	5.01×
direct design	0.55	(90.31, 85.66)	(373, 315)	21.8M	14.82ms	10.59×	7.48×

* Measured with 10 batch size and 30 unrolled steps.
† The reduction of multiplication-add operations in matrix multiplication. Defined as (original Mult-add)/(left Mult-add)



- Training: (1500-373) and (1500-315) hidden dimensions can be removed w/o using perplexity
- Deployment:
  - Construct stacked LSTMs with hidden sizes of 373 in *LSTM 1* and 315 in *LSTM 2*
  - Use learned weights to initialize the constructed small LSTMs
- Inference: Dense computation with smaller hidden sizes

Baseline: Recurrent Highway Networks (RHNs) (Zilly et al. (2017))

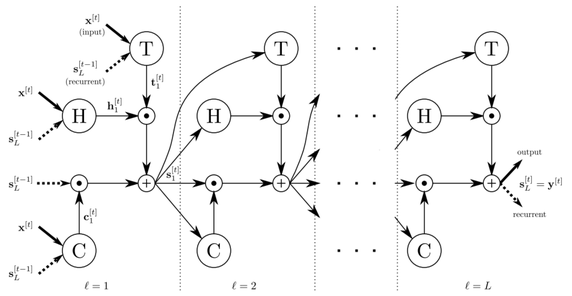


Table 2: Learning ISS sparsity from scratch in RHNs.

Method	λ	Perplexity (validate, test)	RHN width	Parameter #
baseline	0.0	(67.9, 65.4)	830	23.5M
ISS*	0.004	(67.5, 65.0)	726	18.9M
ISS*	0.005	**(68.1, 65.4)**	**517**	**11.1M**
ISS*	0.006	(70.3, 67.7)	403	7.6M
ISS*	0.007	(74.5, 71.2)	328	5.7M

* All dropout ratios are multiplied by 0.6×.

Baseline:
BiDAF model (LSTMs + attention) on SQuAD Question Answering (Seo et al. (2017))

Table 4: Remaining ISS components in BiDAF by training from scratch.

EM	F1	ModFwd1	ModBwd1	ModFwd2	ModBwd2	OutFwd	OutBwd	weight #	Total time*
67.98	77.85	100	100	100	100	100	100	2.69M	6.20ms
67.36	77.16	87	81	87	92	74	96	2.29M	5.83ms
66.32	76.22	51	33	42	58	37	26	1.17M	4.46ms
65.36	75.78	20	33	40	38	31	16	0.95M	3.59ms
64.60	74.99	23	22	35	35	25	14	0.88M	2.74ms

* Measured with batch size 1.

Histogram of vector lengths of ISS weight groups in BiDAF