Simple Recurrent Units for Highly Parallelizable Recurrence

Tao Lei tao@asapp.com

Yu Zhang ngyuzh@google.com

Sida I. Wang sidaw@cs.princeton.edu

Hui Dai hd@asapp.com

Yoav Artzi yoav@cs.cornell.edu

Motivation

Recurrent networks scale poorly

- The computation of $\mathbf{c}_t$ is suspended until $\mathbf{c}_{t-1}$ becomes completely available.
- This sequential dependency breaks computation into a successive execution of relative small computation for each $\mathbf{c}_t$.
- As a result, RNNs cannot utilize the full parallelization power of hardware and runs much slower than attention and convolution.

Contribution

Simple Recurrent Unit (SRU), a recurrent unit that is no longer a parallelization bottleneck.

- exhibits the same parallelism as convolution and attention.
- retrains modeling capacity as LSTM and GRU etc.

Open source code: https://github.com/taolei87/sru

SRU

Basic architecture

SRU involves relatively few computation, which decomposes into two sub-components:

(i) light gated recurrence

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{v}_f \odot \mathbf{c}_{t-1} + \mathbf{b}_f\right)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1-\mathbf{f}_t) \odot (\mathbf{W}\mathbf{x}_t)$$

(ii) highway connection

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r\mathbf{x}_t + \mathbf{v}_r \odot \mathbf{c}_{t-1} + \mathbf{b}_r\right)$$
$$\mathbf{h}_t = \mathbf{r}_t \odot \mathbf{c}_t + (1-\mathbf{r}_t) \odot \mathbf{x}_t$$

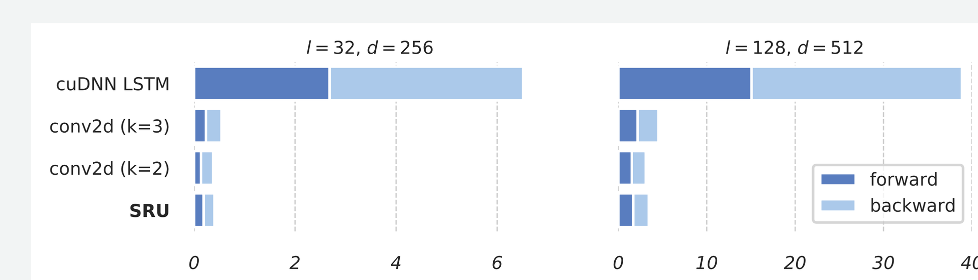
We use element-wise multiplication (e.g. $\mathbf{v}_f \odot \mathbf{c}_{t-1}$) for hidden-to-hidden connection.

Optimizations

The architecture enables two optimizations that achieve significant speed-up over traditional RNNs: (i) group matrix multiplications across all steps into one single multiplication, and (ii) write a custom function to perform the element-wise operations for computation intensity.

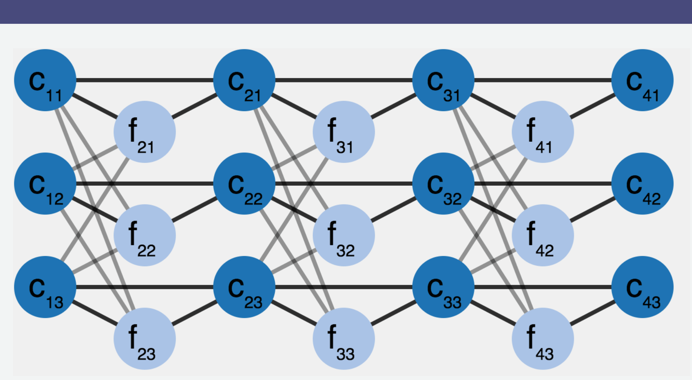
$$\mathbf{U}^\top = \begin{pmatrix} \mathbf{W} \\ \mathbf{W}_f \\ \mathbf{W}_r \end{pmatrix} [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L]$$

Grouped multiplications

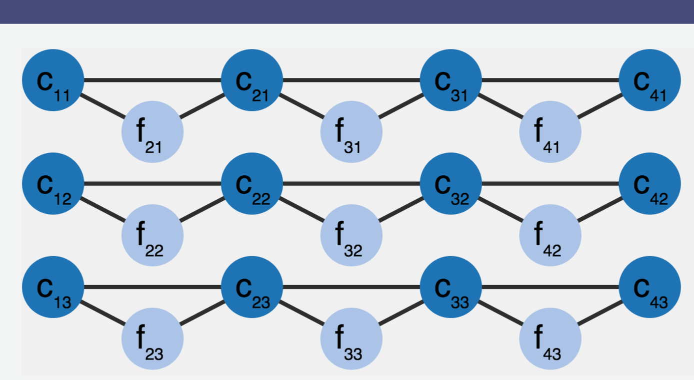


Computation time

SRU vs. LSTM



LSTM: all dimensions of $\mathbf{c}_t$ is needed to compute each of $\mathbf{f}_{t+1}$.



SRU: only 1 dimension of $\mathbf{c}_t$ is needed to compute each of $\mathbf{f}_{t+1}$.

While LSTM also uses a *light gated recurrence* from $\mathbf{g}_t$ to $\mathbf{c}_t$, it uses a full recurrence from $\mathbf{c}_t$ to $\mathbf{g}_{t+1}$ which intuitively seems wasteful.

$g \in \{f, r, i, o\}$ is a gate,

full: $\mathbf{g}_t = \sigma(\mathbf{W}_g\mathbf{x}_t + \boxed{\mathbf{V}_g\ \ }\mathbf{c}_{t-1} + \mathbf{b}_g)$

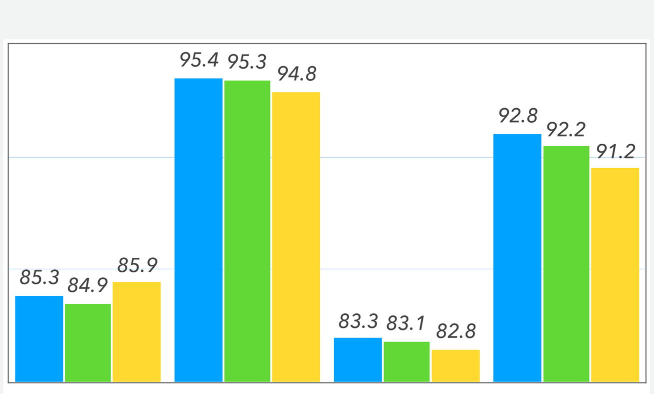
light: $\mathbf{g}_t = \sigma(\mathbf{W}_g\mathbf{x}_t + \boxed{\mathbf{v}_g \odot}\mathbf{c}_{t-1} + \mathbf{b}_g)$

Standard NN uses matrix multiplications to stack layers. SRU uses highway connections shown to be effective in ResNet/highway networks.

Results

Ablation analysis

Successively disable components in SRU to confirm the impact of our design choices.



Comparison between the full SRU (left), the variant without element-wise hidden-to-hidden connections (middle) and the variant without highway connection on classification datasets

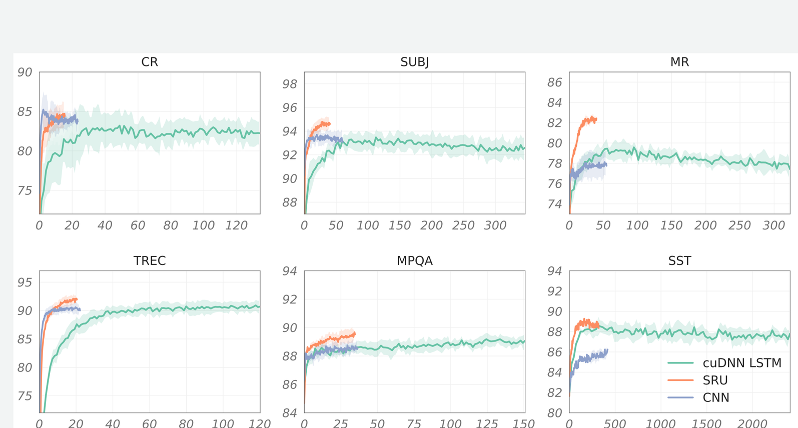
Model	4 layers	6 layers
SRU (full)	70.7	71.4
– remove $\mathbf{v} \odot \mathbf{c}_{t-1}$	70.6	71.1
– remove α-scaling	70.3	71.0
– remove highway	69.4	69.1

Comparison on SQuAD between the full SRU and variants by successively removing parts of the architecture.

Classification

Tested on 6 sentence classification benchmarks. SRU operates 5-9x faster than cuDNN LSTM, achieving on par or better results than various baselines.

Model	Size	CR	SUBJ	MR	TREC	MPQA	SST
Best reported results:							
Wang and Manning (2013)		82.1	93.6	79.1	-	86.3	-
Kalchbrenner et al. (2014)		-	-	-	93.0	-	86.8
Kim (2014)		85.0	93.4	81.5	93.6	89.6	88.1
Zhang and Wallace (2017)		84.7	93.7	81.7	91.6	89.6	85.5
Zhao et al. (2015)		86.3	**95.5**	**83.1**	92.4	**93.3**	-
Our setup (default Adam, fixed word embeddings):							
CNN	360k	83.1±1.6	92.7±0.9	78.9±1.3	93.2±0.8	89.2±0.8	85.1±0.6
LSTM	352k	82.7±1.9	92.6±0.8	79.8±1.3	93.4±0.9	89.4±0.7	88.1±0.8
QRNN (k=1)	165k	83.5±1.9	93.4±0.6	82.0±1.0	92.5±0.5	90.2±0.7	88.2±0.4
QRNN (k=1) + highway	204k	84.0±1.9	93.4±0.8	82.1±1.2	93.2±0.6	89.6±1.2	88.9±0.2
SRU (2 layers)	204k	84.9±1.6	93.5±0.6	82.3±1.2	94.0±0.5	90.1±0.7	89.2±0.3
SRU (4 layers)	303k	85.9±1.5	93.8±0.6	82.9±1.0	**94.8**±0.5	90.1±0.6	**89.6**±0.5
SRU (8 layers)	502k	**86.4**±1.7	93.7±0.6	**83.1**±1.0	94.7±0.5	90.2±0.8	88.9±0.6



Question answering

Tested on SQuAD benchmark using DrQA (Chen et al. 2017) as the model architecture. SRU exhibit 5x speed-up over LSTM and obtains better EM and F1 scores.

Model	# layers	Size	Dev EM	Dev F1	Time per epoch RNN	Time per epoch Total
LSTM (Chen et al., 2017)	3	4.1m	69.5	78.8	316s	431s
QRNN (k=1) + highway	4	2.4m	70.1 ± 0.1	79.4 ± 0.1	113s	214s
	6	3.2m	70.6 ± 0.1	79.6 ± 0.2	161s	262s
SRU	3	2.0m	**70.2** ± 0.3	**79.3** ± 0.1	58s	159s
SRU	4	2.4m	**70.7** ± 0.1	**79.7** ± 0.1	72s	173s
SRU	6	3.2m	**71.4** ± 0.1	**80.2** ± 0.1	100s	201s

Machine translation

Evaluated on WMT English->German dataset. Compared with Transformer by substituting the feed-forward net with SRU.

Model	# layers	Size	BLEU score Valid	BLEU score Test	Speed (toks/sec)	Hours per epoch
Transformer (base)	6	76m	26.6±0.2 (26.9)	27.6±0.2 (27.9)	20k	2.0
Transformer (+SRU)	4	79m	26.7±0.1 (26.8)	27.8±0.1 (28.3)	**22k**	**1.8**
Transformer (+SRU)	5	90m	**27.1**±0.0 (**27.2**)	**28.3**±0.1 (**28.4**)	19k	2.1