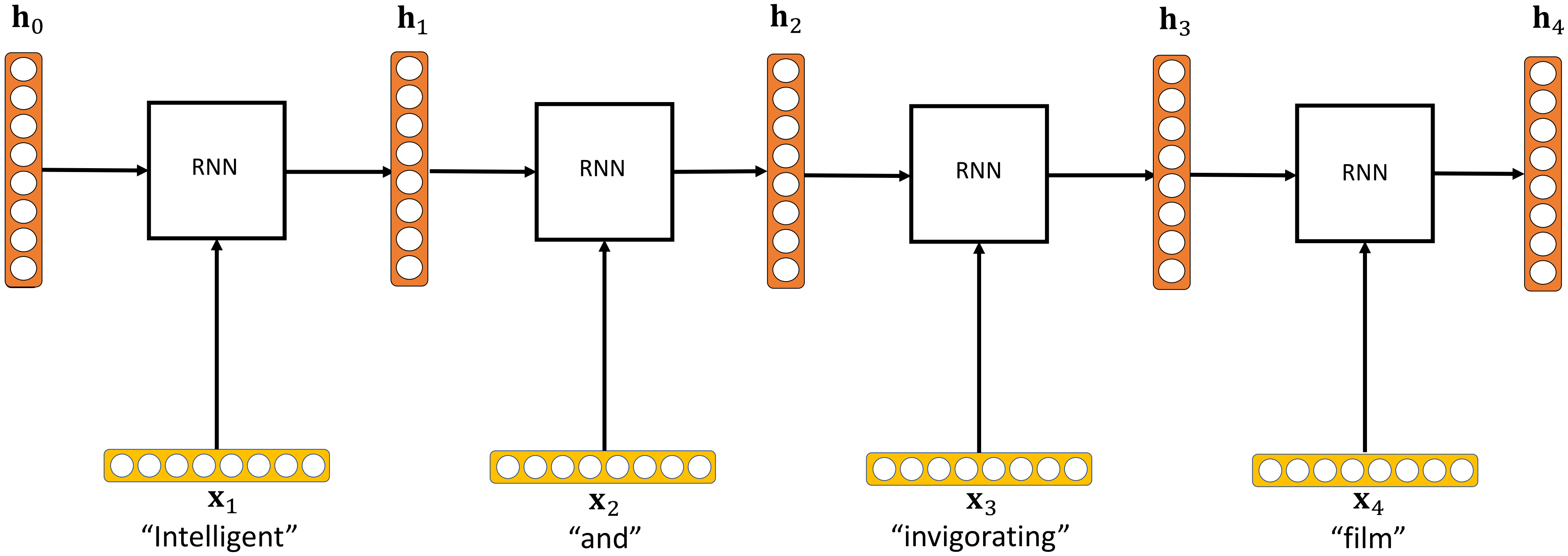
Neural Speed Reading via Skim-RNN

Minjoon Seo[1,2*], Sewon Min[3*], Ali Farhadi[2,4,5], Hannaneh Hajishirzi[2]
NAVER Clova[1], University of Washington[2], Seoul National University[3], Allen Institute for AI[4], XNOR.AI[5]

May 4, 2018

** denotes equal contribution.*

RNN for “reading”



RNN for “reading”

- RNN has become the standard model for reading text
  - Machine Translation
  - Question Answering
  - Classification
  - Syntactic Parsing
  - Natural Language Inference
  - …

RNNs are slow…

- RNNs are slow on GPUs
  - RNNs cannot be parallelized
- CPUs are slow too
  - So many (millions) FLOP in neural networks
- Recent works to replace RNNs:
  - Google’s Attention-based MT (2017)
  - Facebook’s CNN-based MT (2017)

FLOP = Floating-point operations i.e. # of computations

RNNs are slow…

- RNNs are slow on GPUs
  - RNNs cannot be parallelized

- CPUs are slow too
  - So many (millions) FLOP in neural networks

- Recent works to replace RNNs:
  - Google's Transformer (2017)
  - Facebook's CNN-based MT (2017)

Inference speed on CPUs

- CPUs are often more desirable options than GPUs for production
- Small devices often times only have CPUs
- Latency-critical applications
    - CPUs can have lower latency than GPUs.

How can we make RNNs faster on CPUs?

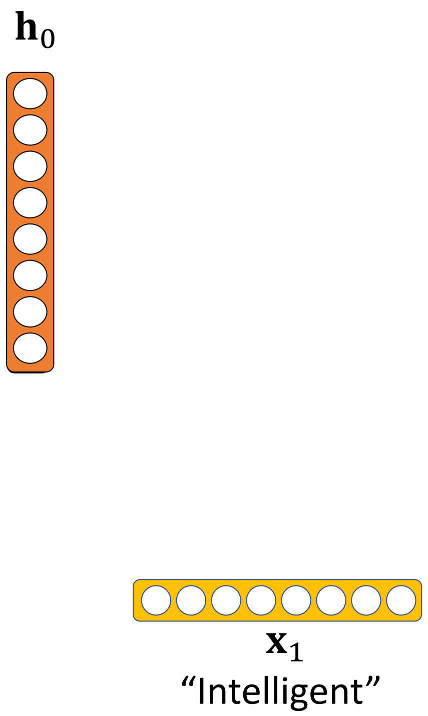
Speed Reading

- ″Readers make longer pauses at points where processing loads are greater … as a function of the involvement of the **various levels** of processing″ (Just & carpenter, 1980).

- ‘Reading’ is similar to *matrix multiplication* in RNN.
  - **Skim**: use small matrix multiplication.
  - **Fully read**: use big matrix multiplication.

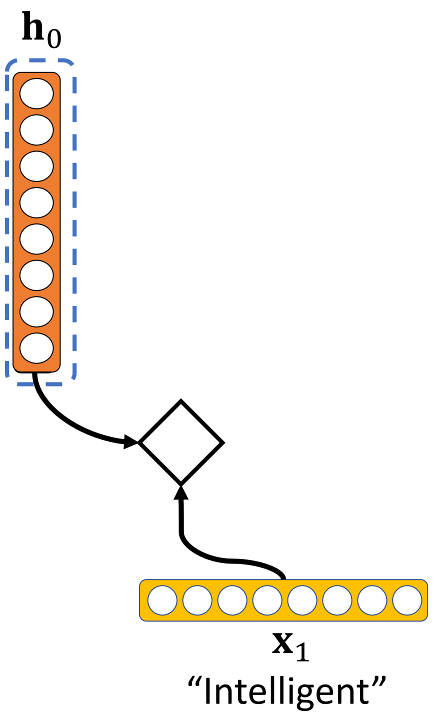
Just & Carpenter. “A theory of reading: From eye fixations to comprehension.” Psychological review 87.4 (1980): 329

Sentiment Classification with Skim-RNN



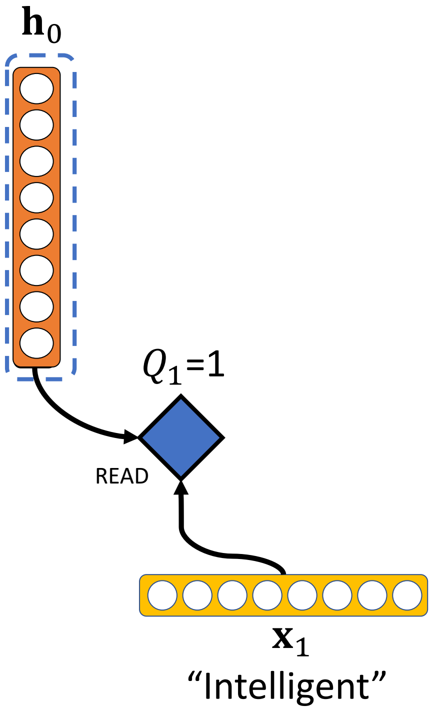
Sentiment Classification with Skim-RNN

Clova Developers

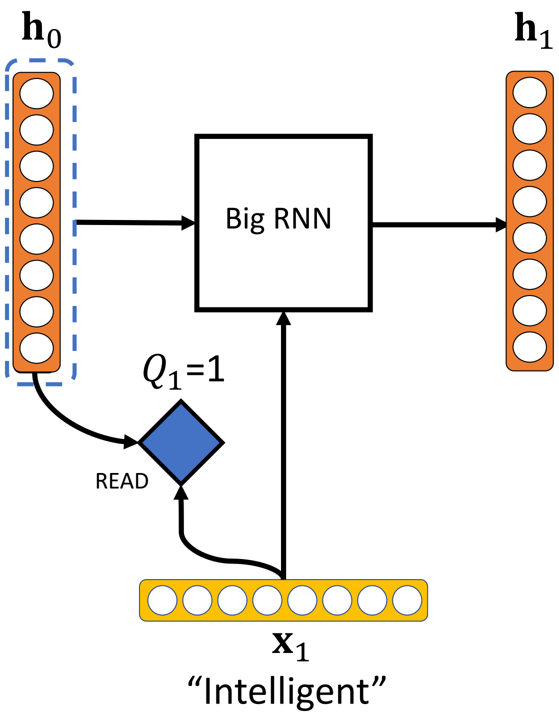


Sentiment Classification with Skim-RNN

Clova Developers

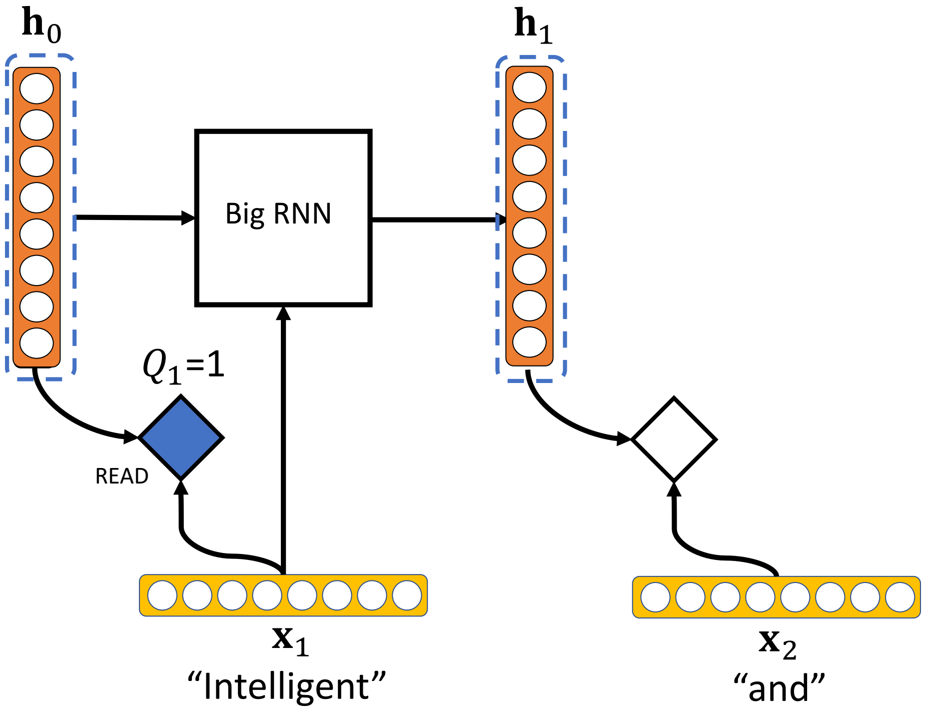


Sentiment Classification with Skim-RNN

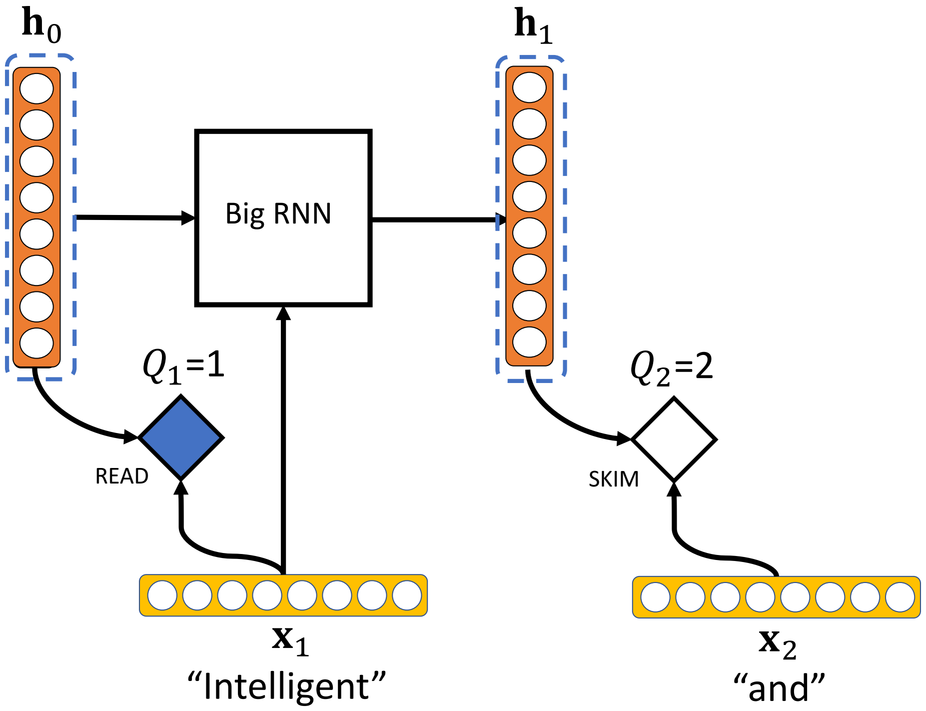


Sentiment Classification with Skim-RNN

Clova Developers

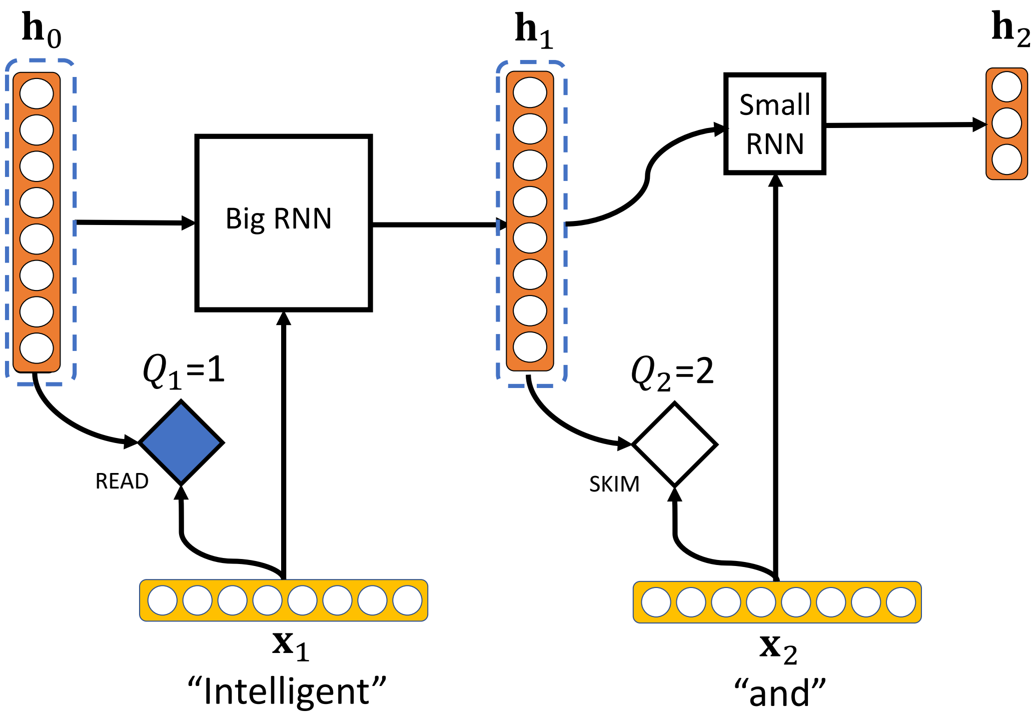


Sentiment Classification with Skim-RNN

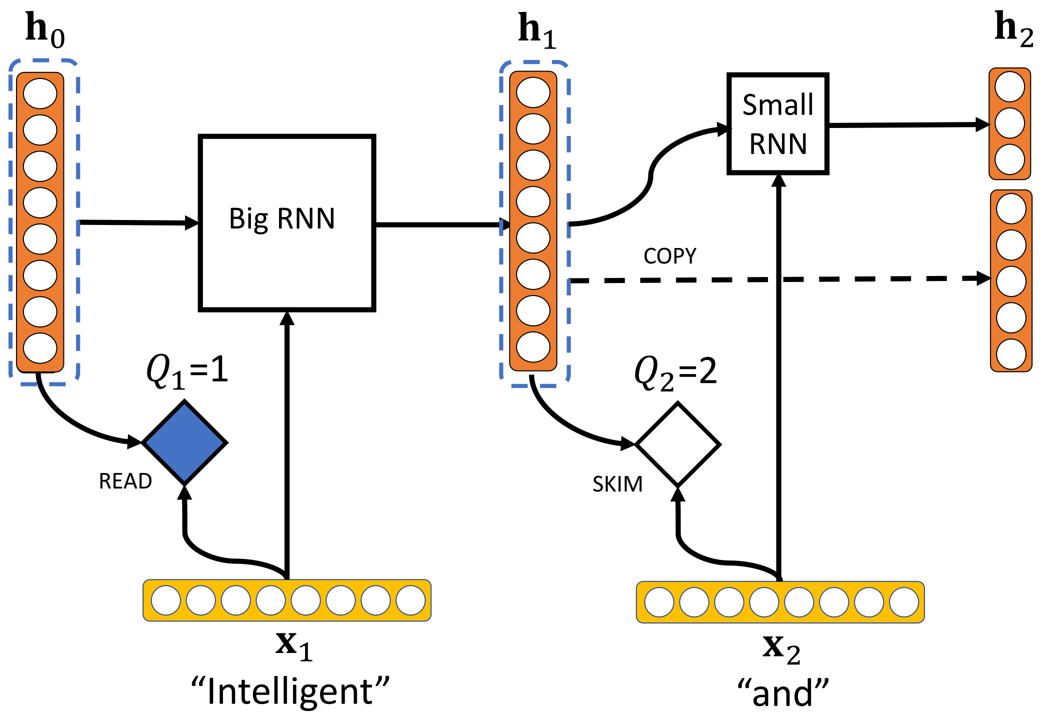


Sentiment Classification with Skim-RNN

Clova Developers

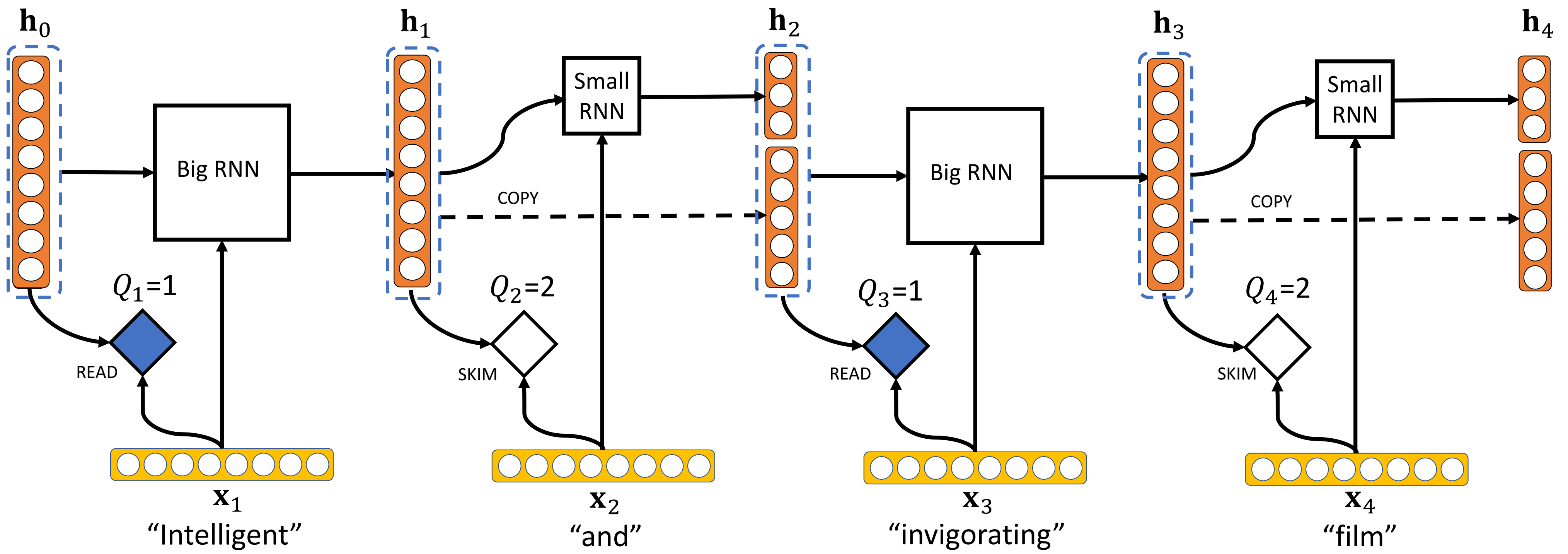


Sentiment Classification with Skim-RNN



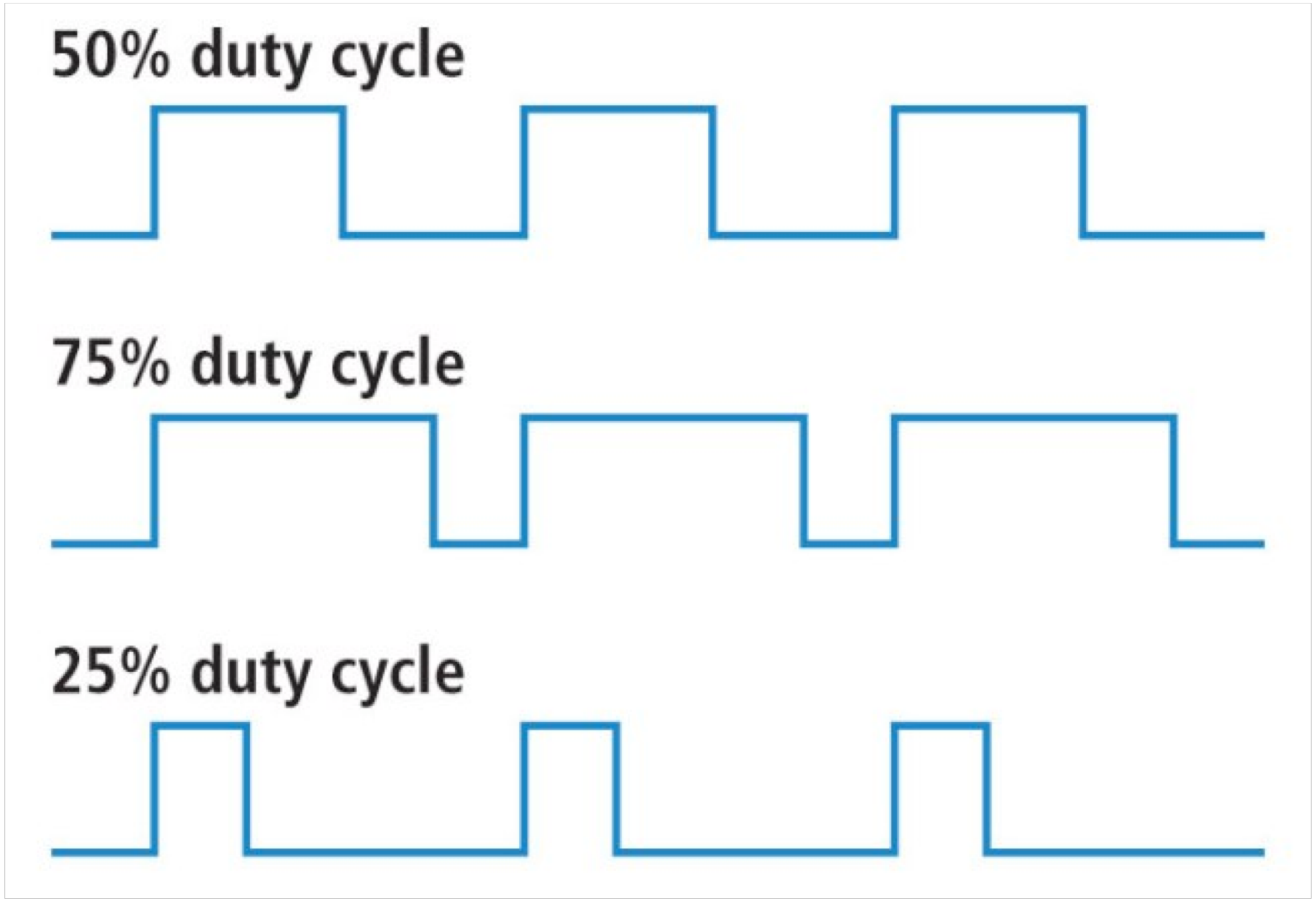
Sentiment Classification with Skim-RNN

Clova Developers



Similar to Pulse Width Modulation

Clova
Developers



Skim-RNN

- Consists of two RNNs:
  - **Big RNN**: hidden state size = d
  - **Small RNN**: hidden state size = d'
  - $d >> d'$ (e.g. d=100, d'=5)
- Hidden state is shared between the RNNs.
- Big RNN updates the entire hidden state.
- Small RNN updates only a small portion of the hidden state.
- When using *small* RNN, the inference requires smaller # of FLOP.
  - $O(d^2) >> O(d'd)$
- Dynamically makes decision on which size of RNN to use

Decision (Q_t) as a random variable

$\mathbf{x}_t$: Input

$\mathbf{h}_{t-1}$: Previous hidden state

$$\mathbf{p}_t = \mathrm{softmax}(\alpha(\mathbf{x}_t, \mathbf{h}_{t-1}))$$

$$Q_t = \mathrm{Multinomial}(\mathbf{p}_t)$$

$$Q = [Q_1, Q_2, \dots, Q_T]$$

$$\mathbb{E}[L(\theta)] = \sum_{\boxed{Q \in \mathcal{Q}}} L(\theta; Q)\Pr(Q)$$

But the sample space is exponentially large!

How to train?

- Computing gradient is intractable

- Policy gradient (Williams, 1992)
  - REINFORCE
  - Unbiased gradient estimation

REINFORCE (Williams, 1992)

Clova Developers

$$\mathbb{E}[L(\theta)] = \sum_{Q \in \mathcal{Q}} L(\theta; Q)\Pr(Q)$$

$$\nabla \log \mathbb{E}[L(\theta)] = \mathbb{E}\underbrace{[\nabla \log L(\theta; Q) + \log L(\theta; Q)\nabla \log \Pr(Q)]}_{\text{Gradient can be } \textit{sampled}}$$

But the sample space is exponentially large!

How to train?

- Computing gradient is intractable

- Policy gradient (Williams, 1992)
  - REINFORCE
  - Unbiased estimation
  - High variance; hard to train

- Gumbel-Softmax (Jang et al., 2017)
  - Biased estimation
  - Low variance; good empirical results
  - Fully differentiable during training via reparameterization

Gumbel-Softmax Reparameterization (Jang et al., 2017)

Clova Developers

- Start with
  - soft decision (attention) **p**
  - Random variable **g** from Gumbel distribution

$$\mathbf{r}_t^i = \frac{\exp((\log(\mathbf{p}_t^i) + g_t^i)/\tau)}{\sum_j \exp((\log(\mathbf{p}_t^j) + g_t^j)/\tau)}$$

$$\mathbf{h}_t = \sum_i \mathbf{r}_t^i \tilde{\mathbf{h}}_t^i$$

- Slowly anneal (decrease τ), making the distribution more discrete
- Near 0 temperature, identical to categorical distribution
- Sampling g gives stochasticity
- Reparameterization allows differentiation with stochasticity
- *Shake & Anneal*

Experiments

- 4 Classification Tasks
  - Stanford Sentiment Treebank (SST)
  - Rotten Tomatoes (RT)
  - IMDb
  - AG News

- 2 Question Answering Tasks
  - Stanford Question Answering Dataset (SQuAD)
  - Children Book Test (CBT)

Baselines

- Regular RNN (LSTM)
- LSTM-Jump (Yu et al., 2017)
  - Learns to *skip* inputs
- Variable-Computation RNN (VCRNN) (Jernite et al, 2017)
  - Variable number of hidden state units to update

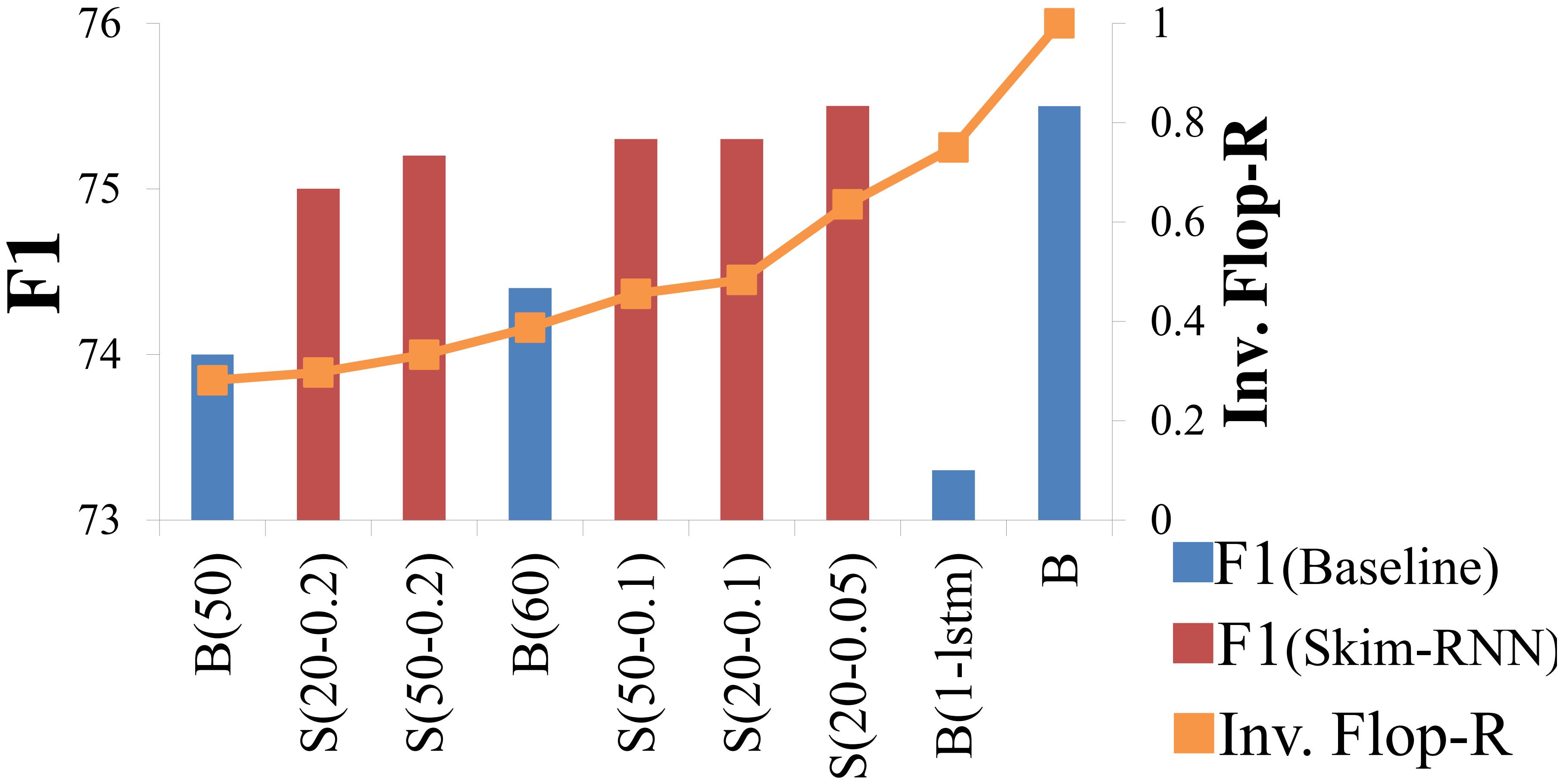
SST and RT Results

Model	SST	Rotten Tomatoes
Baseline (LSTM)	86.4%	82.5%
LSTM-Jump	-	79.3% / 1.6x Speed
VCRNN	81.9% / 2.6x FLOP	-
Skim-RNN	86.4% / 3.0x FLOP	84.2% / 1.3x Speed

Question Answering Results

	F1	EM	FLOP-R
Baseline (LSTM+Att)	75.5%	67.0%	1.0x
VCRNN	74.9%	65.4%	1.0x
Skim-RNN	75.0%	66.0%	2.3x

F1 vs FLOP across diff configs.



Visualization on IMDb Sentiment Classification

Positive	I **liked** this movie, not because Tom Selleck was in it, but **because** it was a **good** story about baseball and it also had a semi-over **dramatized** view of some of the issues that a BASEBALL player coming to the end of their time in Major League sports must face. I also **greatly enjoyed** the cultural differences in American and Japanese baseball and the small facts on how the games are played differently. **Overall**, it is a **good movie** to watch on Cable TV or rent on a cold winter's night and watch about the "Dog Day's" of summer and know that spring training is only a few months away. A **good** movie for a baseball fan as well as a good "DATE" movie. Trust me on that one! *Wink*
Negative	**No**! **no** - **No** - **NO**! My **entire** being is **revolting** against this **dreadful** remake of a classic movie. I **knew** we were heading for trouble from the moment Meg Ryan appeared on screen with her **ridiculous** hair and clothing - literally looking like a scarecrow in that garden she was digging. Meg Ryan playing Meg Ryan - how **tiresome** is that?! And it **got worse** ... so much **worse**. The **horribly cliché** lines, the stock characters, the increasing sense I was **watching** a spin-off of "The First Wives Club" and the ultimate **hackneyed** schtick in the delivery room. How many times have I seen this movie? Only once, but it **feel** like a dozen times - **nothing** original or fresh about it. For shame!

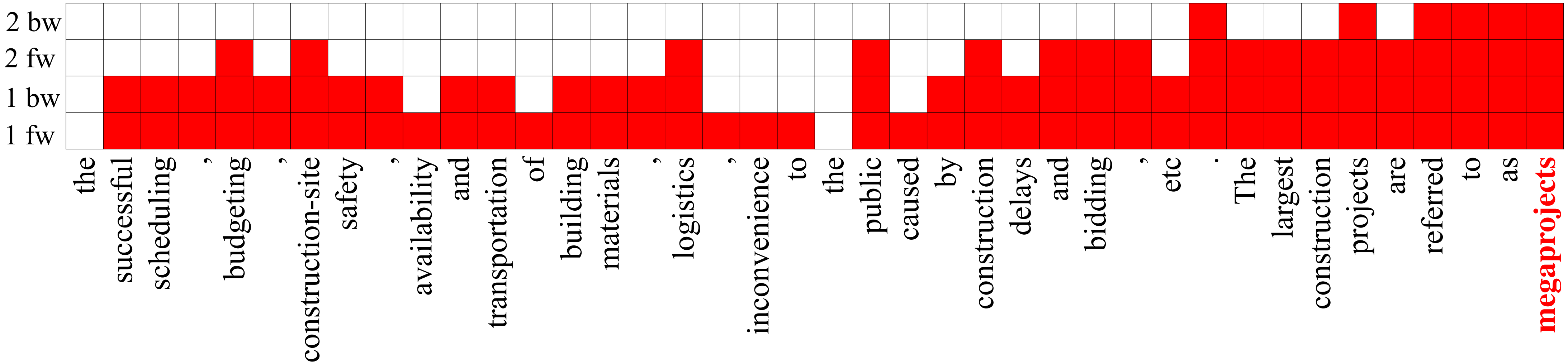
*Black words are skimmed (small RNN), blue words are fully read.

Visualization on Stanford Question Answering Dataset

Q	What is one straightforward case of a probabilistic test?
C	A particularly **simple** example of a **probabilistic test** is the **Fermat primality test**, which **relies** on the fact (**Fermat**'s little theorem) that $np \equiv n$ (**mod** p) for any n if p is a prime number. If you have a number **b** that we want to **test** for **primality**, then we work out nb (**mod** b) for a random value of n as our **test**. A **flaw** with this **test** is that there are some composite numbers (the **Carmichael** numbers) that satisfy the **Fermat** identity even though they are not prime, so the **test** has no way of distinguishing between prime numbers and Carmichael numbers. **Carmichael** numbers are substantially rarer than prime numbers, though, so this **test** can be useful for practical purposes. More powerful extensions of the **Fermat primality test**, such as Baillie-PSW, Miller-Rabin, and Solovay-Strassen **tests**, are guaranteed to fail at least some of the time when applied to a composite number.

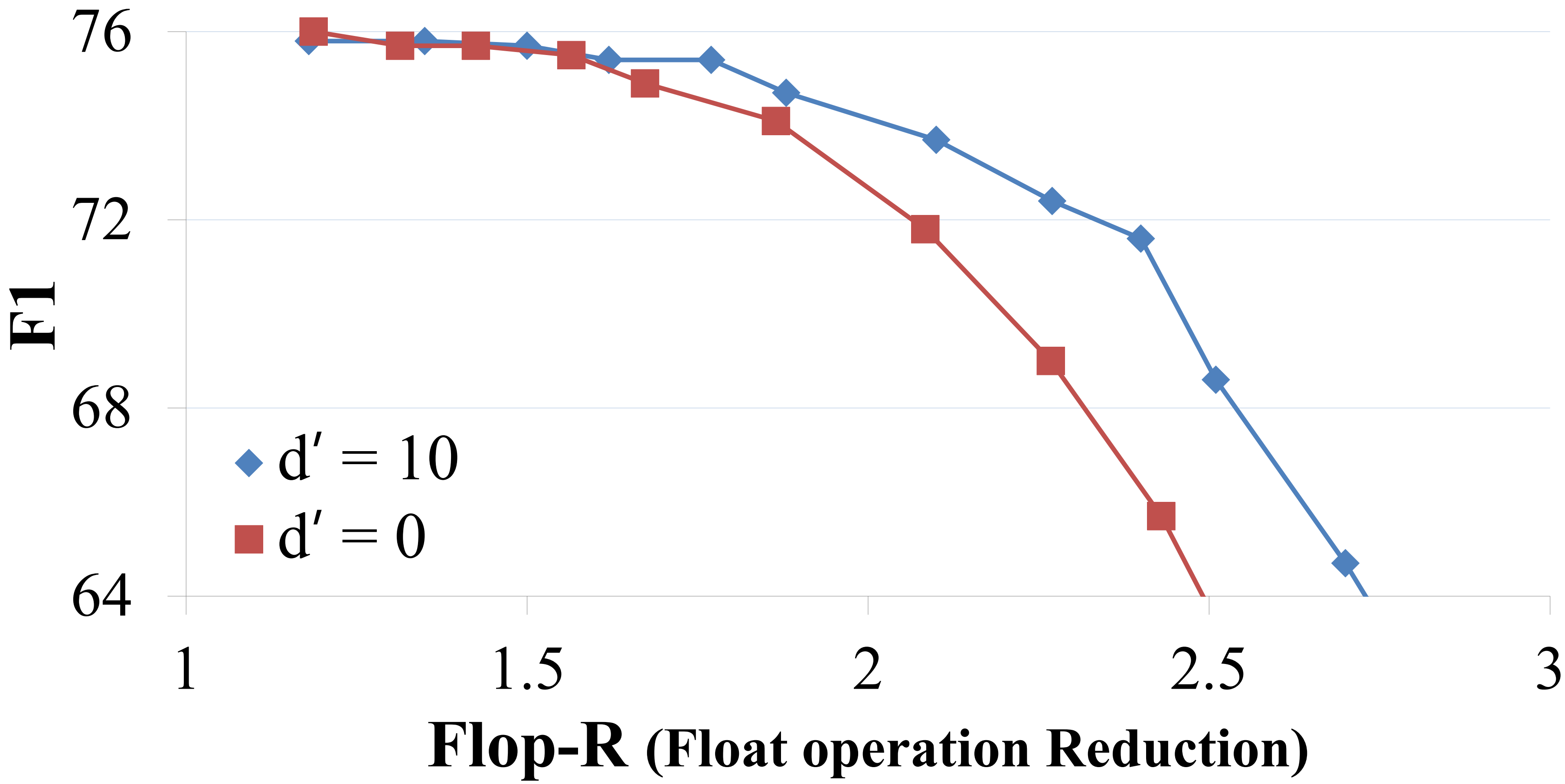
*Black words are skimmed (small RNN), blue words are fully read.

Layer-wise Skim Visualization (SQuAD)



Most RNN steps at higher layer is redundant!

Dynamically controlling # of FLOP



Conclusion

- **Skim-RNN**: switching between two different-size RNNs with shared hidden state.
  - Can be generalized to multiple RNNs.
- Speed gain can be substantial.
- Especially useful for **latency.**

Future Work

- Using multiple granularities of RNNs (not just two)
- Extension to latency-critical applications
  - Speech
  - Video
- Low-level implementation

Thanks! Questions?

Clova Developers

- minjoon.seo@navercorp.com
- http://seominjoon.github.io