Focused Hierarchical RNNs

Karol Kaczmarek

Adam Mickiewicz University
Poznań

Applica.ai
Warsaw

2018

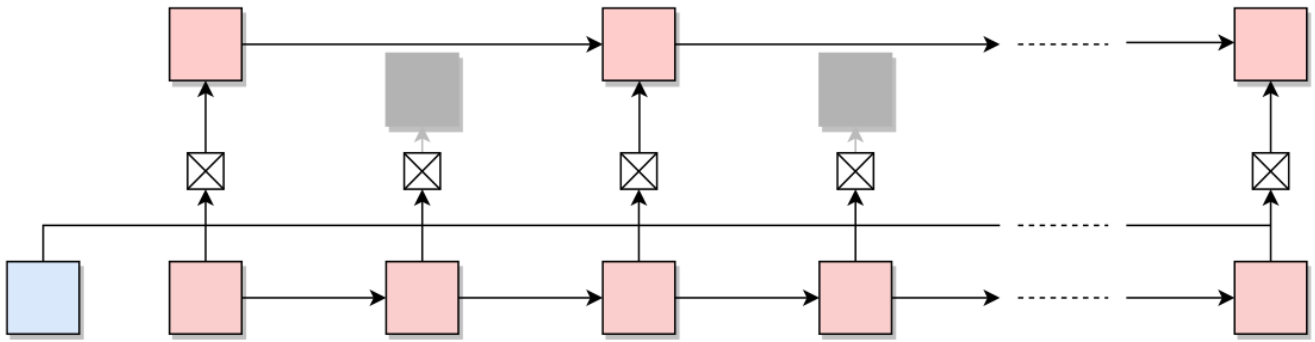
Ideas

- reading a Wikipedia article and trying to identify information that is relevant to answering a question
  - before one knows what the question is
  - where context or question is given before reading the article

Architecture

- Focused Hierarchical RNNs for Conditional Sequence Processing [1]
  - focused hierarchical encoder (FHE)
  - decoder
- Focused Hierarchical Encoder = two-layer LSTM
  - lower layer operates at the input token level
  - upper layer focuses on tokens relevant to the context
  - conditional boundary gate to decide, depending on the context or question whether it is useful to update the upper-level LSTM

Focused Hierarchical Encoder - visualization



- lower layer LSTM processes each step
- for each token, the boundary gate decides (based on the current lower layer LSTM state and question embedding) if information should be stored in the upper level
- higher layer LSTM states update only when the corresponding gate is open

Lower-level Layer

- $h_t^l, c_t^l = \text{LSTM}(x_t, h_{t-1}^l, c_{t-1}^l)$
- sequence of input tokens - $P = (x_1, ..., x_n)$
- LSTM hidden state - h_t^l
- LSTM cell state - c_t^l
- time - t
- may be also augmented with other available information (question encoding)

Conditional Boundary Gate

- decides if information at the current time step should be stored in the upper-level representation
- question is essential in deciding how to represent the passage
- question embedding - q
- output of the boundary gate is a scalar $b_t \in (0, 1)$ that is taken to be the parameter of a Bernoulli distribution $b_t \sim \text{Bernoulli}(b_t)$
- time - t

Conditional Boundary Gate

- in the simplest case, the boundary gate forward pass is formulated as $\tilde{b}_t = \sigma(w_b^\top \text{LReLU}(W_b z_t + b_b))$
- trainable weights - W_b, b_b, w_b
- leaky ReLU - $\text{LReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$
- input - $z_t = [q \odot h_t^l, h_t^l, q]$
- element-wise product - $\odot$
- lower-layer hidden states - h_t^l
- question embedding - q

Upper-level Layer

- $\tilde{h}_t^u, \tilde{c}_t^u = \text{LSTM}(h_t^l, h_{t-1}^u, c_{t-1}^u)$
- $b_t \sim \text{Bernoulli}(\tilde{b}_t)$
- $c_t^u = \tilde{b}_t \tilde{c}_t^u + (1 - \tilde{b}_t) c_{t-1}^u$
- $h_t^u = \tilde{b}_t \tilde{h}_t^u + (1 - \tilde{b}_t) h_{t-1}^u$
- LSTM hidden state - h_t^u
- LSTM cell state - c_t^u
- time - t

Baseline

- LSTM1 (1-layer LSTM) - equivalent to FHE with the always closed ($b_t = 0$ for each t)
- LSTM2 (2-layer LSTM) - equivalent to FHE with the boundary gate fully open ($b_t = 1$ for each t)

Picking task

- given a sequence of randomly generated digits of length n
- the goal is to determine the most frequent digit within the first k digits ($k \leq n$)
- where value k is the question
- tested for $n \in \{100, 200, 400\}$

Picking task - examples

Input Sequence	k	Target Mode
random examples		
80560201708283837170131630447 3	10	**0**
6387332908903966902559379864 85	23	**3**
1645519375793738968139811259 82	26	**1**
malicious examples		
6663336662888828888199999999 90	6	**6**
6663336662888828888199999999 90	10	**6**
6663336662888828888199999999 90	20	**8**
6663336662888828888199999999 90	30	**9**

Picking task - hyper-parameters

- hidden $= 128$, learning rates $= 0.0001$, Adam optimizer
- fix hyper-parameters to a small value (for examples $\beta = 0.1$ and $\gamma = 0.25$) - gates can almost freely open
  - once the desired accuracy has been reached, enforce constraints on our hyper-parameters
  - provides a level of control over the accuracy-sparsity tradeoff
  - this approach with the requirement of achieving a desired accuracy $a \in \{80\%, 90\%, 95\%, 98\%\}$, called: FHE80, FHE90, FHE95, FHE98
- fix hyper-parameters more restricted than previous (for examples $\beta = 1$ and $\gamma = 0.1$), called FHE-fixed

Picking task - results

LENGTH	LSTM1	LSTM2	FHE-FIXED
100	99.4	**99.7**	99.5
200	97.0	99.2	**99.4**
400	92.9	**97.5**	96.9

LENGTH	FHE80	FHE90	FHE95	FHE98
100	93.4	94.2	96.6	**98.7**
200	92.3	92.4	**93.6**	**93.6**
400	87.2	90.5	90.0	**91.0**

Picking task - test for longer sequence length

- models trained on short sequences ($n = 200$)
- evaluated on longer sequences and $k \leq 200$

LENGTH	LSTM1	LSTM2	FHE-FIXED
200	97.1	99.2	**99.4**
400	55.9	61.4	**97.6**
800	39.6	39.7	**95.6**
1600	29.5	28.6	**93.3**
10000	18.5	14.8	**66.8**

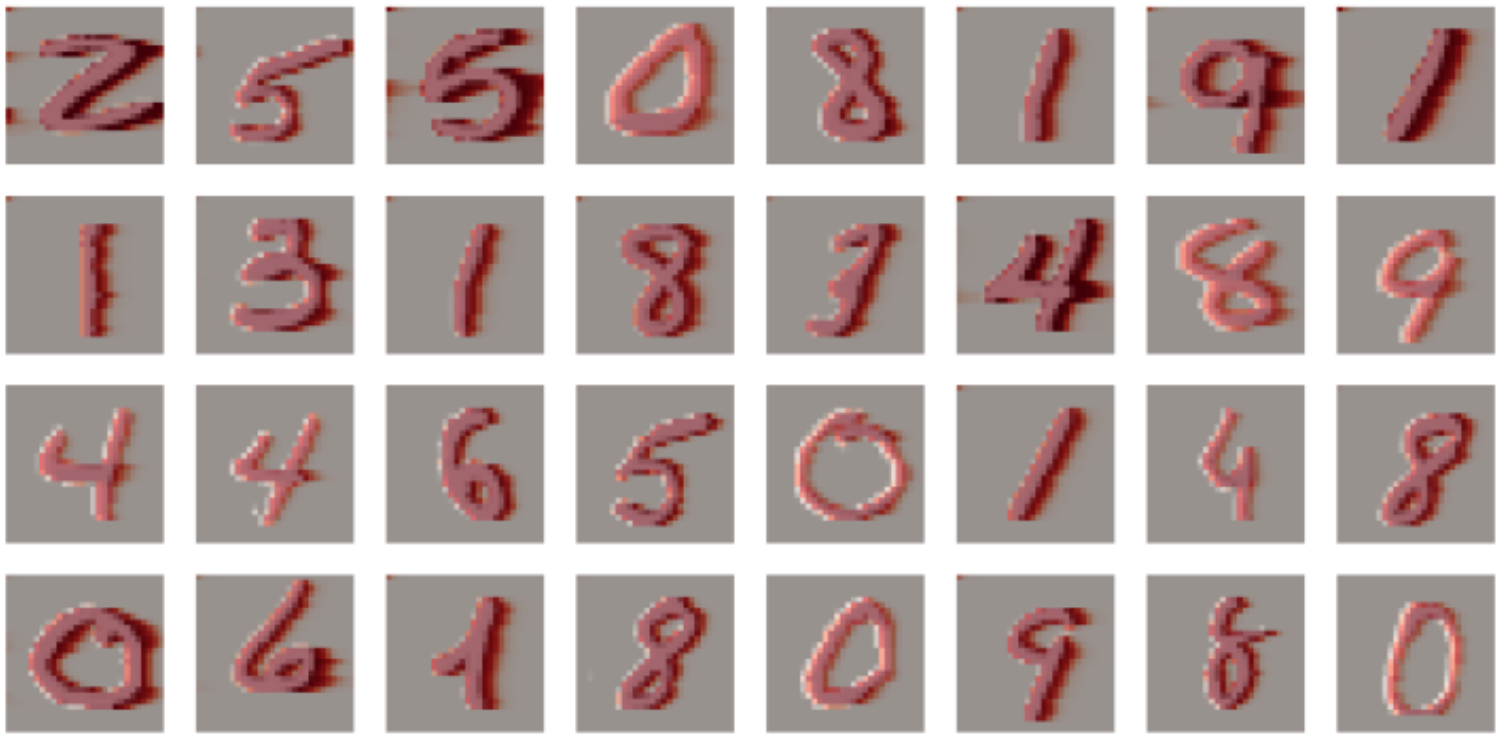
Pixel-by-Pixel MNIST QA task

- adapt the Pixel-by-Pixel MNIST classification task to the question and answering setting
- encoder reads in MNIST digits one pixel at a time
- question asked is whether the image is a specific digit and the answer is either True or False

LSTM1	LSTM2	FHE-FIXED
97.3	98.4	**99.1**

Pixel-by-Pixel MNIST QA task

- model learns to open the boundary gate almost always around the digit
- gates do not depend on the question



Natural Language QA Tasks

- using the MS MARCO and SearchQA datasets and tasks
- use a bidirectional LSTM for question
- hidden $= 300$, learning rate $= 0.001$, Adam optimizer
- to avoid large-vocabulary issues, use the pointer softmax
  - distribution over a shortlist of words ($100, 10000$ most frequent words for SearchQA or MS MARCO tasks) - o_j
  - distribution over words in the document - x_j
  - variable z_j determines how to interpolate between the indices in the document and the shortlist words
  - $P_j = [z_j o_j; (1 - z_j)\alpha_j]$

SearchQA Question and Answering Task

- large scale QA dataset in the form of Question-Context-Answer
- question-answer pairs are real Jeopardy! (crawled from J!Archive)
- contexts are text snippets retrieved by Google
- use F1 scores for multi-word answers
- use Exact Match (EM) for single word answers

SearchQA Question and Answering Task - results

Models	Validation F1	Validation EM	Test F1	Test EM
TF-IDF Max (Dunn et al., 2017)	-	13.0	-	12.7
ASR (Dunn et al., 2017)	24.1	43.9	22.8	41.3
AQA (Buck et al., 2018)	47.7	40.5	45.6	38.7
Human (Dunn et al., 2017)	-	-	43.9	-
LSTM1 + pointer softmax	52.8	41.9	48.7	39.7
LSTM2 + pointer softmax	55.3	44.7	51.9	41.7
Our Model	56.7	**49.6**	53.4	**46.8**
Concurrent work				
AMANDA (Kundu & Ng, 2018)	**57.7**	48.6	**56.6**	**46.8**

MS MARCO Question and Answering Task

- one of the largest publicly available QA datasets
- example in the dataset consists of a query
- several context passages retrieved by the Bing search engine
- several human generated answers (synthesized from the given contexts)
- contains many rare words, with around 90% of words appealing less than 20 times in the dataset

MS MARCO Question and Answering Task

- span-based
  - state of the art according to Bleu-1 and Rouge scores
  - trained using ”gold-spans”, obtained by a preprocessing step which selects the passage in the document maximizing the Bleu-1 score with the answer
  - they cannot answer questions where the answer is not contained in the passage
- generative systems
  - synthesize a novel answer for the given question
  - could learn a disentangled representation, and therefore generalize better

MS MARCO Question and Answering Task - span-based systems

- state of the art according to Bleu-1 and Rouge scores
- trained using ”gold-spans”, obtained by a preprocessing step which selects the passage in the document maximizing the Bleu-1 score with the answer
- they cannot answer questions where the answer is not contained in the passage

MS MARCO Question and Answering Task - results

Generative Models	Validation		Test	
	Bleu-1	Rouge-L	Bleu-1	Rouge-L
Seq-to-Seq (Nguyen et al., 2016)	-	8.9	-	-
Memory Network (Nguyen et al., 2016)	-	11.9	-	-
Attention Model (Higgins & Nho, 2017)	9.3	12.8	-	-
LSTM1 + pointer softmax	24.8	26.5	28	28
LSTM2 + pointer softmax	24.3	23.3	27	28
Our Model	**27.3**	**26.7**	**30**	**30**
Ablation study				
Our Model – dot-product between question and context	18.5	19.3	-	-
Our Model – pointer softmax	20.5	18.7	-	-
Our Model – learned boundaries	23.5	24	-	-

References I

[1] N. R. Ke, K. Żołna, A. Sordoni, and et al, “Focused hierarchical rnns for conditional sequence processing,” 2018.