Bidirectional attention flow for machine comprehension

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi

Machine learning

Francesca Del Lungo
Prof. Paolo Frasconi

17 September 2020

Contents overview

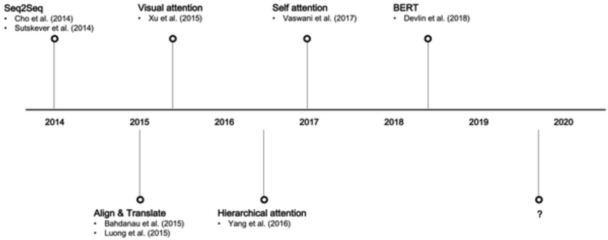
Overview

Question answering

MC/ QA

Machine Comprehension (MC) / Question Answering (QA): understand unstructured text and then answer questions

Attention before BiDAF



- (2014) Encoder–decoders (fixed length)
- (2015) *"Neural Machine Translation by Jointly Learning to Align and Translate"* Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio
- (2018) Bi-directional attention flow (BiDAF) model

Objective

Context
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

Query	**Answer**
Where do water droplets collide with ice crystals to form precipitation?	*within a cloud*

BiDAF Model

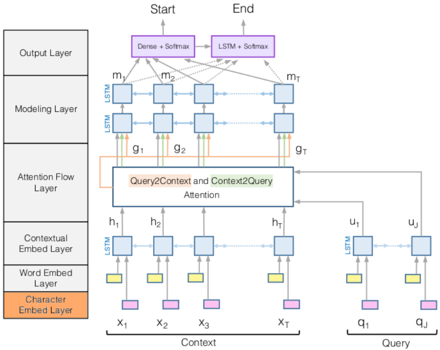
BiDAF model

The model is composed of six layers

Three main parts:

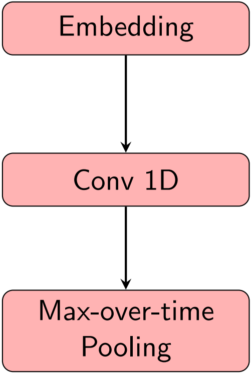
- **Embedding Layers**: (three levels of granularity)
    - character-level embedding
    - word-level embedding
    - context-based embedding
- **Attention and Modeling Layers**: Query and Context information are fused
- **Output Layer**: answer extraction

Character Embedding Layer I



- $x_1, \ldots, x_T$ context words
  $q_1, \ldots, q_J$ query words
- map each word to a high-dimensional vector space
- query and context

Character Embedding Layer II



- context:

$$X \in \mathbb{R}^{K \times T} \rightarrow X \in \mathbb{R}^{d_1 \times T}$$

- query:

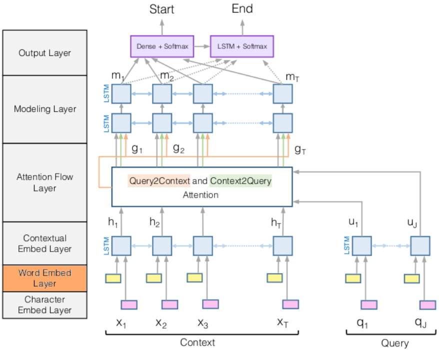
$$Q \in \mathbb{R}^{K \times J} \rightarrow Q \in \mathbb{R}^{d_1 \times J}$$

Character Embedding Layer III

Conv1D

- 100 1D filters, each with a width of 5
- out-of-vocabulary (OOV) word
- extract information from sub-parts of each word

Word Embedding Layer I



- maps each word to a vector space using a pre-trained word embedding model (**GloVe**)
- each word mapped to d-dimensional vector
- query and context

Word Embedding Layer II

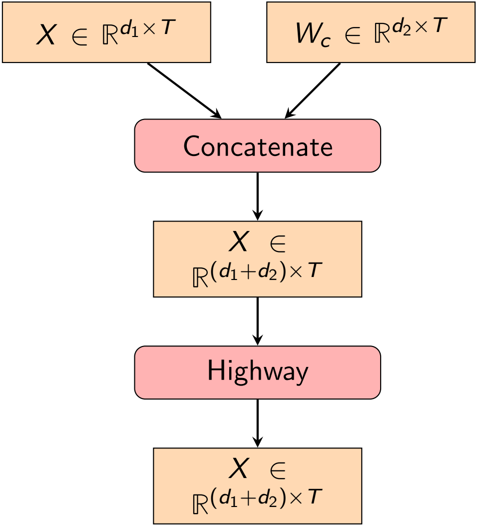
- context:

$$W_c \in \mathbb{R}^{d_2 \times T}$$

- query:

$$W_q \in \mathbb{R}^{d_2 \times J}$$

Highway Net



- $d' = (d_1 + d_2)$
- Context:

$$X \in \mathbb{R}^{d' \times T}$$

- Query:

$$Q \in \mathbb{R}^{d' \times J}$$

Highway Network I

$$y = H(x, W_H)$$

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C)$$

where non-linear transformations:

- H: general transformation
- T: **transform gate**

$$T(x) = \sigma(W_T^T x + b_T) \quad \in (0, 1)$$

- C: **carry gate** C = 1 - T

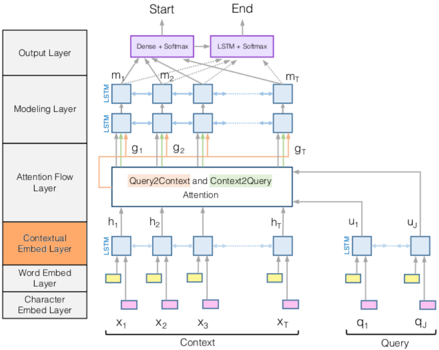
Highway Network II

$$y = \begin{cases} x, & \text{if } T(x, W_T) = 0 \\ H(x, W_H), & \text{if } T(x, W_T) = 1 \end{cases}$$

Why Highway Net

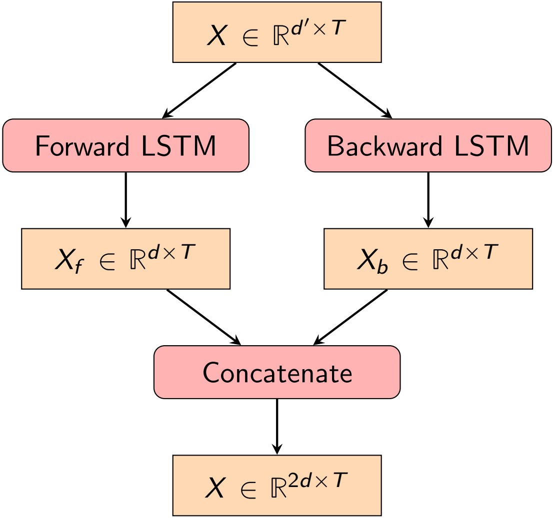
Adjust the relative contribution from the word embedding and the character embedding

Contextual Embedding Layer I



- Bidirectional LSTM
- Model temporal interactions between words
- Contextual meaning of words (eg "tear")

Contextual Embedding Layer II



- context:

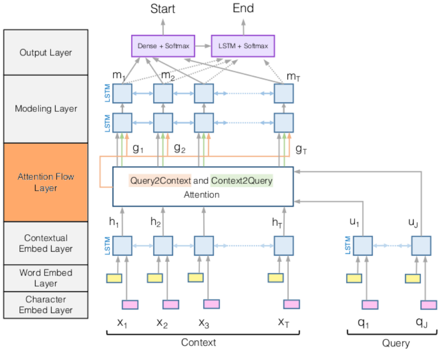
  $X \in \mathbb{R}^{d' \times T} \rightarrow H \in \mathbb{R}^{2d \times T}$

- query:

  $Q \in \mathbb{R}^{d' \times J} \rightarrow U \in \mathbb{R}^{2d \times J}$

- *semantic*, *syntactic* and *contextual* meaning

Attention Flow Layer



- Fuse infos from context and query words
- Goal: create multiple representations of the context that also contain information from the query
- Steps:
  1. Similarity matrix
  2. Context-to-query attention
  3. Query-to-context attention
  4. Merge

1. Similarity matrix

S_{tj} represents the similarity of t-th context word and j-th query word.

Compute similarity matrix $S \in \mathbb{R}^{T \times J}$

$$\mathbf{S}_{tj} = \alpha(h, u) = w_{(s)}^{T}[h; u; h \circ u] \in \mathbb{R}$$

where:

- $h = \mathbf{H}_{:t} \in \mathbb{R}^{2d}$
- $u = \mathbf{U}_{:j} \in \mathbb{R}^{2d}$
- $w_{(s)}^{T} \in \mathbb{R}^{6d}$ trainable weight vector
- $\circ$ element-wise multiplication
- $[;]$ vector concatenation

2. Context-to-query attention

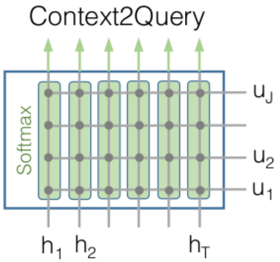
C2Q attention

Context-to-query (C2Q) attention: which query words are most relevant to each context word.

$\widetilde{U} \in \mathbb{R}^{2d \times T}$

$$a_t = softmax(\mathbf{S}_{t:}) \in \mathbb{R}^J$$

$$\widetilde{U}_{:t} = \sum_j a_{tj} \mathbf{U}_{:j}$$



3. Query-to-context (Q2C) attention

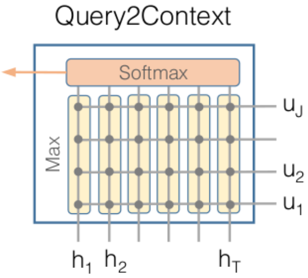
Q2C attention

Which context words have the closest similarity to one of the query words

$\widetilde{H} \in \mathbb{R}^{2d \times T}$

$$\mathbf{b} = softmax(max_{row}(\mathbf{S})) \in \mathbb{R}^{T}$$

$$\widetilde{h} = \sum_{t} b_t \mathbf{H}_{:t} \in \mathbb{R}^{2d}$$



4. Merge

Combine contextual embeddings and attention vectors

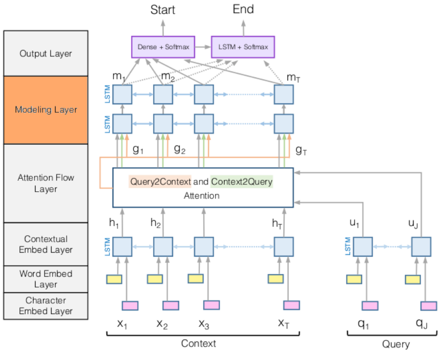
$G \in \mathbb{R}^{8d \times T}$

$$\mathbf{G}_{:t} = \beta(\mathbf{h}, \widetilde{\mathbf{u}}, \widetilde{\mathbf{h}}) = [\mathbf{h}; \widetilde{\mathbf{u}}; \mathbf{h} \circ \widetilde{\mathbf{u}}; \mathbf{h} \circ \widetilde{\mathbf{h}}] \in \mathbb{R}^{8d}$$

where:

- $\mathrm{h} = \mathbf{H}_{:t}$
- $\widetilde{u} = \widetilde{\mathbf{U}}_{:t}$
- $\widetilde{h} = \widetilde{\mathbf{H}}_{:t}$
- $\circ$ element-wise multiplication
- $[;]$ vector concatenation

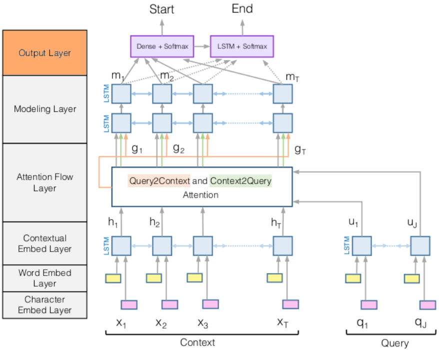
Modeling Layer I



- Captures the interaction among the context words *conditioned on* the query
- Different from the contextual emb layer
- 2 layers Bidirectional-LSTM
- Modeling layer:

$$G \in \mathbb{R}^{8d \times T} \rightarrow M \in \mathbb{R}^{2d \times T}$$

Output Layer I



- Application-specific
- Answer is a span in the context
- Predict start and end indices

Output Layer II

Start index

$$\mathbf{p}^1 = softmax(w^T_{(p^1)}[\mathbf{G}; \mathbf{M}])$$

where:

- $w^T_{(p^1)} \in \mathbb{R}^{10d}$ trainable weight vector
- $\mathbf{G} \in \mathbb{R}^{8d \times T}$
- $\mathbf{M} \in \mathbb{R}^{2d \times T}$
- $[;]$ vector concatenation

Output Layer III

End index

$$\mathbf{p}^2 = softmax(w_{(p^2)}^T [\mathbf{G}; \mathbf{M}^2])$$

where:

- $w_{(p^2)}^T \in \mathbb{R}^{10d}$ trainable weight vector
- $\mathbf{G} \in \mathbb{R}^{8d \times T}$
- $\mathbf{M}^2 = \text{bidirectional-LSTM(M)} \in \mathbb{R}^{2d \times T}$
- $[;]$ vector concatenation

Training

Loss (minimize)

$$L(\theta) = -\frac{1}{N}\sum_{i}^{N} log(\mathbf{p}^1_{y_i^1}) + log(\mathbf{p}^2_{y_i^2})$$

where:

- θ : all trainable weights
- y_i^1: true start index of the i-th example
- y_i^2: true end index of the i-th example
- $\mathbf{p}_k$: k-th value of the vector $\mathbf{p}$

Details

Preprocessing:

- PTB Tokenizer: regular-expression-based word tokenizer

Model details:

- AdaDelta optimizer
- Glove word embedding $d_2 = 100$
- Hidden state size of the model d = 100
- CNN char embedding: 100 1D filters, width 5
- Dropout 0.2 (CNN, all LSTM, output layer)

Evaluation metrics:

- EM
- F1

Exact match (EM)

- Binary measure: the output **exactly matches** the ground truth answer
- Fairly strict metric

Example

Ground truth: "Albert Einstein"; Predicted answer: "Einstein"

EM score: 0

F1 or F-Measure

- Harmonic mean of precision and recall

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$precision = \frac{tp}{tp + fp} \quad recall = \frac{tp}{tp + fn}$$

Example

Ground truth: "Albert Einstein"; Predicted answer: "Einstein"

100% precision, 50 % recall

F1 score: $\frac{2 \times 50 \times 100}{100 + 50} = 66.67\%$

Experiments

Experiments

- BiDAF model
- Ablations on BiDAF model
- BiDAF model on different datasets
- BiDAF and competing approaches

Results

Results

Model	EM	F1
Bidaf [1]	67.7%	77.3%
Bidaf [2]	66.3 $\pm$0.6%	76.1 $\pm$0.7%

[1] Original bidaf implementation by the authors of the paper
[2] Reproduced bidaf model (10 runs)

Dataset comparison I

Datasets:

- (context, question, answer) triples
- Closed: answer is a segment of text (span)
- **SQUAD** (2016):
  - 100K question-answer pairs
  - 500+ articles
  - 75% of answers less than 4 words long
- **TriviaQA** (2017):
  - 650K question-answer-evidence triples
  - complex, compositional questions
  - Web
  - Wikipedia
    - train 110K
    - dev 14K, verified dev 305
  - 14 tokens avg query length

Dataset comparison II

Model	Dataset	EM	F1
Bidaf [1]	Squad	67.7%	77.3%
Bidaf [2]	Squad	66.3 ±0.6%	76.1 ±0.7%
Bidaf [1]	TriviaQA (Wiki-verified)	47.4%	53.7%
Bidaf [2]	TriviaQA (Wiki-verified)	45.2 ±0.9%	51.1 ±1.0%
Bidaf [1]	TriviaQA (Wiki)	40.2%	45.7%
Bidaf [2]	TriviaQA (Wiki)	39.7 ±0.7%	44.7 ±0.8%
Bidaf [2]	TriviaQA (Wiki[3])	44.5 ±0.5%	49.2 ±0.7%

Human accuracy on TriviaQA (Wiki-verified): 79.6%

[1] Original bidaf implementation by the authors of the paper
[2] Reproduced bidaf model (5 runs)
[3] TriviaQA all doc contains answers

Ablations I

- No **char embedding**: better handle out-of-vocab (OOV) or rare words
- No **word embedding**: better at representing the semantics of each word as a whole
- No **C2Q attention**
- No **Q2C attention**

Ablations II

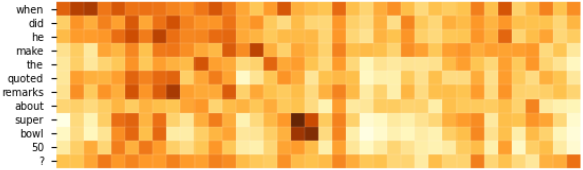
Ablation	EM	F1
No char emb[1]	65.0%	75.4%
No char emb[2]	63.7 ±0.7%	73.2 ±0.9%
No word emb[1]	55.5%	66.8%
No word emb[2]	53.2 ±0.7%	64.5 ±0.7%
No C2Q attention[1]	57.2%	67.7%
No C2Q attention[2]	54.5 ±1.0%	65.0 ±1.1%
No Q2C attention[1]	63.6%	73.7%
No Q2C attention[2]	61.7 ±0.3%	72.6 ±0.4%

[1]Original bidaf implementation by the authors of the paper
[2]Reproduced bidaf model (10 runs)

Attention matrix I

- Context:
  ...in early 2012 , nfl commissioner Roger Goodell stated that the league planned to make the 50th super bowl "spectacular" and that it would be "an important game for us as a league"...
- Question:
  when did he make the quoted remarks about super bowl 50 ?
- Answer:
  early 2012

Attention matrix II



- when : ['2012', 'early', 'years', 'league', 'nfl', '50th']
- did : ['stated', 'commissioner', 'goodell']
- he : ['commissioner', 'goodell', 'us', 'planned', 'stated', 'roger']
- make : ['make', 'planned', 'stated', 'goodell', 'spectacular', 'commissioner', 'to']
- the : [ ]
- quoted : ['stated', 'goodell', 'league', 'roger']
- remarks : ['stated', 'commissioner', 'planned', 'goodell', 'spectacular']
- about : [ ]
- super : ['super', 'bowl', 'commissioner', 'nfl', 'goodell', 'league', 'spectacular']
- bowl : ['bowl', 'super', 'commissioner', 'goodell', 'spectacular', 'game']
- 50 : ['roger', 'nfl']
- ? : ['stated', 'commissioner']

Compare models

Comparison of different models (on Squad dataset):

- BiDAF
- Match-LSTM
- Dynamic Coattention Networks

Match-LSTM[3]

Based on:

- Match-LSTM: predict textual entailment

Structure:

1. LSTM Preprocessing layer
2. Match-LSTM layer
3. Answer layer

[3] *Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. 2016*

Match-LSTM[3]

Main differences with BiDAF:

1. No character-embedding
2. Dynamic attention: use match-LSTM recurrently
3. Answer sequence/boundary model

[3] *Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. 2016*

Dynamic Coattention Networks[4]

Structure:

- Query and context encoders
- Coattention encoder
- Answer extraction

Main differences with BiDAF:

1. No character-embedding
2. Early attention summarization

[4] *Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. 2016*

Results on different models

Model	EM	F1
Bidaf [1]	67.7%	77.3%
Bidaf [2]	66.3 ±0.6%	76.1 ±0.7%
Dynamic Coattention Networks	65.4%	75.6%
Match-LSTM	64.1%	73.9%

[1] Original bidaf implementation by the authors of the paper

[2] Reproduced bidaf model

Conclusion

Conclusion

- BiDAF model:
  - Character-level embedding
  - Bidirectional attention
  - Attention flow
- Ablations
- Comparison of different models
- Comparison of BiDAF model on different datasets

Future works

- Unanswerable questions
- More complex models

Thanks for the attention

Match-LSTM I

Steps:

1. LSTM Preprocessing Layer
2. Match-LSTM Layer
3. Answer Pointer Layer

Match-LSTM II

Passage (or context):

$$\mathbf{P} \in \mathbb{R}^{d \times P}$$

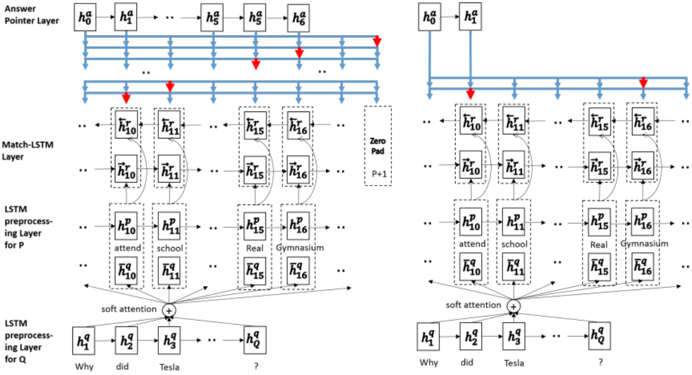
Question (or query):

$$\mathbf{Q} \in \mathbb{R}^{d \times Q}$$

where:

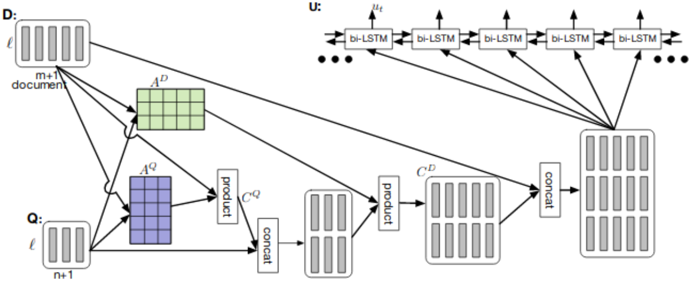
- P: length (number of tokens) of the passage
- Q: length of the question
- d: word embeddings dimensionality

Match-LSTM III



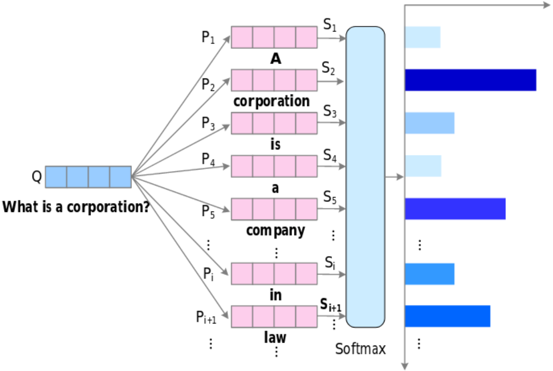
(a) Sequence Model (b) Boundary Model

Dynamic Coattention Networks



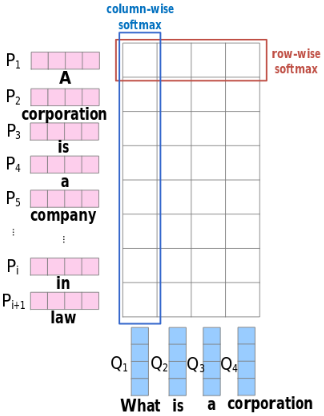
Unidirectional attention

- From query to context (usually)
- Similarity of context word C_i: $S_i = f(C_i, Q)$

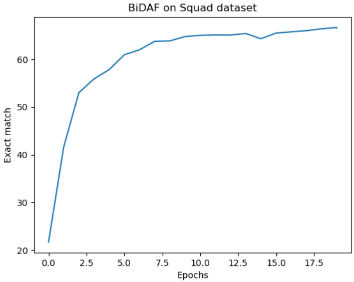


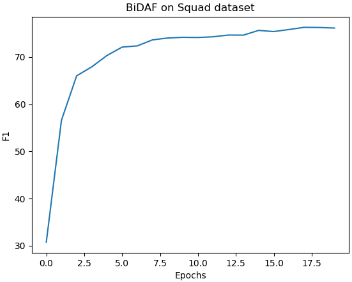
Bidirectional attention

- From query to context and context to query



BiDAF model on Squad dataset





AdaDelta

- Improve AdaGrad
- Scale learning rate based on historical gradient (taking into account only recent time window, not the whole history, like AdaGrad)