딥러닝 자연어 처리 개요

Kyomin Jung

Department of Electrical and Computer Engineering
Seoul National University

Table of Contents

Recurrent Neural Network (RNN)

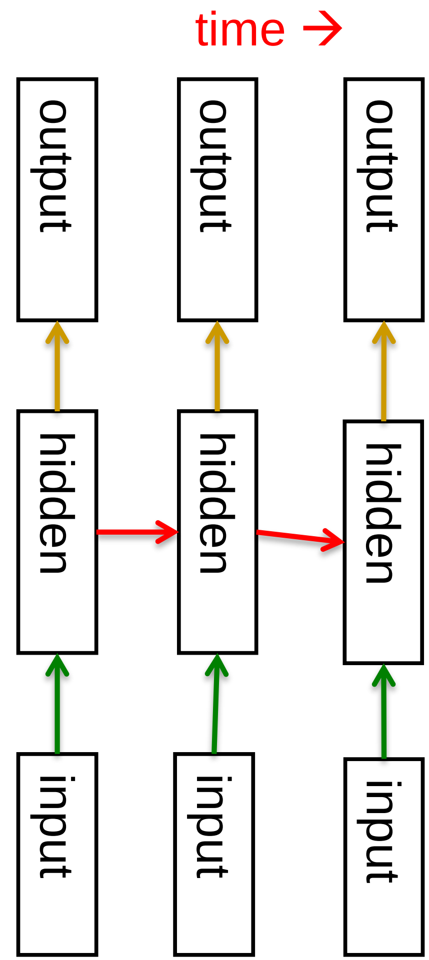
- Feed forward neural networks (including CNN)
  - Information only flows one direction (acyclic directed graph structure)

    cycle이 없는 graph
    === layer 번호 assign 가능
  - One input produces the same output
  - No sense of time (or memory of previous state)

- Recurrence
  - Nodes are allowed to connect back to a previous nodes and/or to themselves (graph contains directed cycles)
  - Sense of time and memory

- Biological nervous systems show many recurrences or cycles

Recurrent Neural Network (RNN) for Modeling Sequences



- RNN consider input sequence over discrete time.
- RNN have the ability to remember information in their hidden states for a long time.

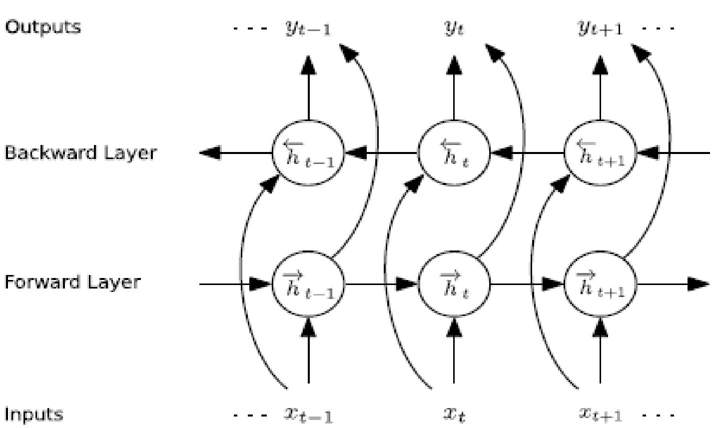
- RNNs are very natural way to model sequential data:
  - They are equivalent to very deep nets with a hidden layer per one time step.
  - Except that they use the same weights at every time step and they get input at every time step.

Remind of Basic RNN formulation

- Input sequence : $x = (x_1, \ldots, x_T)$
- Hidden vector sequence : $h = (h_1, \ldots, h_T)$
- Output vector sequence : $y = (y_1, \ldots, y_T)$
- Hidden vector update function :
  $h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$ W는 time independent @RNN
- Output vector update function :
  $y_t = \sigma(W_{hy}h_t + b_y)$

Bidirectional Recurrent Neural Networks

- Traditional RNNs only model the dependence of the current state on the previous state.
- BRNN extends to model dependence on both past states and future states.
- For example: to predict a missing word in a sequence, look at both the left and the right context in the sentence.



An BRNN

$$\overrightarrow{h}_t = f(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}})$$

$$\overleftarrow{h}_t = f(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$

two separate recurrent hidden layers

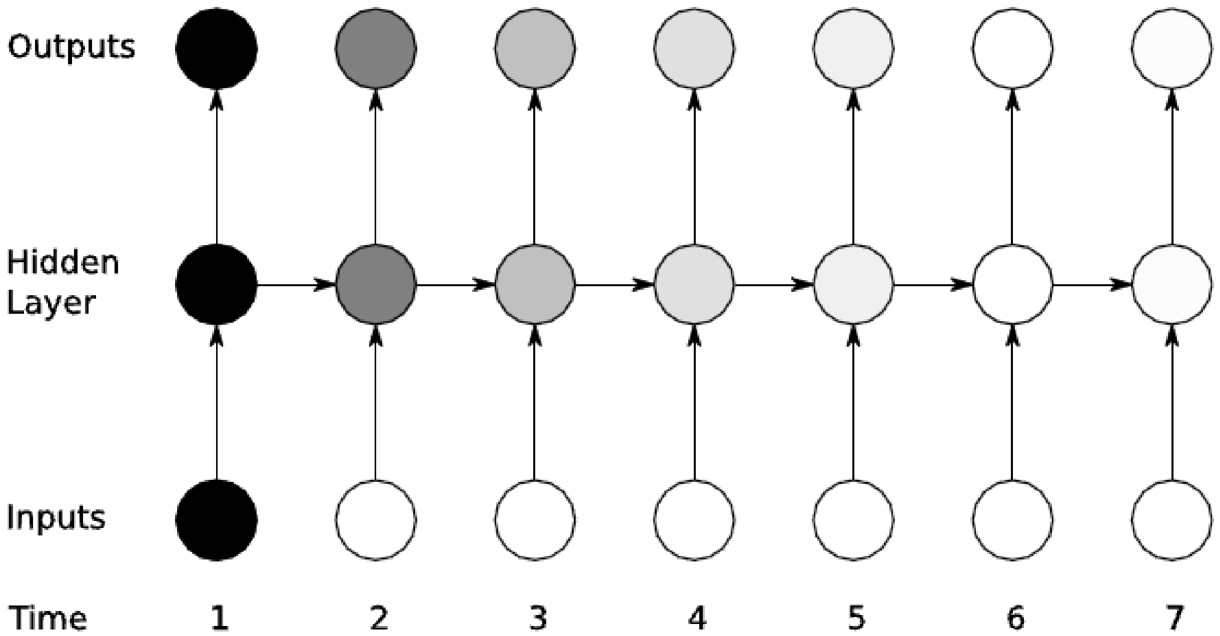
$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y$$

실시간은 아니다

past and future context determines the output

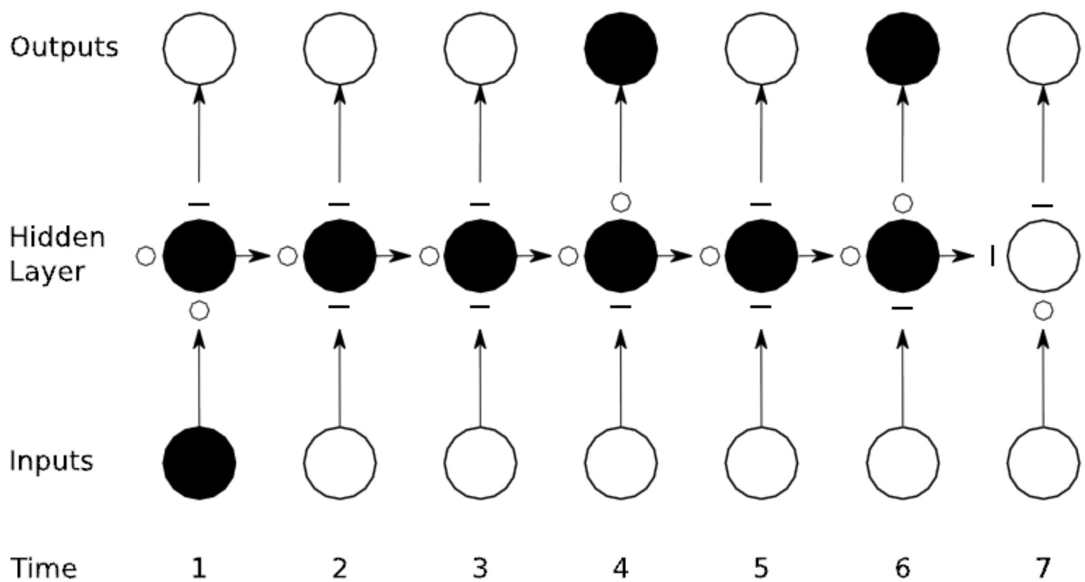
Vanishing Gradients

- Vanishing Gradients problem for basic RNNs
  - Influence of the inputs at time t decreases and vanishes over time



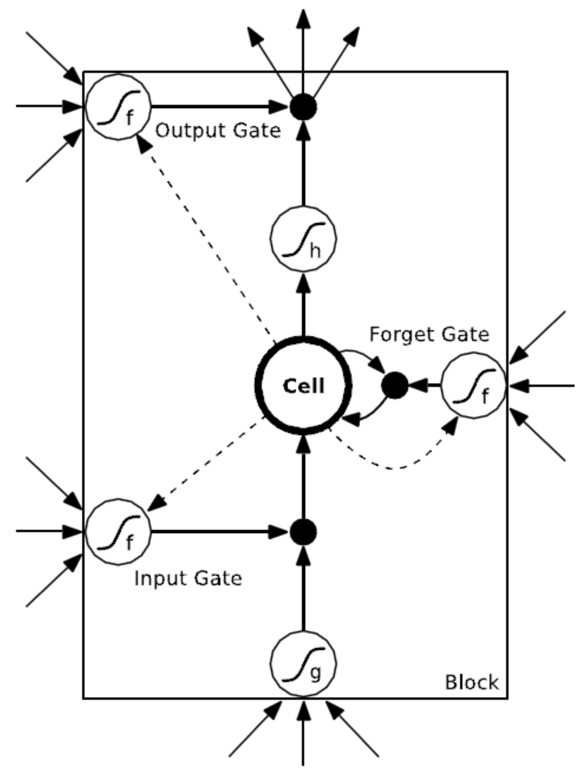
LSTM-RNNs

- LSTM can preserve gradient information
  - Hidden layer units formed with **Long Short-Term Memory (LSTM)** cells can store and access information over longer periods of time



Long Short-Term Memory

- LSTM block architecture
  - 3 gates
    - Input gate adjust the influence from input to cell
    - Forget gate adjust the influence from cell to cell over time
    - Output gate adjust the influence from cell to output



Long Short-Term Memory

- Notations for the next slides
  - Every weights from m to n : w_{mn}
  - Every inputs to unit j : a_j
  - Every outputs to unit j : b_j
    - State of cell c : s_c
  - Subscripts for units
    - input gate : ι
    - output gate : ω
    - forget gate : ϕ
  - Upper script t is for denoting time steps :

    e.g) a_j^t, b_j^t, δ_j^t

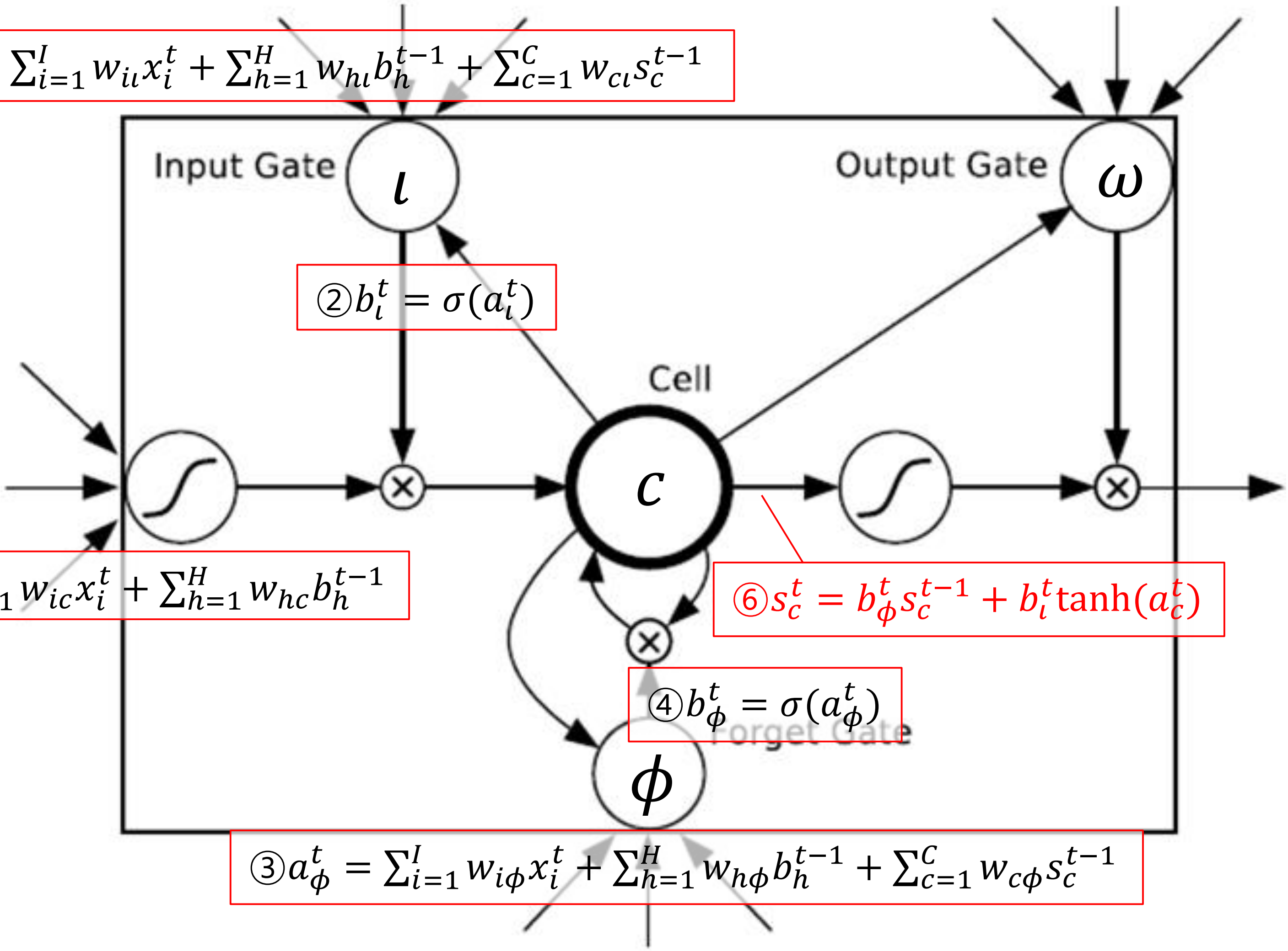
x_i^t: Input from the previous layer

b_h^t:Output to the next layer

s_c^t:Cell state value at time t

Long Short-Term Memory

$\sigma(x)$ *is the sigmoid function.*



①$a_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1} + \sum_{c=1}^{C} w_{c\iota} s_c^{t-1}$

②$b_\iota^t = \sigma(a_\iota^t)$

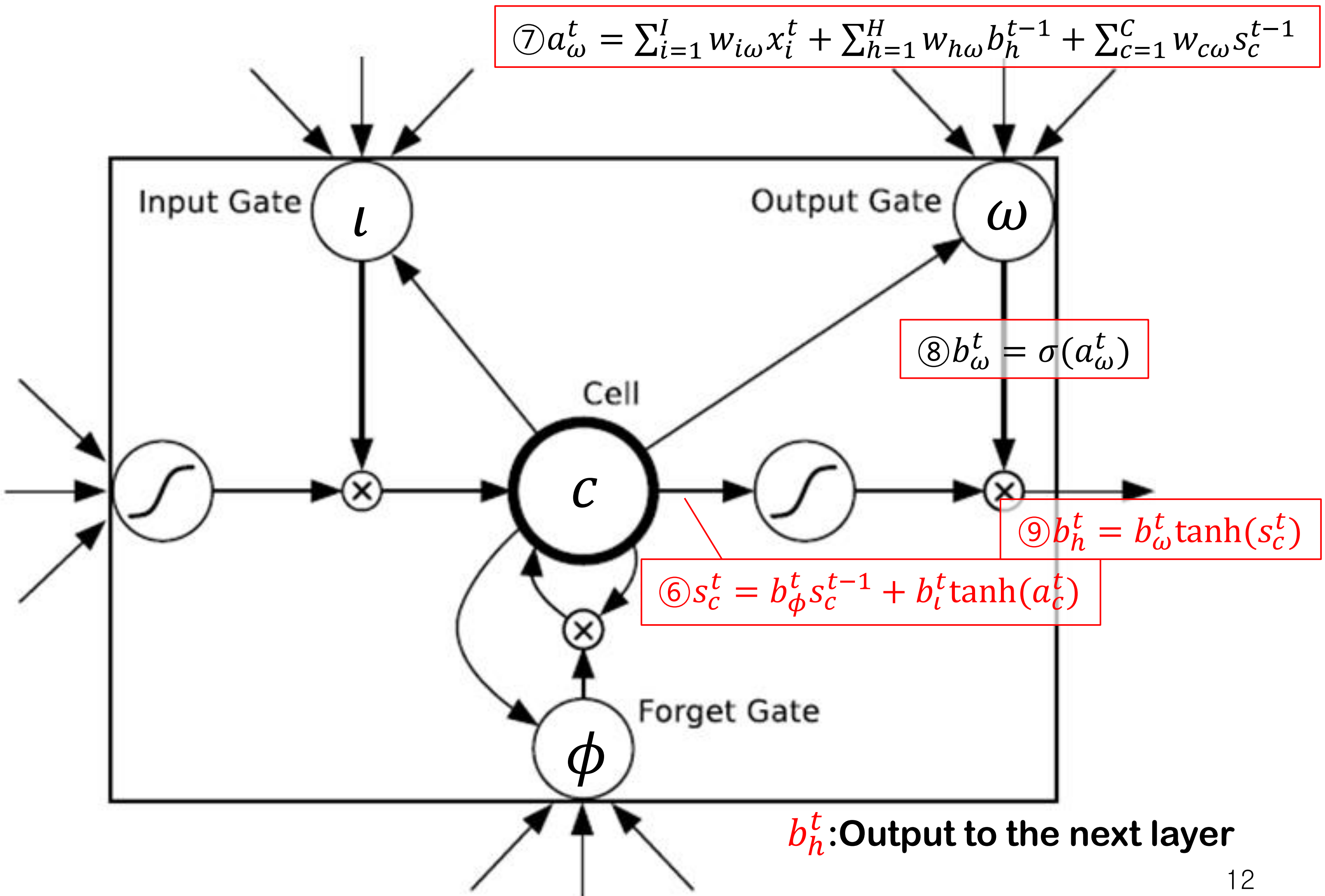
③$a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1}$

④$b_\phi^t = \sigma(a_\phi^t)$

⑤$a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$

⑥$s_c^t = b_\phi^t s_c^{t-1} + b_\iota^t \tanh(a_c^t)$

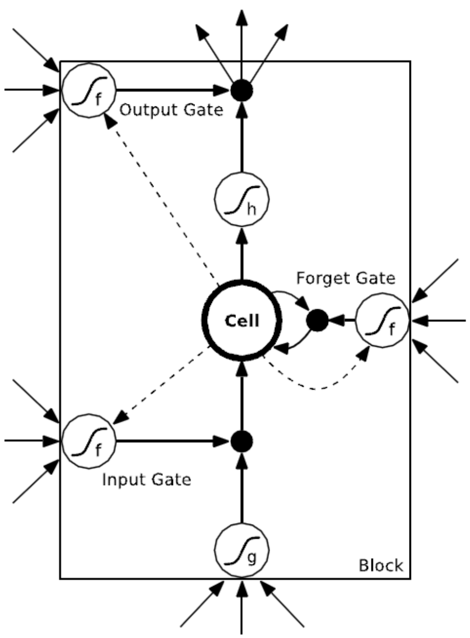
Long Short-Term Memory



b_h^t:**Output to the next layer**

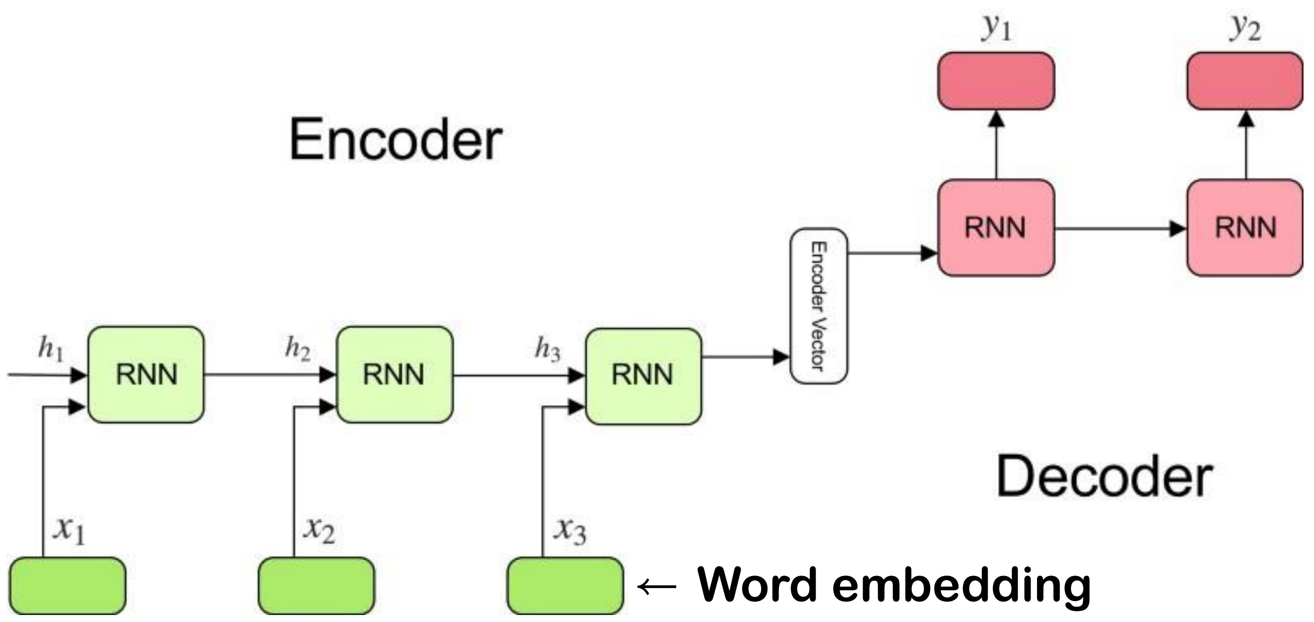
Long Short-Term Memory

- LSTM block architecture
  - LSTM block – input, output and forget gates
    - $b_\iota^t = \sigma(\sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1} + \sum_{c=1}^{C} w_{c\iota} s_c^{t-1})$
    - $b_\phi^t = \sigma(\sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1})$
    - $s_c^t = b_\phi^t s_c^{t-1} + b_\iota^t \tanh(\sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1})$
    - $b_\omega^t = \sigma(\sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{h=1}^{H} w_{h\omega} b_h^{t-1} + \sum_{c=1}^{C} w_{c\omega} s_c^{t-1})$
    - $b_h^t = b_\omega^t \tanh(s_c^t)$



Seq2Seq Encoder-Decoder using RNN

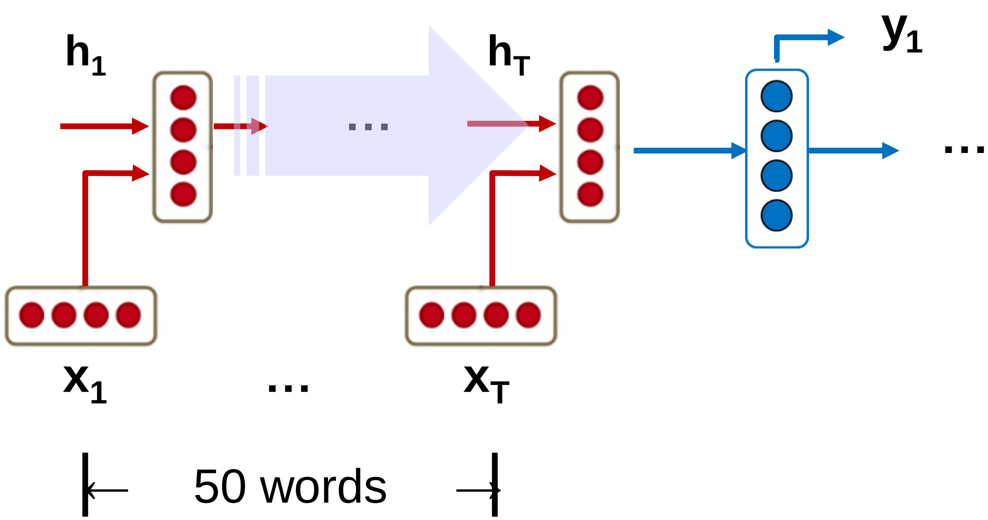
- We need to understand seq2seq encoder-decoder model to know the motivation of '**attention mechanism**'.
- Encoder: from **word sequence** to **sentence representation (a real-valued vector).**
- Decoder: from **representation** to **word sequence distribution**



← **Word embedding**

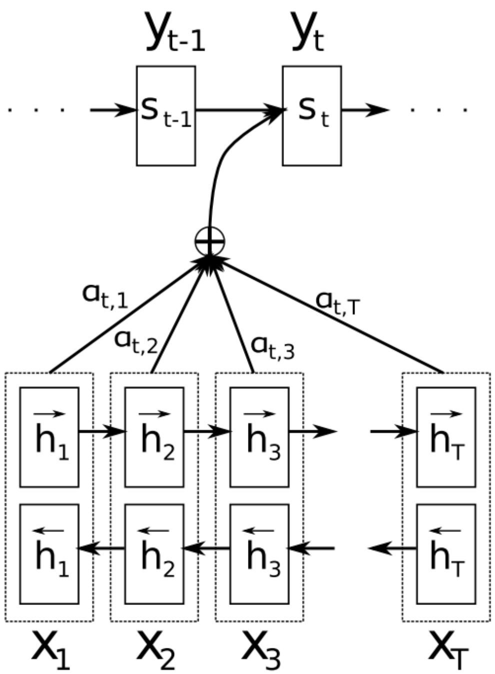
Attention Model – Motivation

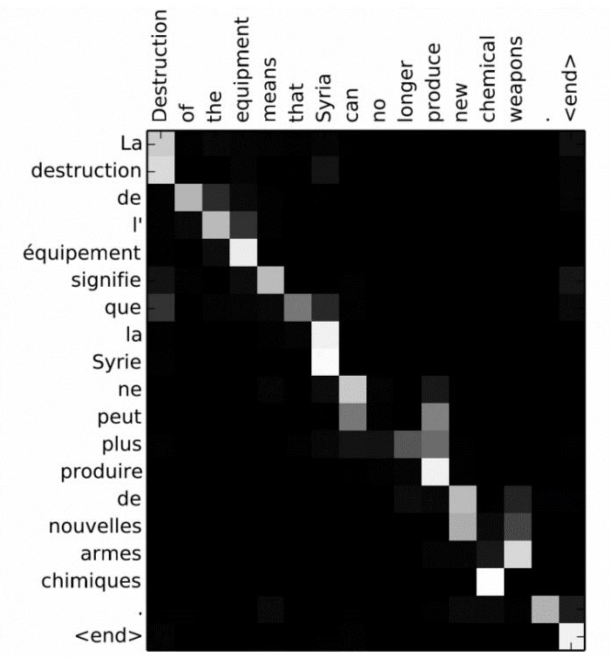
- Challenge in vanilla seq2seq for long sentences
  - Decoder generates a translation solely based on the last hidden state.
  - Information about the first word needs to be encoded in the last hidden state.



Intuition of Attention Mechanism

- Attention mechanism in decoder
  - The decoder decides which different parts of the source sentence to pay "attention" at each step of the output generation.





* Bahdanau, ICLR'15

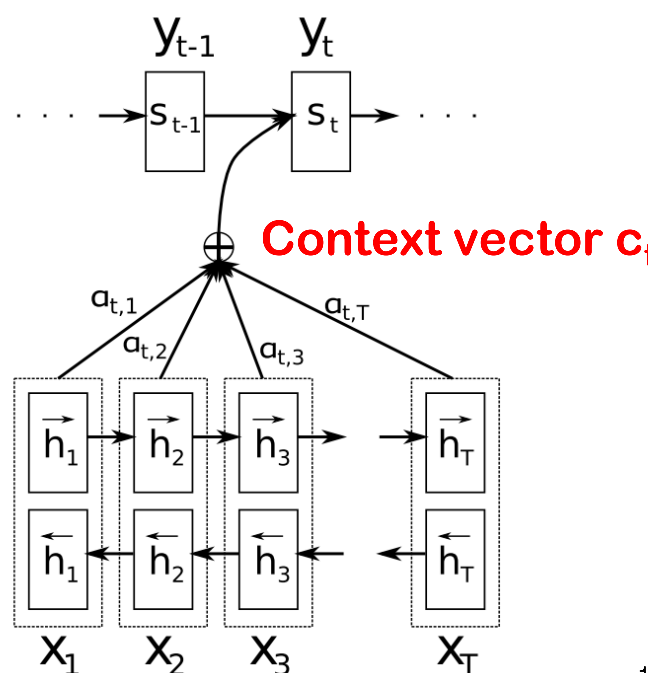
Attention Mechanism with Seq2Seq

- **RNN hidden state of the decoder at i:** $s_i = f(s_{i-1}, y_{i-1}, c_i)$

- **The context vector c_i is computed as a weighted sum of annotations $(h_1, \dots, h_T)$:**

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j$$

- **How to get attention weight α_{ij}:**
  Alignment score function



Attention Mechanism – Normalization

- Let α_{ij} be the probability that the target word y_i is aligned to (or translated from) a source word x_j.

- α_{ij} is computed by normalizing the probabilities with a softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

Attention Mechanism – Scoring

- **Alignment score function** $e_{ij} = \text{score}(s_{i-1}, h_j)$

  where s_{i-1} is the RNN hidden state just before emitting i th word, and h_j is the j th RNN hidden state of the input sentence.

- **It scores how well the inputs around position j and the output at position i match.**

- $$\text{score}(s_{i-1}, h_j) = \begin{cases} s_{i-1}{}^{\top} h_j \\ s_{i-1}{}^{\top} W_a h_j \\ v_a{}^{\top} \tanh(W_a [s_{i-1}; h_j]) \end{cases}$$

  single hidden layer network

Family of Attention Mechanisms

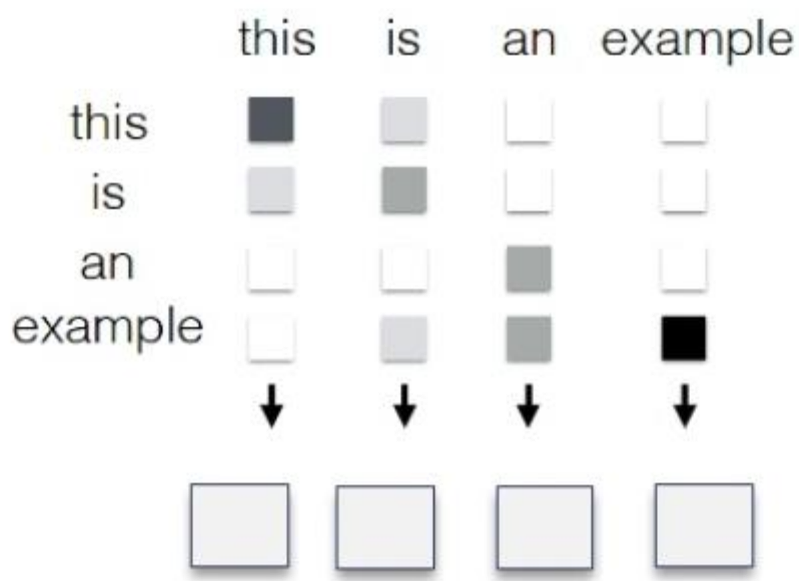
- **Popular attention mechanisms and their alignment score functions:**

Name	Alignment score function	Citation
Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	Graves2014
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.	Luong2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017

* https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

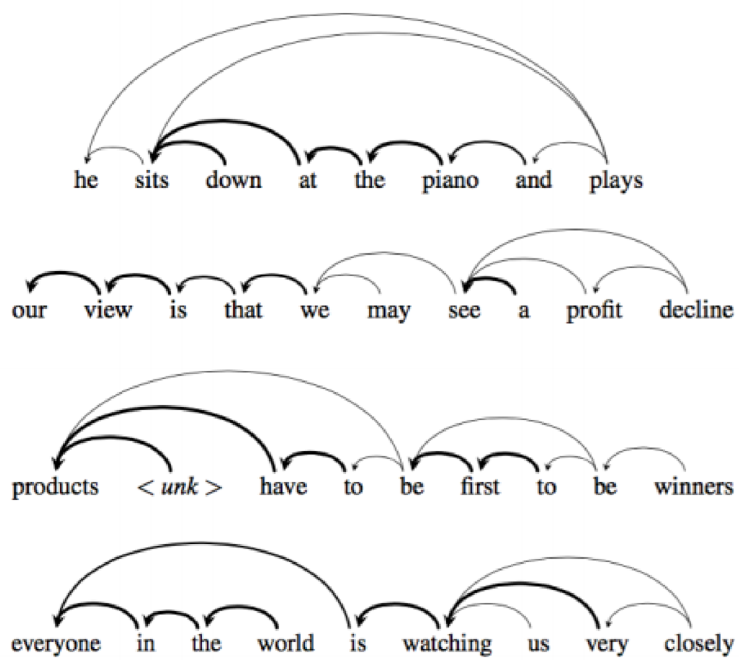
Self-Attention

- Self-attention (intra-attention) relates different positions of a single sequence.
  - Each element in the sentence attends to other elements from the same sentence -> context-sensitive encodings
  - Self-attention enhances the automatic understanding of text



Self-Attention

- Useful in machine reading
  - Self-attention enhances the automatic understanding of text
  - Tasks: language modeling or sentiment analysis



* **Long Short-Term Memory-Networks for Machine Reading, Cheng et al., 2016**

Evaluation Metrics : Bleu Score

- BLEU score evaluates the similarity between two sentences by counting matching n-grams.
  - Generally calculated as an average of unigram, bigram, trigram and 4-gram score.

hypothesis	reference
I like dogs.	I do like dogs.
(I,like), (like,dogs)	(I,do), (do, like), (like,dogs)
2-gram BLEU: 1/2 * (penalty of length) = 1/2 * exp(1- 4/3)	

Penalty of length: $$exp\left(1 - \frac{\text{lengh of reference}}{\text{length of hypothesis}}\right)$$

Other Evaluation Metrics

- **Perplexity** measures how well the learned probability distribution of words matches that of the input text.
  - Inverse probability of the test set, normalized by the number of words.
  - Often used for language modelling.

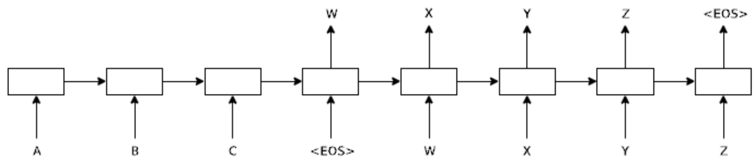
- **METEOR** is similar to BLEU but includes additional steps, like considering synonyms and comparing the stems of words.
  - Unlike BLEU, it is explicitly designed to compare sentences rather than corpora.
  - Ex: “running” and “runs” are counted as matches.

Applications of RNN – Machine translation

*experiment done by Ilya Sutskever et al.

- On WMT English to French dataset
  - 12M sentences with 348M French words and 304M English words
  - 4 layers of 1000 LSTM blocks each

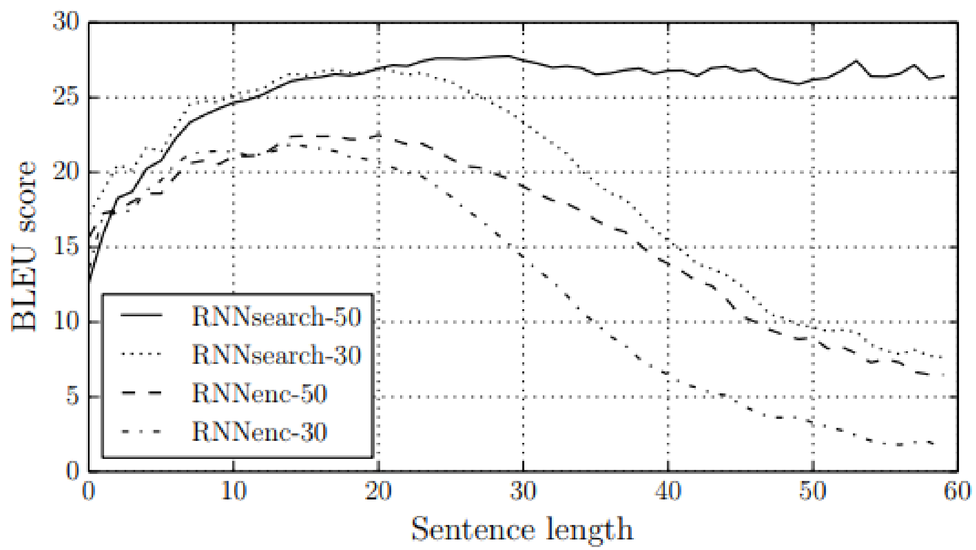
Type	Sentence
Our model	Ulrich UNK , membre du conseil d’ administration du constructeur automobile Audi , affirme qu’ il s’ agit d’ une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d’ administration afin qu’ ils ne soient pas utilisés comme appareils d’ écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d’ administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu’ ils ne puissent pas être utilisés comme appareils d’ écoute à distance , est une pratique courante depuis des années .
Our model	“ Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu’ ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu’ ils pourraient interférer avec les tours de téléphone cellulaire lorsqu’ ils sont dans l’ air ” , dit UNK .
Truth	“ Les téléphones portables sont véritablement un problème , non seulement parce qu’ ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d’ après la FCC , qu’ ils pourraient perturber les antennes-relais de téléphonie mobile s’ ils sont utilisés à bord ” , a déclaré Rosenker .
Our model	Avec la crémation , il y a un “ sentiment de violence contre le corps d’ un être cher ” , qui sera “ réduit à une pile de cendres ” en très peu de temps au lieu d’ un processus de décomposition “ qui accompagnera les étapes du deuil ” .
Truth	Il y a , avec la crémation , “ une violence faite au corps aimé ” , qui va être “ réduit à un tas de cendres ” en très peu de temps , et non après un processus de décomposition , qui “ accompagnerait les phases du deuil ” .



Machine Translation with Attention

*Bahdanau, ICLR'15

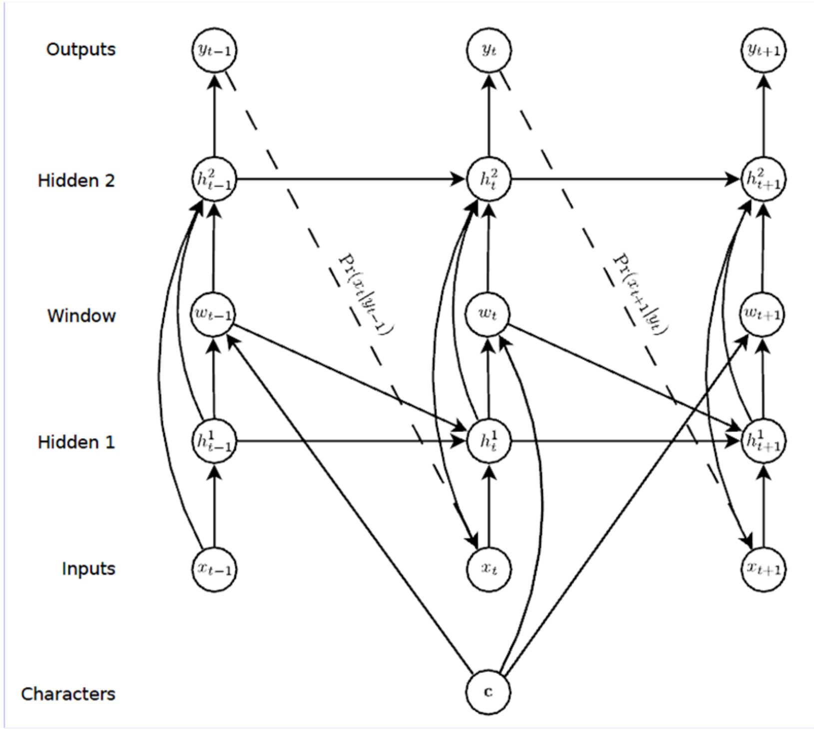
- BLEU score outperformed the conventional RNN enc-dec.
  - Robust to the sentence length when attention is applied.



Applications of RNN – Text sequence generation

*experiment done by Alex Graves et al.

- Demo available at
  - http://www.cs.toronto.edu/~graves/handwriting.html
- On IAM-OnDB
  - Using character-level transcriptions
  - 57 distinct characters
  - 3 layers of 400 LSTM blocks each



Applications of RNN
– Speech Recognition

- On TIMIT database
  - Audio data – phoneme classification
- 3 layers with 250 hidden LSTM block each
- Beats HMM (Hidden Markov Model) based models

```
3050 5723 she
5723 10337 had
9190 11517 your
11517 16334 dark
16334 21199 suit
21199 22560 in
22560 28064 greasy
28064 33360 wash
33754 37556 water
37556 40313 all
40313 44586 year
```

Applications of RNN – Question Answering (QA)

- We can assess questions and answers by encoding them with RNNs.
- QA tasks and their datasets:
  - Search over knowledge bases

    WebQuestions, WikiMovies, SimpleQuestions
  - Machine reading

    SQuAD, bAbI tasks, QACNN, CBT, MCTest, MS MARCO, WikiQA

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".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

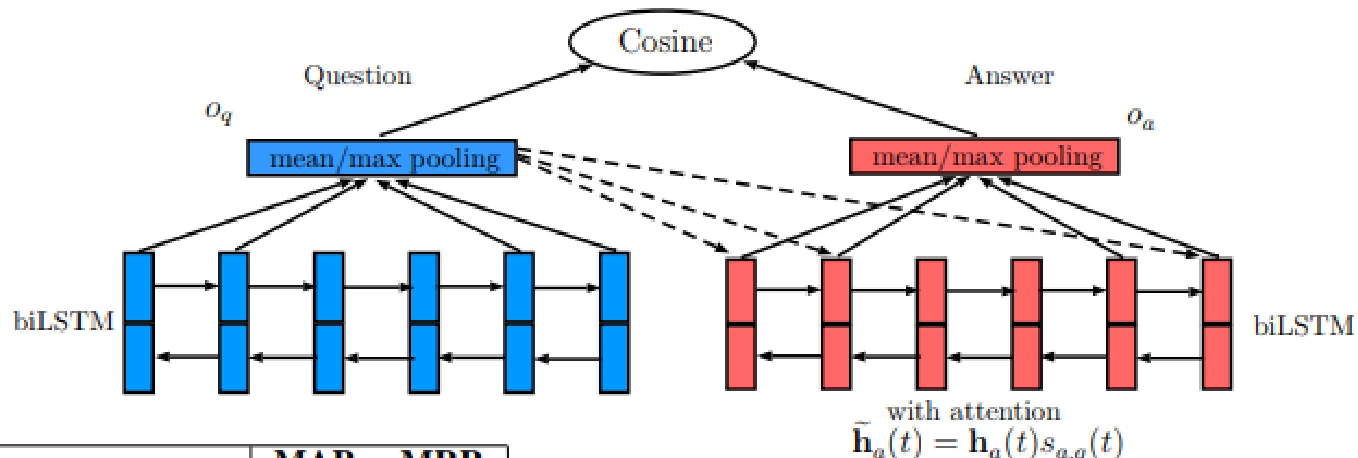
Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Figure 1: Question-answer pairs for a sample passage in the SQuAD dataset. Each of the answers is a segment of text from the passage.

QA with Attention

*Zhou et al., ICLR'16

- QA-LSTM + Attention
  - Attention vector from question and answer combined with question.



	Models	MAP	MRR
A	QA-LSTM (avg-pool)	68.19	76.52
B	QA-LSTM with attention	68.96	78.49
C	QA-LSTM/CNN	70.61	81.04
D	QA-LSTM/CNN with attention	**71.11**	**83.22**
E	QA-LSTM/CNN with attention (LSTM hiddenvector=500)	**72.79**	**82.40**

Applications of RNN – Image to text

- Generate or predict a text sequence from a given image
  - Image embedding is done by a CNN
  - Word embedding is done by an LSTM-RNN
  - Decoding to a text sequence is done by another LSTM-RNN
    - Content vector is multimodal vector in embedded space

Applications of RNN – Image to text

- Demo available at
  - http://www.cs.toronto.edu/~rkiros/lstm_scnlm.html

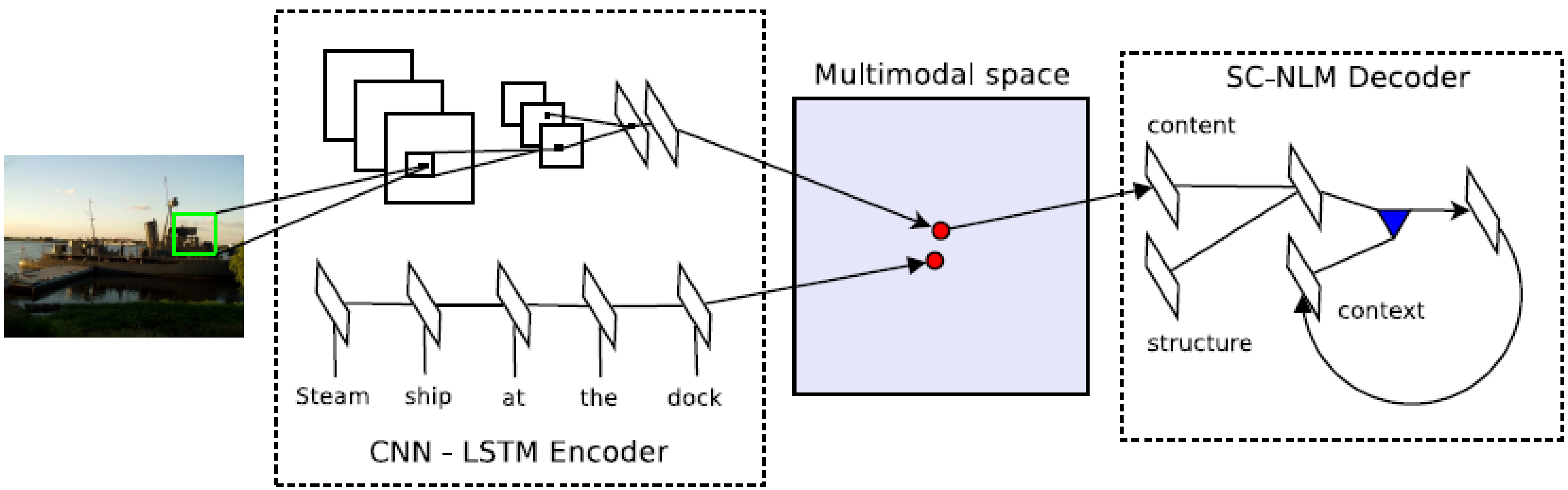
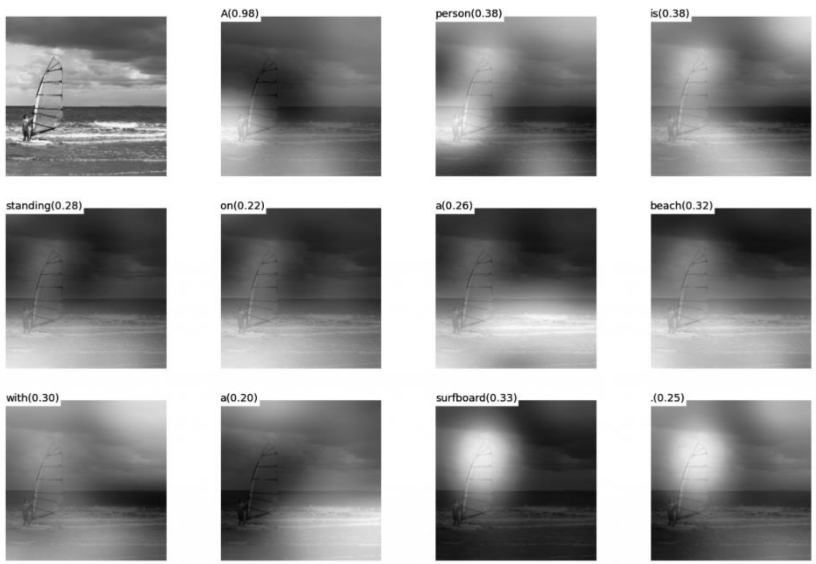
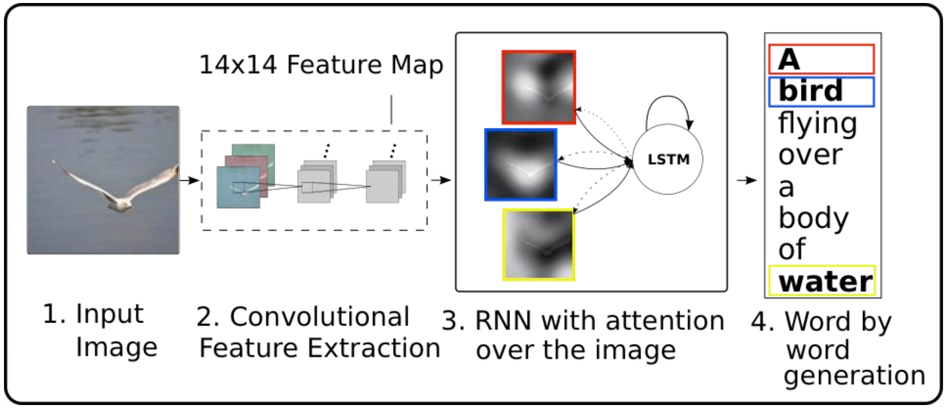


Image-to-Text with Attention

- RNN with self-attention consumes the convolution feature maps to generate the descriptive words one by one.

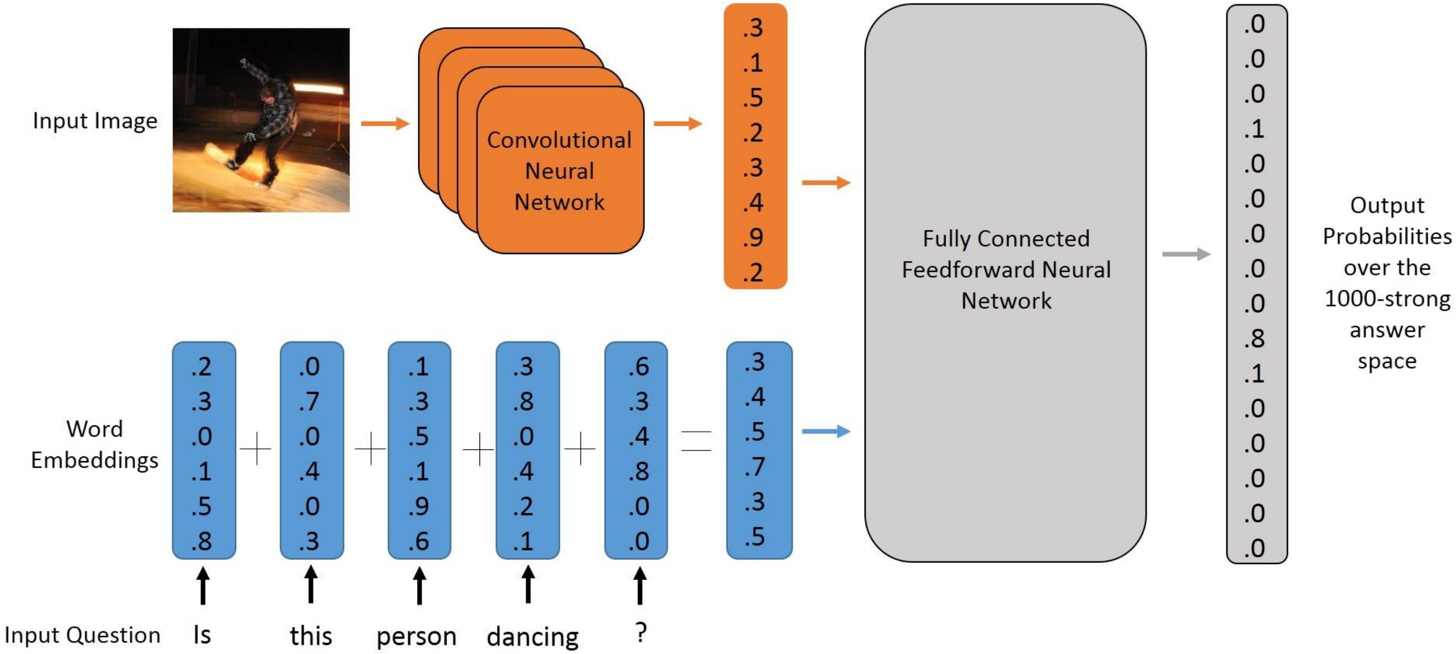


(b) A person is standing on a beach with a surfboard.



*Show, attend and tell, Xu et al., 2015

Applications of RNN
– Visual Question Answering (VQA)



The output is conditioned on both image and text inputs. A CNN is used to encode the image and a RNN is used to encode the sentence.

Sentiment analysis using CNNs

- Analysis based on Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification.
- A simple CNN with one layer of convolution on top of word vectors obtained from an unsupervised neural language model.
- Just use simple CNN with one layer outperform previous methods.
- Good results are obtained by using pre-trained word vector and multiple width CNN.

Sentiment analysis using CNNs

- Use multiple width convolution to obtain multiple features

$$c_i = f(w * x_{i:i+h-1} + b)$$

- After convolution, concatenate features and use softmax function, regularization

$$\boldsymbol{c} = [c_1, c_2, \dots, c_{n-h+1}]$$