Deep learning

Attention models

Hamid Beigy

Sharif University of Technology

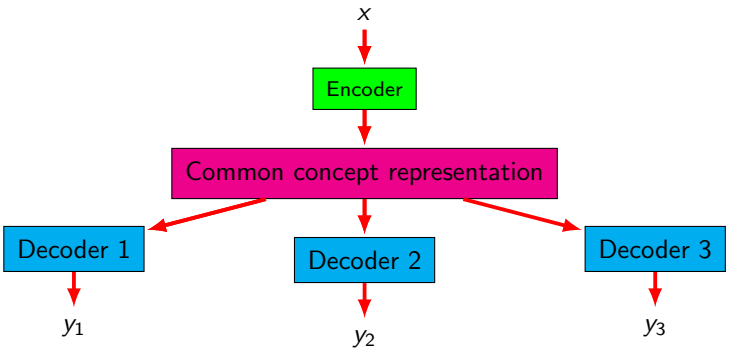
November 23, 2024

Table of contents

Introduction

Common concept representation

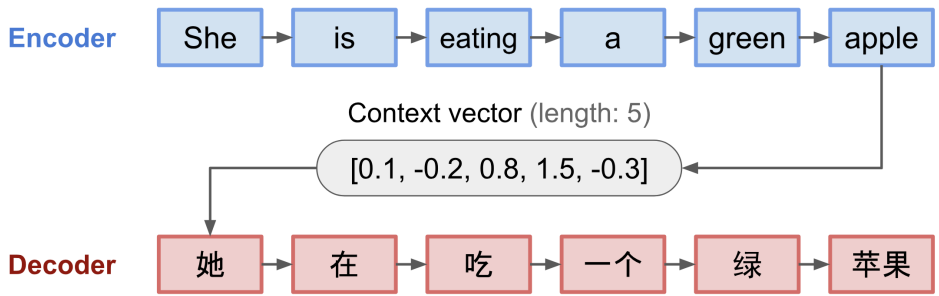
1. Consider the task of transferring a concept from a source domain to different target domains.



2. For example, consider the following tasks
   - A translation from Persian language to English language
   - A translation from Persian language to German language
   - A translation from Persian language to French language

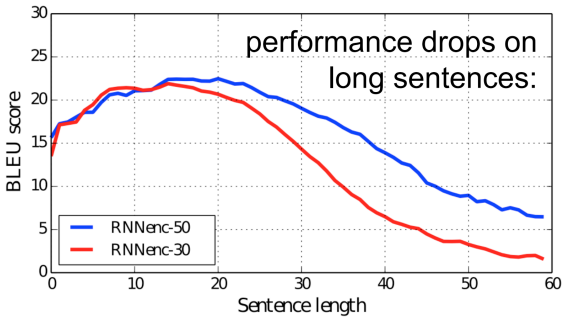
Sequence to sequence models

1. In seq2seq, the idea is to have two recurrent neural networks (RNNs) with an encoder-decoder architecture:
   - read the input words one by one to obtain a vector representation of a fixed dimensionality (encoder), and
   - conditioned on these inputs, extract the output words one by one using another RNN (decoder).
2. Both the encoder and decoder are recurrent neural networks such as LSTM or GRU units.



3. A critical disadvantage of this fixed-length context vector design is incapability of remembering long sentences.

1. RNNs cannot remember longer sentences and sequences due to the vanishing/exploding gradient problem.
2. The performance of the encoder-decoder network degrades rapidly as the length of the input sentence increases.



3. In psychology, attention is the cognitive process of selectively concentrating on one or a few things while ignoring others.

Example (Counting the number of people in a photo)

Counting the number of heads and ignoring the rest.

Attention models (examples)

1. Consider two different tasks : neural machine translation and image captioning.

neural machine translation (heatmap)

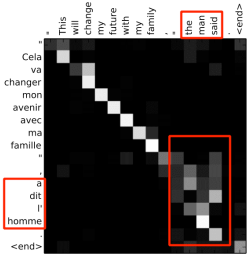
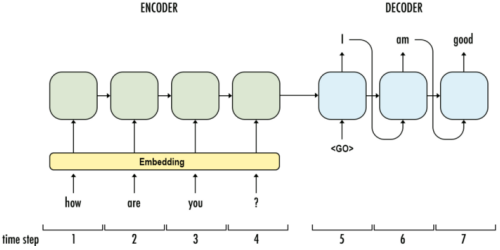
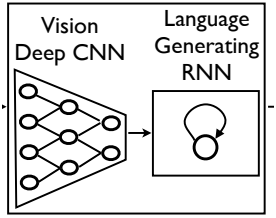


Image captioning

Neural network model



Neural network model



Attention models

Attention models

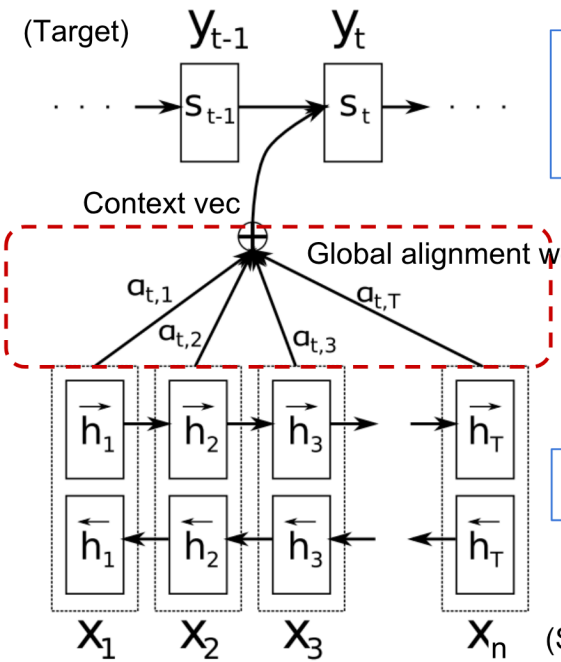
1. The attention mechanism was born to help memorize long source sentences in neural machine translation (NMT) (Bahdanau, Cho, and Bengio 2015).
2. Instead of building a single context vector out of the encoder's last hidden state, the goal of attention is to create shortcuts between the context vector and the entire source input.
3. The weights of these shortcut connections are customizable for each output element.
4. The alignment between the source and target is learned and controlled by the context vector.
5. Essentially the context vector consumes three pieces of information:
   - Encoder hidden states
   - Decoder hidden states
   - Alignment between source and target

1. Assume that we have a source sequence x of length n and try to output a target sequence y of length m

$$x = [x_1, x_2, \ldots, x_n]$$
$$y = [y_1, y_2, \ldots, y_m]$$

2. The encoder is a bidirectional RNN with a forward hidden state $\overrightarrow{h}_i$ and a backward one $\overleftarrow{h}_i$.
3. A simple concatenation of these two hidden states represents the encoder state.
4. The motivation is to include both the preceding and following words in the annotation of one word.

$$\boldsymbol{h}_i = \left[\overrightarrow{\boldsymbol{h}}_i^\top ; \overleftarrow{\boldsymbol{h}}_i^\top\right]^\top \quad i = 1, 2, \ldots, n$$



Decoder: RNN with input from previous state + dynamic context vector.

Attention layer: parameterized by a simple feed-forward network

Encoder: bidirectional RNN

1. The decoder network has hidden state $\boldsymbol{s}_t = f(\boldsymbol{s}_{t-1}, y_{t-1}, \mathbf{c}_t)$ at position $t = 1, 2, \ldots, m$.
2. The context vector $\mathbf{c}_t$ is a **sum** of **hidden states of the input sequence**, weighted by **alignment scores**:

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i \qquad \text{Context vector for output } y_t$$

$$\alpha_{t,i} = \text{align}(y_t, x_i) \qquad \text{How well two words } y_t \text{ and } x_i \text{ are aligned.}$$

$$= \frac{exp\left(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i)\right)}{\sum_{j=1}^{n} \exp\left(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_j)\right)} \qquad \text{Softmax of predefined alignment score.}$$

3. The **alignment model** assigns a score $\alpha_{t,i}$ to the pair of (y_t, x_i) based on how well they match.
4. The set of $\{\alpha_{t,i}\}$ are weights defining how much of each source hidden state should be considered for each output.

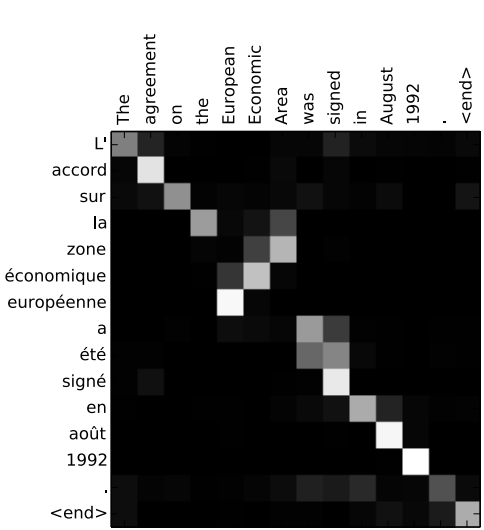
1. The **alignment score** α is parametrized by a **feed-forward network with a single hidden layer** (Bahdanau, Cho, and Bengio 2015).
2. **This network is jointly trained with other parts of the model.**
3. The score function is in the following form.

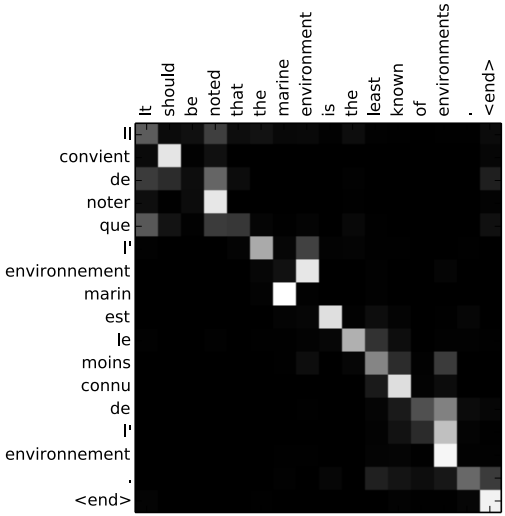
$$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$$

where both $\mathbf{V}_a$ and $\mathbf{W}_a$ are weight matrices to be learned in the alignment model.

Alignment scores

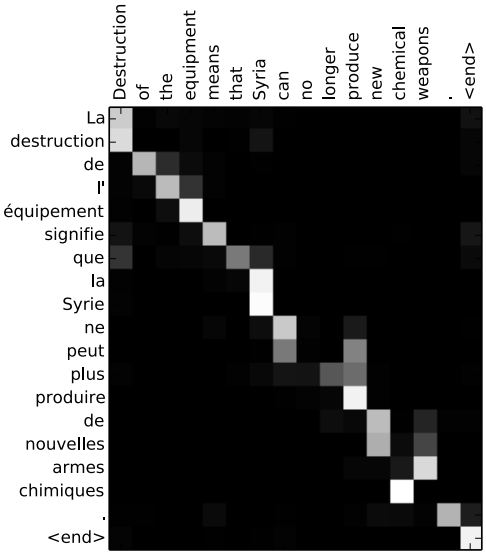
1. The matrix of alignment scores explicitly show the correlation between source and target words.

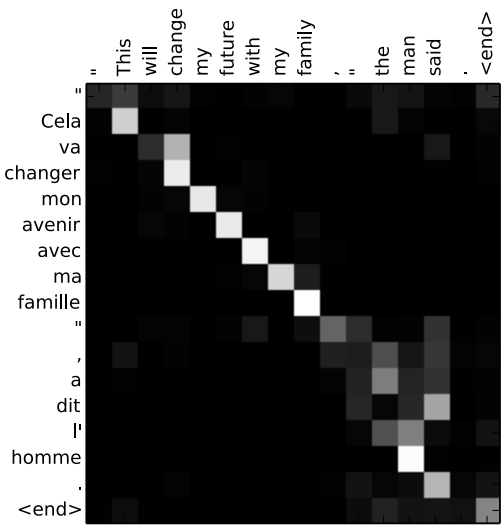




Alignment scores

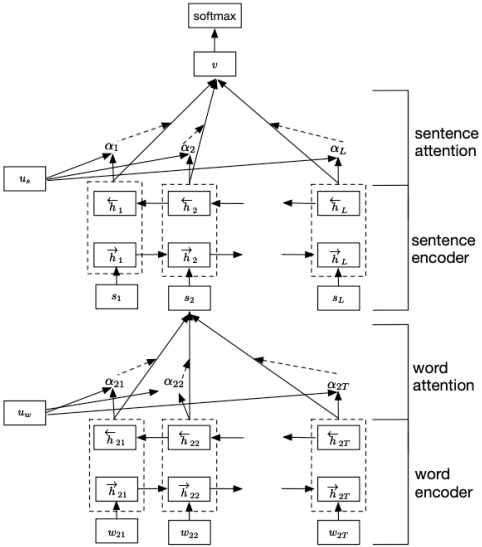
1. The matrix of alignment scores explicitly show the correlation between source and target words.





hierarchical attention network (HAN)

1. Attention can be effectively used on various levels (Yang et al. 2016).
2. HAN applicable to classification problem, not sequence generation.
3. HAN has two encoders: word and sentence.
   - Word encoder processes each word and aligns them a sentence of interest.
   - Then, sentence encoder aligns each sentence with final output.
4. HAN enables hierarchical interpretation of
   - which sentence is crucial in classifying document,
   - which part of a sentence (which words) are salient in that sentence.



1. Consider the following example

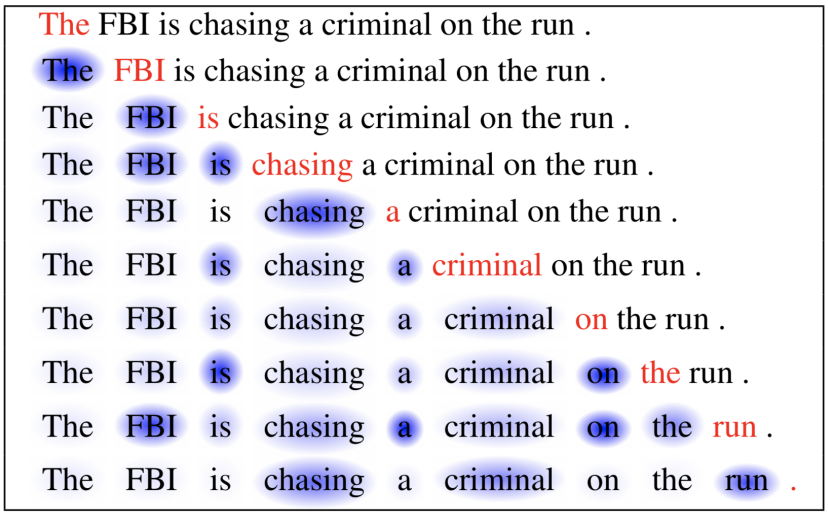
Example (Self-Attention)

- Consider the following sentence

  The animal didn't cross the street because it was too tired.
- What does **it** in this sentence refer to?
- Is it referring to **the street** or to **the animal**?

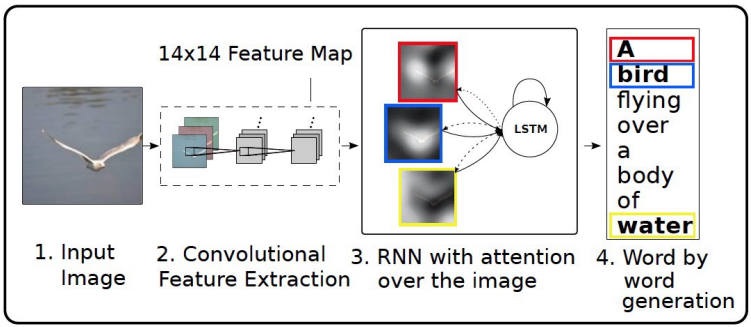
2. Self-attention (intra-attention) is an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence (Cheng, Dong, and Lapata 2016).
3. It is very useful in
   - Machine reading (the automatic, unsupervised understanding of text)
   - Abstractive summarization
   - Image description generation

1. The self-attention mechanism enables us to learn the correlation between the current words and the previous part of the sentence.



2. The current word is in red and the size of the blue shade indicates the activation level.

1. Self-attention is applied to the image to generate descriptions (Xu et al. 2015).



2. Image is encoded by a CNN and a RNN with self-attention consumes the CNN feature maps to generate the descriptive words one by one.
3. The visualization of the attention weights clearly demonstrates which regions of the image, the model pays attention to output a certain word.

Self Attention

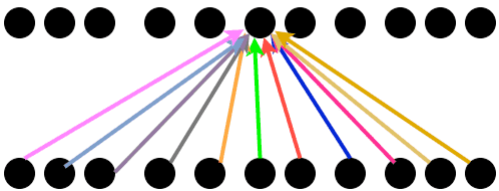
Soft vs Hard Attention

1. The soft vs hard attention is another way to categorize how attention is defined based on whether the attention has access to the entire image or only a patch.
   - Soft Attention: the alignment weights are learned and placed "softly" over all patches in the source image (same idea as in (Bahdanau, Cho, and Bengio 2015)).
     - Soft attention, in its simplest variant, is no different for images than for vector-valued features and is implemented exactly.
     - Pro: the model is smooth and differentiable.
     - Con: expensive when the source input is large.
   - Hard Attention: only selects one patch of the image to attend to at a time.
     - Hard attention for images has been known for a very long time: **image cropping**.
     - Pro: less calculation at the inference time.
     - Con: the model is non-differentiable and requires more complicated techniques such as variance reduction or reinforcement learning to train.

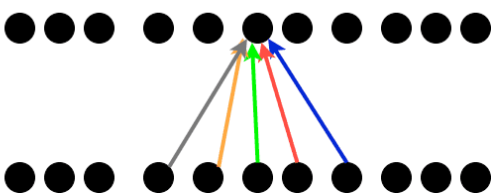
Global vs Local Attention

1. Global and local attention are proposed in (Luong, Pham, and Manning 2015).
2. The idea of a global attentional model is to consider all the hidden states of the encoder when deriving the context vector.

Global attention



Local attention



Global vs Local Attention

1. The global attention has a drawback that it has to attend to all words on the source side for each target word, which is expensive and can potentially render it impractical to translate longer sequences,
2. The local attentional mechanism chooses to focus only on a small subset of the source positions per target word.
3. Local one is an interesting blend between hard and soft, an improvement over the hard attention to make it differentiable:
4. The model first predicts a single aligned position for the current target word and a window centered around the source position is then used to compute a context vector.

$$\boldsymbol{p}_t = n \times \mathit{sigmoid}\left(\mathbf{v}_p^\top \tanh(\mathbf{W}_p \boldsymbol{h}_t)\right)$$

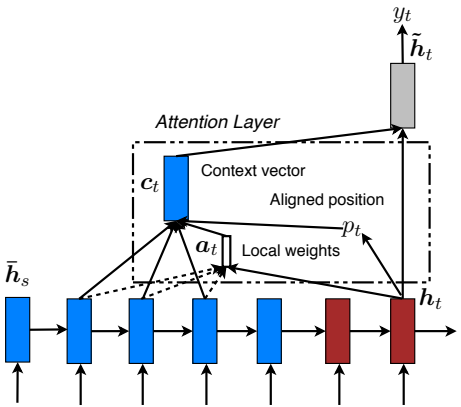
n is length of source sequence. Hence, $\boldsymbol{p}_t \in [0, n]$.

1. To favor alignment points near $\boldsymbol{p}_t$, they placed a Gaussian distribution centered around $\boldsymbol{p}_t$. Specifically, the alignment weights are defined as

$$a_{st} = align(h_t, \bar{h}_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right)$$

and

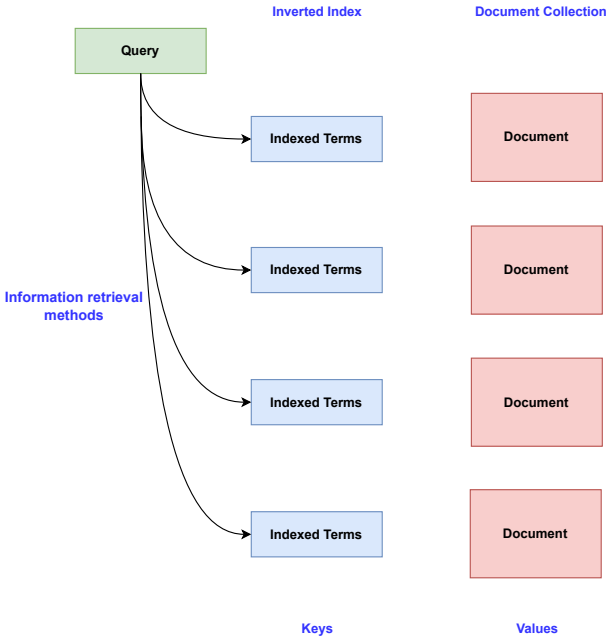
$$\boldsymbol{p}_t = n \times sigmoid\left(\mathbf{v}_p^\top \tanh(\mathbf{W}_p \boldsymbol{h}_t)\right)$$



Generalized model of attention

Retrieving documents in an information retrieval system

1. Consider an information retrieval system,



Retrieving a record from a relational database

1. Consider the following table, called PERSONS, in a relational database.

ID	Name	Family
005123174812	Ali	Ahmadi
015843268901	Mohammad Reza	Ali Mohammadi
005123174823	Ashkan	Mohammadi

2. Now consider the following queries.
   - SELECT ID, Name, Family FROM PERSONS WHERE ID='015843268901'
   - SELECT ID, Name, Family FROM PERSONS WHERE ID like '00512317%'
3. Here, concepts of **query**, **key**, and **value** become, and the result is retrieved using the following similarity function.

$$Similarity(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \sum_i Similarity(\mathbf{q}, \mathbf{k}_i) \times \mathbf{v}_i$$

Retrieving a value from neural Turing machine memory

1. Consider the following memory in the neural Turing machine.

Key	Value
key 1	Value 1
key 2	Value 2
key 3	Value 3

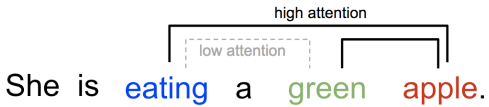
2. When reading from the memory at time t, an attention vector of size p, $\mathbf{w}_t$ controls how much attention to assign to different memory locations.
3. The read vector $\mathbf{r}_t$ is a sum weighted by attention intensity:

$$\mathbf{r}_t = \sum_{i=1}^{p} w_t(i)\mathbf{M}_t(i)$$

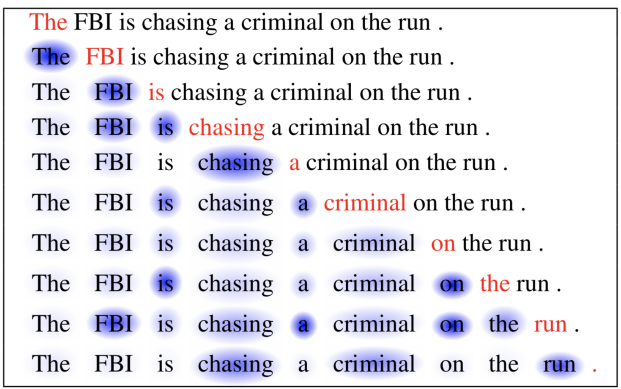
$$\sum_{i=1}^{p} w_t(i) = 1, \forall i : 0 \leq w_t(i) \leq 1$$

Generalized model of attention

1. Consider the following sentence.



2. For calculating the attention of a target word with respect to the input word,
   - we first use the query of the target word and the key of the input word,
   - next calculate a matching score, and
   - finally calculate the weighted sum of value vectors using the matching scores.

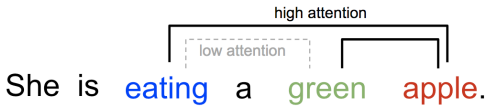


Generalized model of attention

1. Each word is **query**, **key**, and **value**.
2. Each word w is represented by a vector $\mathbf{x} \in \mathbb{R}^d$ by using an word embedding method.
3. Calculate query ($\mathbf{q} \in \mathbb{R}^p$) for $\mathbf{x} \in \mathbb{R}^d$, which is projection of $\mathbf{x}$ to a new space.
$$\mathbf{q} = \mathbf{w}_q^\top \mathbf{x}.$$
4. Calculate key ($\mathbf{k} \in \mathbb{R}^p$) for $\mathbf{x} \in \mathbb{R}^d$, which is projection of $\mathbf{x}$ to a new space.
$$\mathbf{k} = \mathbf{w}_k^\top \mathbf{x}.$$
5. Calculate value ($\mathbf{w} \in \mathbb{R}^p$) for $\mathbf{x} \in \mathbb{R}^d$, which is projection of $\mathbf{x}$ to a new space.
$$\mathbf{v} = \mathbf{w}_v^\top \mathbf{x}.$$
6. **A single word x has three different representations.** Sometimes, we look at this word as query, sometimes as key, and sometimes as value.
7. The self-attention means that looking a word as query and compute the similarity of the query with all of the words seen as key.
8. Then use the softmax for computing the weights and compute the weighted average all of the words seen as value.
9. This computes the attention vector.

Generalized model of attention

1. Consider the following sentence.



2. Calculating the attention for word apple.
3. Taking the inner product of the query vector of apple to the key vector of the previous words.

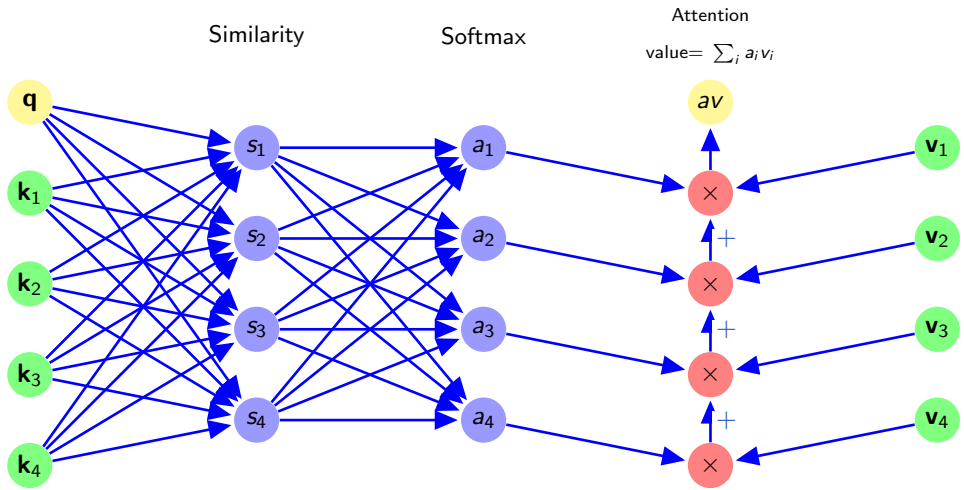
$$\mathbf{a} = softmax\left(\mathbf{q}_{apple}^{\top}\mathbf{k}_{she}, \mathbf{q}_{apple}^{\top}\mathbf{k}_{is}, \mathbf{q}_{apple}^{\top}\mathbf{k}_{eating}, \mathbf{q}_{apple}^{\top}\mathbf{k}_{a}, \mathbf{q}_{apple}^{\top}\mathbf{k}_{green}\right)$$

4. Suppose that we obtain $\mathbf{a} = (0.1, 0.1, 0.5, 0.1, 0.2)$. Then we obtain

$$\mathbf{v}_{apple} = 0.1\mathbf{v}_{she} + 0.1\mathbf{v}_{is} + 0.5\mathbf{v}_{eating} + 0.1\mathbf{v}_{a} + 0.2\mathbf{v}_{green}$$

Generalized model of attention

1. Self-attention uses the following neural network architecture.



Generalized model of attention

1. By defining three different vectors corresponding to each word.
   - Key $\mathbf{k} \in \mathbb{R}^p$ and $\mathbf{k} = \mathbf{W}_k^\top \mathbf{x}$, where $\mathbf{W}_k \in \mathbb{R}^{d \times p}$ and $\mathbf{x} \in \mathbb{R}^d$.
   - Query $\mathbf{q} \in \mathbb{R}^p$ and $\mathbf{q} = \mathbf{W}_q^\top \mathbf{x}$, where $\mathbf{W}_q \in \mathbb{R}^{d \times p}$ and $\mathbf{x} \in \mathbb{R}^d$.
   - Value $\mathbf{v} \in \mathbb{R}^p$ and $\mathbf{v} = \mathbf{W}_v^\top \mathbf{x}$, where $\mathbf{W}_v \in \mathbb{R}^{d \times p}$ and $\mathbf{x} \in \mathbb{R}^d$.
2. By defining the following matrices
   - $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where $\mathbf{X} \in \mathbb{R}^{d \times n}$.
   - $\mathbf{K} = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_n]$, where $\mathbf{K} \in \mathbb{R}^{p \times n}$.
   - $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]$, where $\mathbf{Q} \in \mathbb{R}^{p \times n}$.
   - $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$, where $\mathbf{V} \in \mathbb{R}^{p \times n}$.
3. Then, the new value $\mathbf{Z} \in \mathbb{R}^{p \times n}$ equals to

$$\mathbf{Z} = \mathbf{V}\ \mathit{Softmax}\left(\frac{\mathbf{Q}^\top \mathbf{K}}{\sqrt{p}}\right) = \mathbf{V}\ \frac{\mathbf{Q}^\top \mathbf{K}}{\sqrt{p} \sum_{r \in S} \mathbf{Q}^\top \mathbf{K}_r}$$

where S is input sentence.

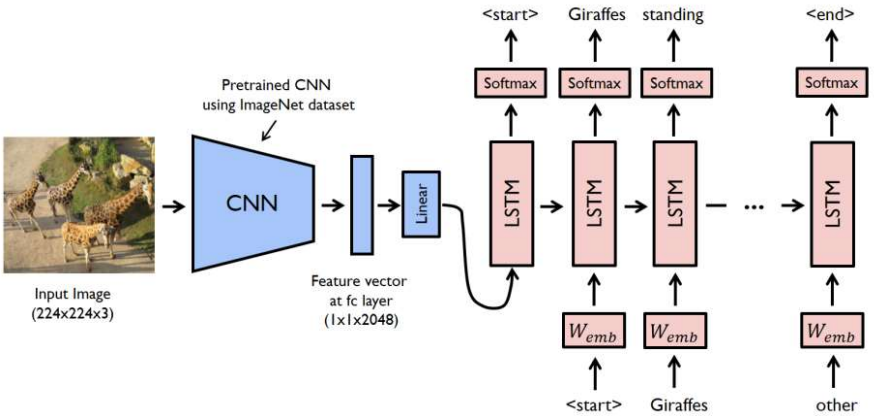
Alignment scores

Name	Alignment score function	Paper
Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	A. Graves, et al. "Neural Turing machines", arXiv, 2014.
Additive	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	D. Bahdanau, et al."Neural machine translation by jointly learning to align and translate", ICLR 2015.
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$	T. Luong, , et al. "Effective Approaches to Attention-based Neural Machine Translation", EMNLP 2015.
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$	Same as the above
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Same as the above
Scaled Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{p}}$	A. Vaswani, et al. "Attention is all you need", NIPS 2017.

Attention in computer vision

Image captioning

1. The natural image caption generator was proposed in (Xu et al. 2015).



2. This network is a combination of CNN and LSTM networks.

1. The outputs of lower layers of CNN are used as representation of values.

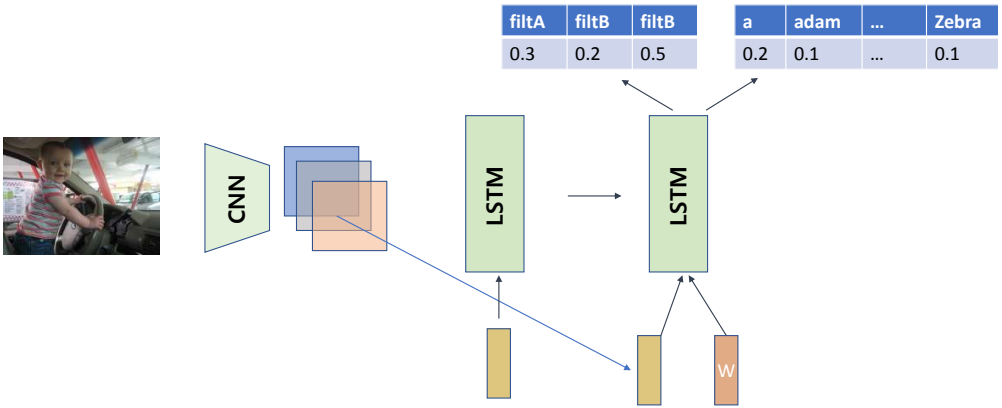
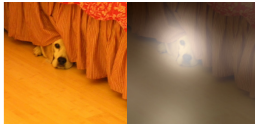


Image captioning results

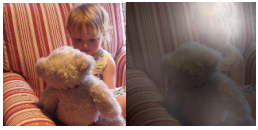
1. Examples of attending to the correct object

A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

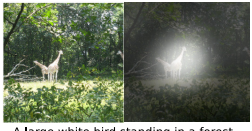


A little girl sitting on a bed with a teddy bear.

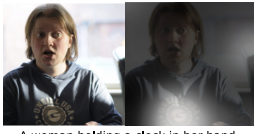
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

2. Examples of mistakes



A large white bird standing in a forest.



A woman holding a clock in her hand.

A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.

A woman is sitting at a table with a large pizza.

A man is talking on his cell phone while another man watches.

Image captioning results

1. There is also a method given in (Vinyals et al. 2015).

Reading

References i

- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *International Conference on Learning Representations*.
- Cheng, Jianpeng, Li Dong, and Mirella Lapata (2016). "Long Short-Term Memory-Networks for Machine Reading". In: *Proceedings of Conference on Empirical Methods in Natural Language Processing, EMNLP*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh, pp. 551–561.
- Luong, Thang, Hieu Pham, and Christopher D. Manning (2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421.
- Vinyals, Oriol et al. (2015). "Show and tell: A neural image caption generator". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164.
- Xu, Kelvin et al. (2015). "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37, pp. 2048–2057.
- Yang, Zichao et al. (2016). "Hierarchical Attention Networks for Document Classification". In: *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489.

Questions?