COMS 4995 Lecture 9: Attention

Richard Zemel

Overview

- We have seen a few RNN-based sequence prediction models.
- It is still challenging to generate long sequences, when the decoders only has access to the final hidden states from the encoder.
  - Machine translation: it's hard to summarize long sentences in a single vector, so let's allow the decoder to peek at the input.
  - Vision: have a network glance at one part of an image at a time, so that we can understand what information it's using
- This lecture will introduce attention that drastically improves the performance on the long sequences.
- We can also use attention to build differentiable computers (e.g. Neural Turing Machines)

Overview

- Attention-based models scale very well with the amount of training data. After 40GB text from reddit, the model generates:

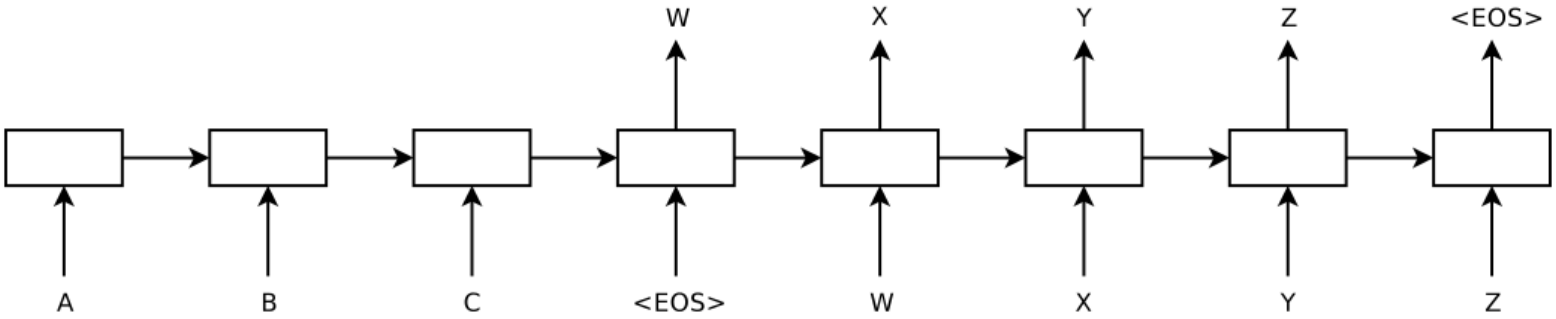
Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

For the full text samples see Radford, Alec, et al. "Language Models are Unsupervised Multitask Learners." 2019.

https://talktotransformer.com/

Attention-Based Machine Translation

- Remember the encoder/decoder architecture for machine translation:



- The network reads a sentence and stores all the information in its hidden units.
- Some sentences can be really long. Can we really store all the information in a vector of hidden units?
  - Let's make things easier by letting the decoder refer to the input sentence.

Attention-Based Machine Translation

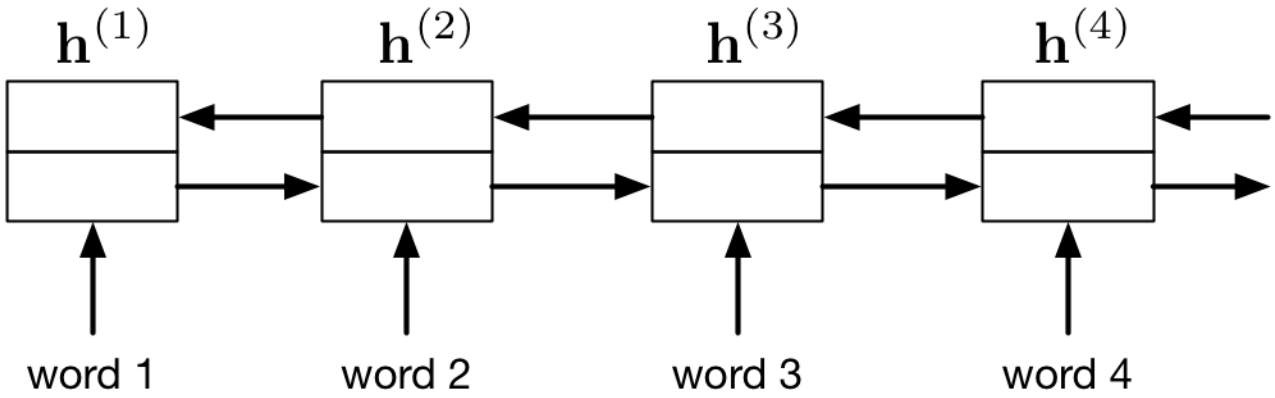
- We'll look at the translation model from the classic paper:

  Bahdanau et al., Neural machine translation by jointly learning to align and translate. ICLR, 2015.

- Basic idea: each output word comes from one word, or a handful of words, from the input. Maybe we can learn to attend to only the relevant ones as we produce the output.

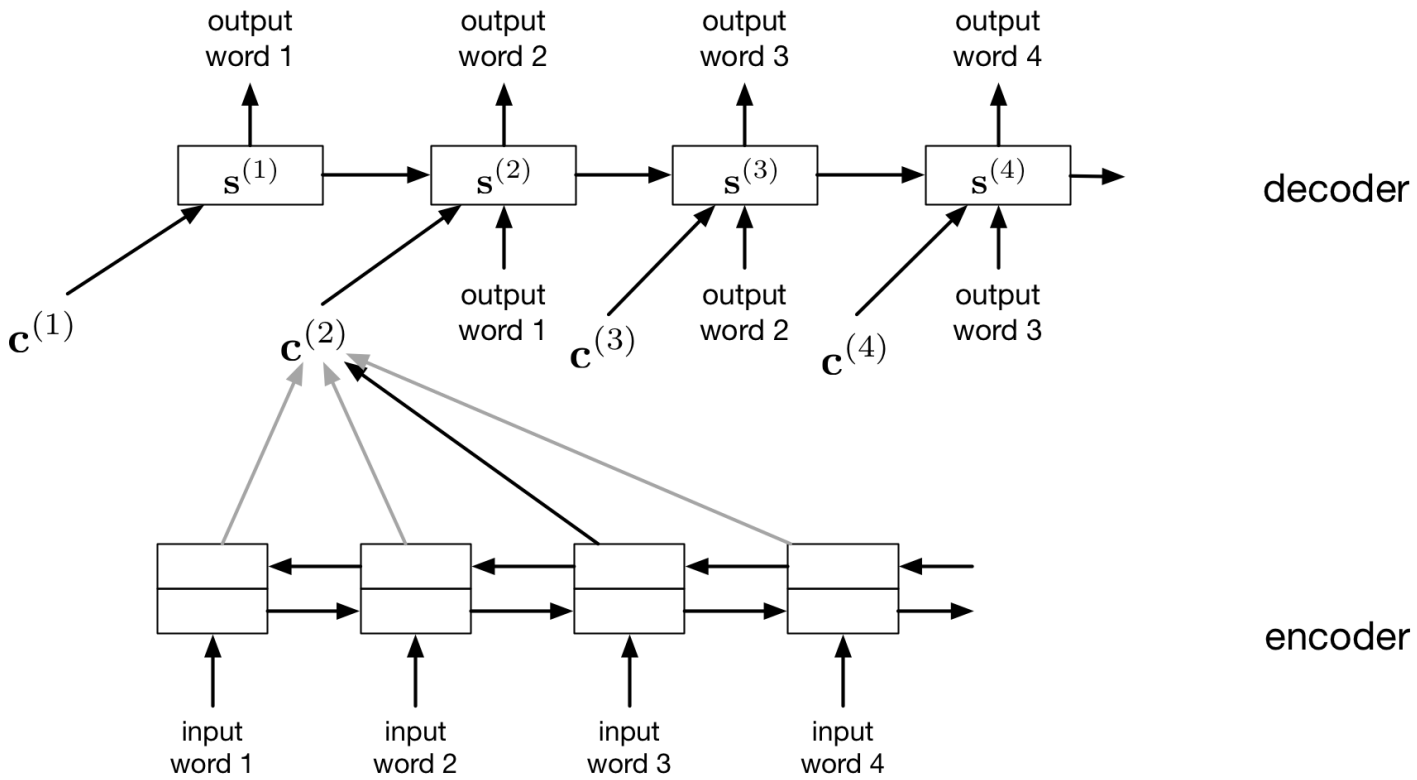
Attention-Based Machine Translation

- The model has both an encoder and a decoder. The encoder computes an annotation of each word in the input.
- It takes the form of a bidirectional RNN. This just means we have an RNN that runs forwards and an RNN that runs backwards, and we concatenate their hidden vectors.
  - The idea: information earlier or later in the sentence can help disambiguate a word, so we need both directions.
  - The RNN uses an LSTM-like architecture called gated recurrent units.



Attention-Based Machine Translation

- The decoder network is also an RNN. Like the encoder/decoder translation model, it makes predictions one word at a time, and its predictions are fed back in as inputs.
- The difference is that it also receives a context vector $\mathbf{c}^{(t)}$ at each time step, which is computed by attending to the inputs.



Attention-Based Machine Translation

- The context vector is computed as a weighted average of the encoder's annotations.

$$\mathbf{c}^{(i)} = \sum_j \alpha_{ij} h^{(j)}$$

- The attention weights are computed as a softmax, where the inputs depend on the annotation and the decoder's state:

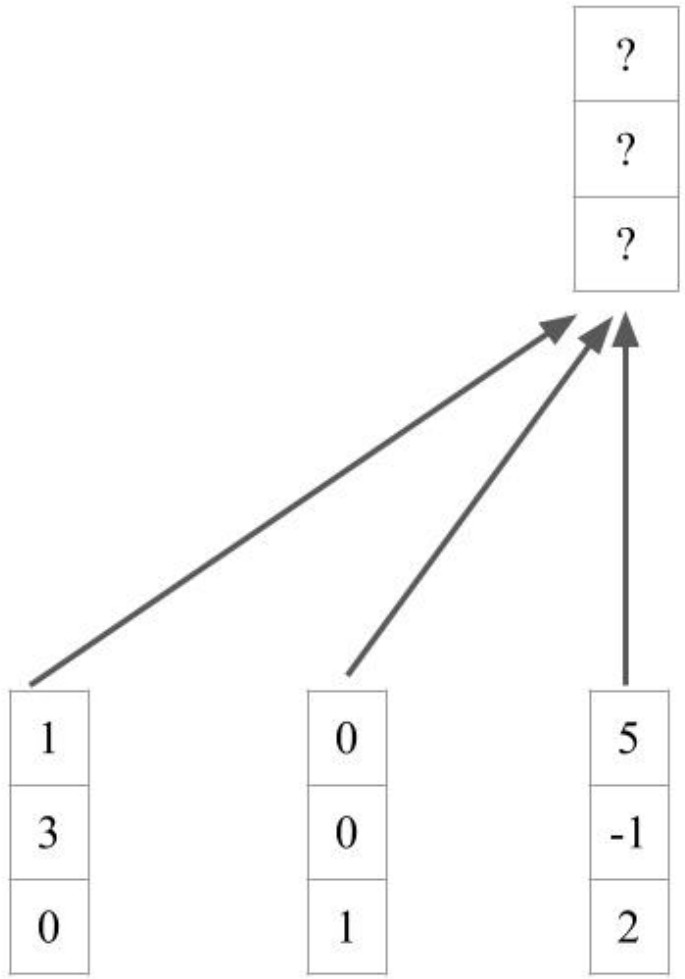
$$\alpha_{ij} = \frac{\exp(\tilde{\alpha}_{ij})}{\sum_{j'} \exp(\tilde{\alpha}_{ij'})}$$

$$\tilde{\alpha}_{ij} = f(\mathbf{s}^{(i-1)}, \mathbf{h}^{(j)})$$

- Note that the attention function, f depends on the annotation vector, rather than the position in the sentence. This means it's a form of content-based addressing.
  - My language model tells me the next word should be an adjective. Find me an adjective in the input.

Example: Pooling

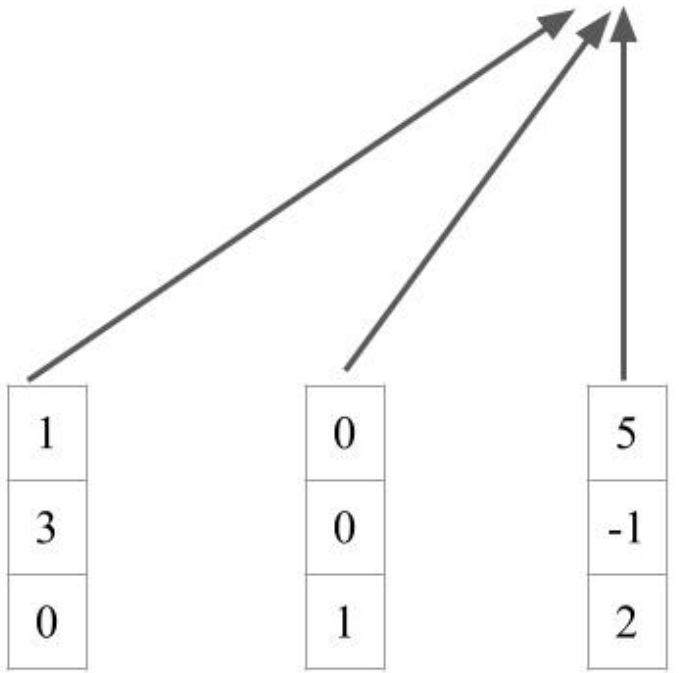
Consider obtain a context vector from a set of annotations.



Example: Pooling

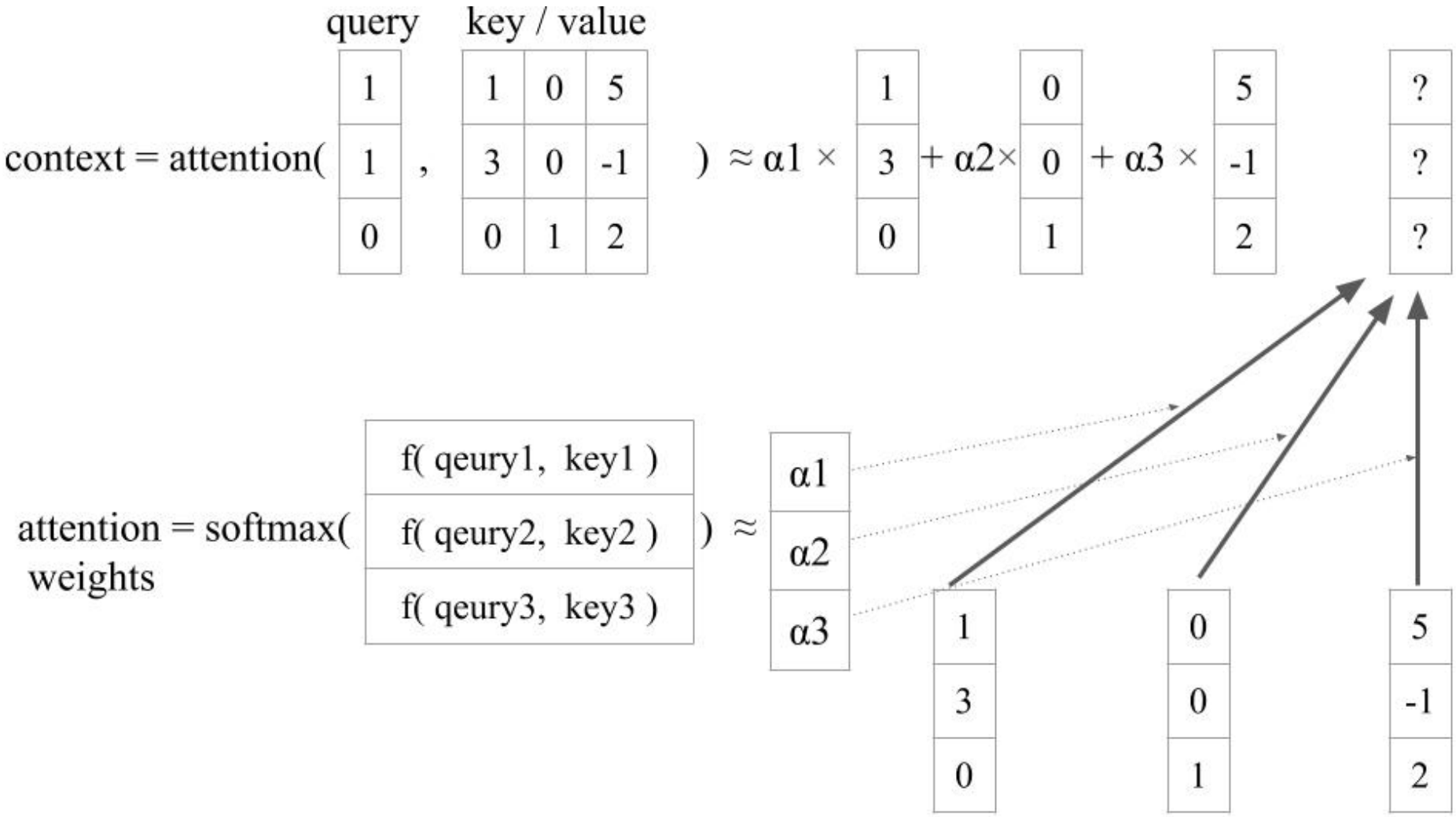
We can use average pooling but it is content independent.

$$\text{context} = \text{avg-pooling}\left(\begin{bmatrix}1 & 0 & 5\\ 3 & 0 & -1\\ 0 & 1 & 2\end{bmatrix}\right) = 0.33\times\begin{bmatrix}1\\3\\0\end{bmatrix} + 0.33\times\begin{bmatrix}0\\0\\1\end{bmatrix} + 0.33\times\begin{bmatrix}5\\-1\\2\end{bmatrix} = \begin{bmatrix}2\\0.6\\1\end{bmatrix}$$



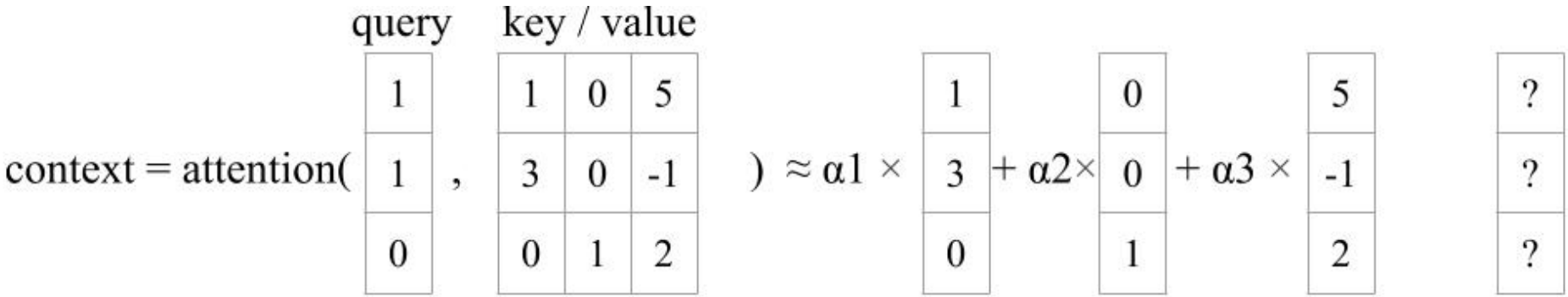
Example1: Bahdanau's Attention

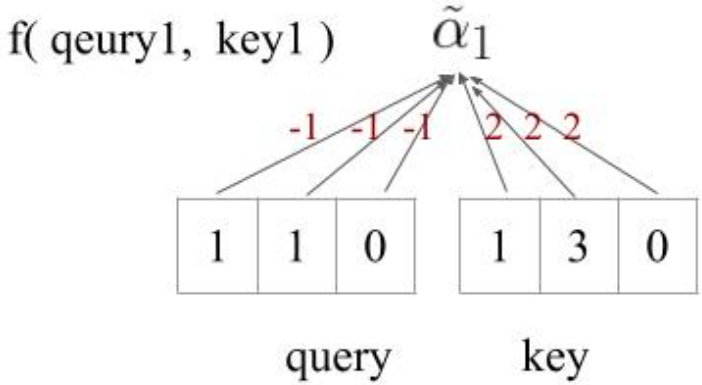
Content-based addressing/lookup using attention.

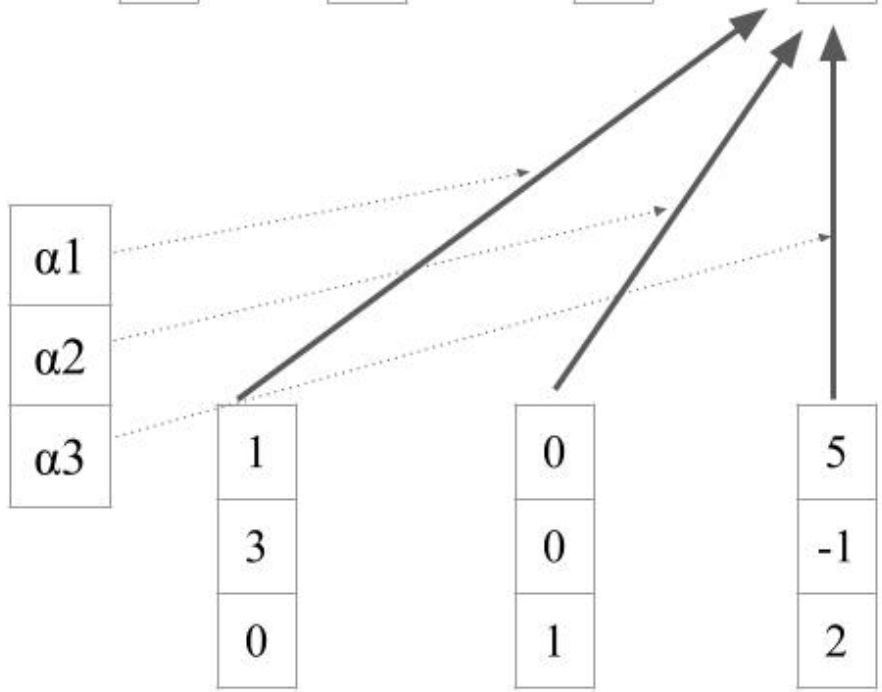


Example1: Bahdanau's Attention

Consider a linear attention function, f.

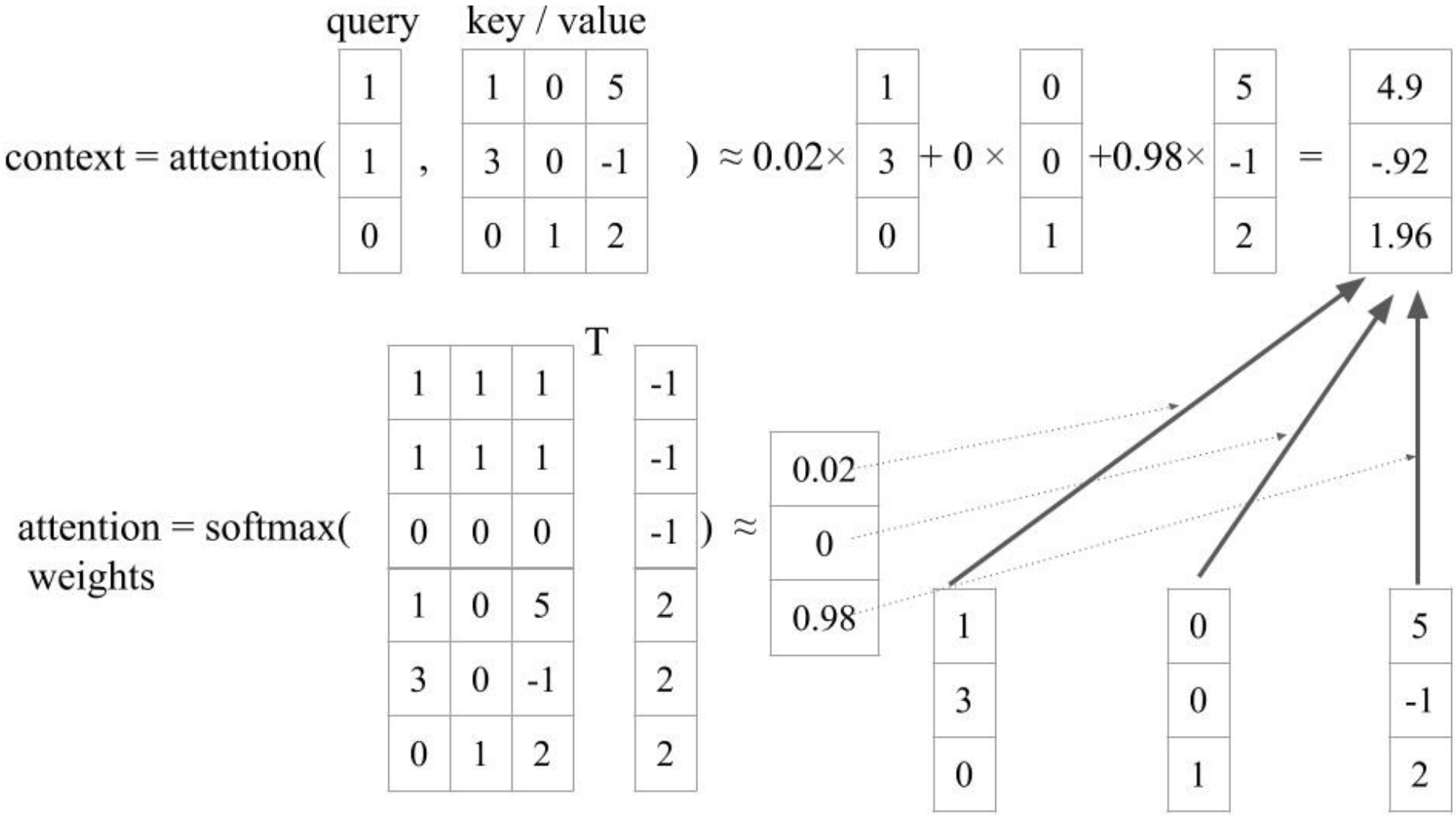






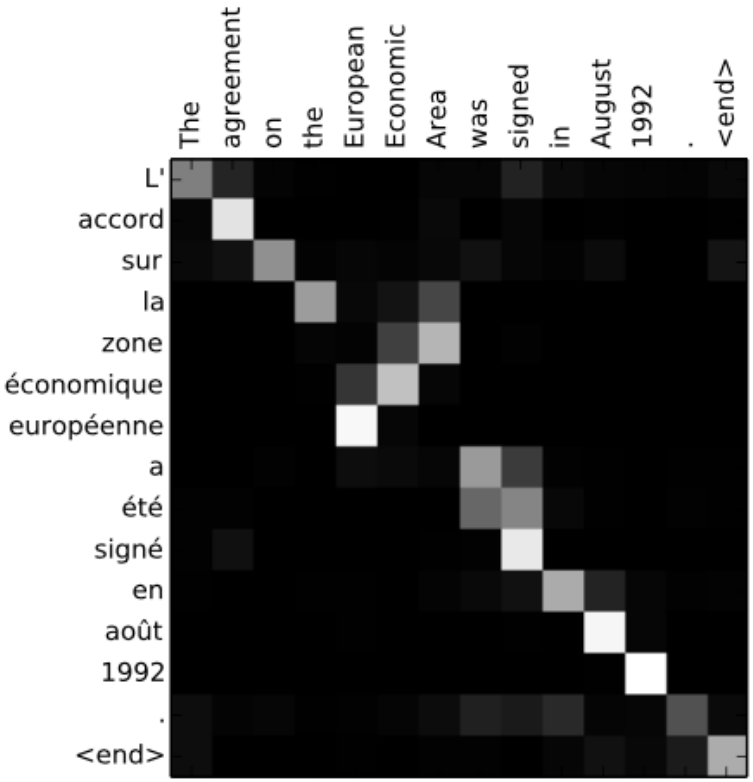
Example1: Bahdanau's attention

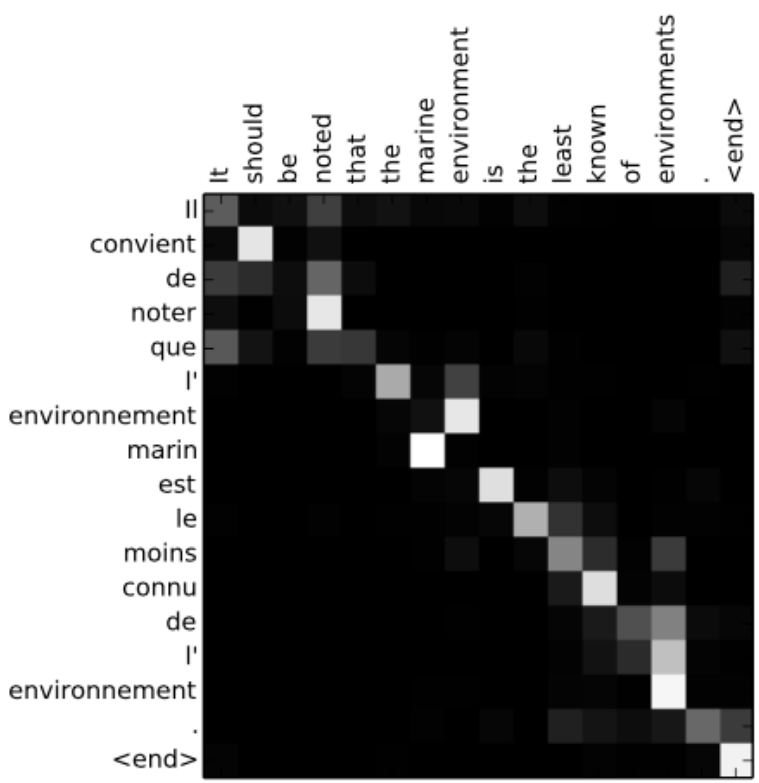
Vectorized linear attention function.



Attention-Based Machine Translation

- Here's a visualization of the attention maps at each time step.

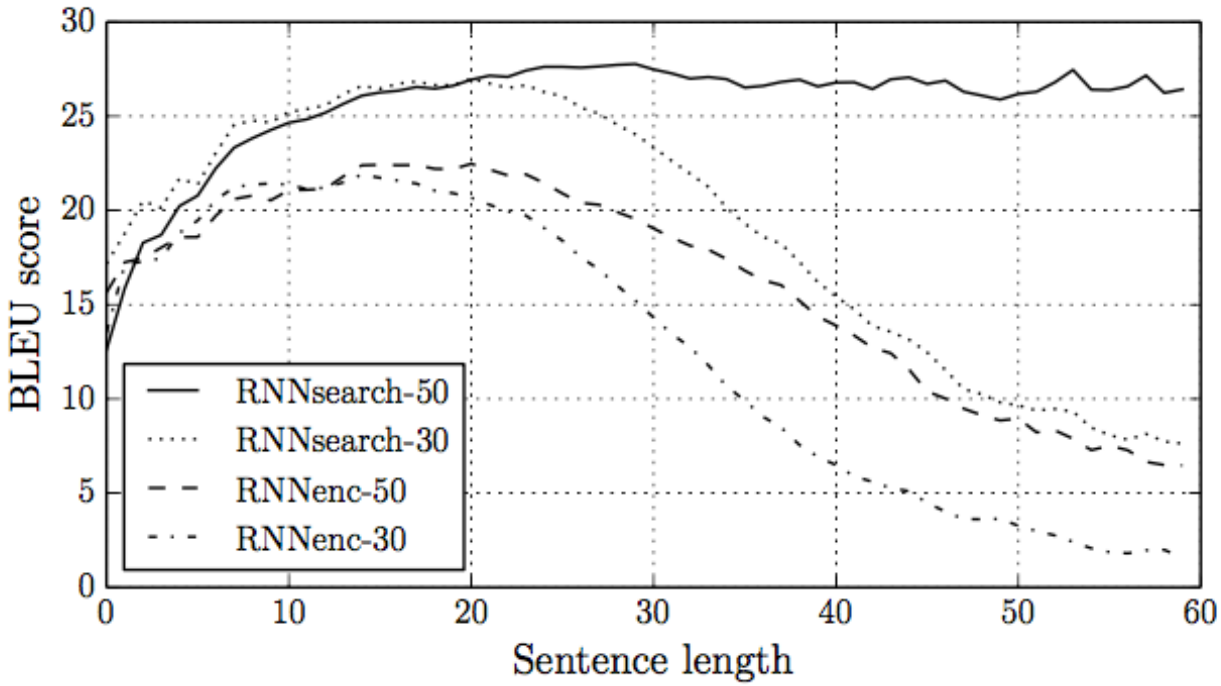




- Nothing forces the model to go linearly through the input sentence, but somehow it learns to do it.
  - It's not perfectly linear — e.g., French adjectives can come after the nouns.

Attention-Based Machine Translation

- The attention-based translation model does much better than the encoder/decoder model on long sentences.



Attention-Based Caption Generation

- Attention can also be used to understand images.
- We humans can't process a whole visual scene at once.
  - The fovea of the eye gives us high-acuity vision in only a tiny region of our field of view.
  - Instead, we must integrate information from a series of glimpses.
- The next few slides are based on this paper from the UofT machine learning group:

  Xu et al. Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention. ICML, 2015.

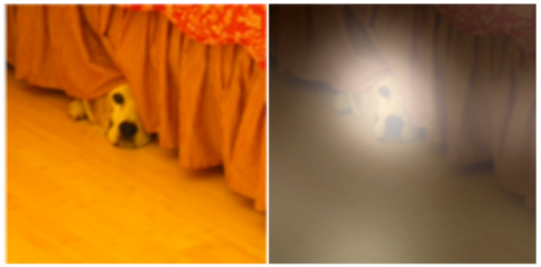
Attention-Based Caption Generation

- The caption generation task: take an image as input, and produce a sentence describing the image.
- **Encoder:** a classification conv net (VGGNet, similar to AlexNet). This computes a bunch of feature maps over the image.
- **Decoder:** an attention-based RNN, analogous to the decoder in the translation model
  - In each time step, the decoder computes an attention map over the entire image, effectively deciding which regions to focus on.
  - It receives a context vector, which is the weighted average of the conv net features.

Attention-Based Caption Generation

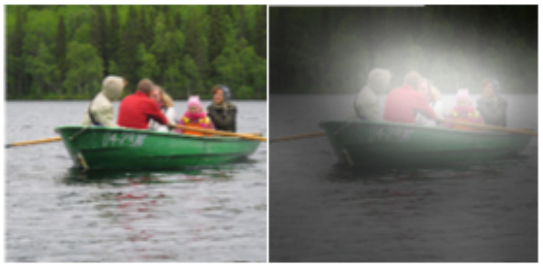
- This lets us understand where the network is looking as it generates a sentence.


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.

Attention-Based Caption Generation

- This can also help us understand the network's 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.