Sequence to Sequence Models

Jon Dehdari

January 25, 2016

Good Morning!

Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?

Sentence Vectors

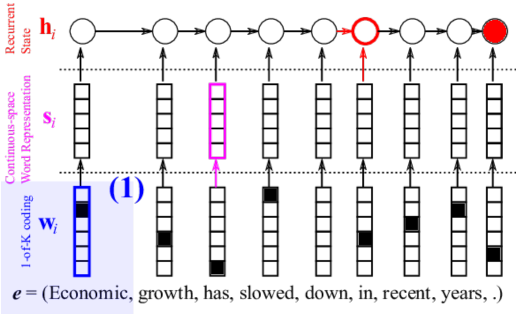
- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not?

Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not? How? From the hidden state at the end of a sentence: $\mathbf{h}_i = \phi_{\text{enc}}(\mathbf{h}_{i-1}, \mathbf{s}_i)$ (ϕ_{enc} = LSTM or GRU)

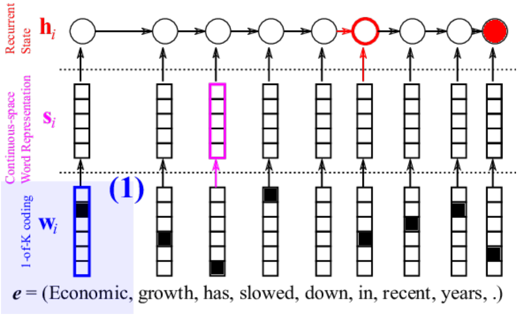
Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not? How? From the hidden state at the end of a sentence: $\mathbf{h}_i = \phi_{\text{enc}}(\mathbf{h}_{i-1}, \mathbf{s}_i)$ (ϕ_{enc} = LSTM or GRU)



Sentence Vectors

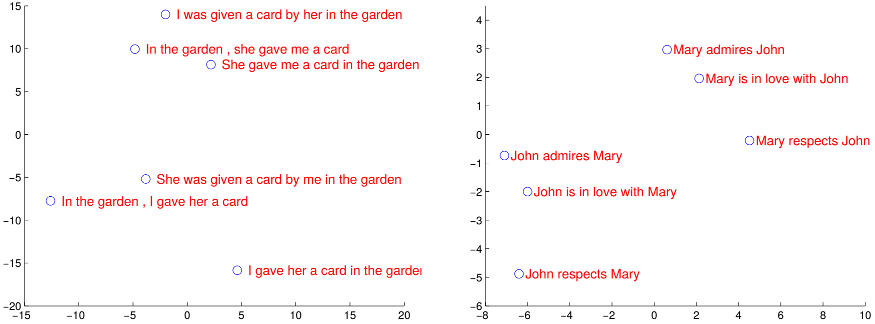
- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not? How? From the hidden state at the end of a sentence: $\mathbf{h}_i = \phi_{\text{enc}}(\mathbf{h}_{i-1}, \mathbf{s}_i)$ $(\phi_{\text{enc}} = \text{LSTM or GRU})$



- Are they any good? For Elman networks (SRNs), not so much. For LSTMs or GRUs, yes, they're pretty good

Image courtesy of http://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2

Sentence Vector Examples

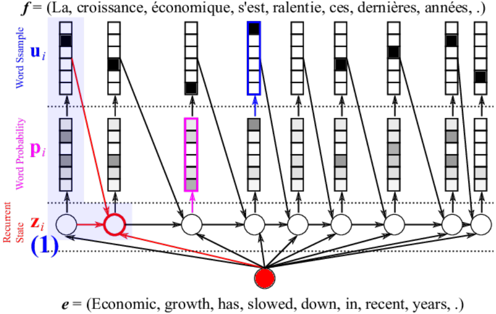


Sentence vectors were projected to two dimensions using PCA

Images courtesy of Sutskever, et al (2014)

Generating Sentences from Vectors

- We can also try to go the other direction, generating sentences from vectors
- How? Use an RNN to **decode**, rather than **encode** a sentence: $\mathbf{z}_i = \phi_{\text{dec}}(\mathbf{z}_{i-1}, \mathbf{u}_{i-1}, \mathbf{h}_T)$

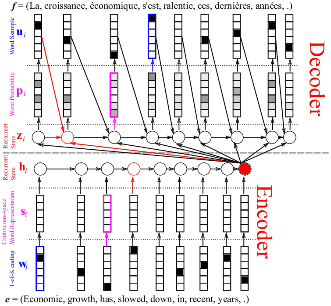


- $\mathbf{h}_T$ ensures global sentence coherency (& adequacy in MT); $\mathbf{u}_{i-1}$ ensures local fluency

Image courtesy of http://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2

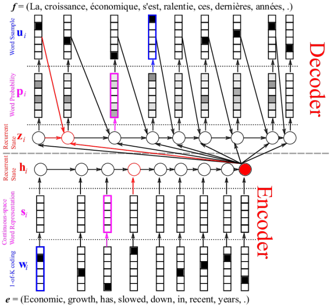
Using Neural Encoders & Decoders to Translate

- We can combine the neural encoder and decoder of previous slides to form an **encoder-decoder model**
- This can be used for machine translation, and other tasks that map sequences to sequences



Using Neural Encoders & Decoders to Translate

- We can combine the neural encoder and decoder of previous slides to form an **encoder-decoder model**
- This can be used for machine translation, and other tasks that map sequences to sequences



- Monolingual word projections (vectors/embeddings) are trained to maximize likelihood of next word
- Source-side word projections ($\mathbf{s}_i$) in an encoder-decoder setting are trained to maximize target-side likelihood

Bidirectional RNNs

- The basic encoder-decoder architecture doesn't handle long sentences very well
- Everything must fit into a fixed-size vector,
- and RNNs remember recent items better

Bidirectional RNNs

- The basic encoder-decoder architecture doesn't handle long sentences very well
- Everything must fit into a fixed-size vector,
- and RNNs remember recent items better
- We can combine left-to-right and right-to-left RNNs to overcome these issues

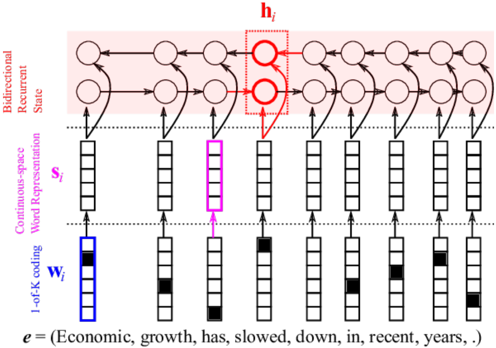


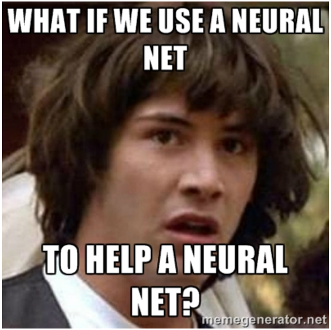
Image courtesy of http://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3

What if . . .

- Even bidirectional encoder-decoders have a hard time with long sentences
- We need a way to keep track of what's already been translated and what to translate next

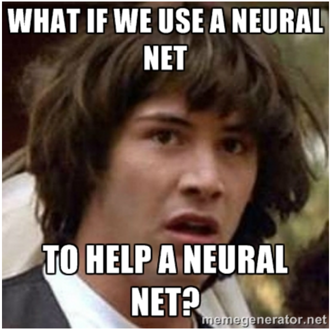
What if . . .

- Even bidirectional encoder-decoders have a hard time with long sentences
- We need a way to keep track of what's already been translated and what to translate next



What if . . .

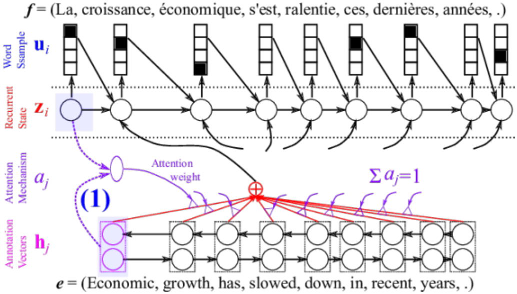
- Even bidirectional encoder-decoders have a hard time with long sentences
- We need a way to keep track of what's already been translated and what to translate next



- For neural nets, the solution is often more neural nets . . .

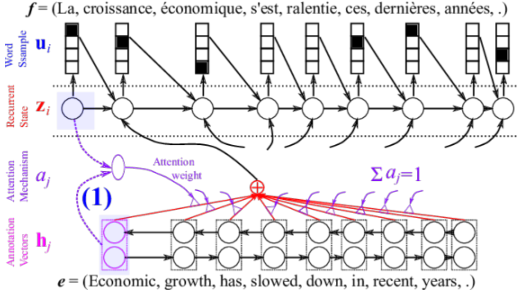
Achtung, Baby!

- Attention-based decoding adds another network (**a**) that takes as input the encoder's hidden state (**h**) and the decoder's hidden state (**z**), and outputs a probability for each source word at each time step (when and where to pay attention) : $e_{i,j} = a(z_{i-1}, h_j) = v_a^\top \tanh(W_a s_{i-1} + U_a h_j)$



Achtung, Baby!

- Attention-based decoding adds another network (**a**) that takes as input the encoder's hidden state (**h**) and the decoder's hidden state (**z**), and outputs a probability for each source word at each time step (when and where to pay attention) : $e_{i,j} = a(z_{i-1}, h_j) = v_a^\top \tanh(W_a s_{i-1} + U_a h_j)$



- The attention weights can also function as soft word alignments. They're trained on target-side MLE

Image courtesy of http://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3

Image Caption Generation

- You can use attention-based decoding to give textual descriptions of images

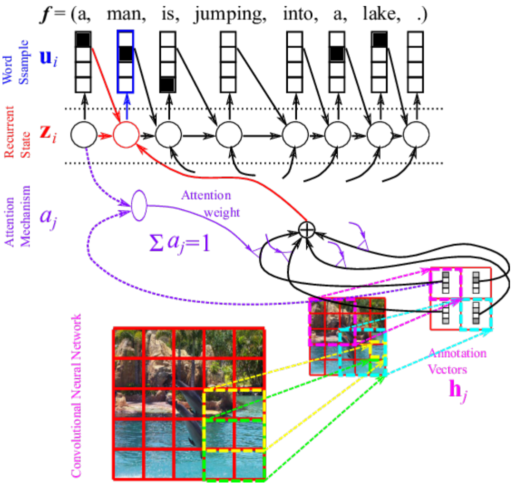
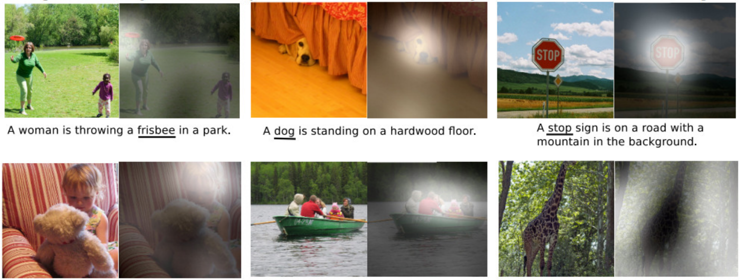


Image courtesy of http://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-3

Image Caption Generation Examples

Figure 4. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



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.

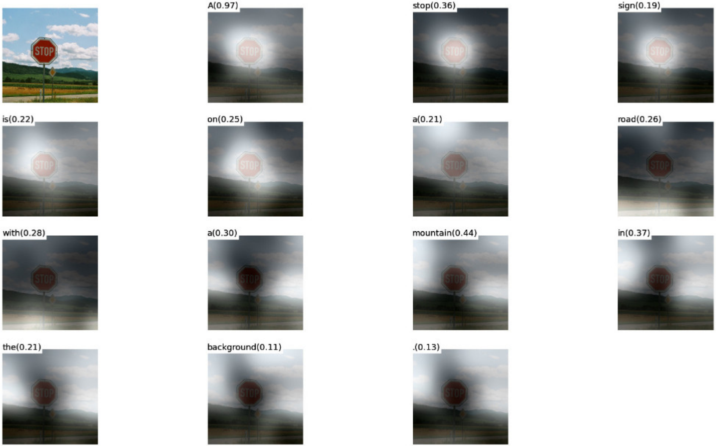
A little girl sitting on a bed with a teddy bear.

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

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

Images courtesy of http://arxiv.org/abs/1502.03044

Image Caption Generation, Step by Step



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

Images courtesy of http://arxiv.org/abs/1502.03044