The Hitchiker's Guide to Sequence-to-Sequence Models

Akash Saravanan

Disclaimer

- DON’T PANIC.
- I have no idea how long this will take.
- Sequences can mean anything from text to video to audio etc. We’re going to mainly deal with text.
- I have some notes written for a bunch of the slides here. I may or may not forget to mention them while presenting, so feel free to come back and read through.
- We’ll be discussing the original seq2seq model utilizing RNNs with only a brief mention of newer approaches. If time permits, we’ll also cover attention.

Agenda

- Overview
- Brief Recap - RNNs & Family
- Seq2Seq
- Attention Please

Overview - Who? What? Where?

Ilya Sutskever, Oriol Vinyals, Quoc V. Le.

Sequence to Sequence Learning with Neural Networks.

NIPS 2014.

Overview - Why?

- For Neural Machine Translation (NMT).
- Converting a message from an input *language* into a target *language*.
- Language - A *sequence* of words that have meaning when taken in order.
- Converting one sequence of words into another sequence of words.
- Converting one sequence into another sequence.

Overview - Why?

- Not just for NMT;
    - Text summarization: Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond
    - Math: Deep Learning for Symbolic Mathematics
    - Video Captioning: Sequence to Sequence -- Video to Text
    - Gmail’s Smart Reply Feature: Computer, respond to this email.
    - Transformers: Attention Is All You Need

Overview - Why RNNs for Sequences?

- Case: Text summarizer.
- Input:

“Space," it says, "is big. Really big. You just won't believe how vastly, hugely, mindbogglingly big it is. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to space, listen…”

- Output:

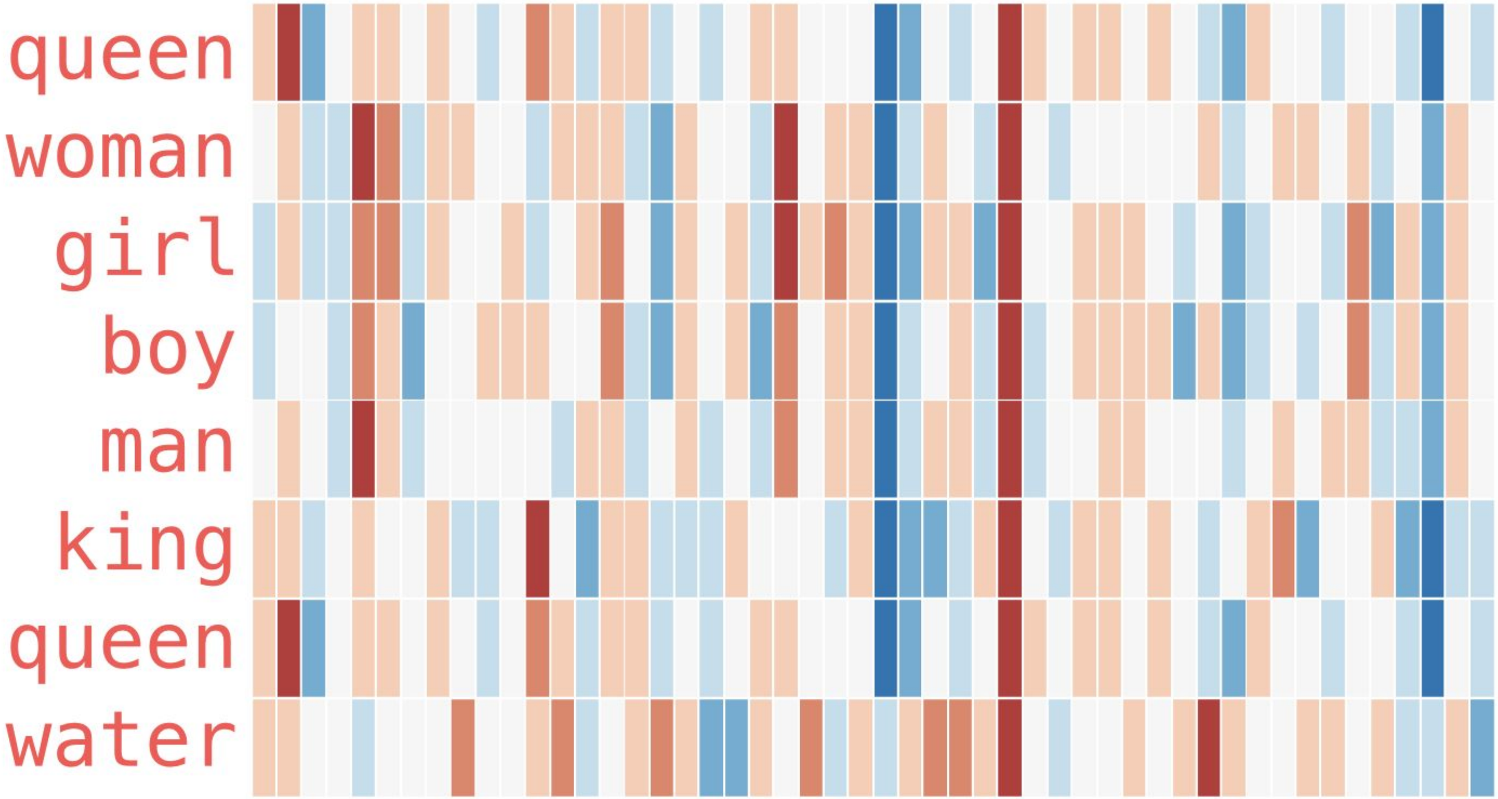
Space is very big.

Brief Recap: Word Embeddings

On a scale of 0-10, rate the following:

1. Pineapple on your pizza.
2. Dipping fries in a milkshake.
3. Sathyam popcorn without the seasoning.
4. Curd rice with ketchup.
5. This questionnaire.

Brief Recap: Word Embeddings

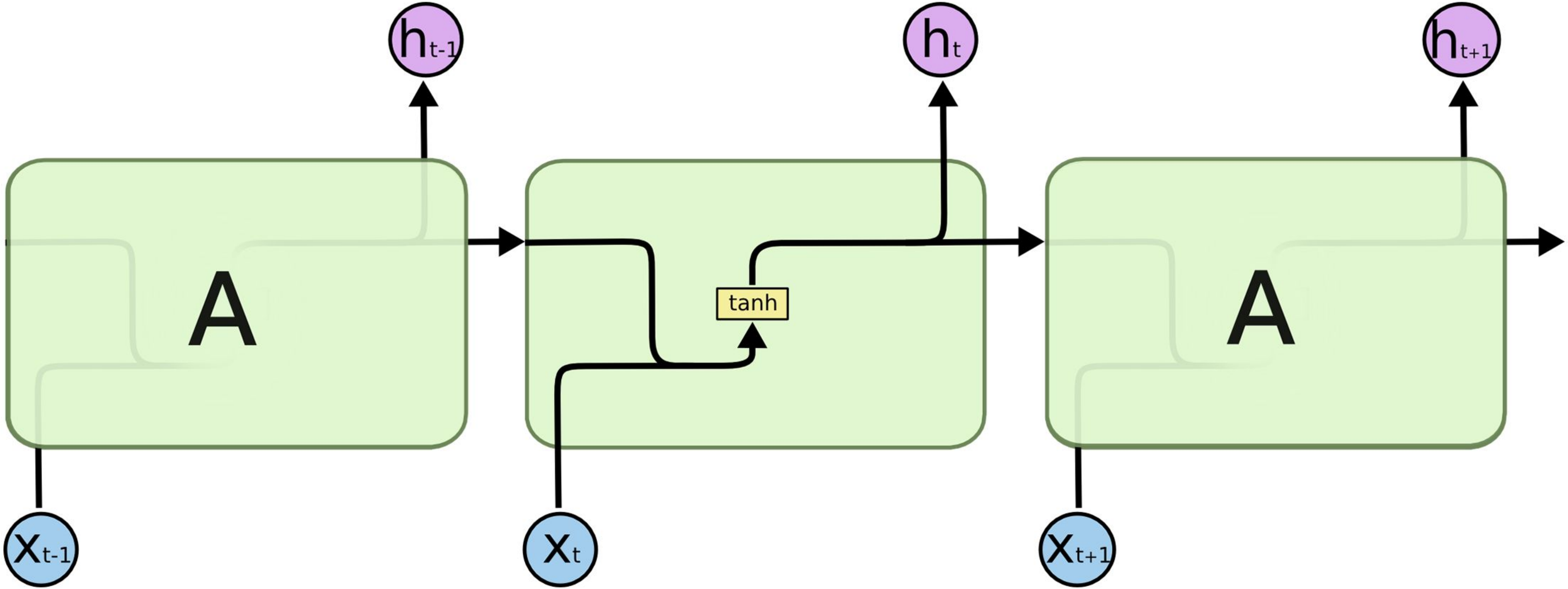


Brief Recap: Word Embeddings

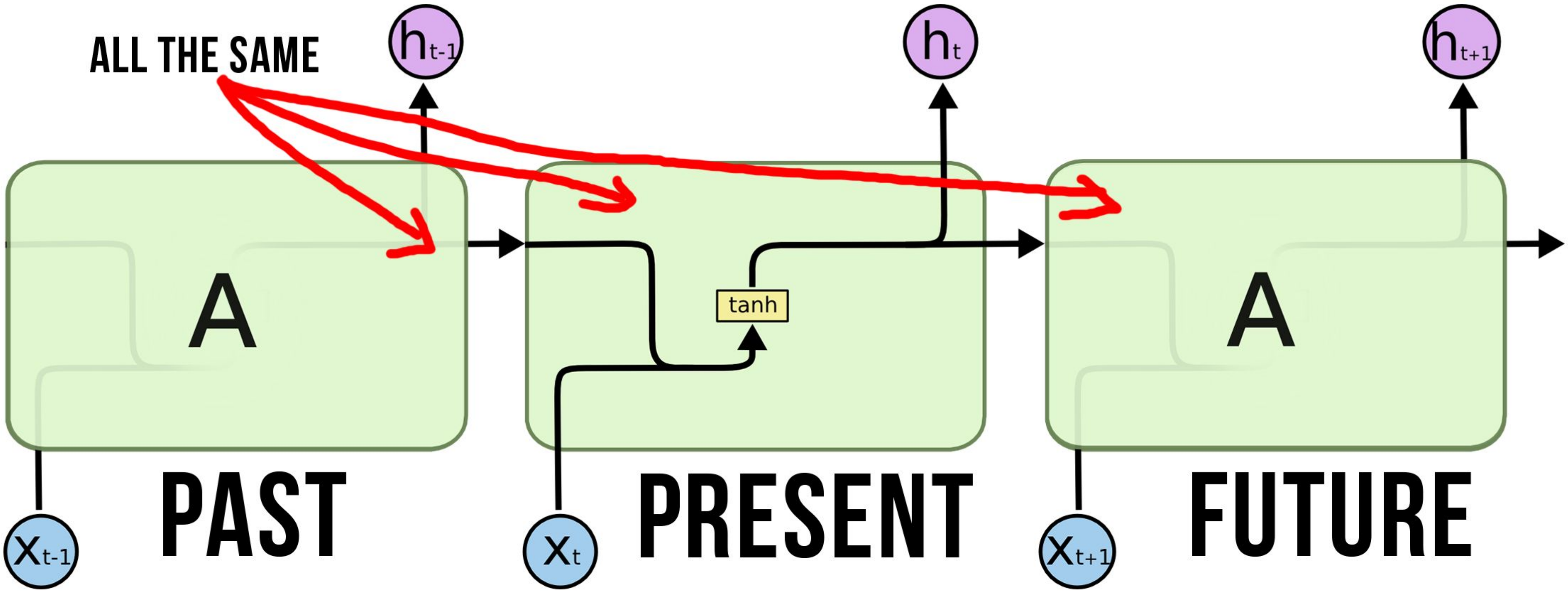
Further Reading:

The Illustrated Word2vec

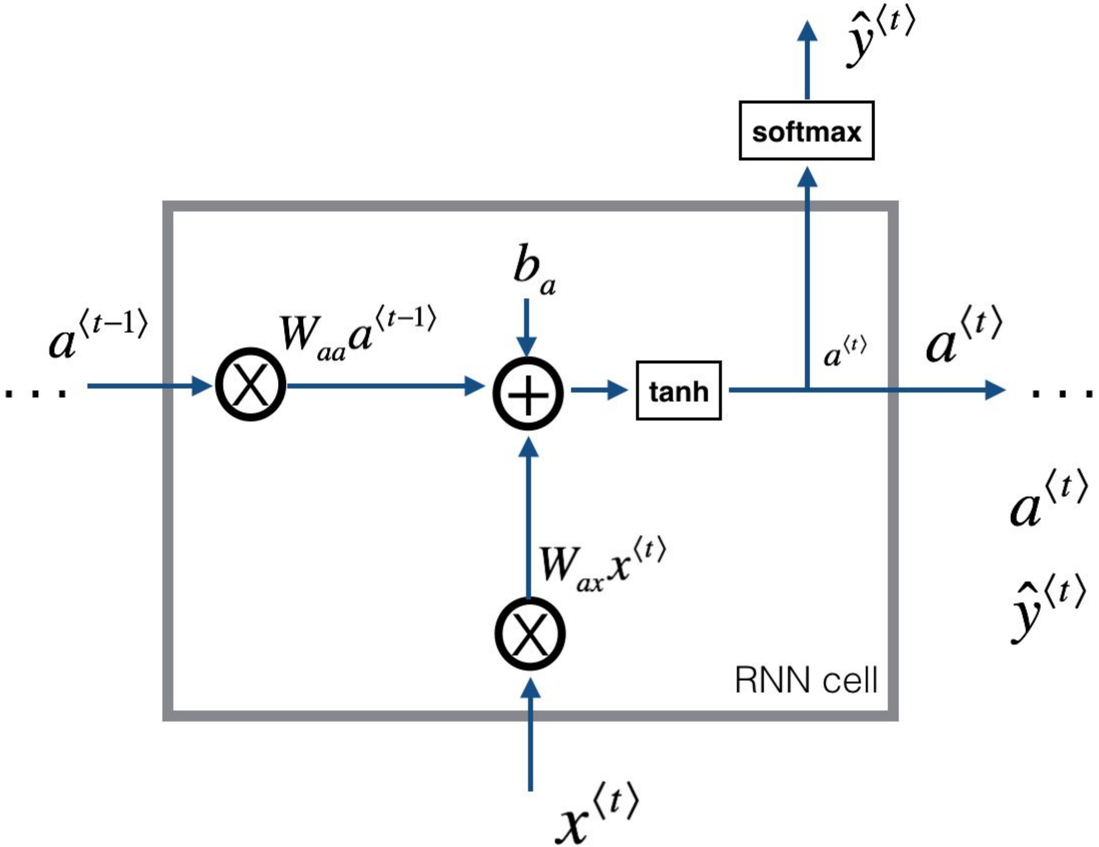
Brief Recap: RNNs & Family



Brief Recap: RNNs & Family



Brief Recap: RNNs & Family

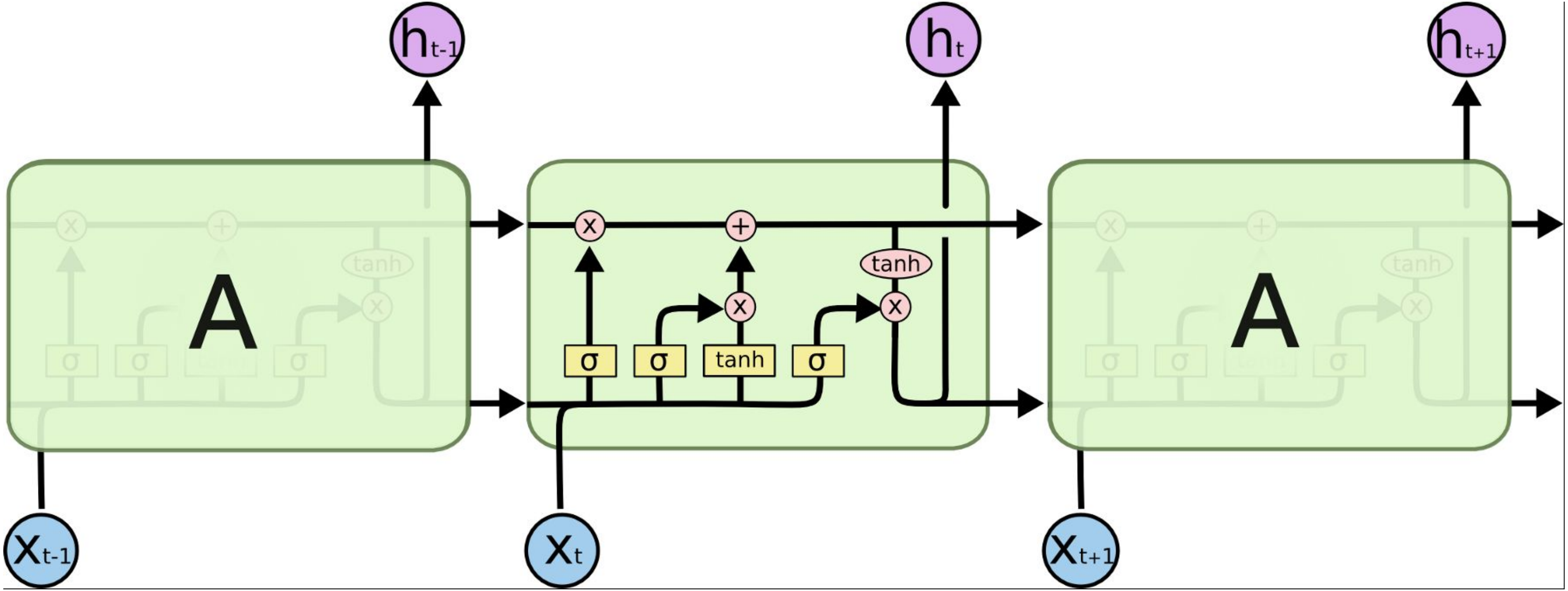


$\hat{y}^{\langle t \rangle}$
softmax
b_a
$a^{\langle t-1 \rangle}$
$W_{aa}a^{\langle t-1 \rangle}$
tanh
$a^{\langle t \rangle}$
$a^{\langle t \rangle}$
$W_{ax}x^{\langle t \rangle}$
RNN cell
$x^{\langle t \rangle}$

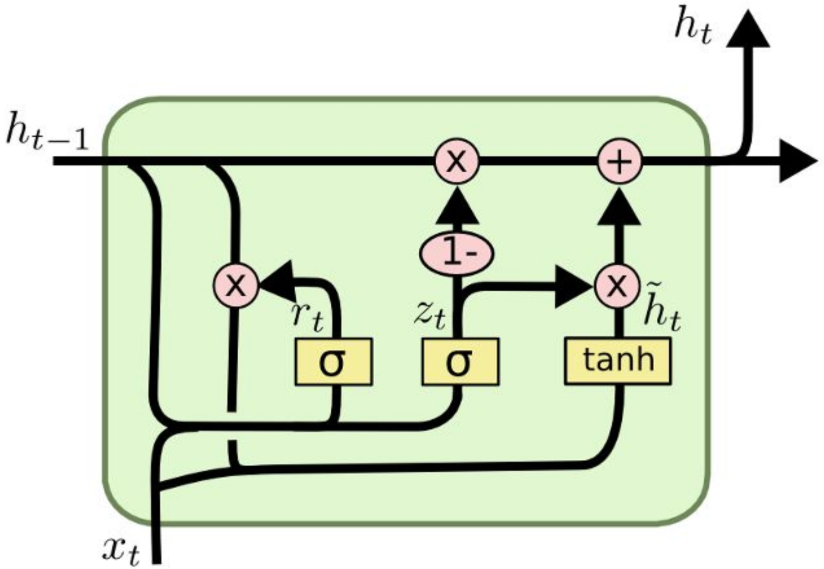
$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$

$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

Brief Recap: RNNs & Family



Brief Recap: RNNs & Family



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Brief Recap: RNNs & Family

For further reading:

Understanding LSTM Networks (Blog)

Recurrent Neural Network (Blog)

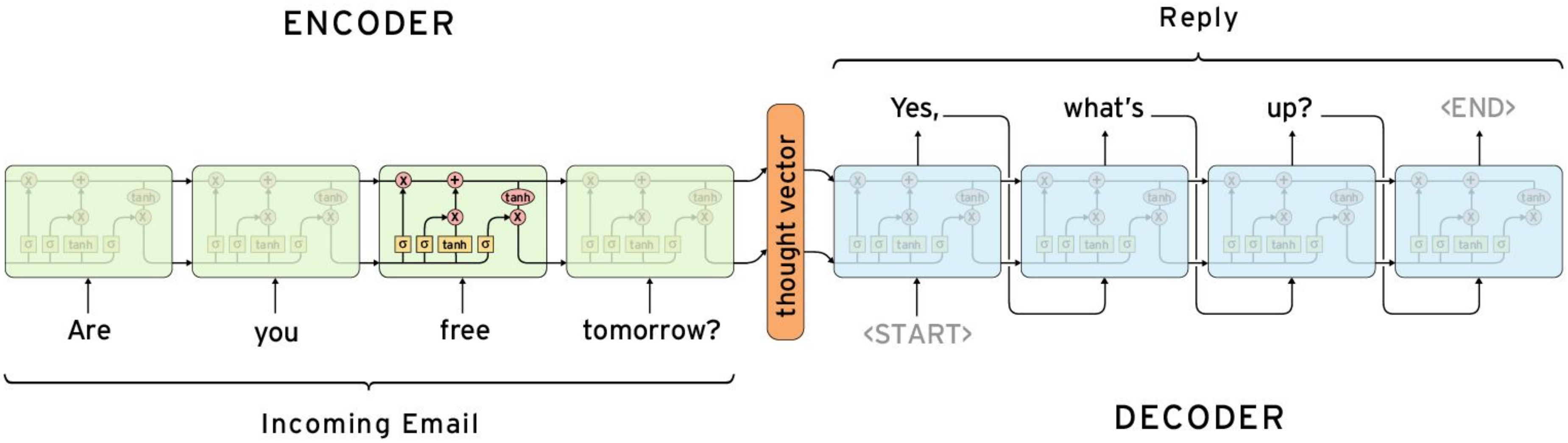
A friendly introduction to Recurrent Neural Networks (Video)

Seq2Seq

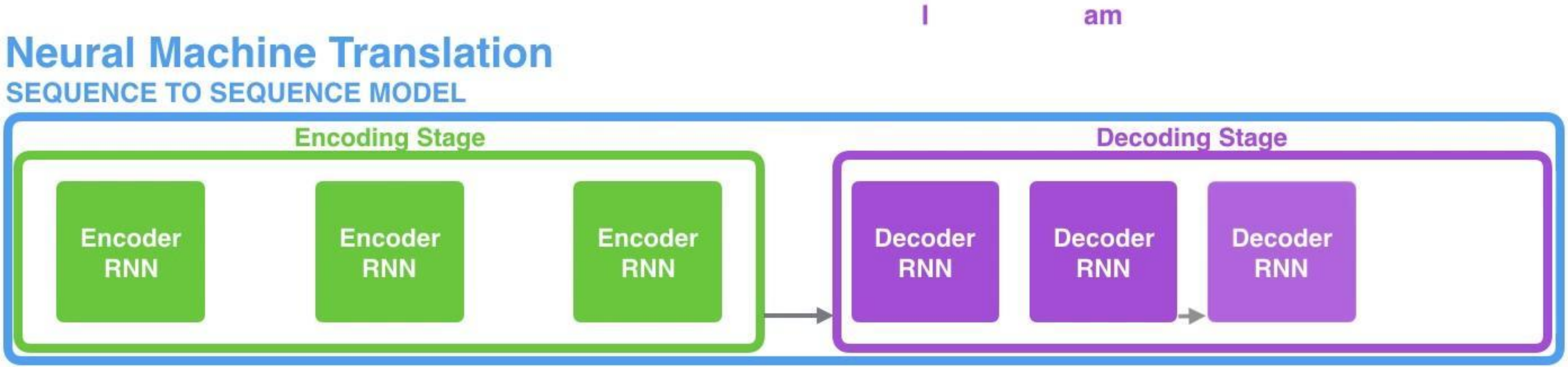
“A useful property of the LSTM is that it learns to map an input sentence of variable length into a fixed-dimensional vector representation.”

“Given that translations tend to be paraphrases of the source sentences, the translation objective encourages the LSTM to find sentence representations that capture their meaning, as sentences with similar meanings are close to each other while different sentences meanings will be far.”

Seq2Seq


ENCODER
Reply
Yes,
what's
up?
<END>
thought vector
Are
you
free
tomorrow?
<START>
Incoming Email
DECODER
σ
σ
tanh
σ
X
+
X
tanh
X

Seq2Seq


I
am
Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL
Encoding Stage
Decoding Stage
Encoder RNN
Encoder RNN
Encoder RNN
Decoder RNN
Decoder RNN
Decoder RNN

Seq2Seq - Training

Two Key Points:

1. They use deep LSTMs. Specifically, 4 layers.
2. They reversed the order of the input sentence.

“So for example, instead of mapping the sentence a, b, c to the sentence α, β, γ, the LSTM is asked to map c, b, a to α, β, γ, where α, β, γ is the translation of a, b, c. This way, a is in close proximity to α, b is fairly close to β, and so on, a fact that makes it easy for SGD to “establish communication” between the input and the output. We found this simple data transformation to greatly boost the performance of the LSTM”

Seq2Seq - Training

Teacher Forcing:

- The input to the decoder at time step t is not the output from time step t - 1.
- Instead, the actual/expected output is fed as the input.
- Essentially, regardless of how right or wrong the output from the previous time step is, we give the actual correct output as the input to the next time step.
- Why? It’s been shown to cause the network to converge faster.

Seq2Seq - Training & Inference

- Deep LSTMs - 4 layers.
- 1000 cells per layer.
- 1000-Dimensional word embeddings.
- Final softmax over a vocabulary of 80,000 words.
- Uses Beam Search for predictions.

Time Check

“I love deadlines. I like the whooshing sound they make as they fly by.”

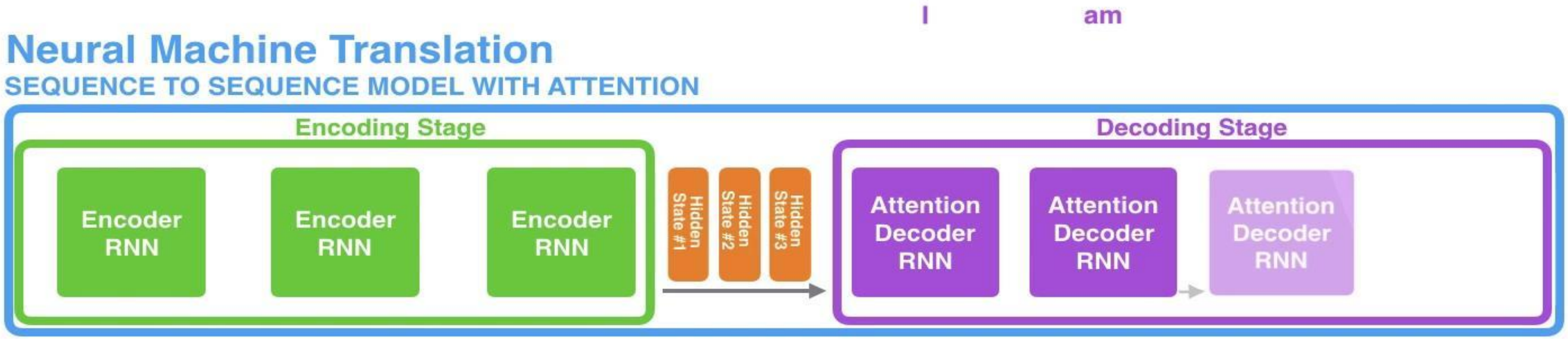
- Douglas Adams

Attention Please

- What if we could get the model to pay attention to particular words while translating?
- If we're translating "Nein, ich bin dein Vater.", it should focus on "Vater" before translating it to "father".
- Essentially, attention amplifies the signal from the relevant part of the input sequence.
- There are a couple of extra things we do in order to incorporate the attention module.

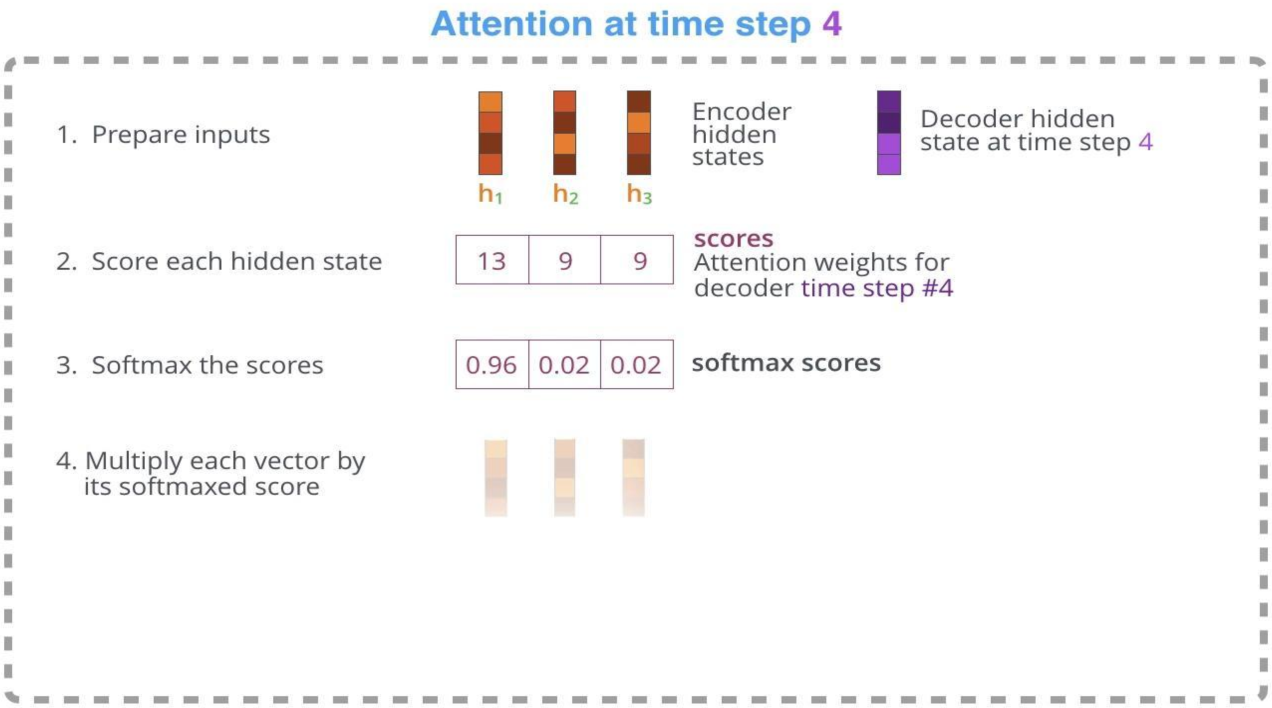
Attention Please

1. We don’t pass only the final hidden state from the encoder. We pass all the hidden states from all time steps.



Attention Please

2. Create a “context” vector at each of the decoder’s time steps.



Attention at time step 4

1. Prepare inputs

2. Score each hidden state

13	9	9

scores
Attention weights for decoder time step #4

3. Softmax the scores

0.96	0.02	0.02

softmax scores

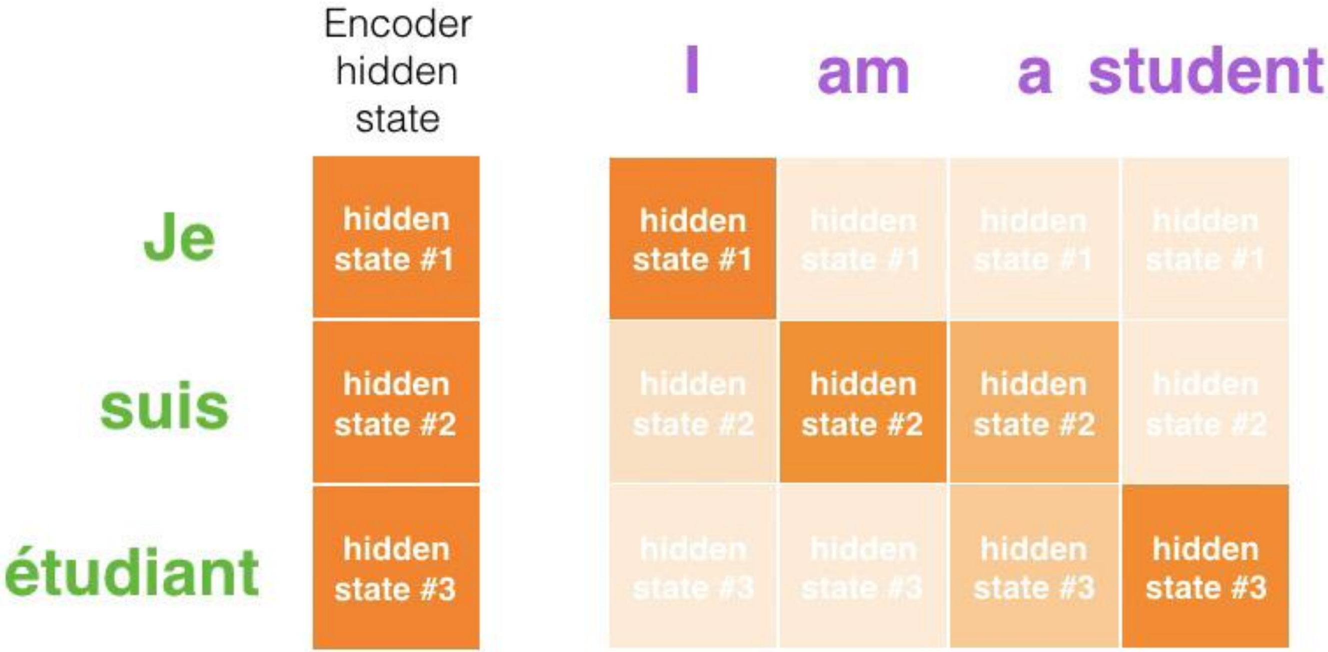
4. Multiply each vector by its softmaxed score

Attention Please

Now how do we use this context vector?

- Ignore the RNN’s output.
- Retrieve it’s output hidden state (what is passed to the next time step).
- Concat it with the context vector.
- Send it to a feedforward neural network (trained jointly with the model)
- Send the output of this feedforward network as the output for this time step.
- Send this output to the next time step of the RNN as input.

Attention Please



	Encoder hidden state	I	am	a	student
Je	hidden state #1	hidden state #1	hidden state #1	hidden state #1	hidden state #1
suis	hidden state #2	hidden state #2	hidden state #2	hidden state #2	hidden state #2
étudiant	hidden state #3	hidden state #3	hidden state #3	hidden state #3	hidden state #3

Attention Please

Further Reading:

NLP From Scratch: Translation with a Sequence to Sequence Network and Attention

Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)

Demo Time

SO LONG

AND

THANKS

FOR ALL

THE FISH