Chatbot: Seq2Seq Model

Shantanu Patil
15JE001628

Mentor: Dr. Anurag Jaiswal
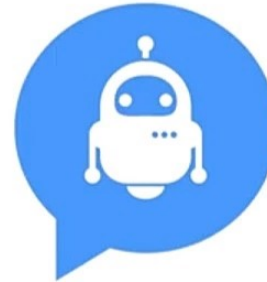
Applications

Speech Transcription

Neural Machine Translation (NMT)
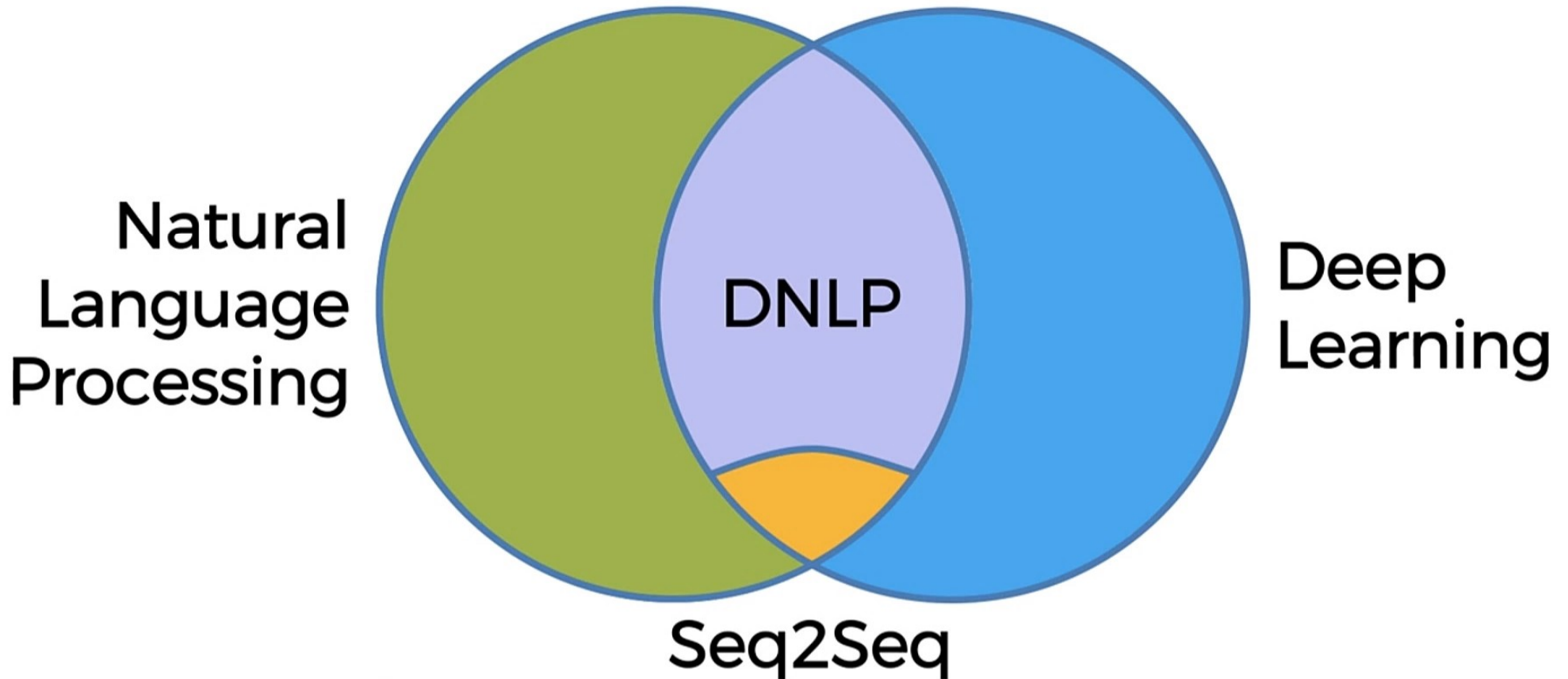
Chatbots

Q&A

Text Summarization

Image Captioning

Video Captioning

Types of NLP

Classical vs Deep Learning Models

Some examples:

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)
3. Bag-of-words model (Classification)
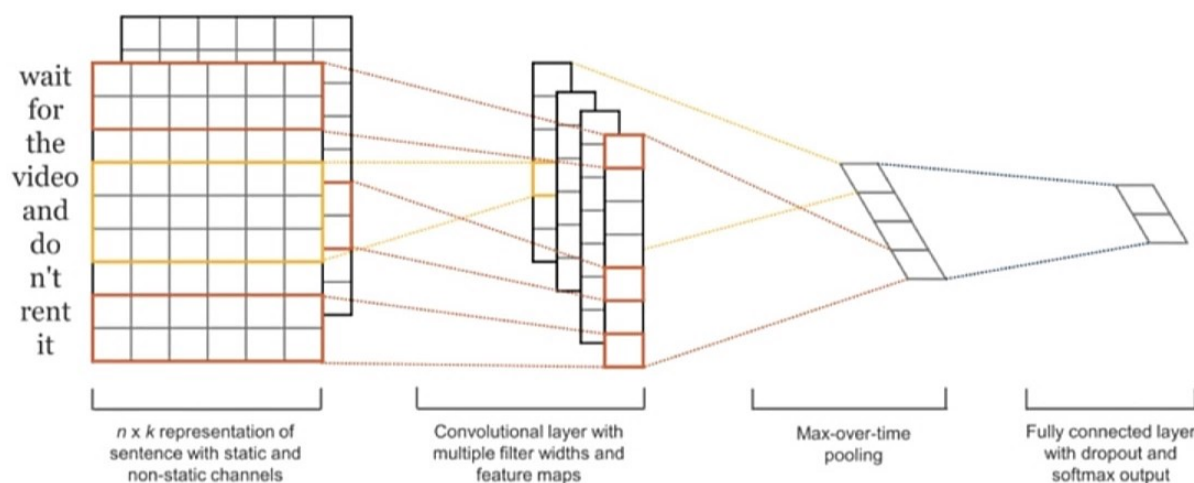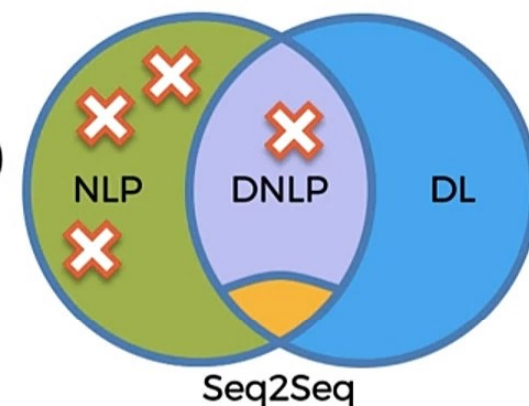4. CNN for text Recognition (Classification)

NLP   DNLP   DL

Seq2Seq

wait
for
the
video
and
do
n't
rent
it

n x k representation of
sentence with static and
non-static channels

Convolutional layer with
multiple filter widths and
feature maps

Max-over-time
pooling

Fully connected layer
with dropout and
softmax output

Image Source: www.wildml.com

Bag-Of-Words

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... , 0]

20,000 elements long

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around ... Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 3]

20,000 elements long

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around ... Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 3] $\longrightarrow$ **Yes / No ?**

20,000 elements long

Training Data:

[1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, ... , 2] $\longrightarrow$ **No**

[1, 1, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 0] $\longrightarrow$ **Yes**

[1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, ... , 1] $\longrightarrow$ **Yes**

[1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, ... , 1] $\longrightarrow$ **No**

[1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, ... , 1] $\longrightarrow$ **Yes**

Image Source: www.helloacm.com
...

$\longrightarrow$ **ooo**

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around ... Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, ... , 3] → **Yes / No ?**
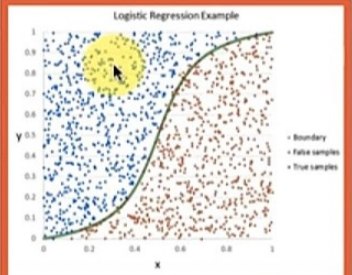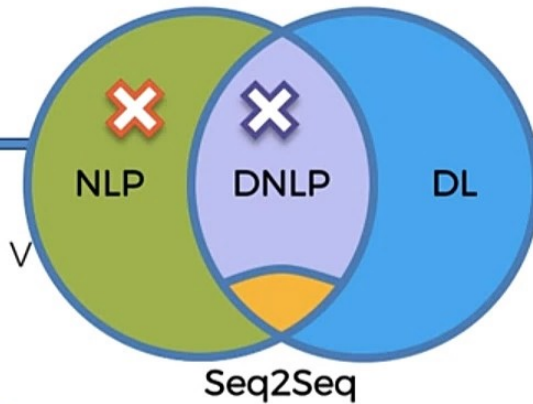
← 20,000 elements long →

Training Data:
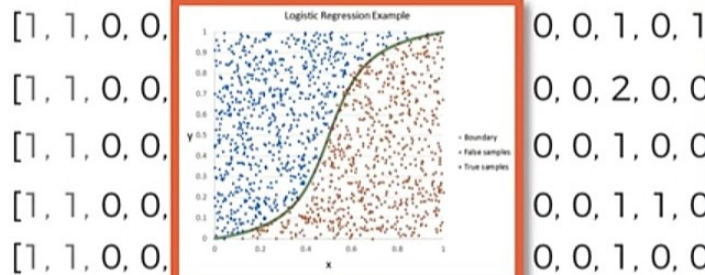
[1, 1, 0, 0,   0, 0, 1, 0, 1
[1, 1, 0, 0,   0, 0, 2, 0, 0
[1, 1, 0, 0,   0, 0, 1, 0, 0
[1, 1, 0, 0,   0, 0, 1, 1, 0
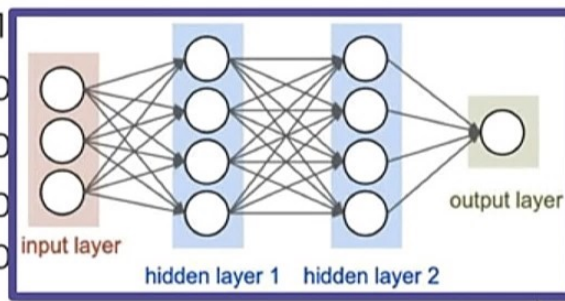[1, 1, 0, 0,   0, 0, 1, 0, 0
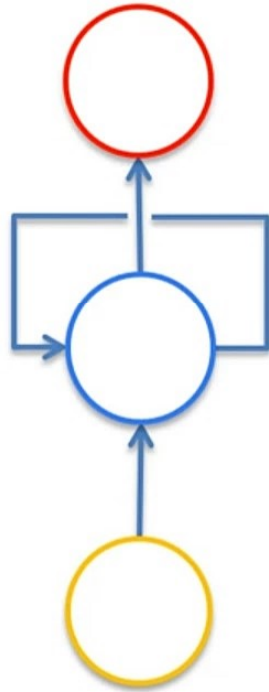
Image Source: www.helloacm.com

No
Yes
Yes
No
Yes
ooo

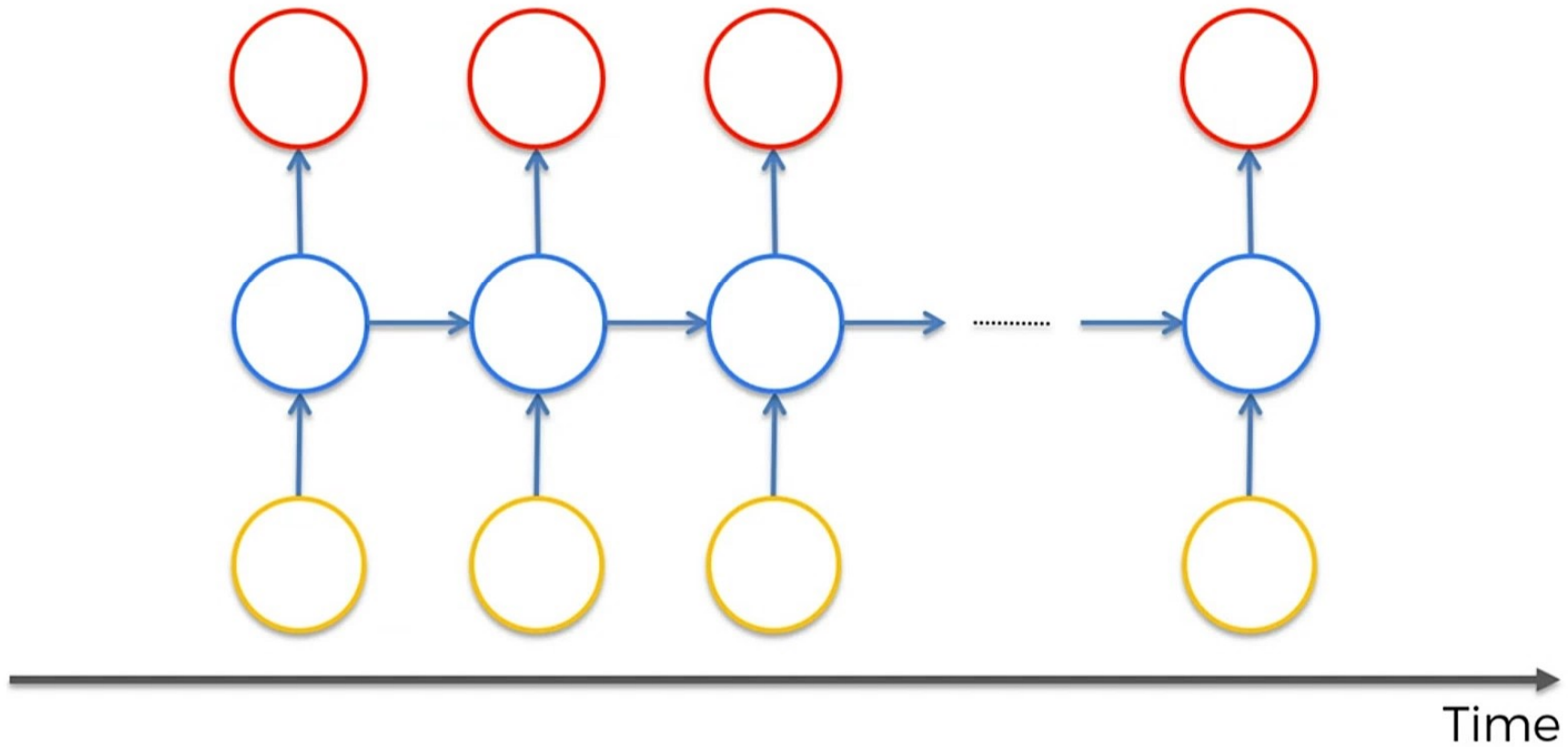# Seq2Seq Architecture

RNNs

Recurrent Neural Networks

Recurrent Neural Networks

Time

Recurrent Neural Networks

Time
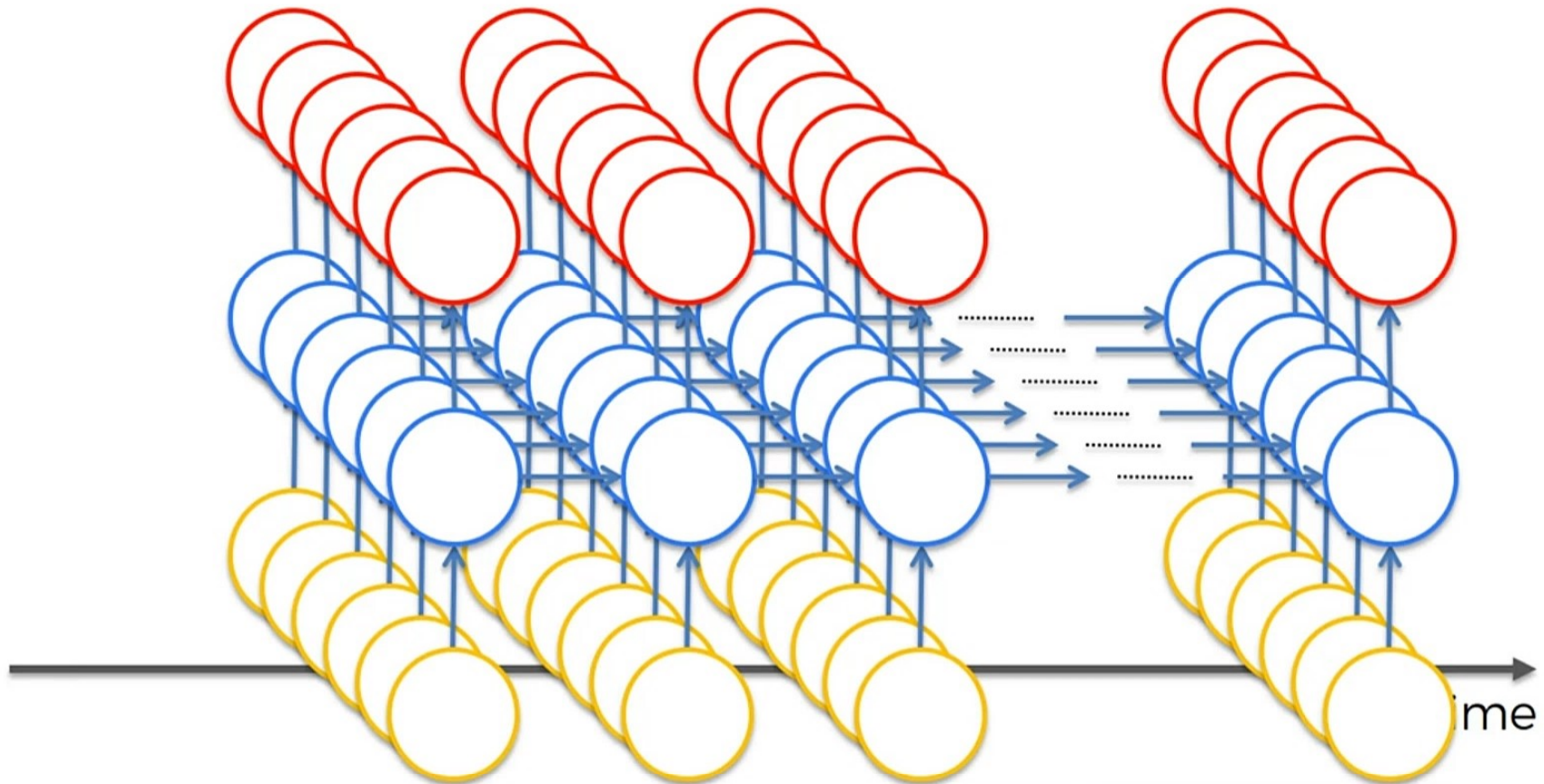
The Vanishing Gradient Problem
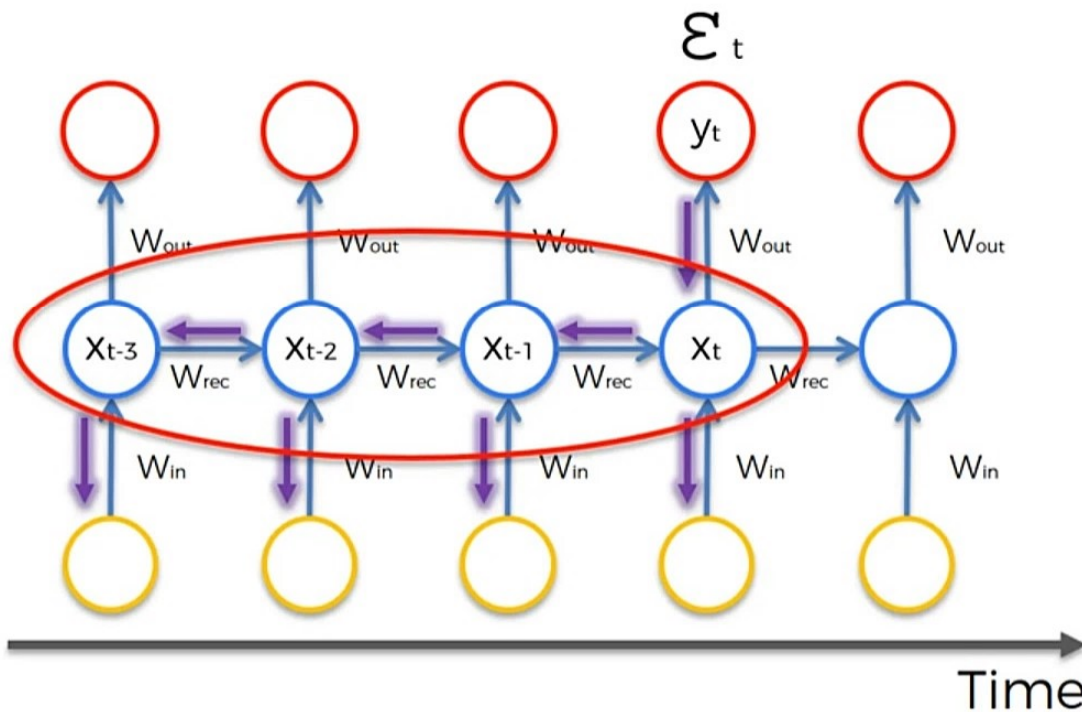
Formula Source: Razvan Pascanu et al. (2013)

The Vanishing Gradient Problem

Solutions:

1. Exploding Gradient

   - Truncated Backpropagation
   - Penalties
   - Gradient Clipping

2. Vanishing Gradient

   - Weight Initialization
   - Echo State Networks
   - Long Short-Term Memory Networks (LSTMs)

Long Short-Term Memory

Long Short-Term Memory

Image Source: colah.github.io
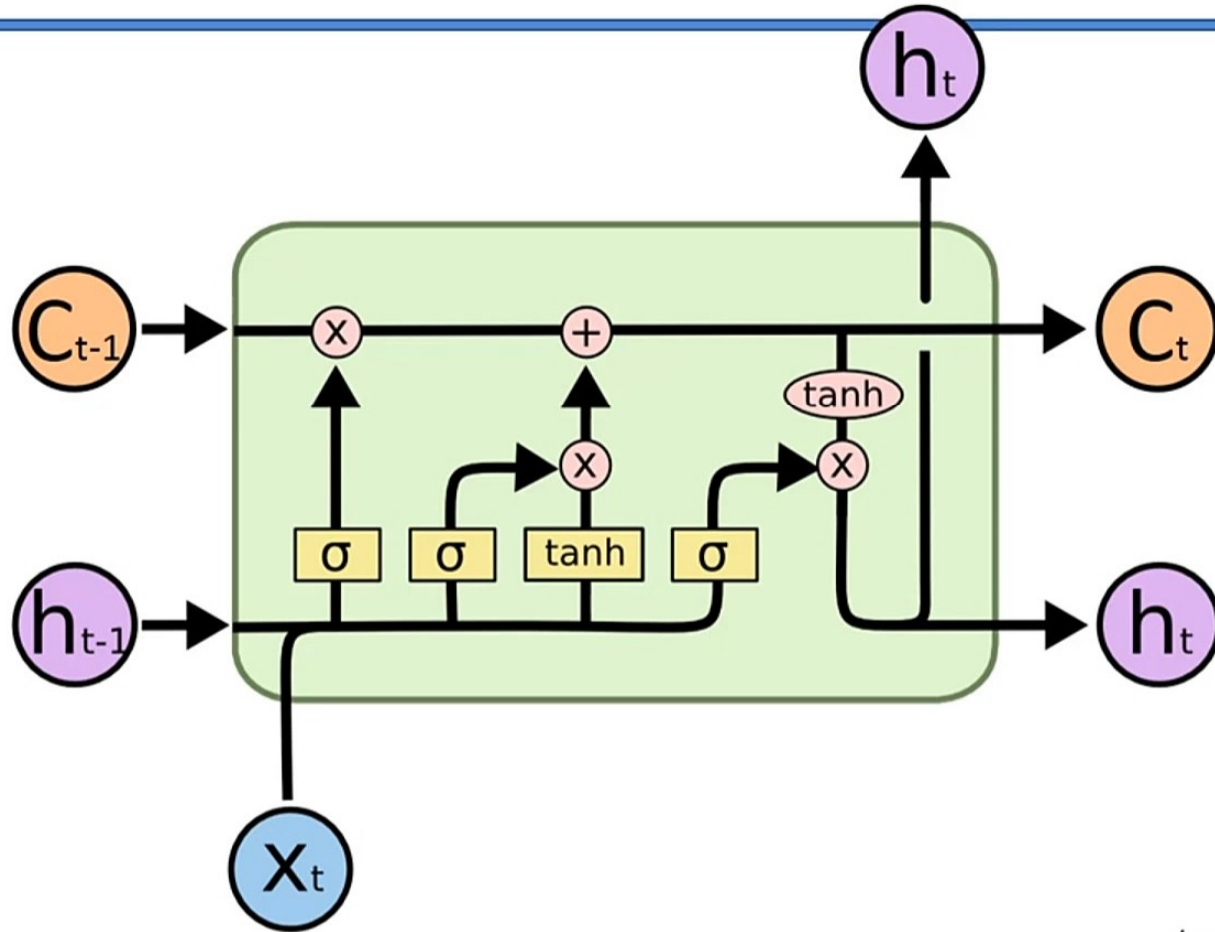
Seq2Seq Architecture

Issues with the Bag-of-words model:

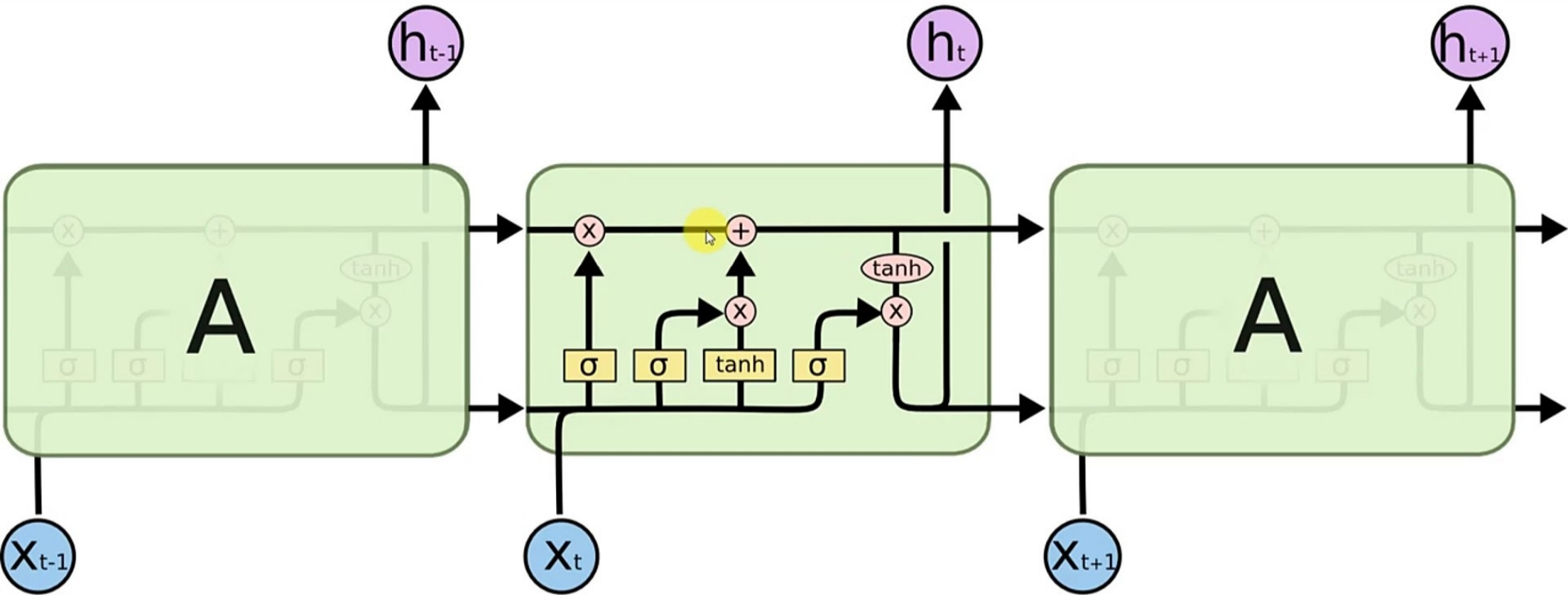1. Fixed-sized input
2. Doesn't take word order into account
3. Fixed-sized output

Seq2Seq Architecture
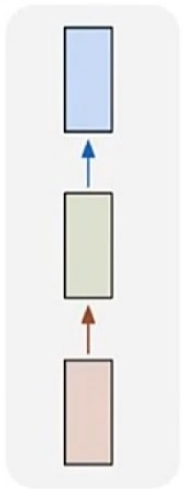
one to one | one to many | many to one | many to many | many to many
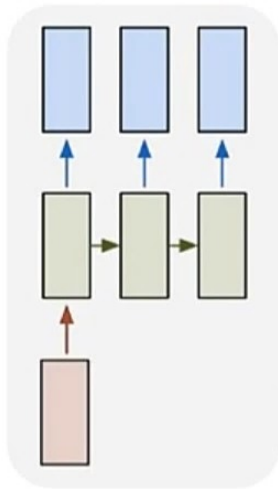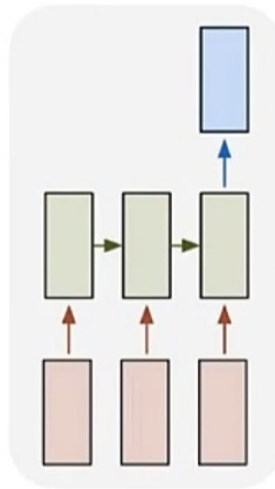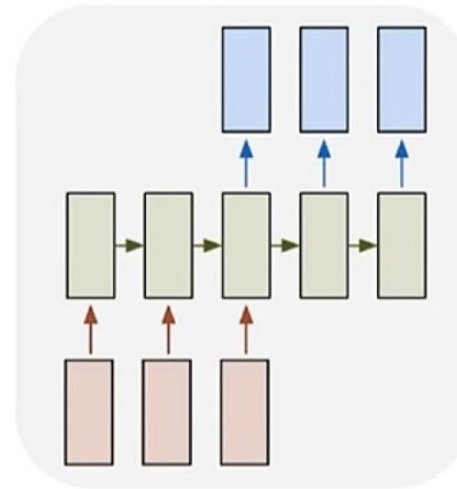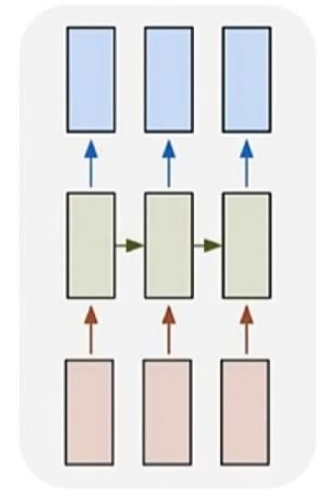
Image Source: karpathy.github.io

Seq2Seq Architecture

Hello Kirill, Checking if you are back to Oz.

[1,   5,   0, 9,   23,   7, 41, 101, 19,   4, 0, 20, 2]

(SOS)   (EOS)

Seq2Seq Architecture
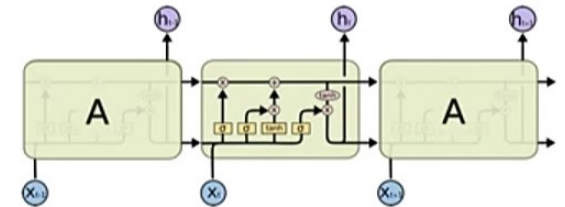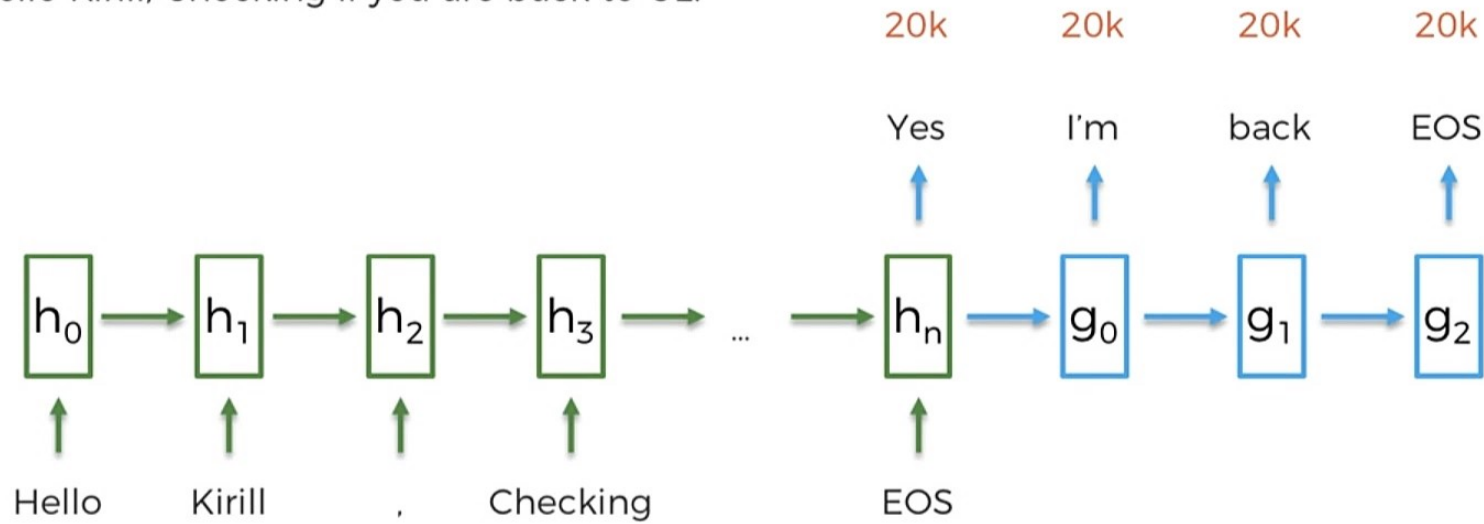
Hello Kirill, Checking if you are back to Oz.

Image Source: colah.github.io

Seq2Seq Architecture

Hello Kirill, Checking if you are back to Oz.
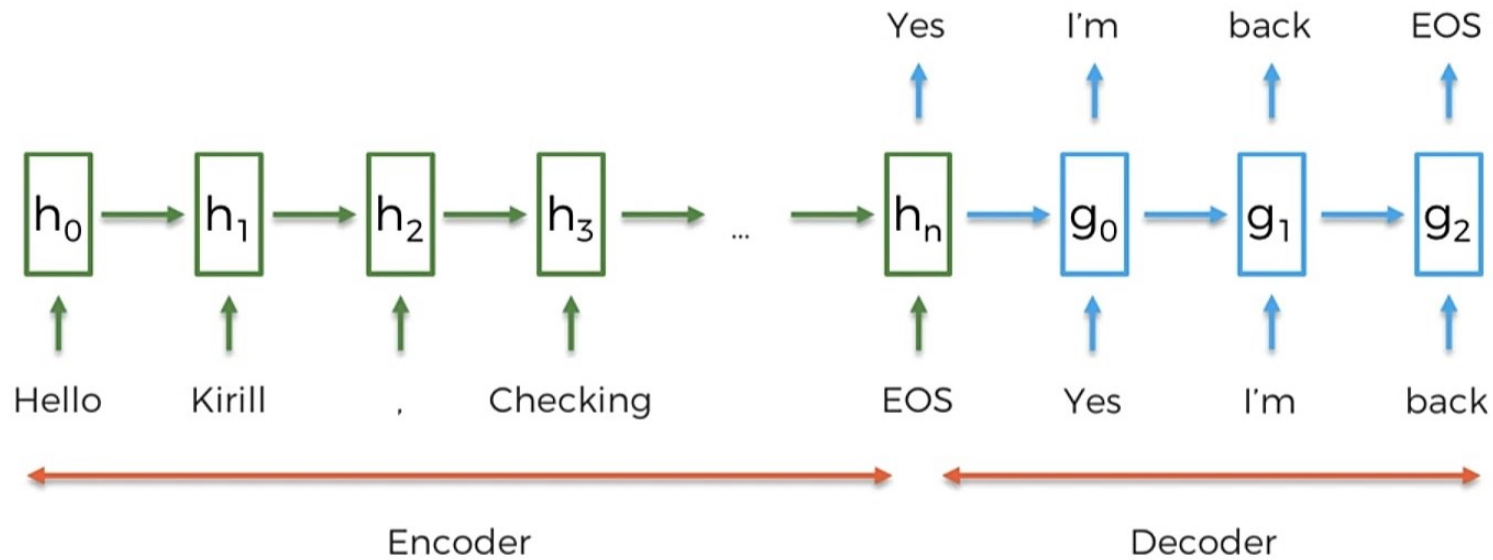
Seq2Seq Training

Did you like that recipe I sent you last week?
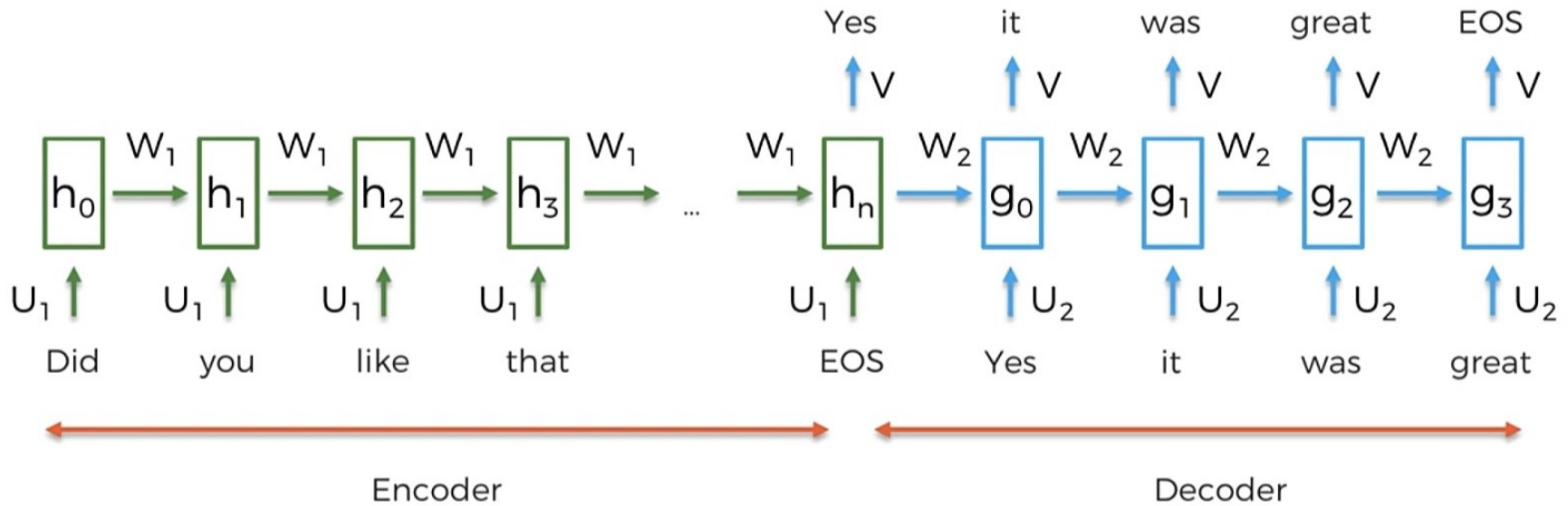
Beam Search Decoding

Attention Mechanisms

Context Vector
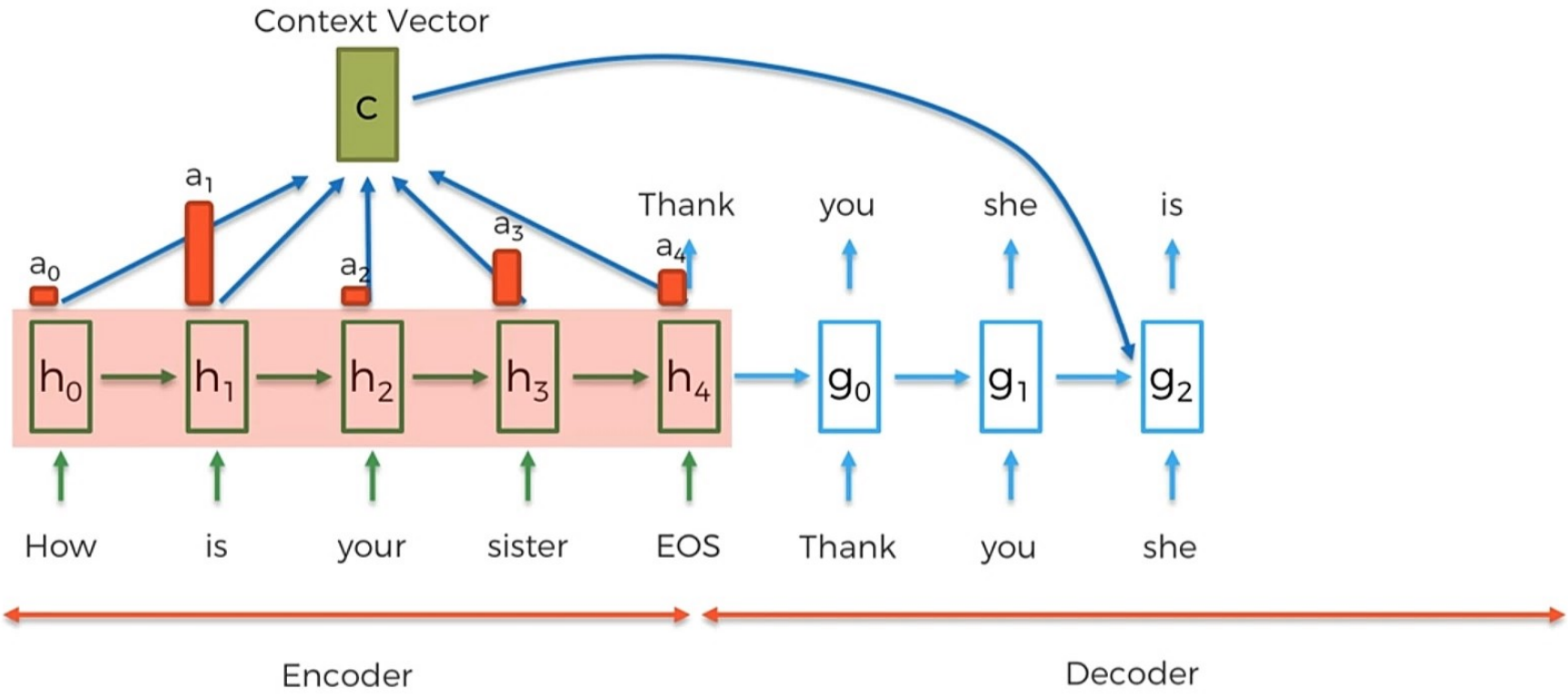
Attention Mechanisms

Image Source: https://arxiv.org/pdf/1508.04025.pdf
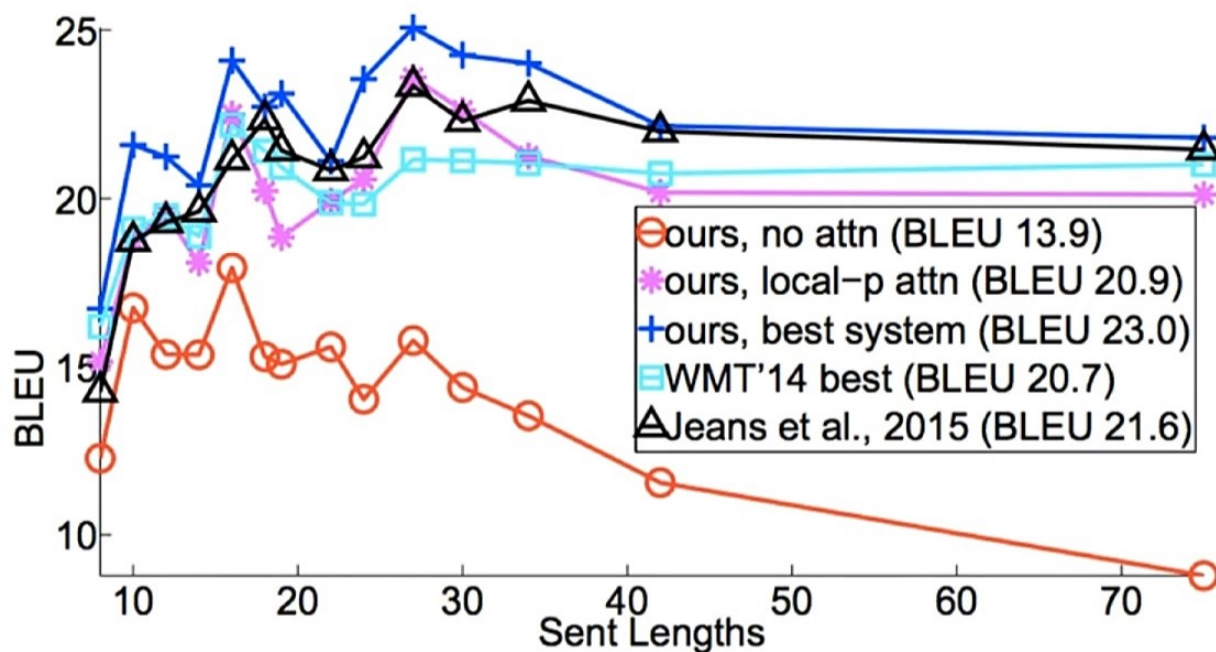
Editor - /Users/Hadelin/Desktop/The Best ChatBot/the_best_chatbot.py

the_best_chatbot.py

```python
28 batch_size = 16
29 vocab_twit = metadata['idx2w']
30 xvocab_size = len(metadata['idx2w'])
31 yvocab_size = xvocab_size
32 emb_dim = 1024
33 idx2w, w2idx, limit = data_utils_2.get_metadata()
34
35
36
37 ########## PART 2 - BUILDING THE SEQ2SEQ MODEL ##########
38
39
40
41 # Building the seq2seq model
42 model = seq2seq_wrapper.Seq2Seq(xseq_len=xseq_len,
43                                 yseq_len=yseq_len,
44                                 xvocab_size=xvocab_size,
45                                 yvocab_size=yvocab_size,
46                                 ckpt_path='./weights',
47                                 emb_dim=emb_dim,
48                                 num_layers=3)
49
50
51
52 ########## PART 3 - TRAINING THE SEQ2SEQ MODEL ##########
53
54
55
56 # See the Training in seq2seq_wrapper.py
57
58
59
60 ########## PART 4 - TESTING THE SEQ2SEQ MODEL ##########
61
62
63
64 # Loading the weights and Running the session
65 sess = model.restore_last_session()
66
67 # Getting the ChatBot predicted answer
68 def respond(question):
69     encoded_question = data_utils_2.encode(question, w2idx, limit['maxq'])
70     answer = model.predict(sess, encoded_question)[0]
71     return data_utils_2.decode(answer, idx2w)
72
73 # Setting up the chat
74 while True :
75     question = input("You: ")
76     answer = respond(question)
77     print ("ChatBot: "+answer)
78
```

File explorer

Name	Size	Kind	Date Modified
▶ 📁 __pycache__	--	Folder	23/12/2017 03:19
📄 data_preprocessing.py	5 KB	py File	05/04/2017 08:08
📄 data_utils_1.py	1 KB	py File	05/04/2017 08:08
📄 data_utils_2.py	1 KB	py File	23/12/2017 01:12
📄 idx_a.npy	10,0 MB	npy File	29/12/2016 18:01
📄 idx_q.npy	10,0 MB	npy File	29/12/2016 18:01
📄 metadata.pkl	2,7 MB	pkl File	29/12/2016 18:01
📄 seq2seq_wrapper.py	7 KB	py File	23/12/2017 01:58
📄 the_best_chatbot.py	1 KB	py File	23/12/2017 02:08
▶ 📁 weights	--	Folder	23/12/2017 01:49

File explorer | Variable explorer | Help

IPython console

Console 1/A

```
    ...:
    ...:     # Getting the ChatBot predicted answer
    ...: def respond(question):
    ...:         encoded_question = data_utils_2.encode(question,
w2idx, limit['maxq'])
    ...:         answer = model.predict(sess, encoded_question)[0]
    ...:         return data_utils_2.decode(answer, idx2w)
    ...:
    ...:
    ...:     # Setting up the chat
    ...: while True :
    ...:     question = input("You: ")
    ...:     answer = respond(question)
    ...:     print ("ChatBot: "+answer)

################Start Chatting Below######################

You: hi
ChatBot: hows the weekend

You: great thanks
ChatBot: youre welcome

You: who won the first presidential debate
ChatBot: trump will be a better time for a man

You: hey happy birthday have a nice day
ChatBot: thank you

You:
```

Resources

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://distill.pub/2016/augmented-rnns/

https://github.com/REDFOX1899/Learn_Machine_Learning_in_3_Months

Online Courses

Deep Learning and **NLP A-Z**™ by Super Data Science

Andrew Ng Stanford Machine Learning

Machine Learning, Udemy (by Kirill Eremenko, Hadelin de Ponteves,

Deep Learning, Udemy (by Kirill Eremenko, Hadelin de Ponteves)

Thank You