భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Deep Learning

15 Encoder-Decoder Models & Attention

Dr. Konda Reddy Mopuri
Dept. of AI, IIT Hyderabad
Jan-May 2023

Encoder-Decoder Model

1. Revisit the ‘language modeling’ problem

Encoder-Decoder Model

1. Revisit the 'language modeling' problem
2. $y^* = \text{argmax } P(y_t/y_1, y_2 \dots y_{t-1})$

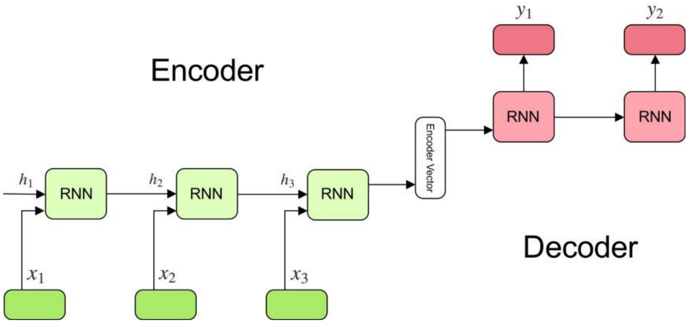
Encoder-Decoder Model

1. Revisit the 'language modeling' problem
2. $y^* = \text{argmax } P(y_t / y_1, y_2 \dots y_{t-1})$
3. We have an RNN consuming the i/p sequence $(y_1^{t-1}) \rightarrow$ **Encoder**

Encoder-Decoder Model

1. Revisit the 'language modeling' problem
2. $y^* = \text{argmax } P(y_t/y_1, y_2 \dots y_{t-1})$
3. We have an RNN consuming the i/p sequence $(y_1^{t-1}) \rightarrow$ **Encoder**
4. We have another RNN predicting the o/p (sequence of words after the i/p) $\rightarrow$ **Decoder**

Encoder-Decoder Model



Credits: Simeon Kostadinov

Encoder-Decoder Model

1. In general, both encoder and decoder use Neural networks

Encoder-Decoder Model

1. In general, both encoder and decoder use Neural networks
2. Based on the application, need modifications to the framework

Encoder-Decoder Model

1. In general, both encoder and decoder use Neural networks
2. Based on the application, need modifications to the framework
3. Basis for a lot of applications
    - Image (or video) captioning
    - Textual entailment
    - Machine translation
    - Transliteration
    - Document summarization
    - VQA: Visual Question Answering
    - Video classification
    - Chatbot for dialog

Encoder-Decoder Model

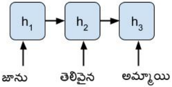
1. In general, both encoder and decoder use Neural networks
2. Based on the application, need modifications to the framework
3. Basis for a lot of applications
    - Image (or video) captioning
    - Textual entailment
    - Machine translation
    - Transliteration
    - Document summarization
    - VQA: Visual Question Answering
    - Video classification
    - Chatbot for dialog
4. Let's consider machine translation...

Encoder-Decoder for Machine Translation

Input sequence: $x_1, x_2, \ldots. x_T$

Output sequence: $y_1, y_2, \ldots. y_{T'}$

Encoder: $h_t = E(x_t, h_{t-1})$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

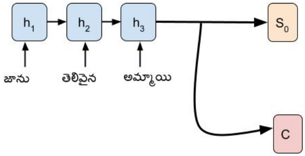
Encoder-Decoder for Machine Translation

Input sequence: $x_1, x_2, \dots x_T$

Output sequence: $y_1, y_2, \dots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Encoder: $h_t = E(x_t, h_{t-1})$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

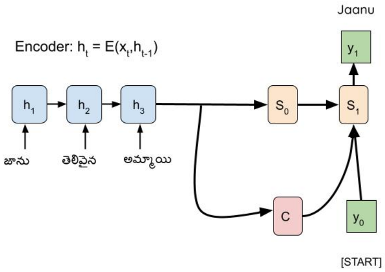
Input sequence: $x_1, x_2, \dots x_T$

Output sequence: $y_1, y_2, \dots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$

Encoder: $h_t = E(x_t, h_{t-1})$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

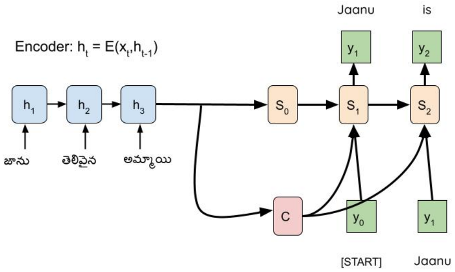
Input sequence: $x_1, x_2, \ldots x_T$

Output sequence: $y_1, y_2, \ldots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$

Encoder: $h_t = E(x_t, h_{t-1})$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

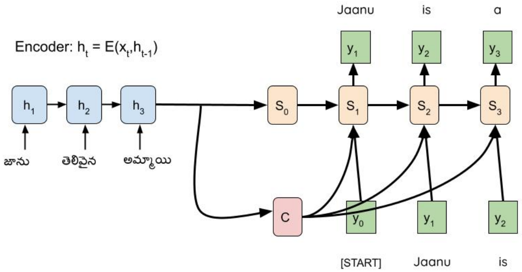
Encoder-Decoder for Machine Translation

Input sequence: $x_1, x_2, \ldots, x_T$

Output sequence: $y_1, y_2, \ldots, y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

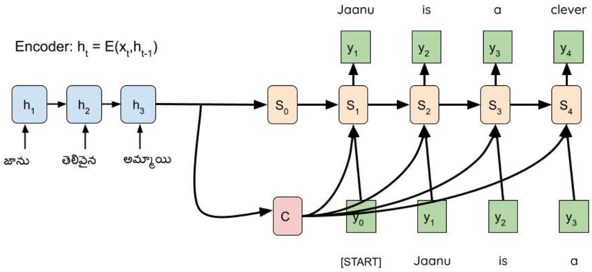
Input sequence: $x_1, x_2, \ldots x_T$

Output sequence: $y_1, y_2, \ldots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$

Encoder: $h_t = E(x_t, h_{t-1})$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

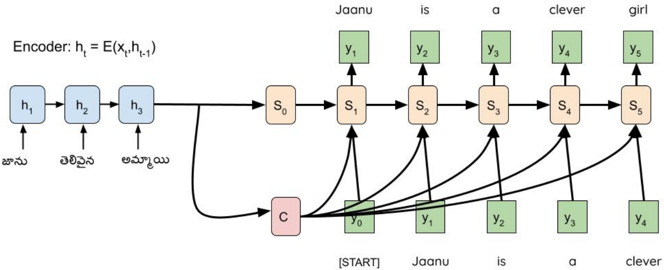
Input sequence: $x_1, x_2, \dots x_T$

Output sequence: $y_1, y_2, \dots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$

Encoder: $h_t = E(x_t, h_{t-1})$



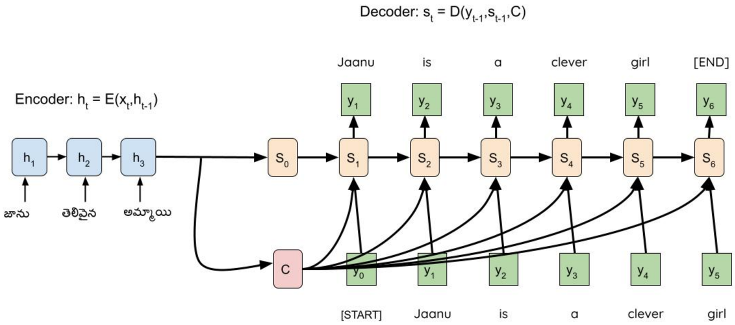
Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

Input sequence: $x_1, x_2, \dots x_T$

Output sequence: $y_1, y_2, \dots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$



Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

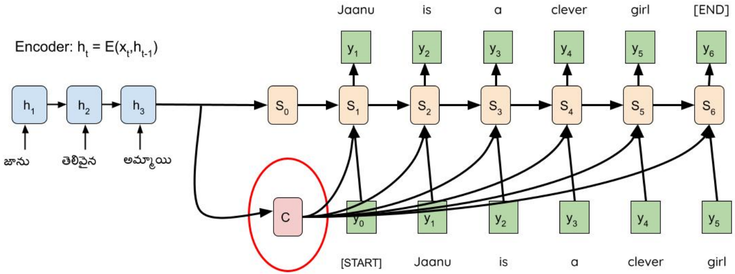
Input sequence: $x_1, x_2, \ldots\, x_T$

Output sequence: $y_1, y_2, \ldots\, y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C
E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$

Encoder: $h_t = E(x_t, h_{t-1})$



Bottleneck: Entire input is summarized by this vector!

Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation

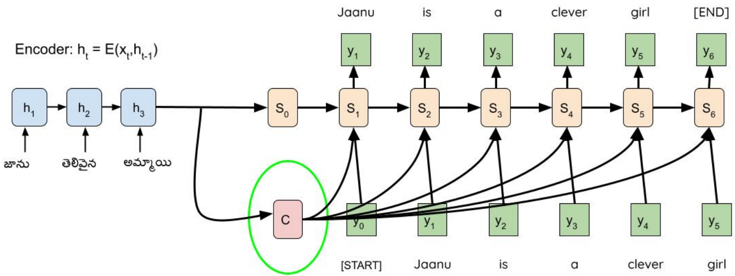
Input sequence: $x_1, x_2, \ldots x_T$

Output sequence: $y_1, y_2, \ldots y_{T'}$

Last hidden state $h_T \rightarrow$ Initial state of the Decoder S_0 and the context information C

E.g. $S_0 \leftarrow h_T$ + dense layers, and $C \leftarrow h_T$

Decoder: $s_t = D(y_{t-1}, s_{t-1}, C)$



Solution: use different context at each time step!

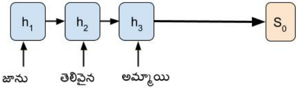
Sequence to sequence learning by Sutskever et al. NeurIPS 2014

Encoder-Decoder for Machine Translation with Attention

Input sequence: $x_1, x_2, \ldots x_T$

Input sequence: $y_1, y_2, \ldots y_{T'}$

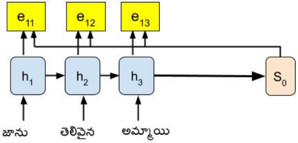
Encoder: $h_t = E(x_t, h_{t-1})$



Encoder-Decoder for Machine Translation with Attention

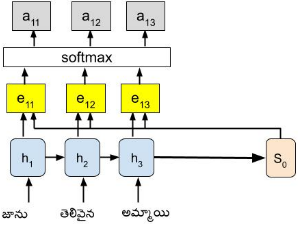
Compute the alignment scores

$e_{t,i} = f_{att}(s_{t-1}, h_i)$ f_{att} - couple of dense layers



Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention

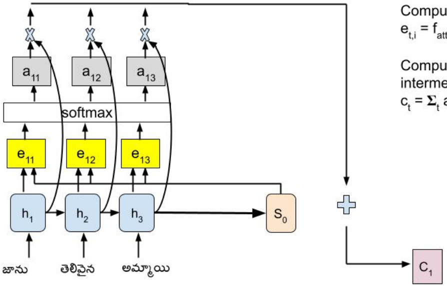


Compute the alignment scores

$e_{t,i} = f_{att}(s_{t-1}, h_i)$ f_{att} - couple of dense layers

Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention



Compute the alignment scores

$e_{t,i} = f_{att}(s_{t-1}, h_i)$ f_{att} - couple of dense layers

Compute the context as a linear combination of intermediate hidden states

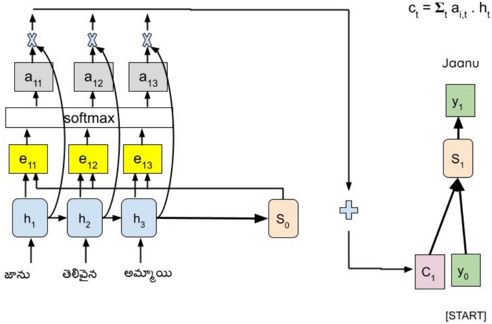
$c_t = \Sigma_t \, a_{i,t} \cdot h_t$

Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention

Compute the alignment scores
$e_{t,i} = f_{att}(s_{t-1}, h_i)$ f_{att} - couple of dense layers

Compute the context as a linear combination of intermediate hidden states
$c_t = \Sigma_t \, a_{i,t} \cdot h_t$



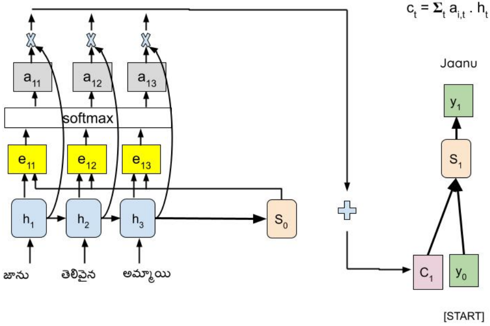
Decoder: $s_t = D(y_{t-1}, C_t)$

Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention

Compute the alignment scores
$e_{t,i} = f_{att}(s_{t-1}, h_i)$ f_{att} - couple of dense layers

Compute the context as a linear combination of intermediate hidden states
$c_t = \Sigma_t a_{i,t} . h_t$

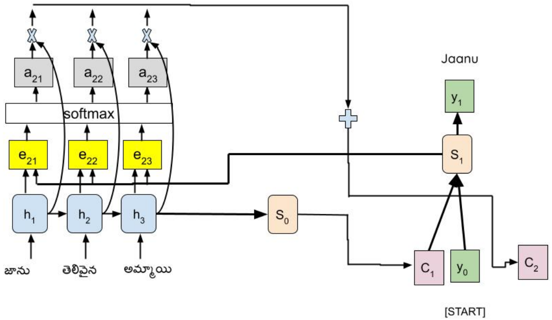


Decoder: $s_t = D(y_{t-1}, C_t)$

All these operations are differentiable!
Attention is learned using backprop!!

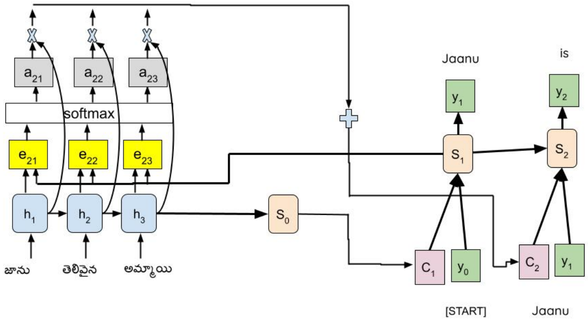
Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention



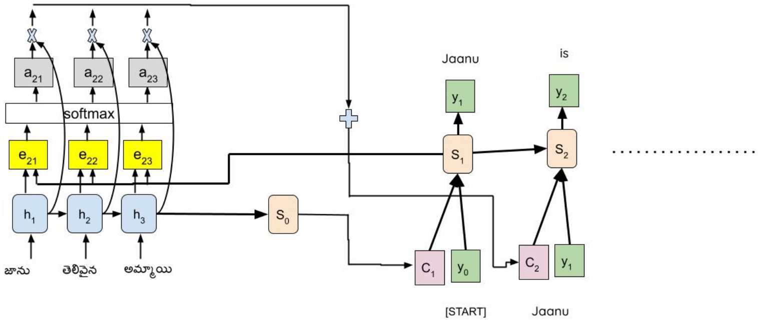
Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention



Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention



Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention

- Employs a different context at each time step of decoding

Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention

- Employs a different context at each time step of decoding
- No more bottleneck-ing of the input

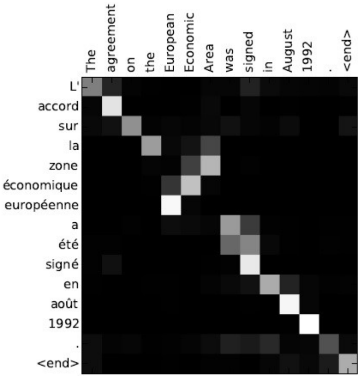
Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

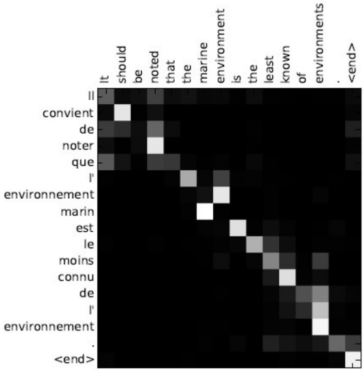
Encoder-Decoder for Machine Translation with Attention

- Employs a different context at each time step of decoding
- No more bottleneck-ing of the input
- Decoder can ‘attend’ to different portions of the input at each time step

Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

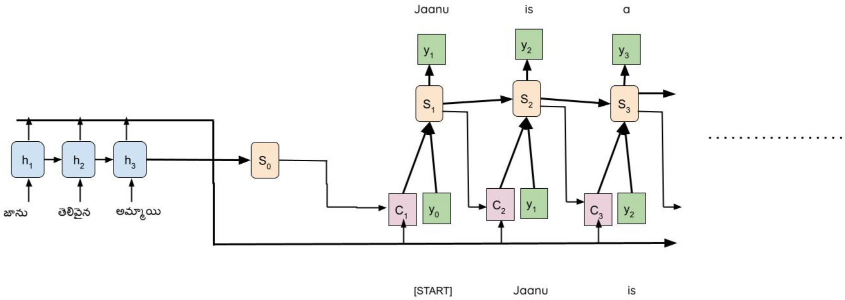
Encoder-Decoder for Machine Translation with Attention





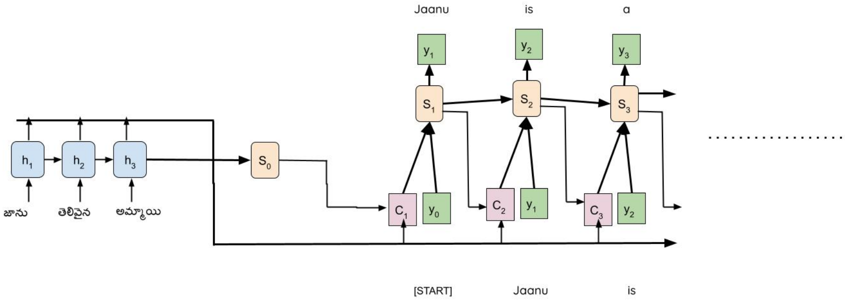
Neural Machine Translation with aligning by Bahdanau et al. ICLR 2015

Encoder-Decoder for Machine Translation with Attention



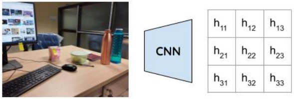
- Decoder doesn't consider the h_i to be an ordered set

Encoder-Decoder for Machine Translation with Attention



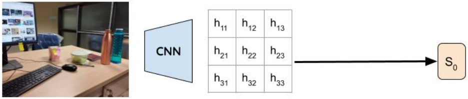
- Decoder doesn't consider the h_i to be an ordered set
- This architecture can be exploited to process a set of inputs h_i

Image captioning using RNNs with Attention



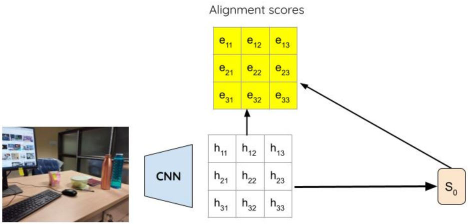
Show Attend and Tell by Xu et al. 2015

Image captioning using RNNs with Attention



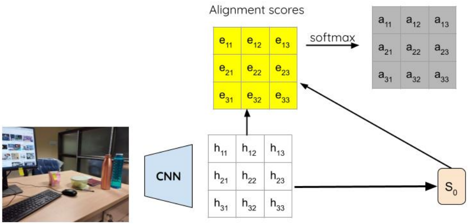
Show Attend and Tell by Xu et al. 2015

Image captioning using RNNs with Attention



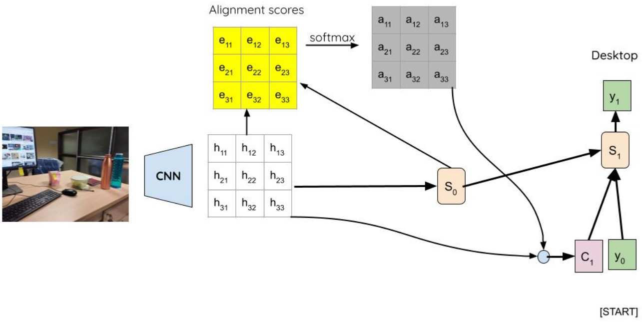
Show Attend and Tell by Xu et al. 2015

Image captioning using RNNs with Attention



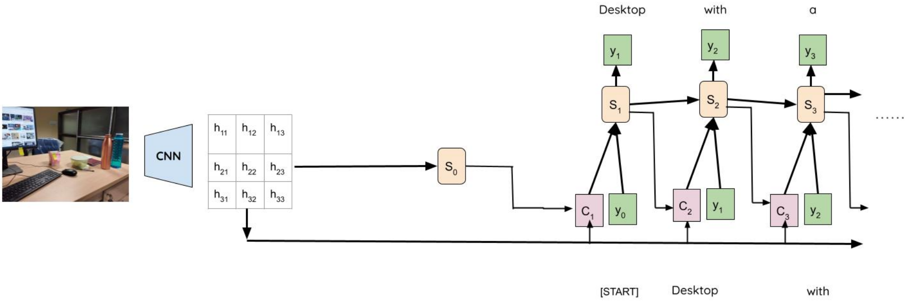
Show Attend and Tell by Xu et al. 2015

Image captioning using RNNs with Attention



Show Attend and Tell by Xu et al. 2015

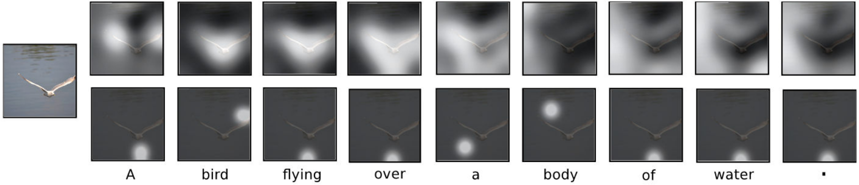
Image captioning using RNNs with Attention



Show Attend and Tell by Xu et al. 2015

Image captioning using RNNs with Attention

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)



Show Attend and Tell by Xu et al. 2015

Image captioning using RNNs with Attention

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.

Show Attend and Tell by Xu et al. 2015