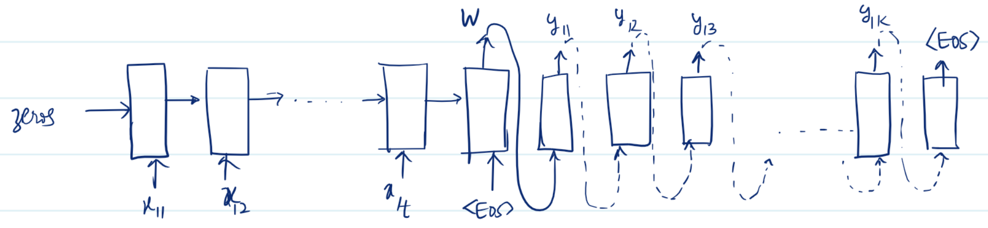
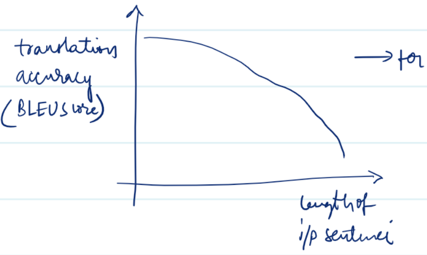
Attention Based Models :-

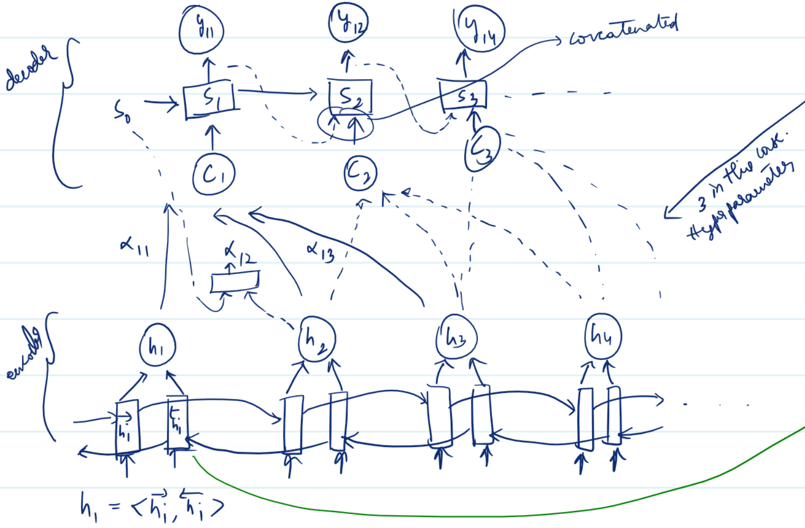


→ W on it's own is not able to capture the essence of the entire sentence if it's long.



→ for simple encoder-decoder models.

⇒ We use bidirectional RNN because it might depend on words before/after a word in encoder. Decoder is unidirectional RNN



$h_1 = <\vec{h}_1, \overleftarrow{h}_1>$

content vector $c_1 = \sum_{i=1}^{T_x} h_i \alpha_{1i}$

weights, $\sum_{i=1}^{T_x} \alpha_{1i} = 1$ & $\alpha_{1i} \geq 0$

$S_0 = \tanh(\underbrace{W_s}_{\text{weight matrix}} * \overleftarrow{h}_1)$

How to Compute α_{ij}'s :-

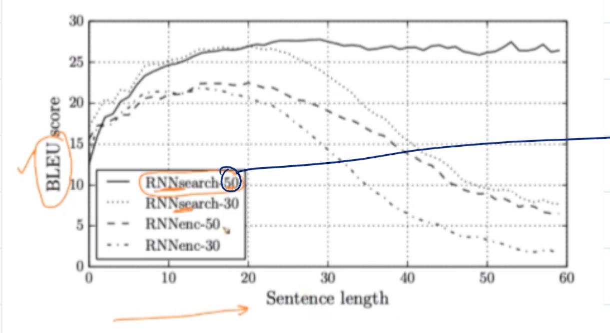
$$\alpha_{ij} = \frac{\overbrace{\exp(e_{ij})}^{\text{always +ve/0.}}}{\underbrace{\sum_{k=1}^{T_x} \exp(e_{ik})}_{\text{denom will be same so sum = 1}}}$$

$e_{ij} = a(s_{i-1}, h_j)$

attention model (how much attention needs to be given to encoder). We can use a simple feed forward NN for this.

ex:- $e_{12} = a(\overset{\text{prev s}}{S_0}, h_2)$ & current h

→ Train using Backprop through time using adam & let it converge.

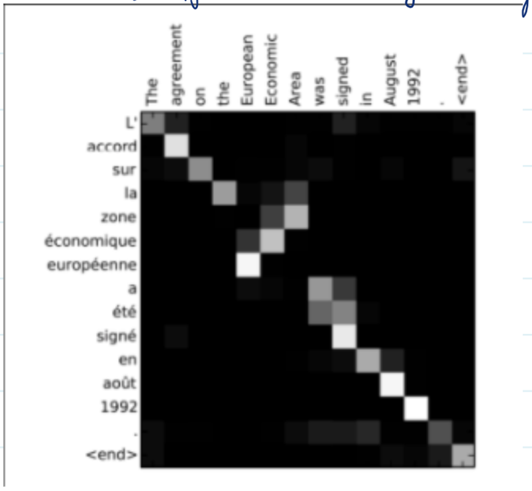


→ T_x

RNNSearch = Attention Model.

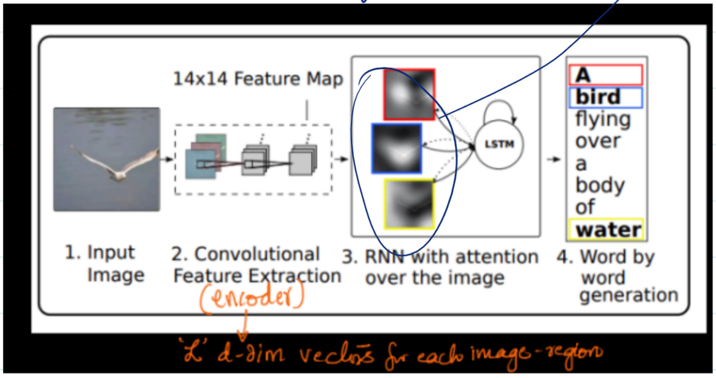
Drawback:- Time Complexity $= O(\underbrace{K_1}_{\text{length of i/p}} . \overbrace{K_2}^{\text{length of o/p.}})$

→ We can visualize α_{ij}'s like this to debug. Accuracy Measurement can be done by using metrics like BLEU score.



→ Image caption generation using Visual Attention :-

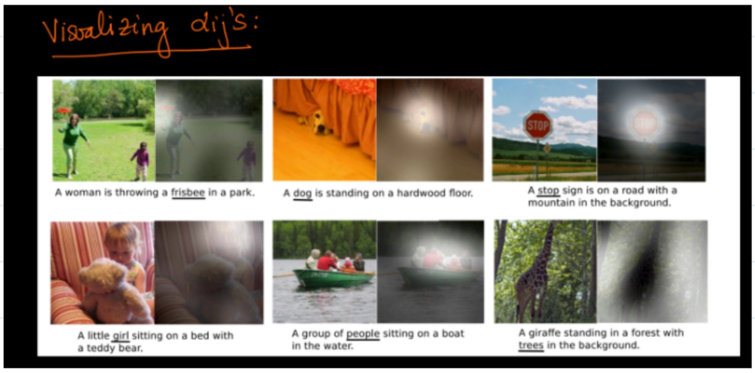
→ every region is basically like a word. if region_j is white ⇒ α of that region is large.


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

(encoder)

'L' d-dim vectors for each image-region

→


Visualizing α_{ij}'s:

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.

(Can also be used as localization)