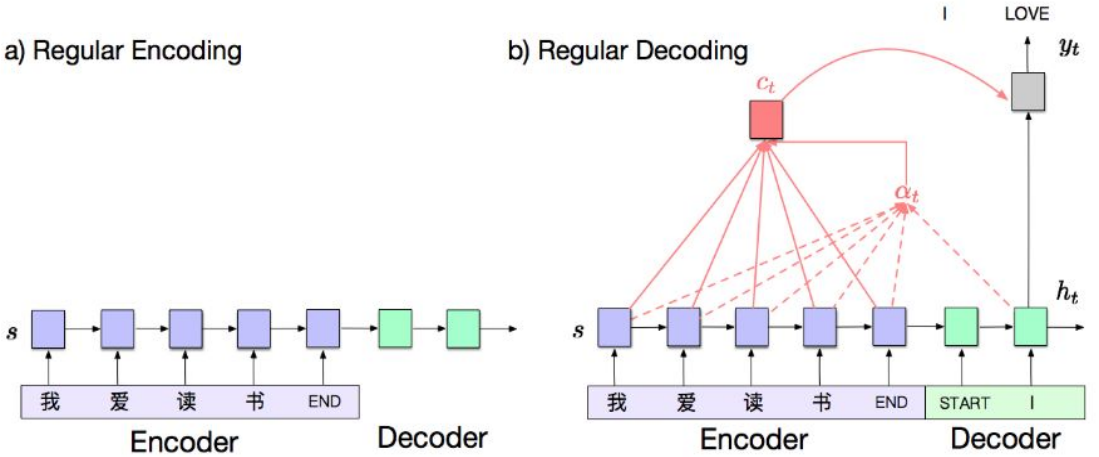
Attention on Hierarchical Representation of Inputs

Mayank Gupta and Bazil Ahmed

Background



$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{W} \bar{\boldsymbol{h}}_s & \text{[Luong's multiplicative style]} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_1} \boldsymbol{h}_t + \boldsymbol{W_2} \bar{\boldsymbol{h}}_s\right) & \text{[Bahdanau's additive style]} \end{cases}$$

Attention mechanism is used in a lot of applications to improve the performance of standard seq-to-seq model.

Attention mechanism calculates context vector for each decoder time step. There exists many ways to calculate context vectors. Bahdanau's and Luong's attention are the most commonly used.

With standard Attention mechanism, we have to look at complete encoder outputs to create a context vector for each decoder time step which makes it inefficient for longer sequences.

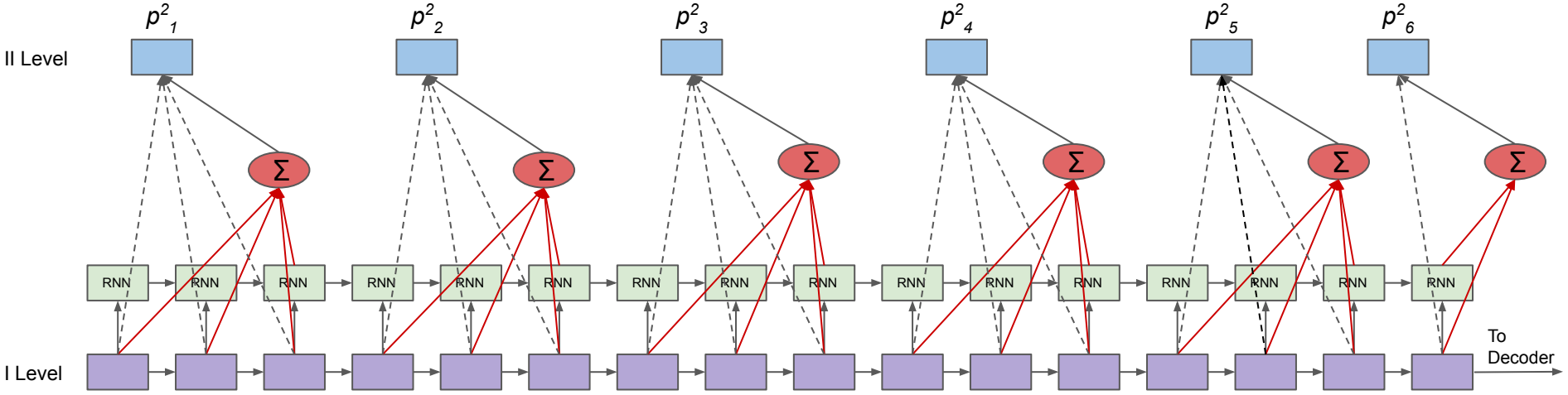
Proposed Approach

- In this work, we propose a multi-level attention mechanism that has better time complexity than the standard attention mechanism.
- In our technique, we represent the input sequence using different levels where the first level is the hidden states obtained by running an LSTM over the input sequence. And the length of vectors in the subsequent level decreases by a constant factor. Using this level structure we can describe the input compactly where each level contains coarser details than the previous level.
- For each decoder time stamp, we will start by attending to the top-most level representation of input and will subsequently go down a particular branch based on the computed attention scores.

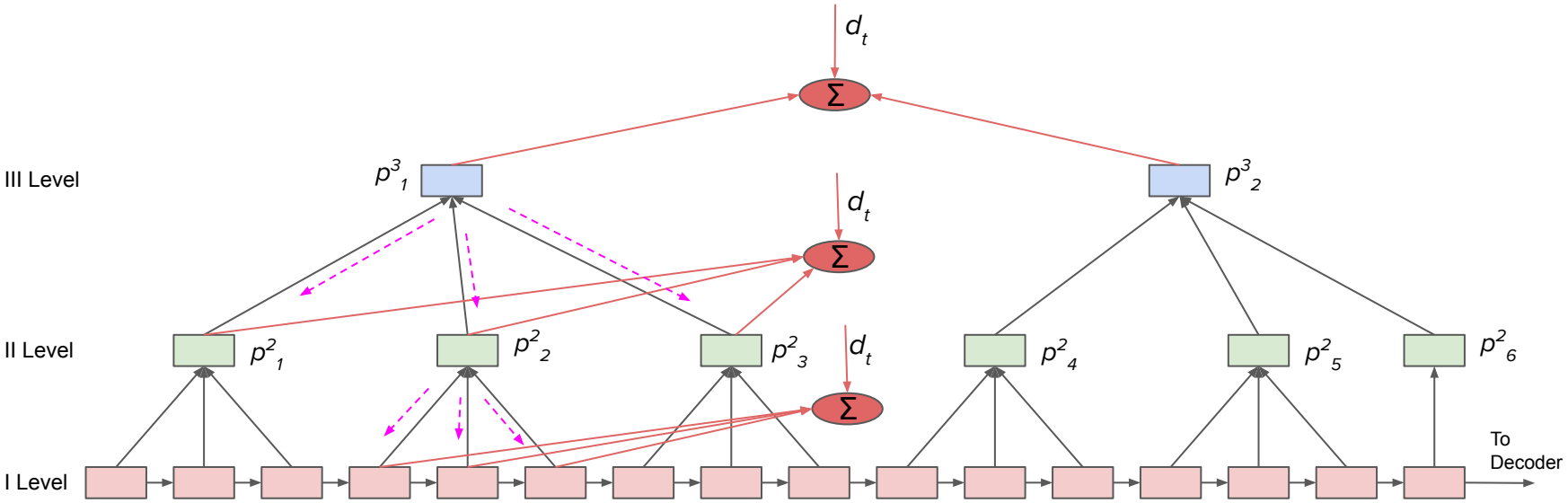
❖ We propose three different approaches to get the representation of input sequence:
  - ➢ RNN
  - ➢ CNN
  - ➢ ACT

Technical Details

- Encoder hidden states are divided into blocks and a representative hidden vector for each block at the next level are computed.
- Figure shows encoder for 2 levels and block size =3 for RNN based hierarchy
- CNN works in similar fashion with fixed block sizes while with ACT block sizes are dynamically adapted for each input sequence



- At decoder, we find attention on the top most level representative hidden vectors, and go deeper along the branches whose representative hidden vector gets the most attention score by the decoder
- Final context vector is a convex combination of all hidden states of Encoder



Contributions (Novelty)

- We proposed a new attention mechanism in which we build multi-level representation of inputs.
- We implemented 3 ways to make the representation of input for our attention mechanism
  - Using RNN
  - Using CNN
  - Using ACT

- The proposed attention mechanism is computationally efficient than the standard attention mechanism.

Model	Complexity
Seq2Seq	$\mathcal{O}(D^2(S+T))$
Bahdanau	$\mathcal{O}(D^2(ST))$
2-Levels	$\mathcal{O}(D^2(ST)/k)$
$\log_k^T$ Levels	$\mathcal{O}(D^2(Sk\log_k^T))$

Results & Conclusion

Model	k	levels	loss	PPI	Bleu
Bahdanau	-	-	3.214	24.872	29.33
CNN	3	3	3.411	30.293	23.82
RNN	6	2	3.358	28.727	24.02
ACT (3 epochs)	3	3	3.764	43.105	19.39

Table 1. Results of NMT Task on Multi30k Dataset

Model	k	levels	loss	PPI
Bahdanau	-	-	3.214	24.872
CNN	5	2	1.842	6.312
RNN	5	3	2.056	7.811

Table 2. Results of Sequence Copy Task for sequence length in [10, 100]

- All models share the same hyperparameters and has been trained for 10 epochs unless explicitly mentioned.
- ACT was highly time consuming hence trained for only 3 epochs.
- Parallelizable Operations could not be reduced to Matrix operations, hence our mechanism turned out to be slower in our experiments.
- Better accuracy can be achieved if multiple branches are explored during computation of context vector.

Attention Visualized

Attention flow is shown for word ‘black’, where source sequence is 'ein schwarzer hund und ein gefleckter hund kämpfen.' and the target sequence is 'a black dog and a spotted dog are fighting'.

Decoder word: ‘black’