PRESENTATION BY

ZUBIN BHUYAN

ENHANCED LSTM FOR NATURAL LANGUAGE INFERENCE

QIAN CHEN, XIAODAN ZHU, ZHENHUA LING, SI WEI, HUI JIANG, DIANA INKPEN

ACL (2017)

https://arxiv.org/abs/1609.06038v3

OUTLINE

- Natural Language Inference
- Hybrid Neural Inference Models
  - Overview of inference networks
  - Architecture Details
- Results

NATURAL LANGUAGE INFERENCE

- Semantic concepts of entailment and contradiction.
- Given two strings *predict the logical relationship between them*.

NATURAL LANGUAGE INFERENCE

- Two sentences, a premise and a hypothesis:
  - $\mathbf{a} = (a_1, \ldots, a_{l_a})$ (Premise)
  - $\mathbf{b} = (b_1, \ldots, b_{l_b})$ (Hypothesis)
    - a_i and $b_j \in \mathbb{R}^l$ are embedding of *l*-dimensional vector
- Goal: Predict label *y* which indicates the logic relationship between **a** and **b**.

SNLI

- Stanford Natural Language Inference corpus
  - collection of sentence pairs labeled for:
    - entailment
    - contradiction and
    - semantic independence.

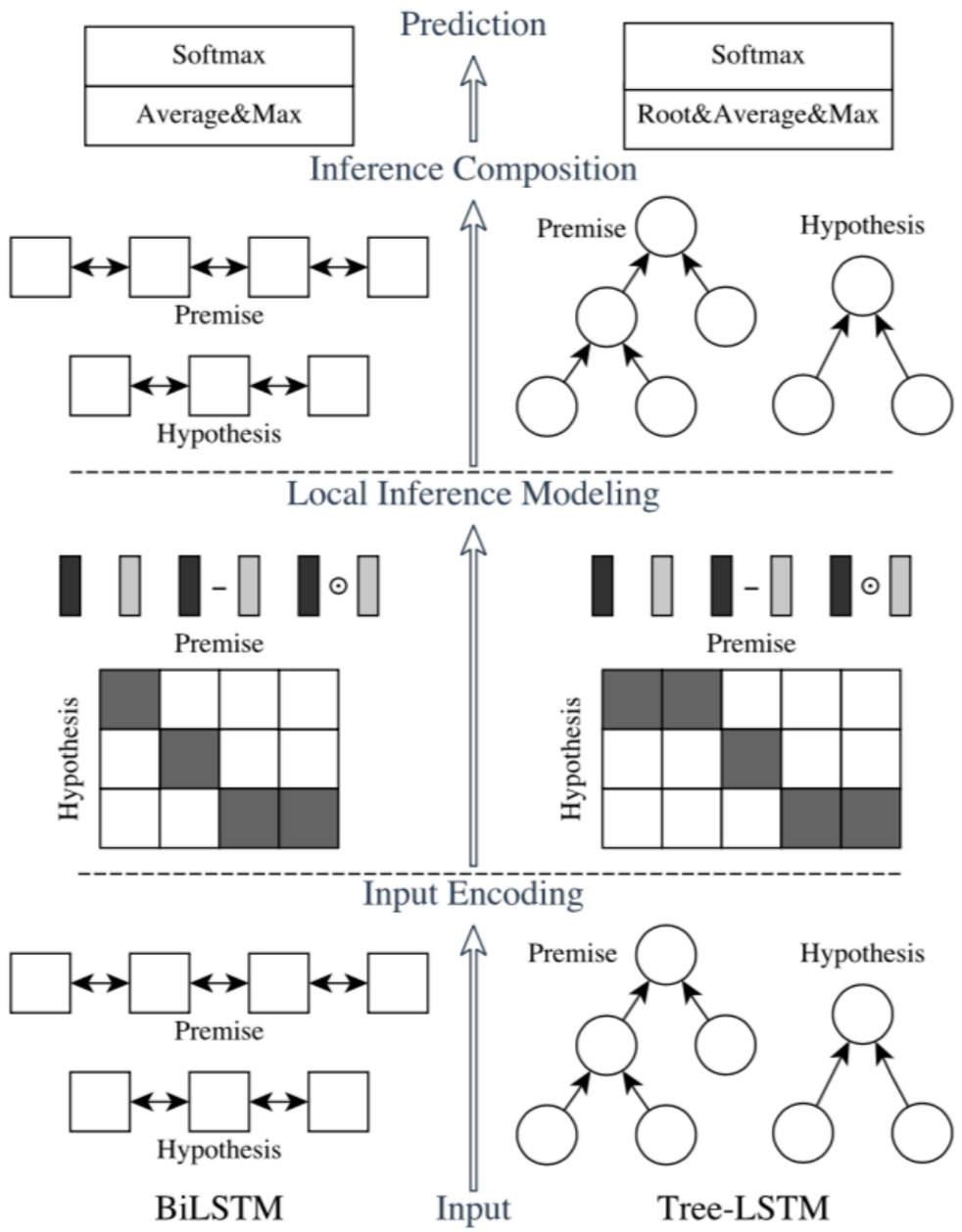
SNLI

- 570,152 sentence pairs.

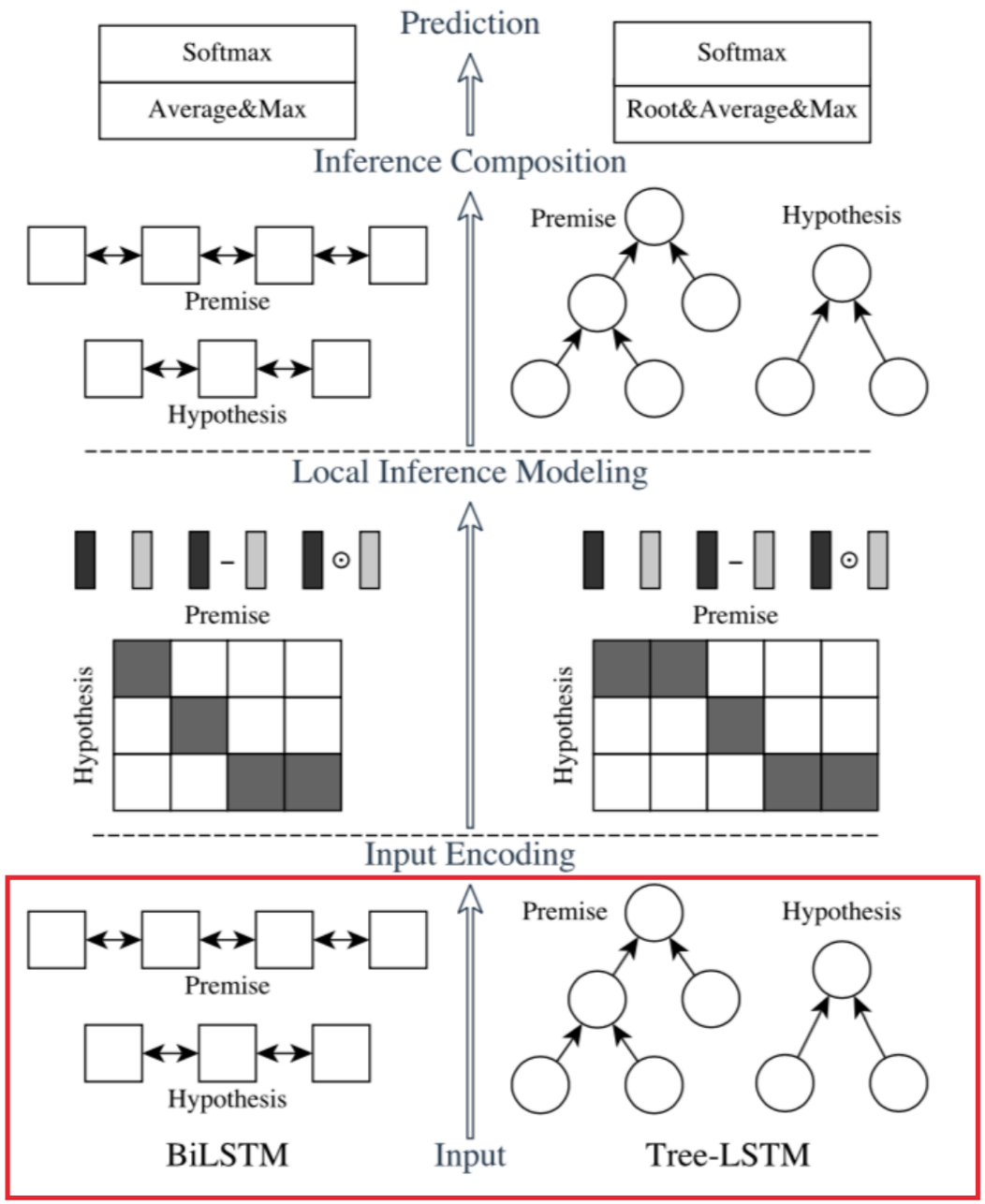
A man inspects the uniform of a figure in some East Asian country.	**contradiction** C C C C C	The man is sleeping
An older and younger man smiling.	**neutral** N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	**contradiction** C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	**entailment** E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	**neutral** N N E C N	A happy woman in a fairy costume holds an umbrella.

- Bowman, et al., A large annotated corpus for learning natural language inference, 2015

OVERVIEW OF THE HYBRID NEURAL INFERENCE NETWORK

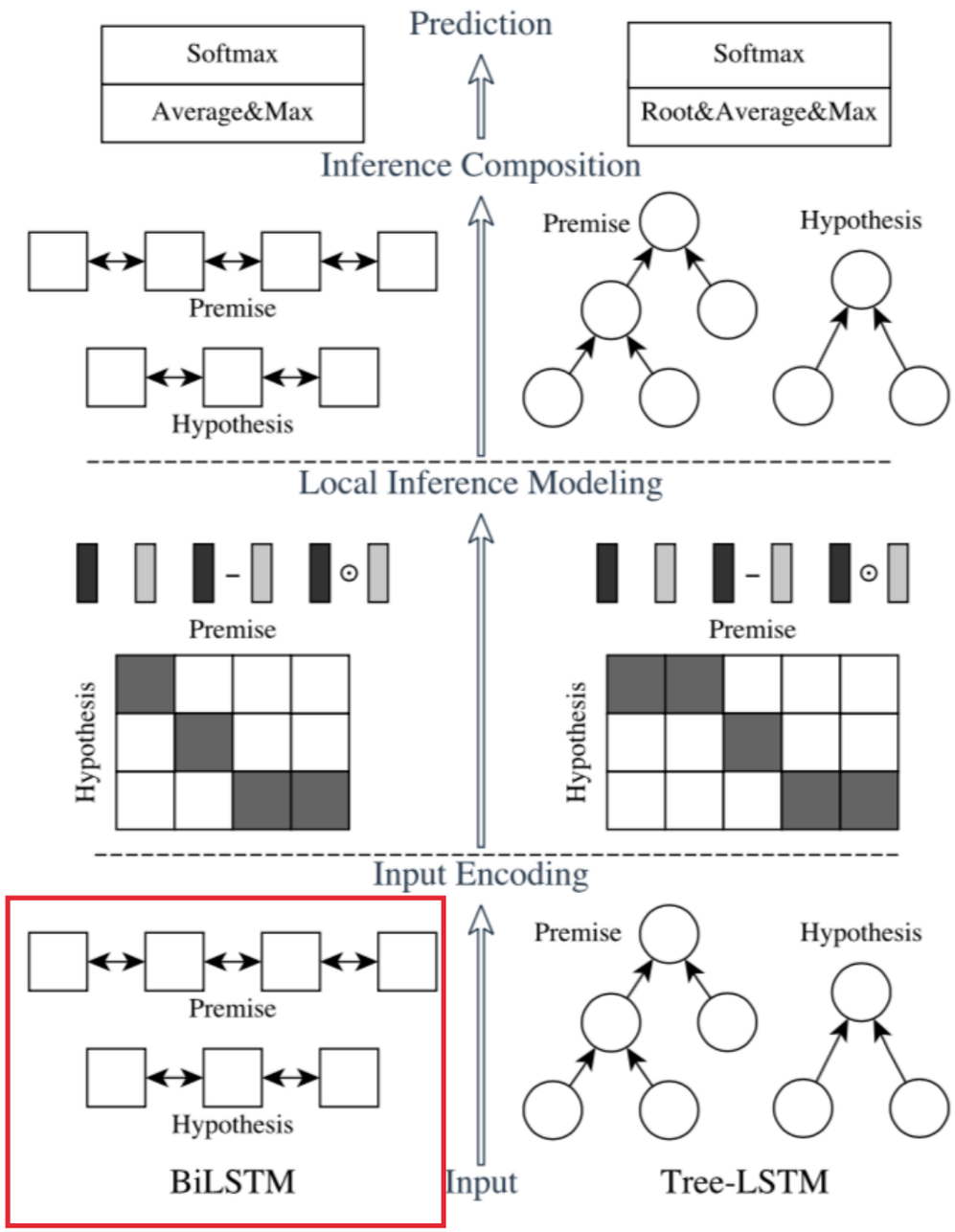


NEURAL INFERENCE NETWORK



- Input encoding:
  - Initialize word embeddings with pertained 300-D Glove 840B vectors
- Bi-directional LSTM basic building block.
- All hidden states of LSTMs and word embeddings have 300 dimension.

NEURAL INFERENCE NETWORK

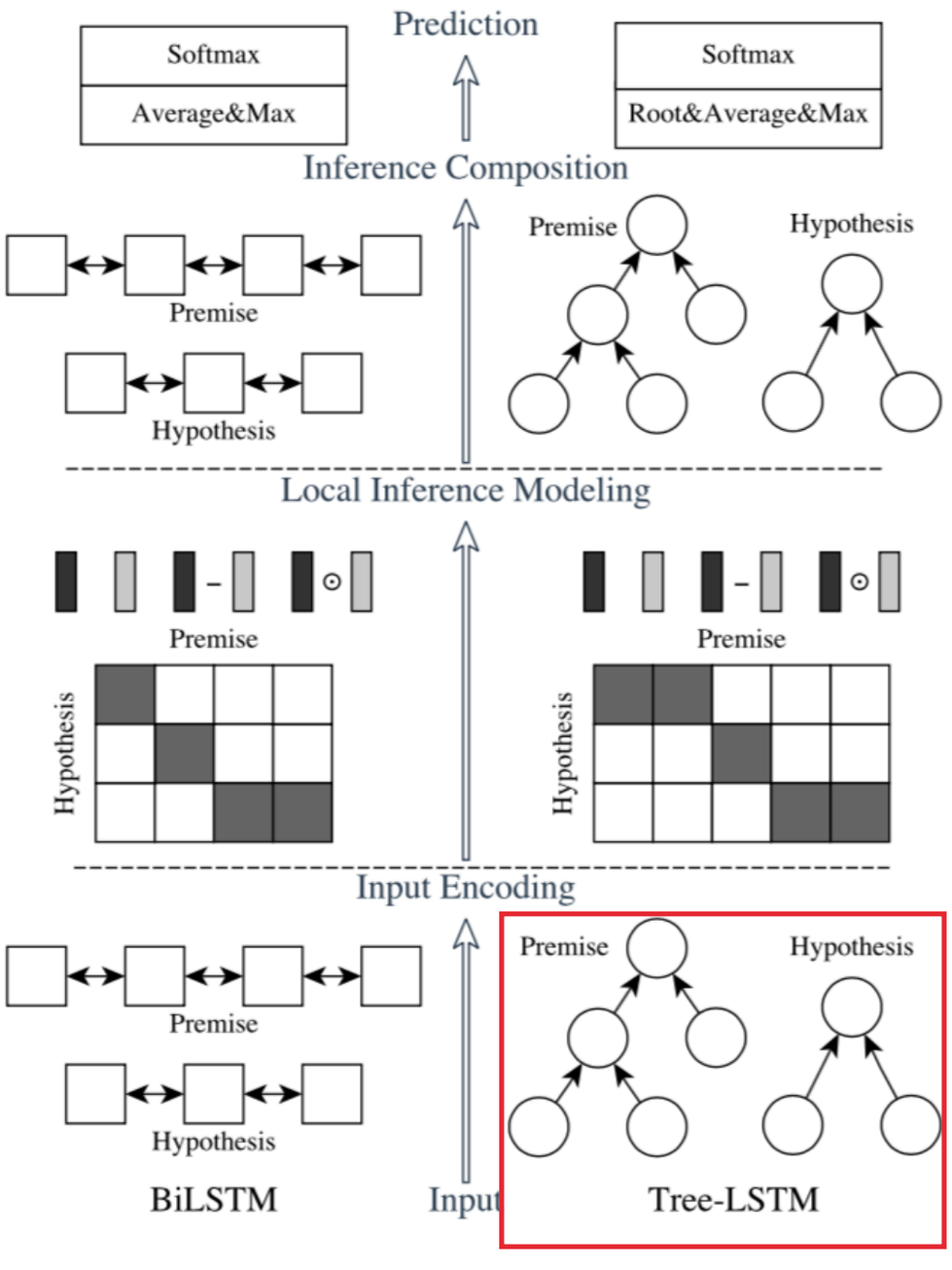


- Bi-LSTM learns to encode words and context.

$$\bar{\mathbf{a}}_i = \mathbf{BiLSTM}(\mathbf{a}, i), \forall i \in [1, \ldots, \ell_a],$$
$$\bar{\mathbf{b}}_j = \mathbf{BiLSTM}(\mathbf{b}, j), \forall j \in [1, \ldots, \ell_b].$$

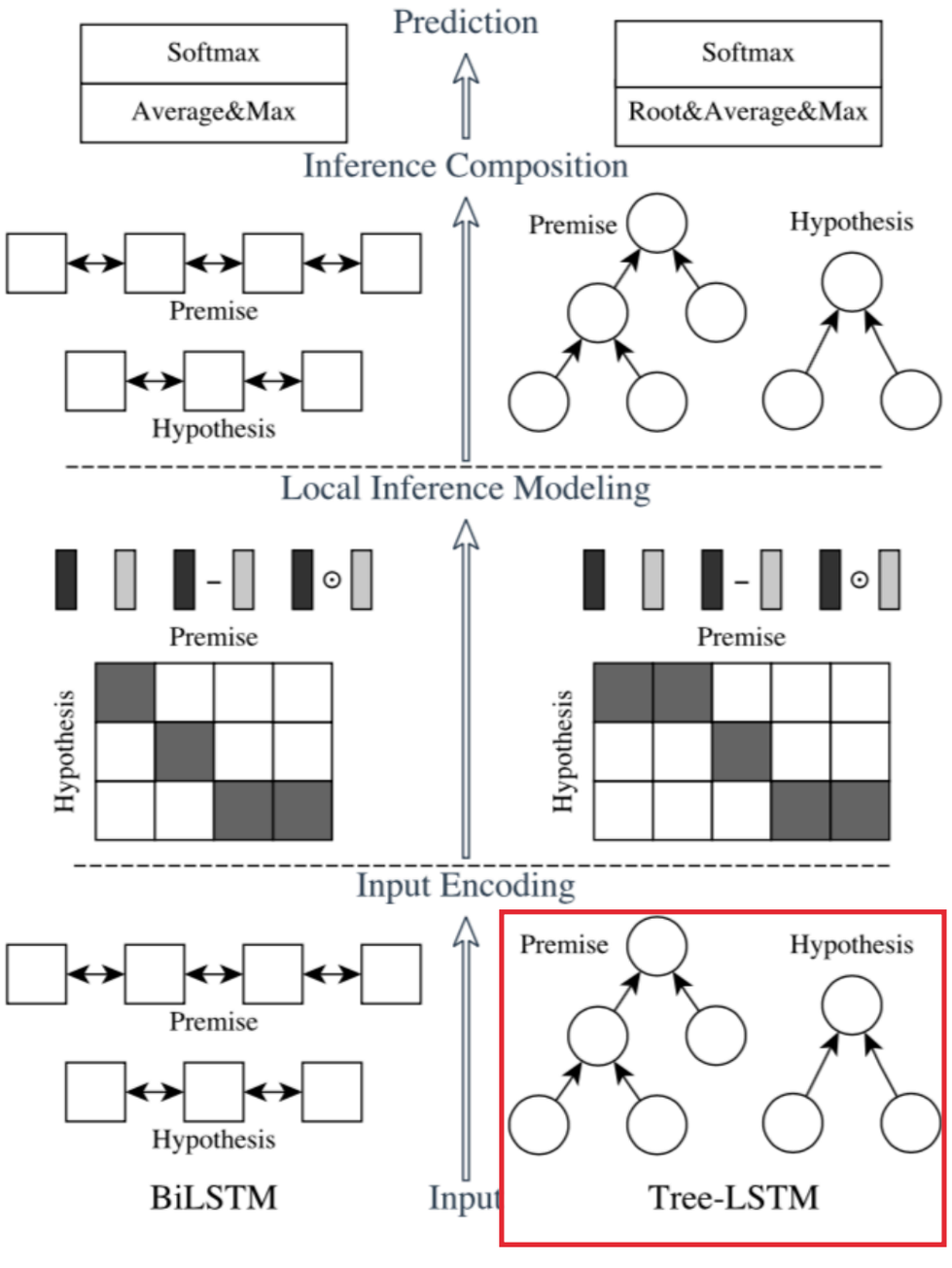
- Hidden states generated by these two LSTMs at each time step are concatenated to represent that time step and its context.

NEURAL INFERENCE NETWORK

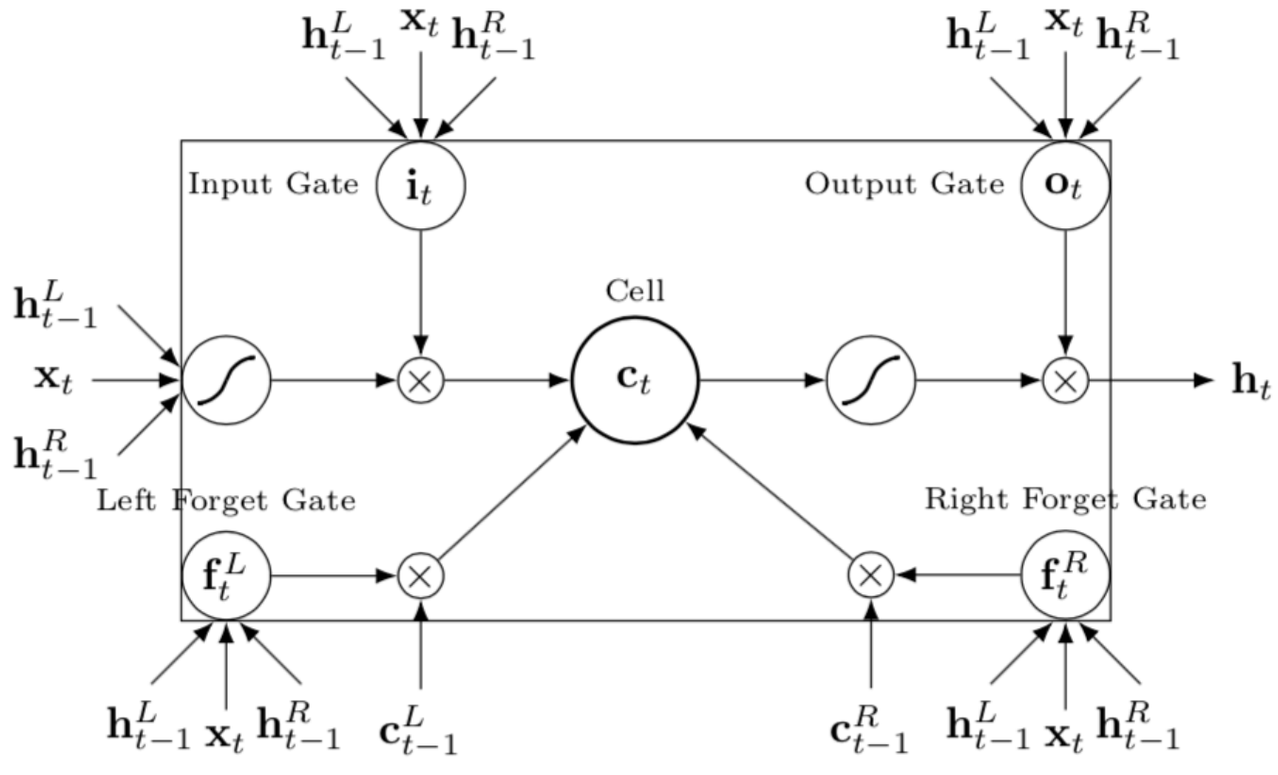


- Tree LSTM:
  - Given the parse of a premise or hypothesis, a tree node is deployed with a tree-LSTM memory block depicted

NEURAL INFERENCE NETWORK

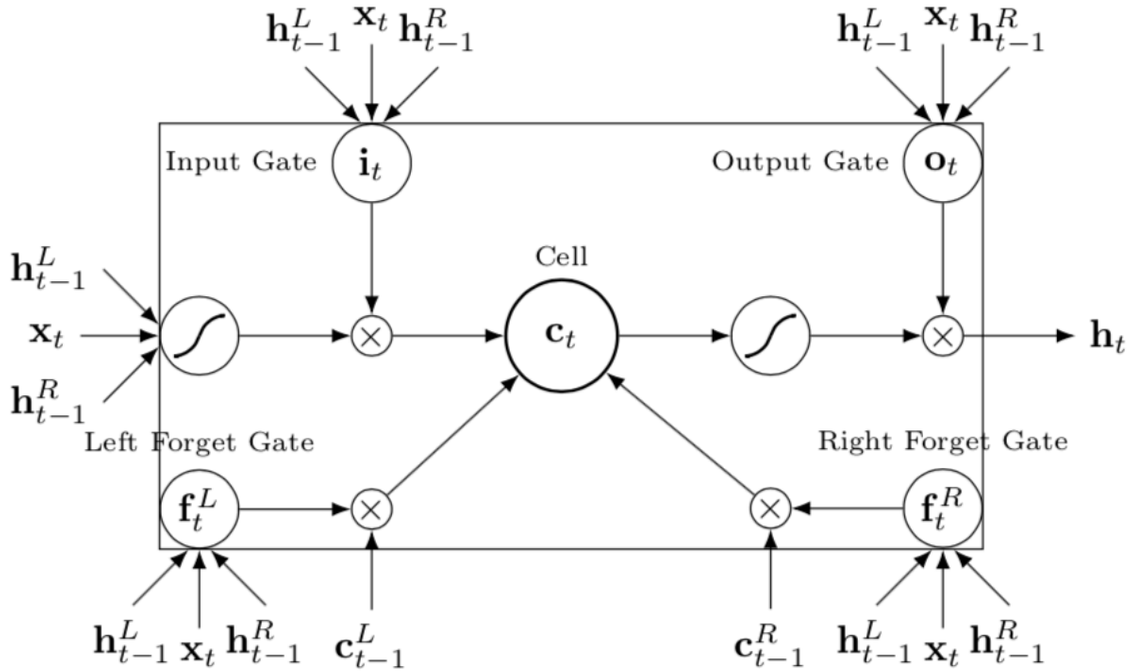


- Tree LSTM
  - A tree-LSTM memory block



NEURAL INFERENCE NETWORK

- A tree-LSTM memory block:
  - Input gate
  - Output gate
  - 2 forget gates



NEURAL INFERENCE NETWORK

- A tree-LSTM memory block:

$$\mathbf{h}_t = \mathrm{TrLSTM}(\mathbf{x}_t, \mathbf{h}^L_{t-1}, \mathbf{h}^R_{t-1}),$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}^L_o\mathbf{h}^L_{t-1} + \mathbf{U}^R_o\mathbf{h}^R_{t-1}),$$

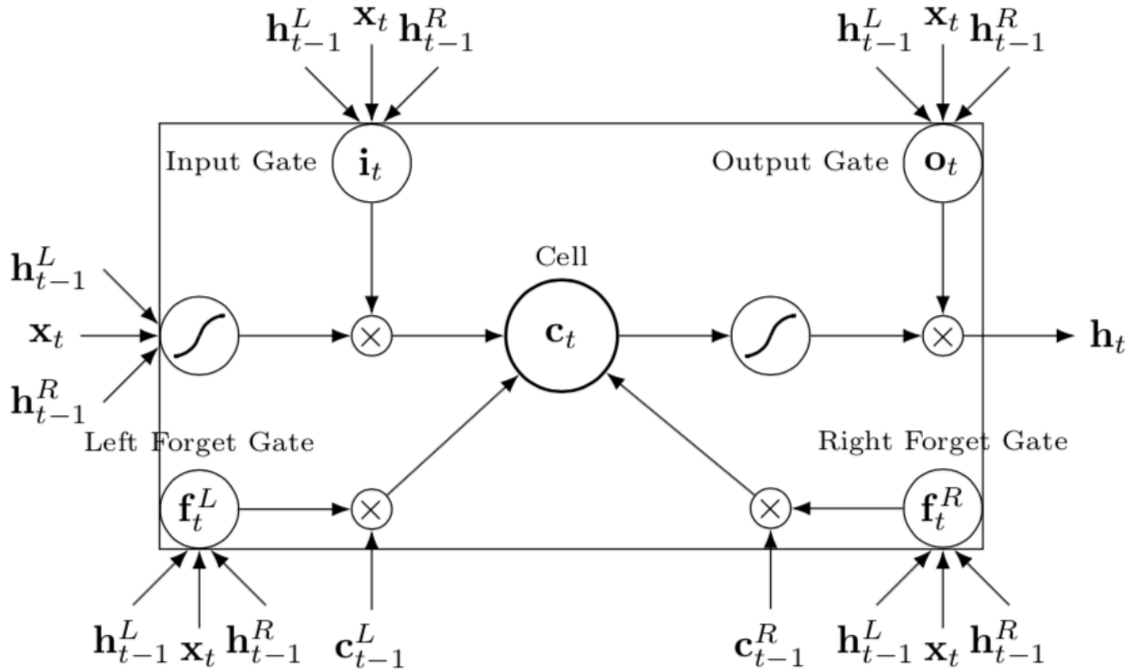
$$\mathbf{c}_t = \mathbf{f}^L_t \odot \mathbf{c}^L_{t-1} + \mathbf{f}^R_t \odot \mathbf{c}^R_{t-1} + \mathbf{i}_t \odot \mathbf{u}_t,$$

$$\mathbf{f}^L_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}^{LL}_f\mathbf{h}^L_{t-1} + \mathbf{U}^{LR}_f\mathbf{h}^R_{t-1}),$$

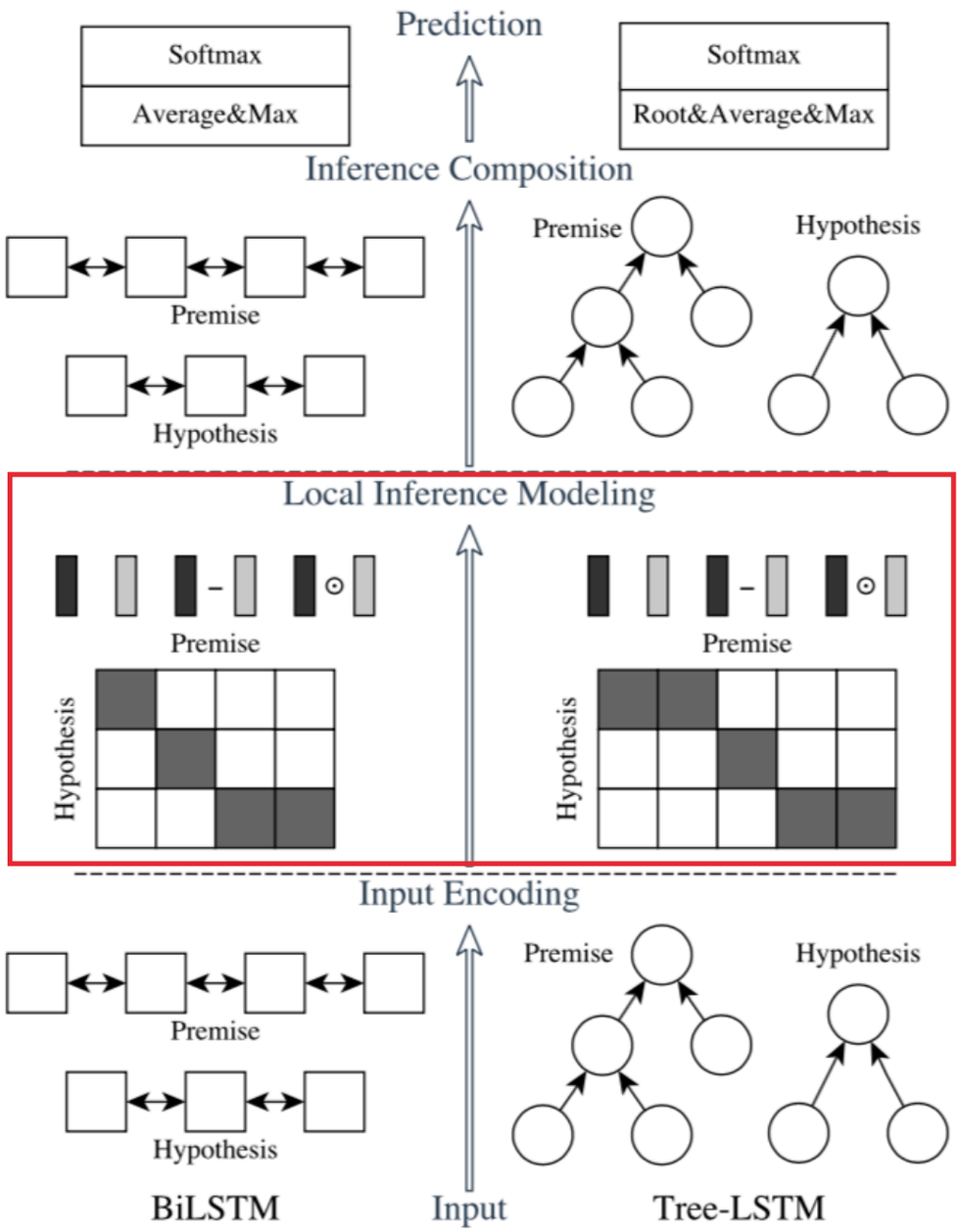
$$\mathbf{f}^R_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}^{RL}_f\mathbf{h}^L_{t-1} + \mathbf{U}^{RR}_f\mathbf{h}^R_{t-1}),$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}^L_i\mathbf{h}^L_{t-1} + \mathbf{U}^R_i\mathbf{h}^R_{t-1}),$$

$$\mathbf{u}_t = \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}^L_c\mathbf{h}^L_{t-1} + \mathbf{U}^R_c\mathbf{h}^R_{t-1}),$$

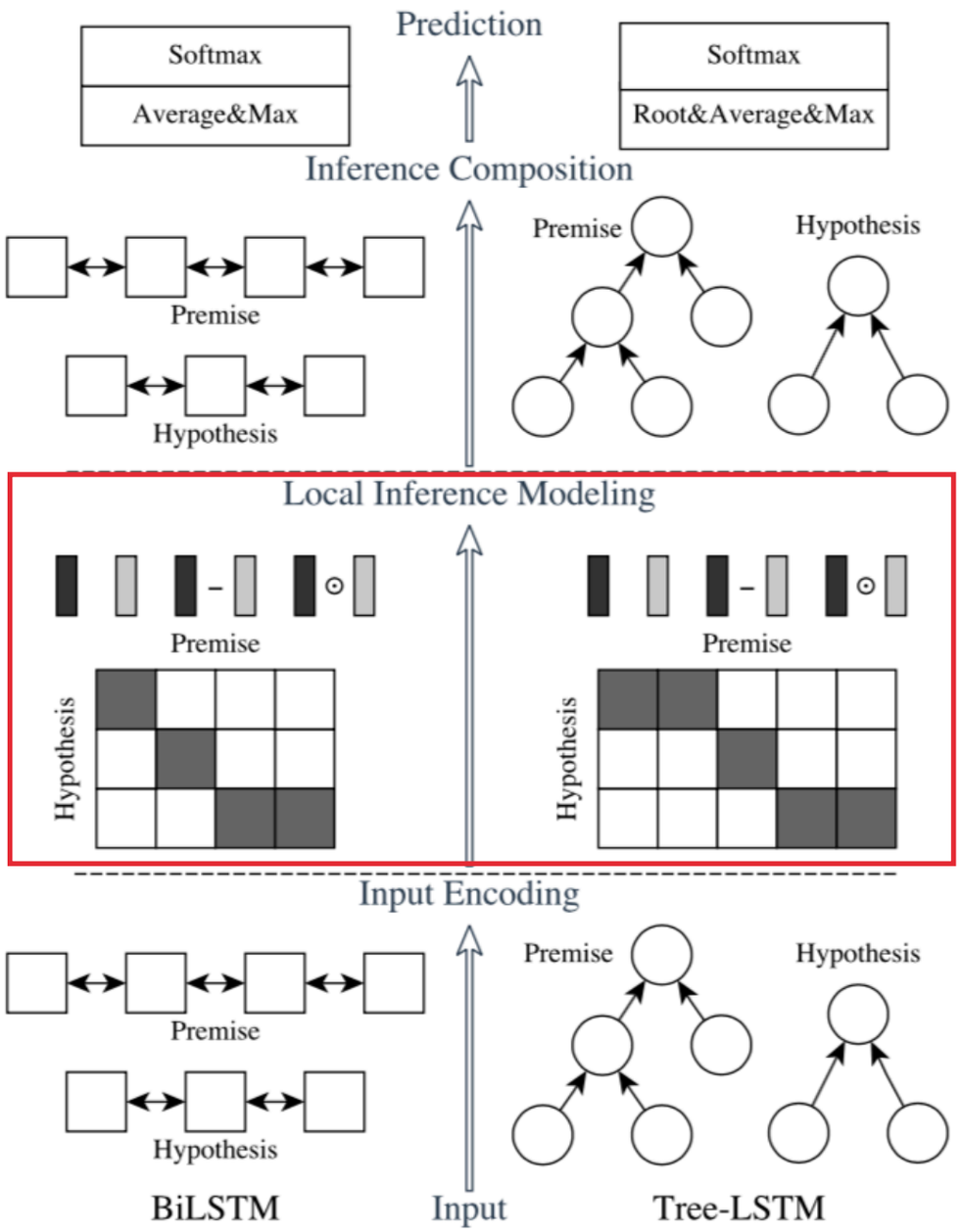


LOCAL INFERENCE MODELLING



- Modeling local inference between premise and a hypothesis

LOCAL INFERENCE MODELLING

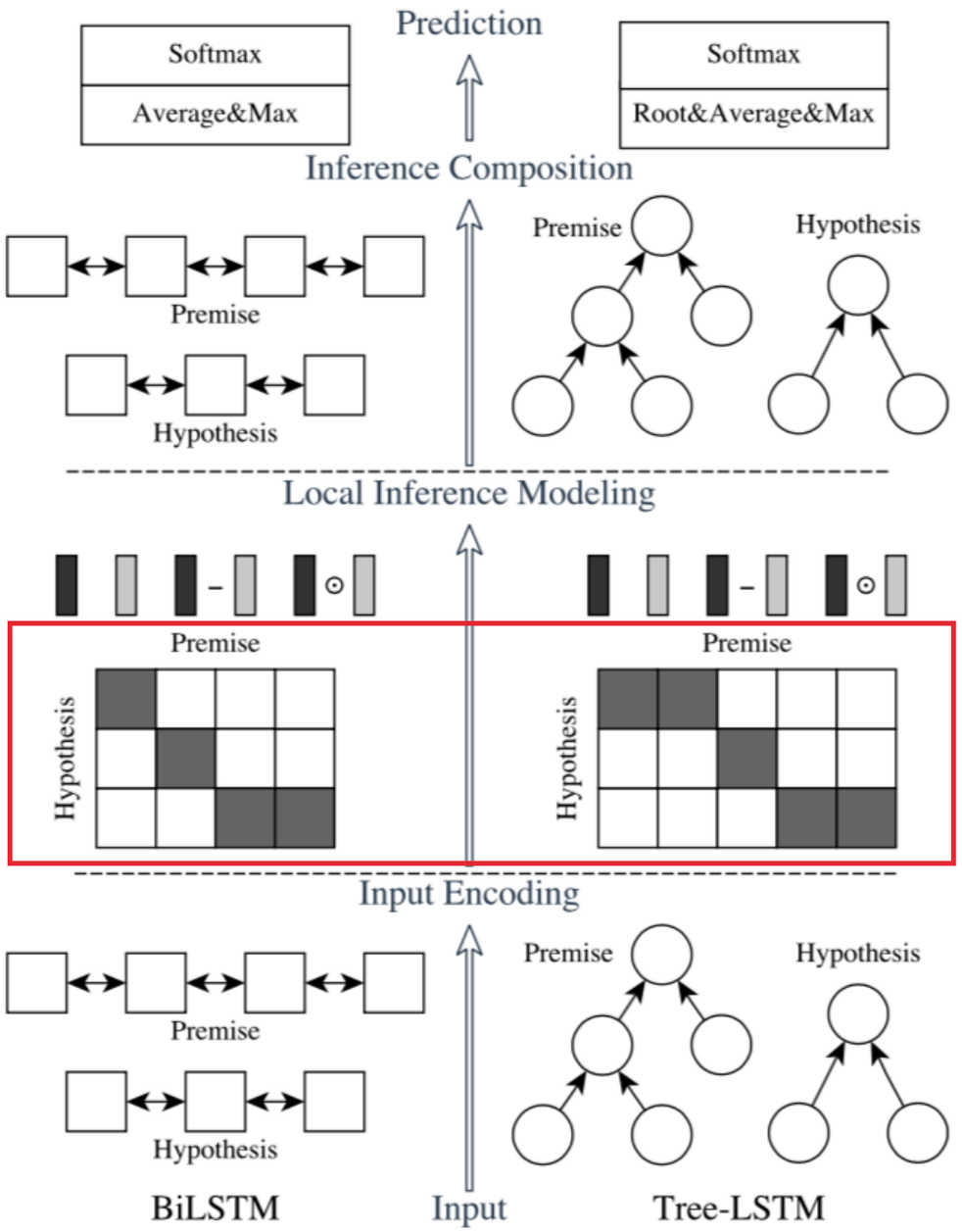


- Modeling local inference between premise and a hypothesis
  - In NN model it is usually achieved with *attention*

$$e_{ij} = \bar{\mathbf{a}}_i^T \bar{\mathbf{b}}_j.$$

- *Attention weight computed as similarity between hidden state tuple <a_i, b_i>*

LOCAL INFERENCE MODELLING (SEQUENTIAL)

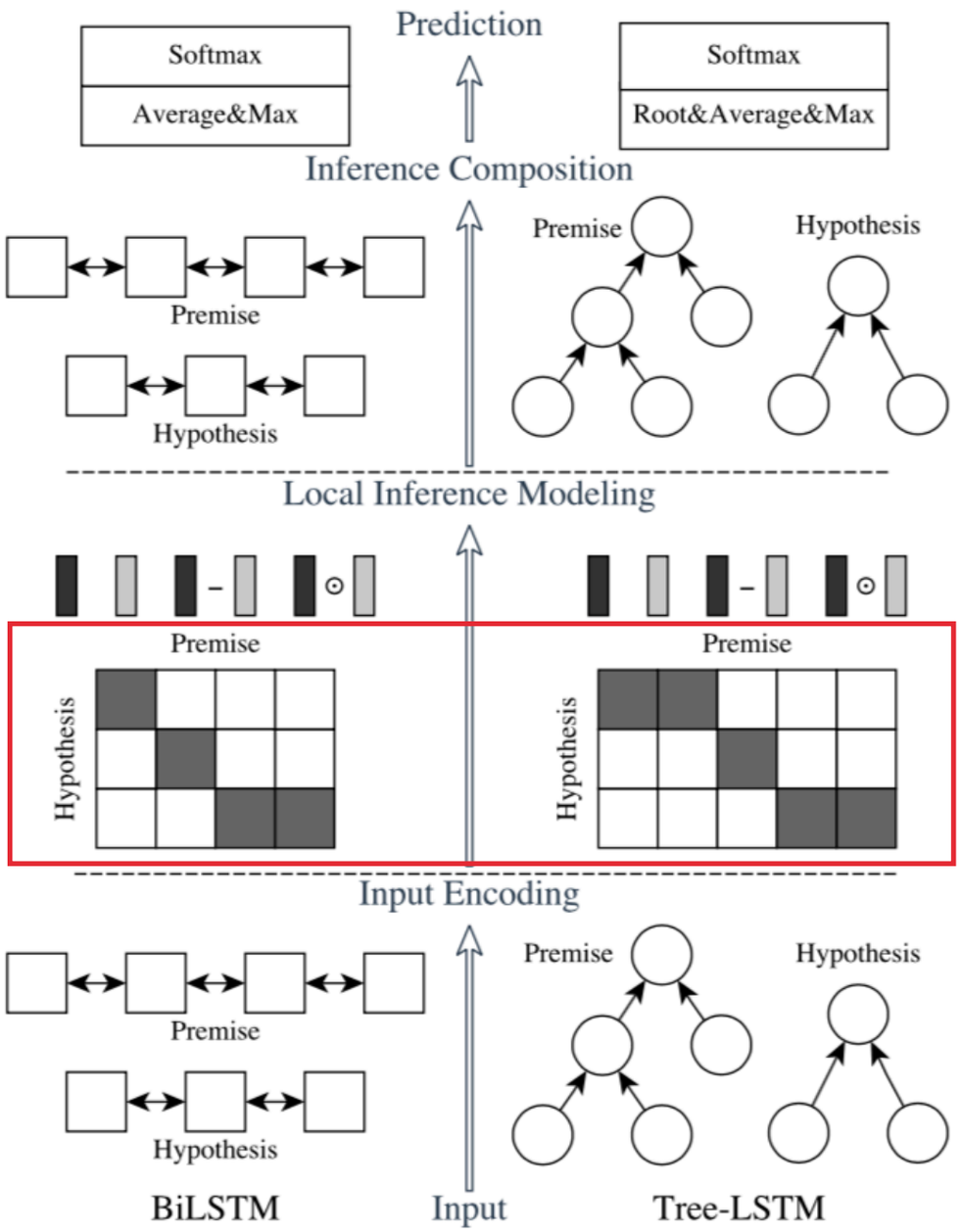


- Local inference determined by attention e_{ij} and is used to obtain local relevance.

$$\tilde{\mathbf{a}}_i = \sum_{j=1}^{\ell_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_b} \exp(e_{ik})} \bar{\mathbf{b}}_j, \forall i \in [1, \ldots, \ell_a],$$

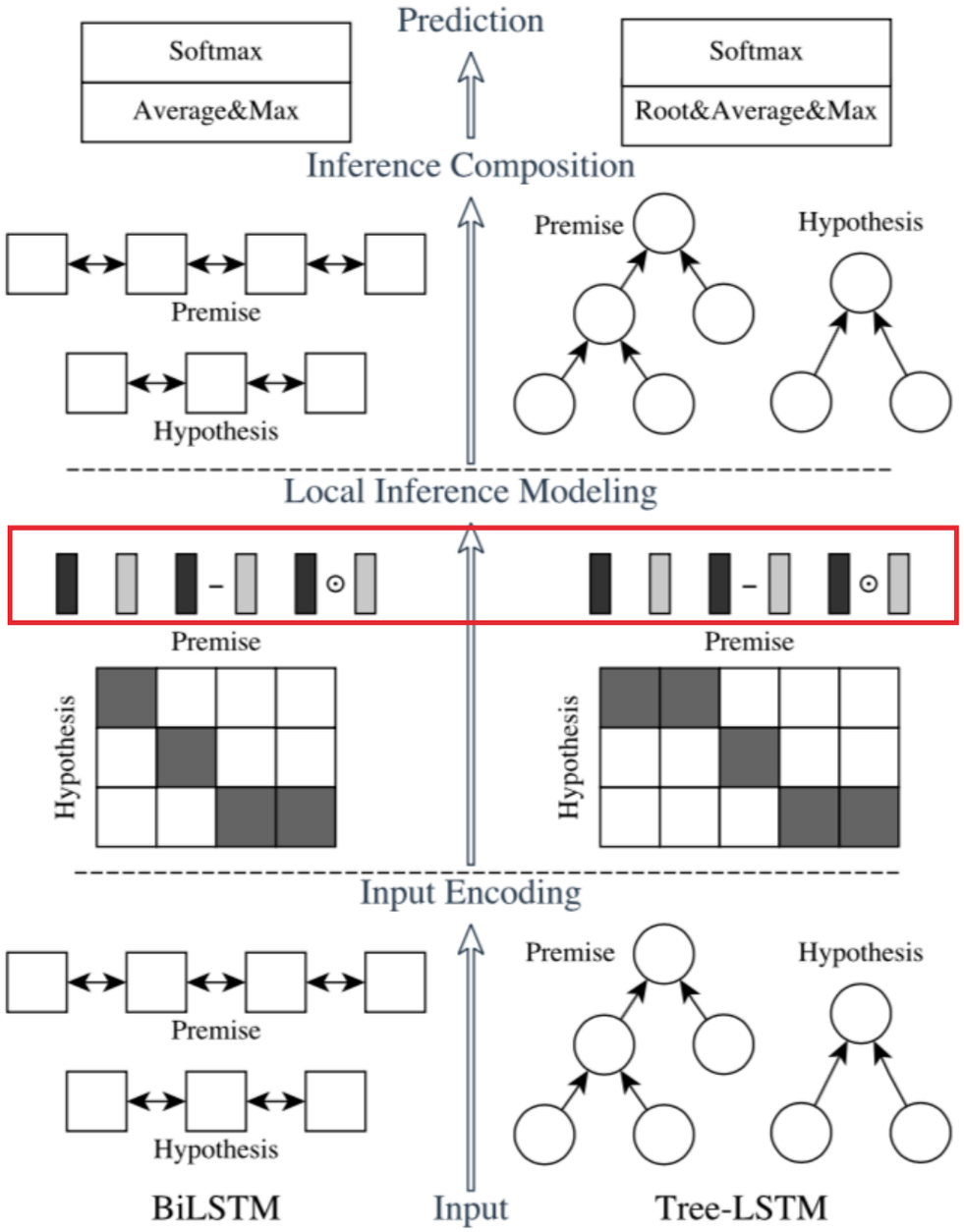
$$\tilde{\mathbf{b}}_j = \sum_{i=1}^{\ell_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_a} \exp(e_{kj})} \bar{\mathbf{a}}_i, \forall j \in [1, \ldots, \ell_b],$$

LOCAL INFERENCE MODELLING (TREE)



- Local inference determined by attention e_{ij} and is used to obtain local relevance. (same as sequential)
- (Treat all states same)

ENHANCEMENT OF LOCAL INFERENCE (SHARPENING)

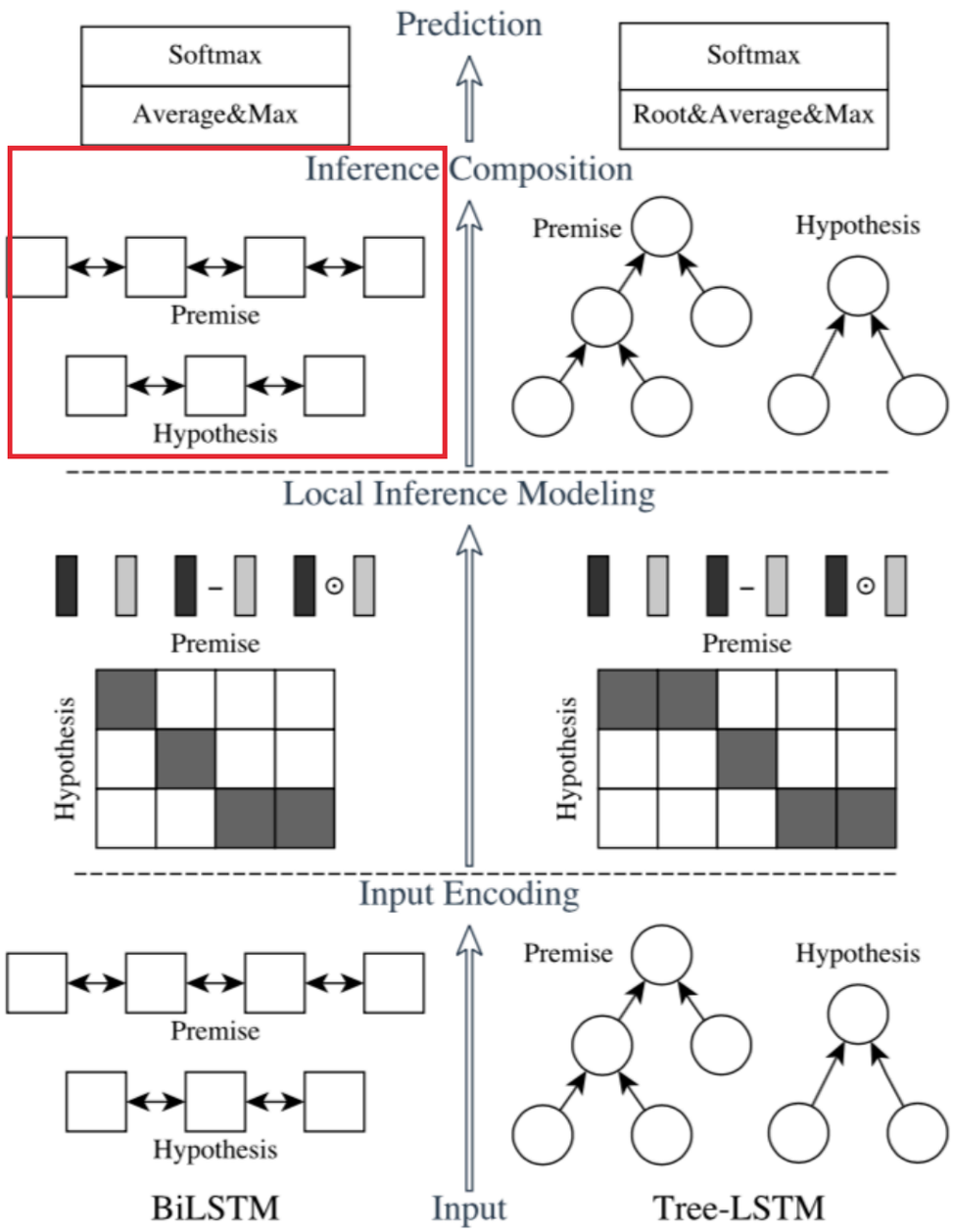


- Concatinate difference and point wise multiplication.

$$\mathbf{m}_a = [\bar{\mathbf{a}}; \tilde{\mathbf{a}}; \bar{\mathbf{a}} - \tilde{\mathbf{a}}; \bar{\mathbf{a}} \odot \tilde{\mathbf{a}}],$$

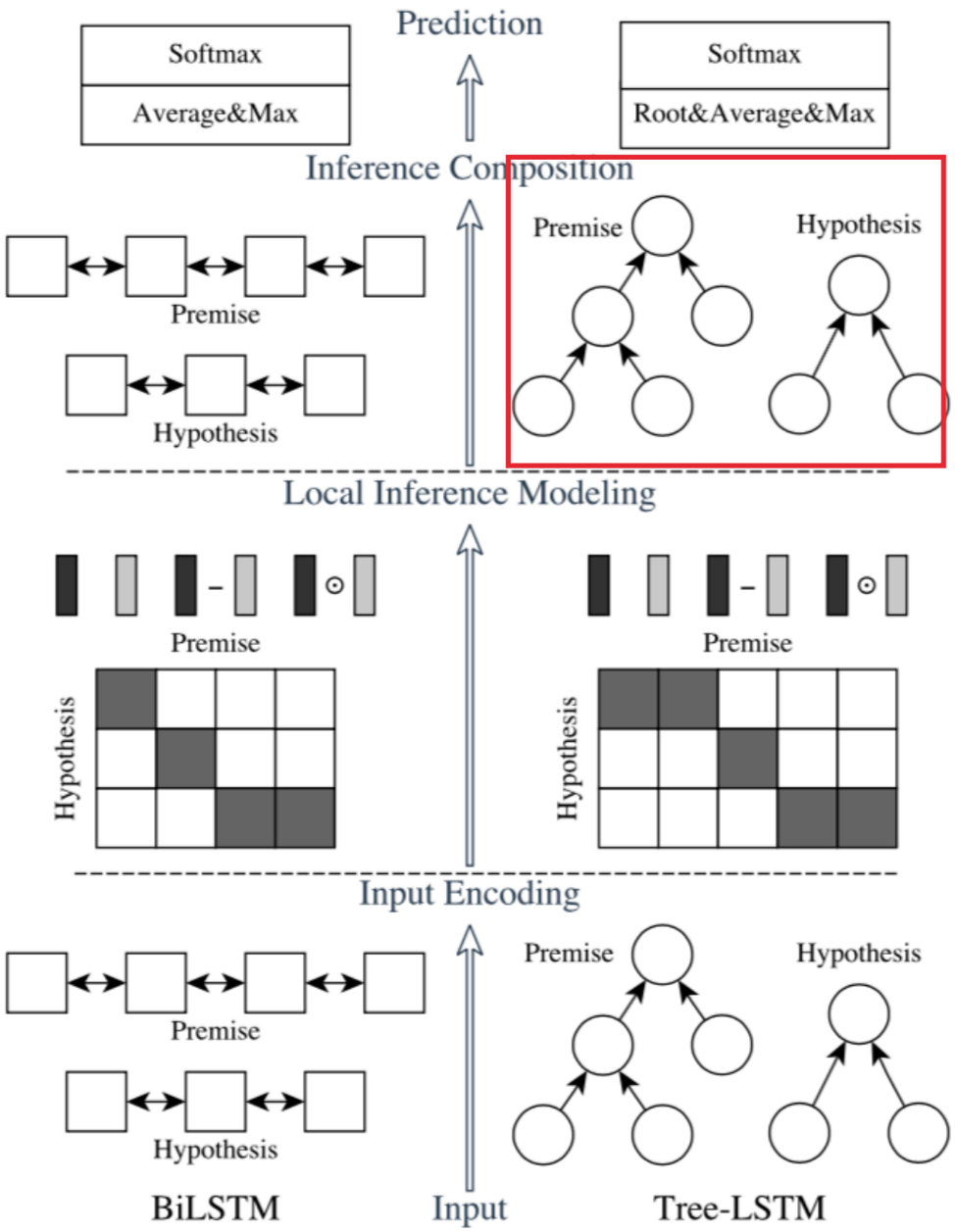
$$\mathbf{m}_b = [\bar{\mathbf{b}}; \tilde{\mathbf{b}}; \bar{\mathbf{b}} - \tilde{\mathbf{b}}; \bar{\mathbf{b}} \odot \tilde{\mathbf{b}}].$$

INFERENCE COMPOSITION



- Compose enhanced local inference:
  - BiLSTM used for m_a and m_b

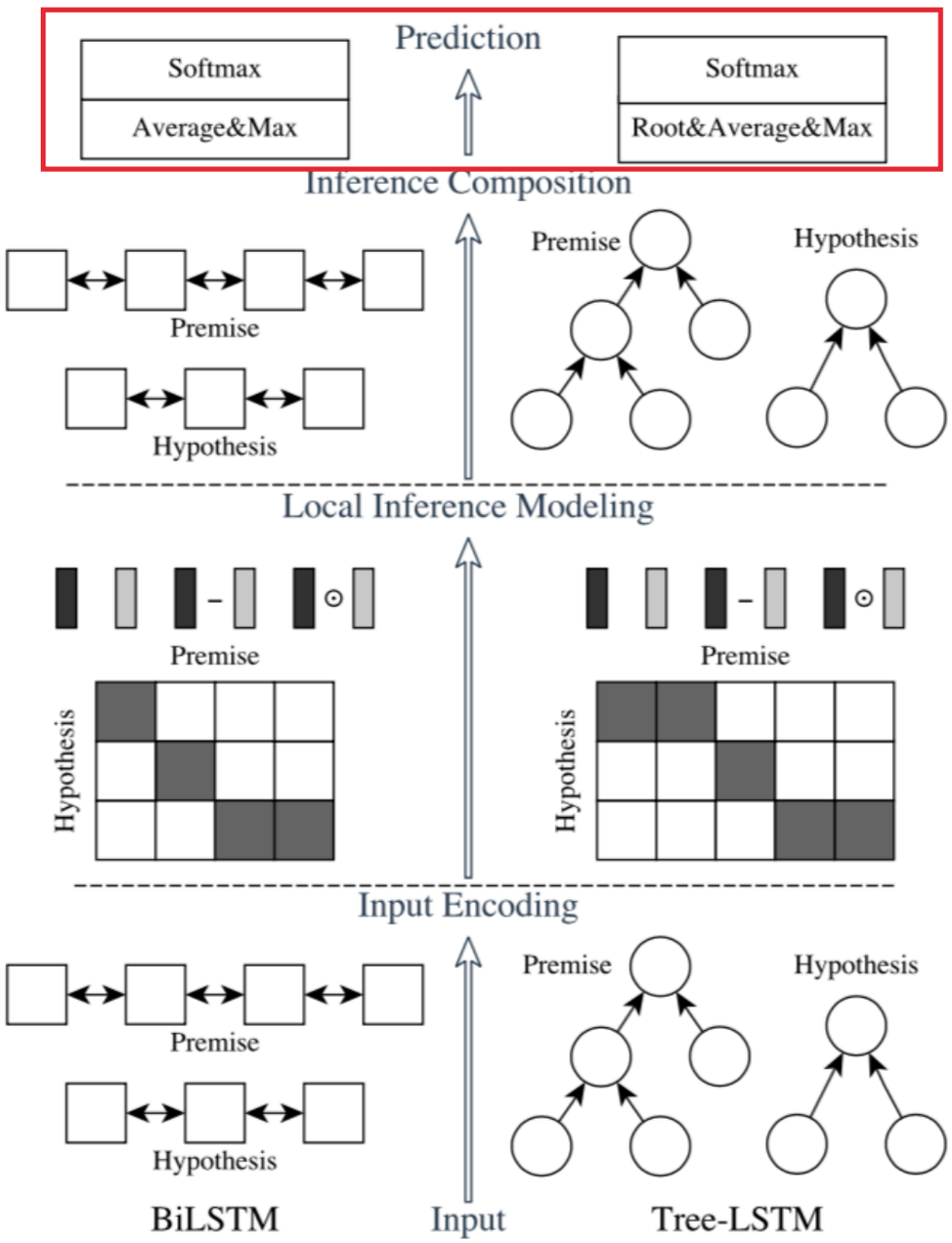
INFERENCE COMPOSITION



- Compose enhanced local inference:

$$\mathbf{v}_{a,t} = \mathrm{TrLSTM}(F(\mathbf{m}_{a,t}), \mathbf{h}^L_{t-1}, \mathbf{h}^R_{t-1}),$$
$$\mathbf{v}_{b,t} = \mathrm{TrLSTM}(F(\mathbf{m}_{b,t}), \mathbf{h}^L_{t-1}, \mathbf{h}^R_{t-1}).$$

- F: 1-layer FF network with ReLU



- Convert to fixed lengt vector
  - Compute both Average and MaxPooling
- Feed to final classifier

RESULTS

Model	#Para.	Train	Test
(1) Handcrafted features (Bowman et al., 2015)	-	99.7	78.2
(2) 300D LSTM encoders (Bowman et al., 2016)	3.0M	83.9	80.6
(3) 1024D pretrained GRU encoders (Vendrov et al., 2015)	15M	98.8	81.4
(4) 300D tree-based CNN encoders (Mou et al., 2016)	3.5M	83.3	82.1
(5) 300D SPINN-PI encoders (Bowman et al., 2016)	3.7M	89.2	83.2
(6) 600D BiLSTM intra-attention encoders (Liu et al., 2016)	2.8M	84.5	84.2
(7) 300D NSE encoders (Munkhdalai and Yu, 2016a)	3.0M	86.2	84.6
(8) 100D LSTM with attention (Rocktäschel et al., 2015)	250K	85.3	83.5
(9) 300D mLSTM (Wang and Jiang, 2016)	1.9M	92.0	86.1
(10) 450D LSTMN with deep attention fusion (Cheng et al., 2016)	3.4M	88.5	86.3
(11) 200D decomposable attention model (Parikh et al., 2016)	380K	89.5	86.3
(12) Intra-sentence attention + (11) (Parikh et al., 2016)	580K	90.5	86.8
(13) 300D NTI-SLSTM-LSTM (Munkhdalai and Yu, 2016b)	3.2M	88.5	87.3
(14) 300D re-read LSTM (Sha et al., 2016)	2.0M	90.7	87.5
(15) 300D btree-LSTM encoders (Paria et al., 2016)	2.0M	88.6	87.6
(16) 600D ESIM	4.3M	92.6	<u>88.0</u>
(17) HIM (600D ESIM + 300D Syntactic tree-LSTM)	7.7M	93.5	**88.6**