Lecture 15: Recurrent Neural Networks
CS480/680 Intro to Machine Learning

2023-3-7

Pascal Poupart
David R. Cheriton School of Computer Science

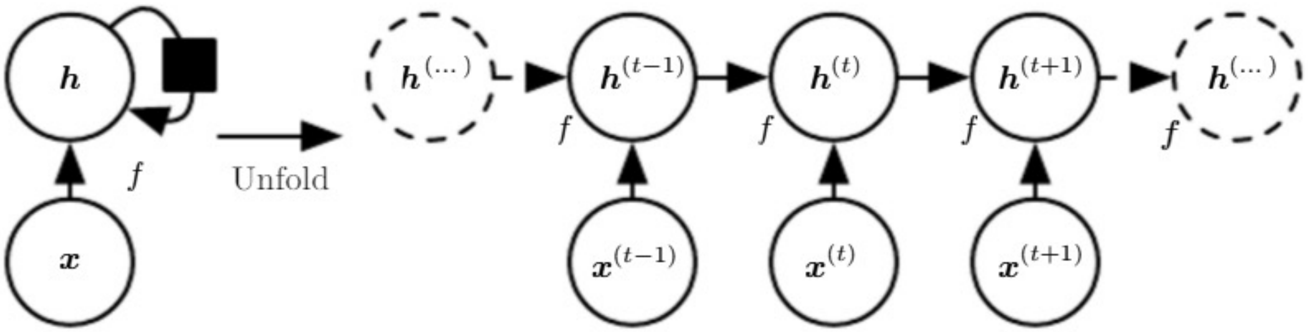
UNIVERSITY OF WATERLOO

Variable length data

- Traditional feed forward neural networks can only handle fixed length data
- Variable length data (e.g., sequences, time-series, spatial data) leads to a variable # of parameters
- Solutions:
    - Convolutional neural networks
    - **Recurrent neural networks**
    - Graph neural networks (including recursive neural networks)

Recurrent Neural Network (RNN)

- In RNNs, outputs can be fed back to the network as inputs, creating a recurrent structure that can be unrolled to handle varying length data.

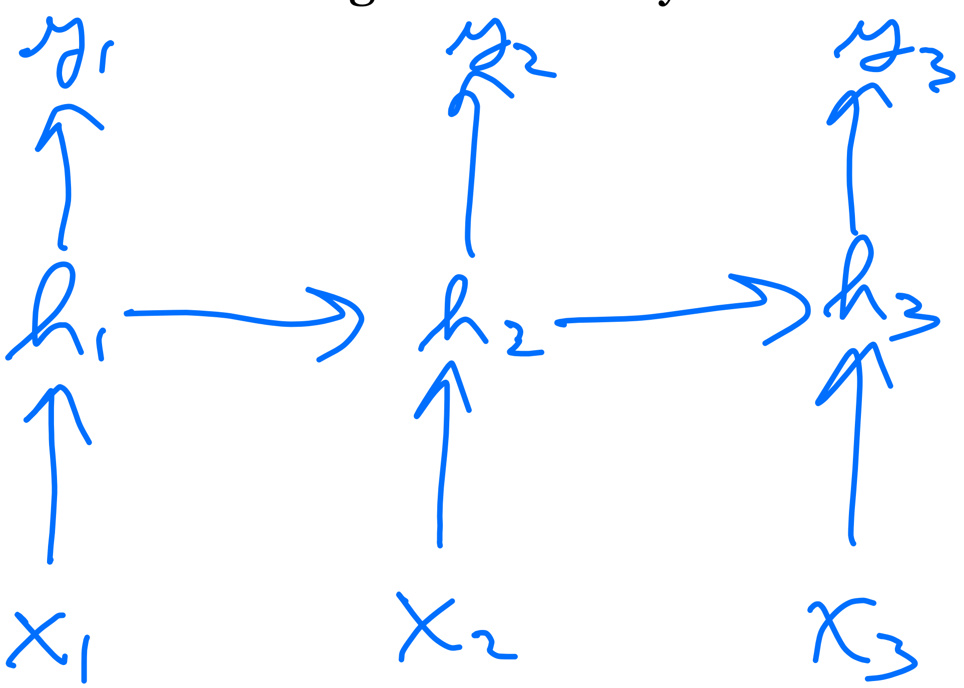


Training

- Recurrent neural networks are trained by backpropagation on the unrolled network
  - E.g. backpropagation through time
- Weight sharing:
  - Combine gradients of shared weights into a single gradient
- Challenges:
  - Gradient vanishing (and explosion)
  - Long range memory
  - Prediction drift

RNN for belief monitoring

- HMM can be simulated and generalized by a RNN

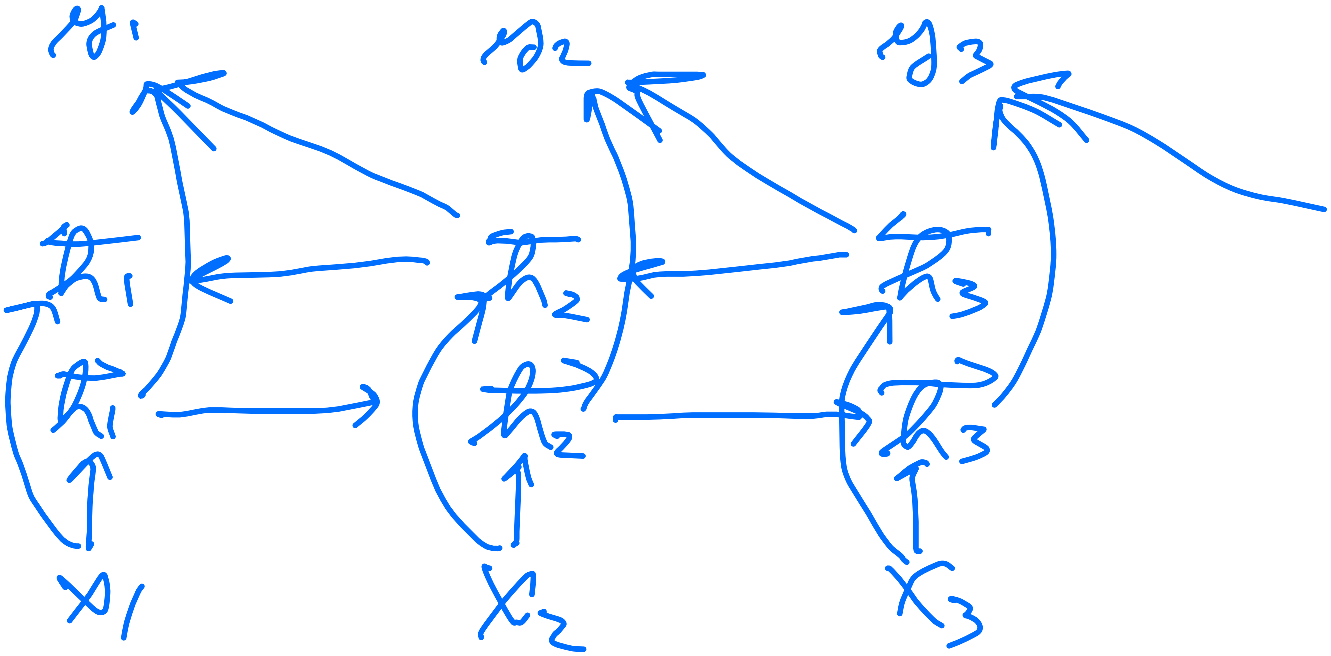


node: computational unit

edge: computational functional dependency

Bi-Directional RNN

- We can combine past and future evidence in separate chains

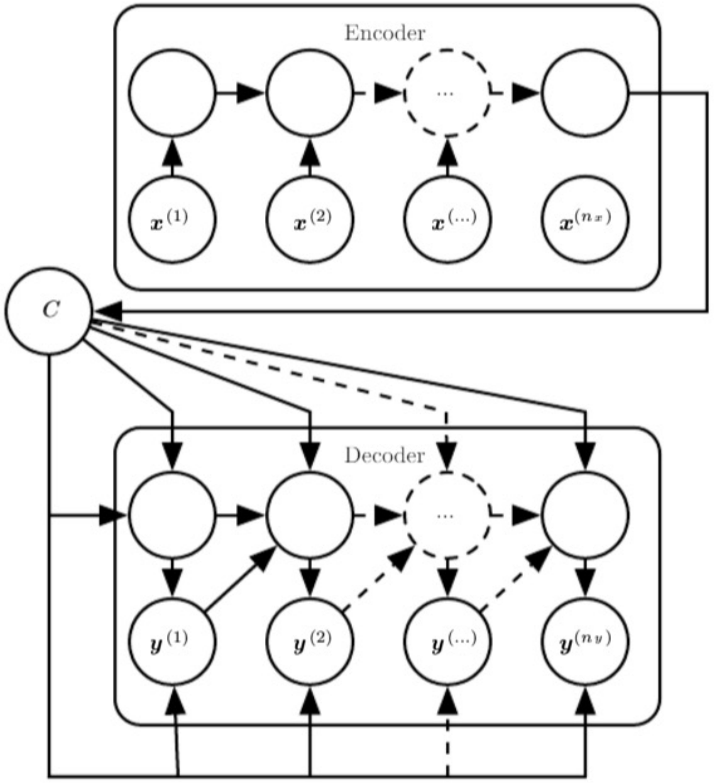


UNIVERSITY OF WATERLOO

Encoder-Decoder Model

- Also known as sequence2sequence
  - $x^{(i)}$: i^{th} input
  - $y^{(i)}$: i^{th} output
  - c: context (embedding)

- Usage:
  - Machine translation
  - Question answering
  - Dialog



Machine Translation

- Cho, van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, Bengio (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

Source	Translation Model	RNN Encoder–Decoder
at the end of the	[a la fin de la] [ƒ la fin des années] [être supprimés à la fin de la]	[à la fin du] [à la fin des] [à la fin de la]
for the first time	[r © pour la premirëre fois] [été donnés pour la première fois] [été commémorée pour la première fois]	[pour la première fois] [pour la première fois ,] [pour la première fois que]
in the United States and	[? aux ?tats-Unis et] [été ouvertes aux États-Unis et] [été constatées aux États-Unis et]	[aux Etats-Unis et] [des Etats-Unis et] [des États-Unis et]
, as well as	[?s , qu'] [?s , ainsi que] [?re aussi bien que]	[, ainsi qu'] [, ainsi que] [, ainsi que les]
one of the most	[?t ?l' un des plus] [?l' un des plus] [être retenue comme un de ses plus]	[l' un des] [le] [un des]

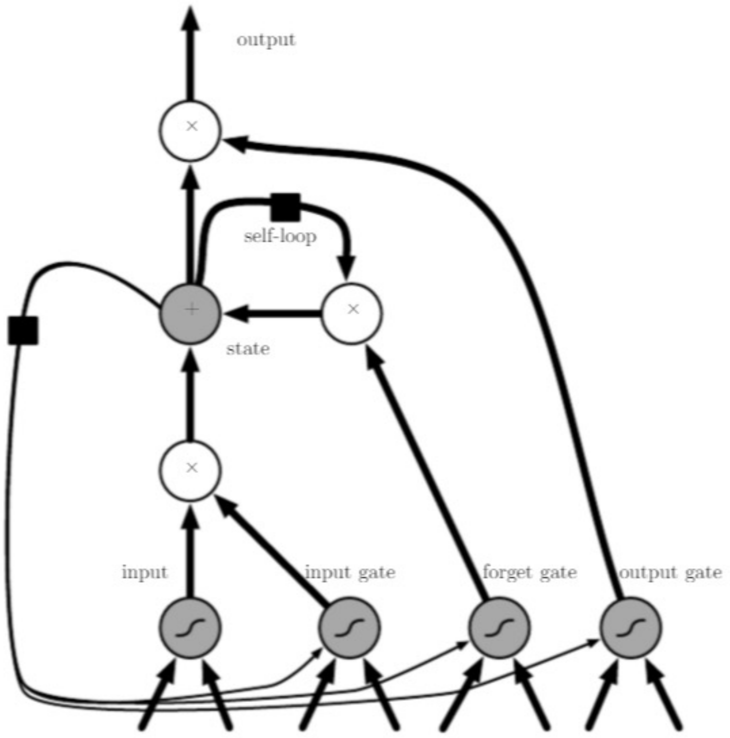
UNIVERSITY OF WATERLOO

Long Short-Term Memory (LSTM)

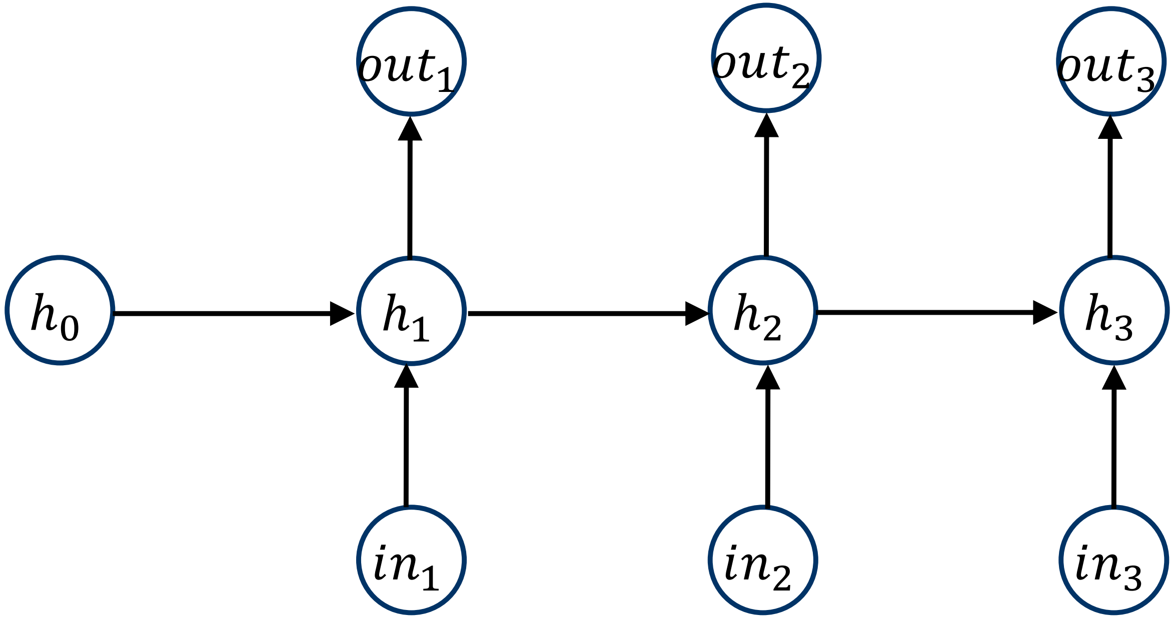
- Special gated structure to control memorization and forgetting in RNNs

- Mitigate gradient vanishing

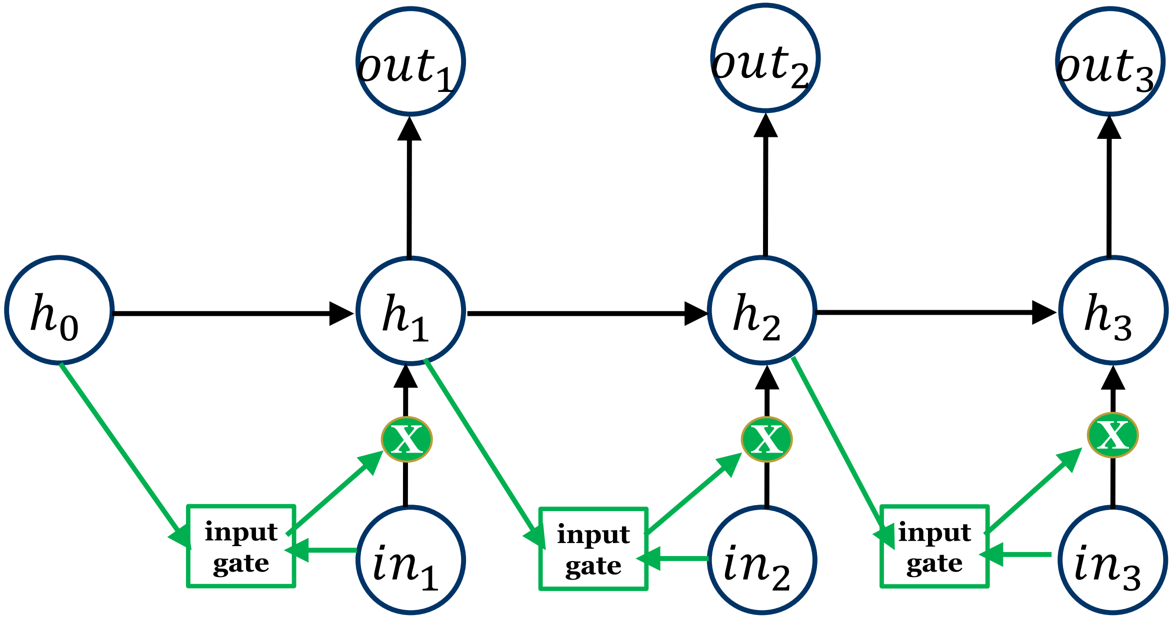
- Facilitate long term memory



Unrolled Long Short-Term Memory

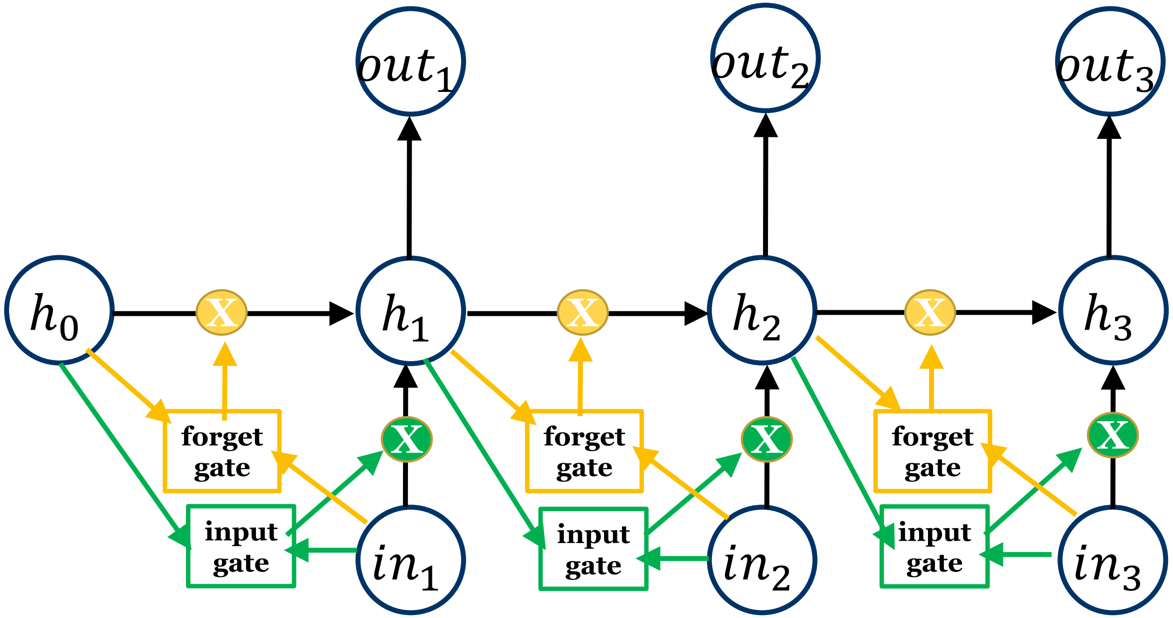


Unrolled Long Short-Term Memory

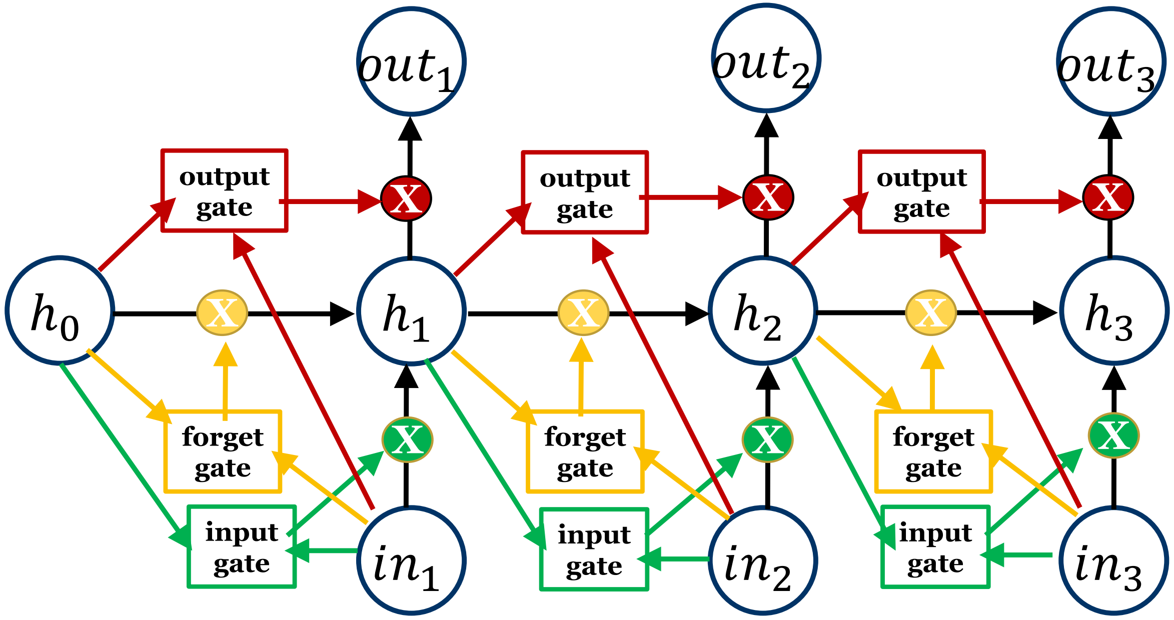


UNIVERSITY OF WATERLOO

Unrolled Long Short-Term Memory



Unrolled Long Short-Term Memory



LSTM cell in practice

- Adjustments:
  - Hidden state h_t called cell state c_t
  - Output y_t called hidden state h_t

- Update equations

Input gate: $i_t = \sigma(W^{(ii)}\bar{x}_t + W^{(hi)}h_{t-1})$

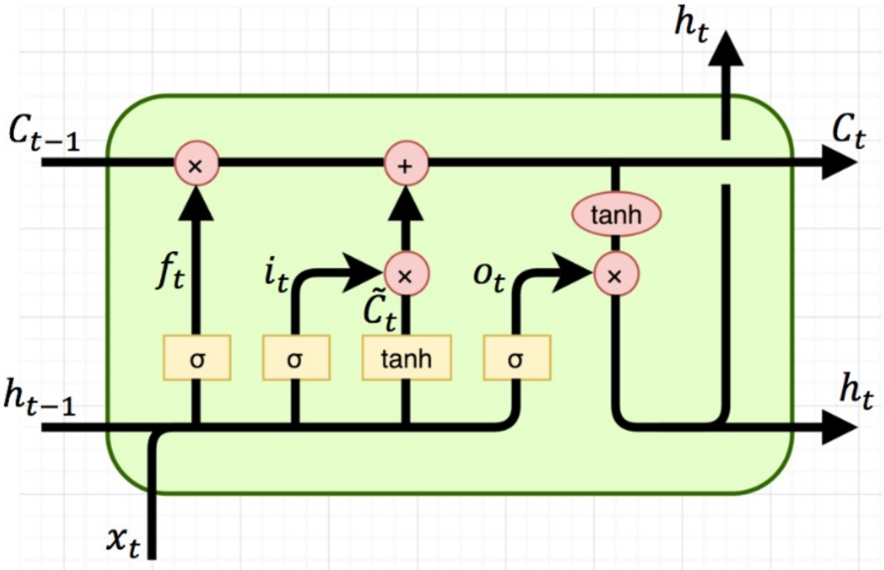
Forget gate: $f_t = \sigma(W^{(if)}\bar{x}_t + W^{(hf)}h_{t-1})$

Output gate: $o_t = \sigma(W^{(io)}\bar{x}_t + W^{(ho)}h_{t-1})$

Process input: $\tilde{c}_t = \tanh(W^{(i\tilde{c})}\bar{x}_t + W^{(h\tilde{c})}h_{t-1})$

Cell update: $c_t = f_t * c_{t-1} + i_t \ * \tilde{c}_t$

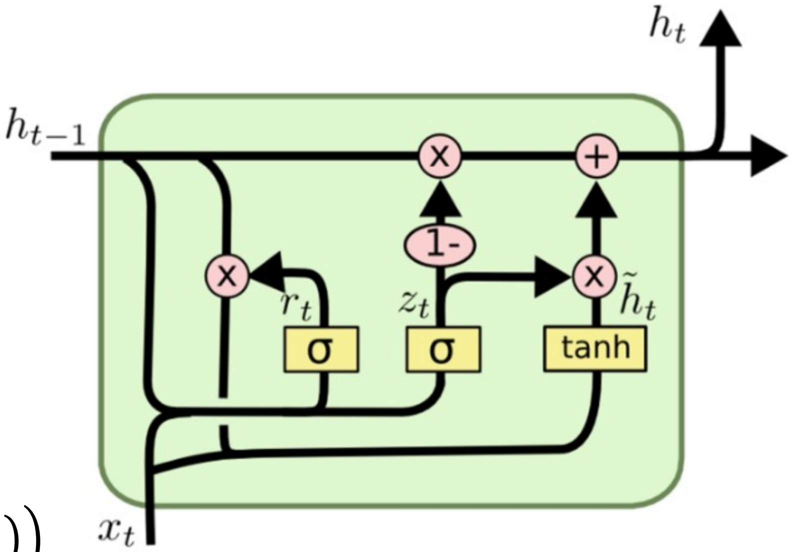
Output: $y_t = h_t = o_t * \tanh(c_t)$



UNIVERSITY OF WATERLOO

Gated Recurrent Unit (GRU)

- Simplified LSTM
  - No cell state
  - Two gates (instead of three)
  - Fewer weights



- Update equations

  Reset gate: $r_t = \sigma(W^{(ir)}\bar{x}_t + W^{(hr)}h_{t-1})$

  Update gate: $z_t = \sigma(W^{(iz)}\bar{x}_t + W^{(hz)}h_{t-1})$

  Process input: $\tilde{h}_t = \tanh\left(W^{(i\tilde{h})}\bar{x}_t + r_t * \left(W^{(h\tilde{h})}h_{t-1}\right)\right)$

  Hidden state update: $h_t = (1 - z_t) * h_{t-1} + z_t \ * \tilde{h}_t$

  Output: $y_t = h_t$

Attention

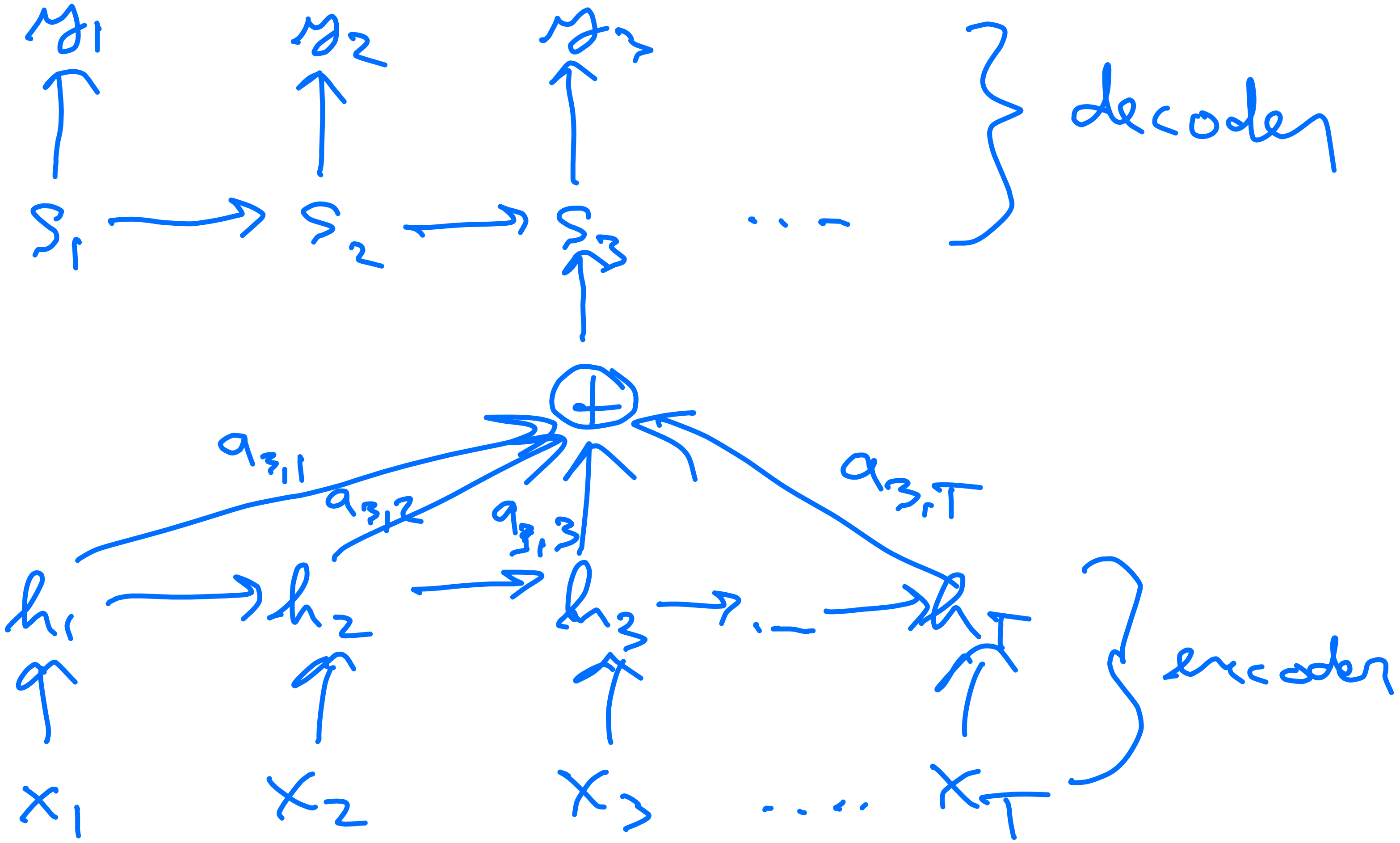
- Mechanism for alignment in machine translation, image captioning, etc.
- Attention in machine translation: align each output word with relevant input words by computing a softmax of the inputs
  - Context vector c_i: weighted sum of input encodings h_j

    $$c_i = \sum_j a_{ij} h_j$$
  - Where a_{ij} is an alignment weight between input encoding h_j and output encoding s_i

    $$a_{ij} = \frac{\exp\left(alignment(s_{i-1}, h_j)\right)}{\sum_{j'} \exp(alignment(s_{i-1}, h_{j'}))} \quad \text{(softmax)}$$
  - Alignment example: $alignment\left(s_{i-1}, h_j\right) = s_{i-1}^T h_j$

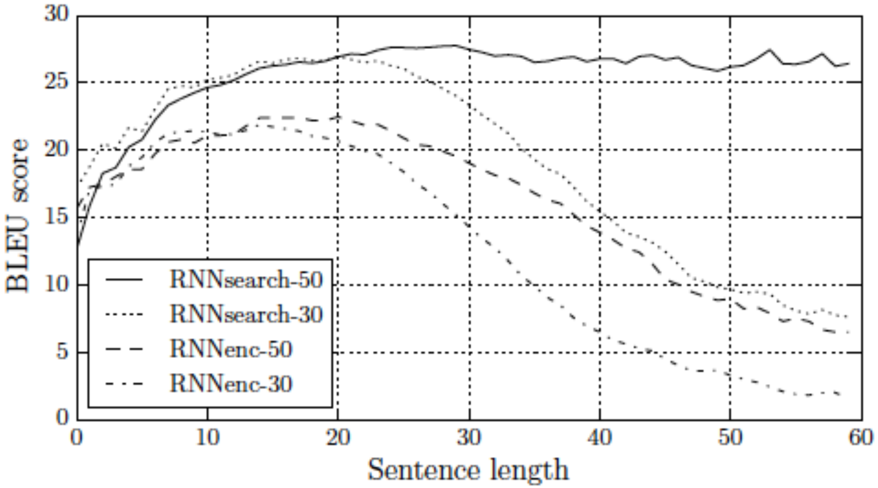
Attention

- Picture



Machine Translation with Bidirectional RNNs, LSTM units and attention

- Bahdanau, Cho, Bengio (ICLR-2015)

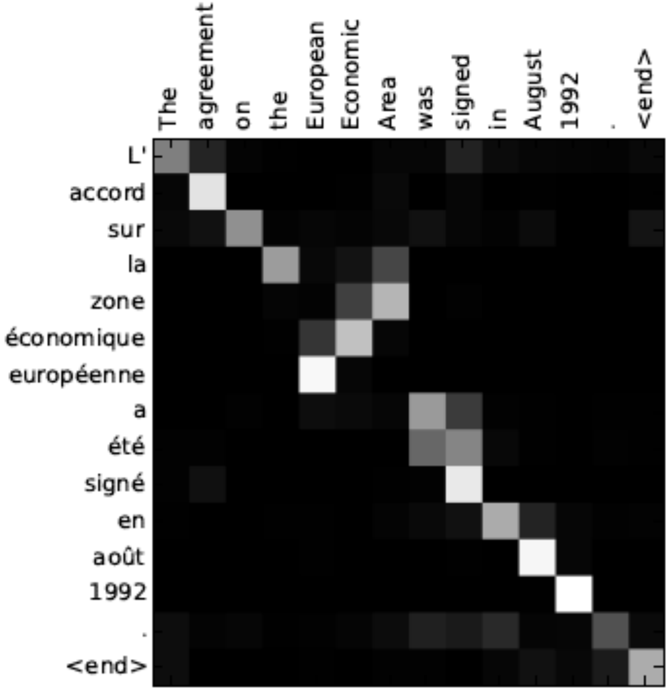


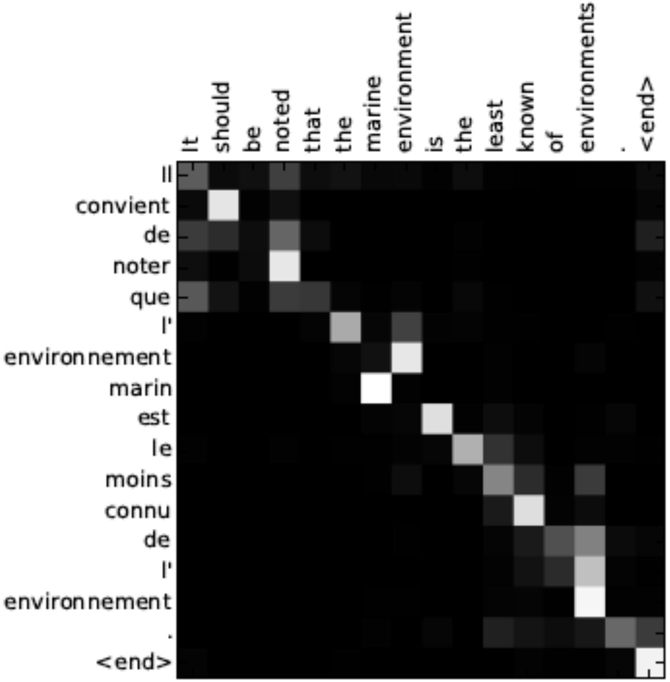
RNNsearch: with attention
RNNenc: no attention

- Bleu: BiLingual Evaluation Understudy
  - Percentage of translated words that appear in ground truth

Alignment example

- Bahdanau, Cho, Bengio (ICLR-2015)



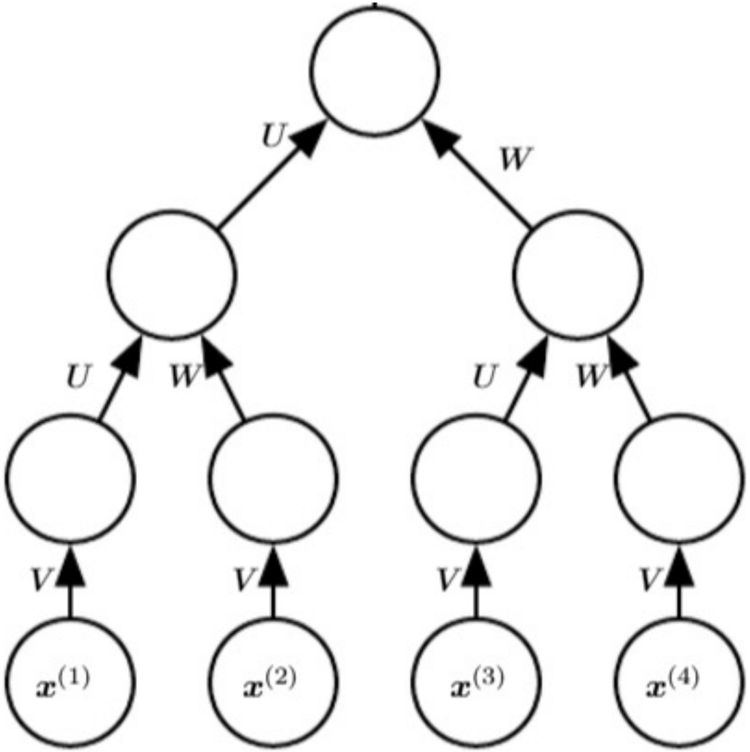


Recursive Neural Network

- Recursive neural networks:
  - generalize RNNs from chains to trees
  - Special case of graph neural nets

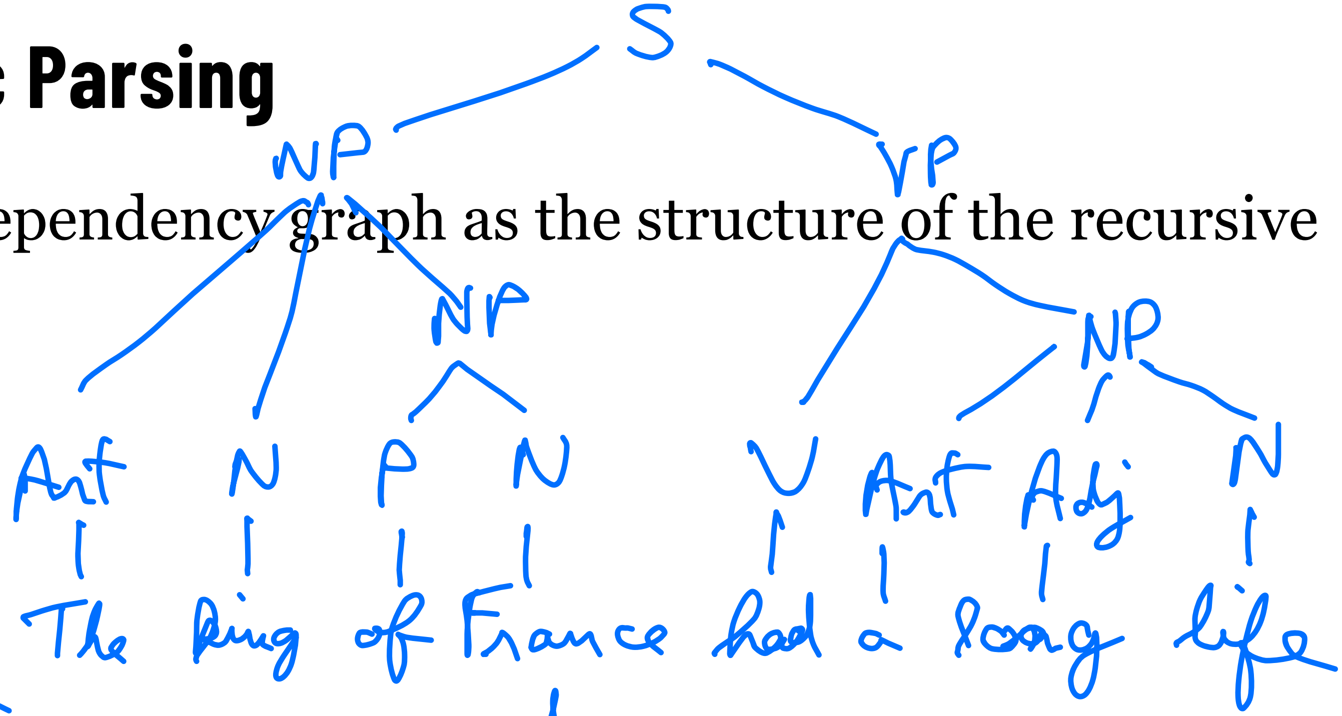
- Weight sharing allows trees of different sizes to fit variable length data.

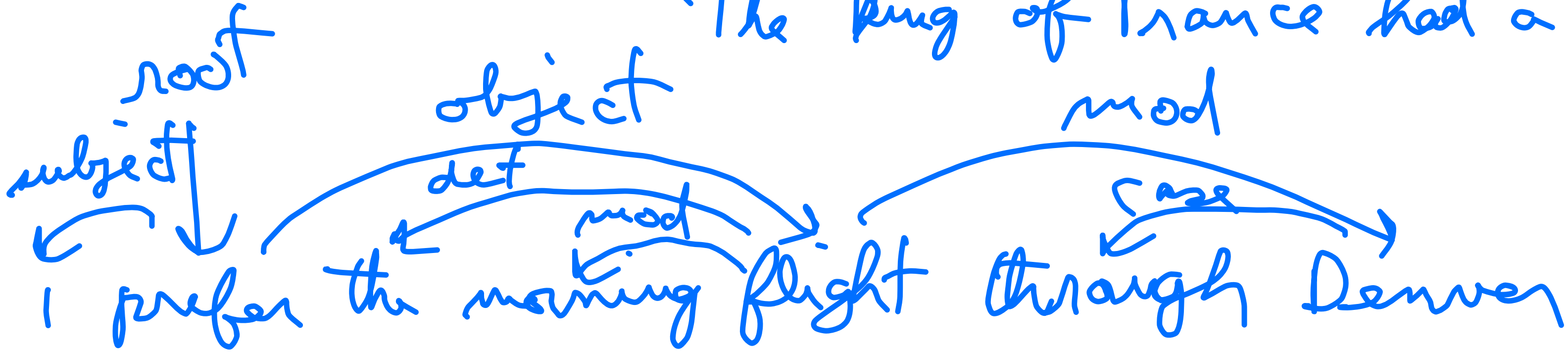
- What structure should the tree follow?



Example: Semantic Parsing

- Use a parse tree or dependency graph as the structure of the recursive neural network
- Example:





UNIVERSITY OF WATERLOO

Application: Sentiment Analysis

- Socher et al., (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**81.8**

Table 2: Accuracy of negation detection. Negated positive is measured as correct sentiment inversions. Negated negative is measured as increases in positive activations.