Sequence Modelling: Part (II)

Deep Learning Reading Group at MPII

Sequence Modelling: Recap

Sequence Modelling: Recap

- Sequence data

Sequence Modelling: Recap

- Sequence data
- Recurrent Neural Nets (RNN)

Sequence Modelling: Recap

- Sequence data
- Recurrent Neural Nets (RNN)
- Training and Design Patterns

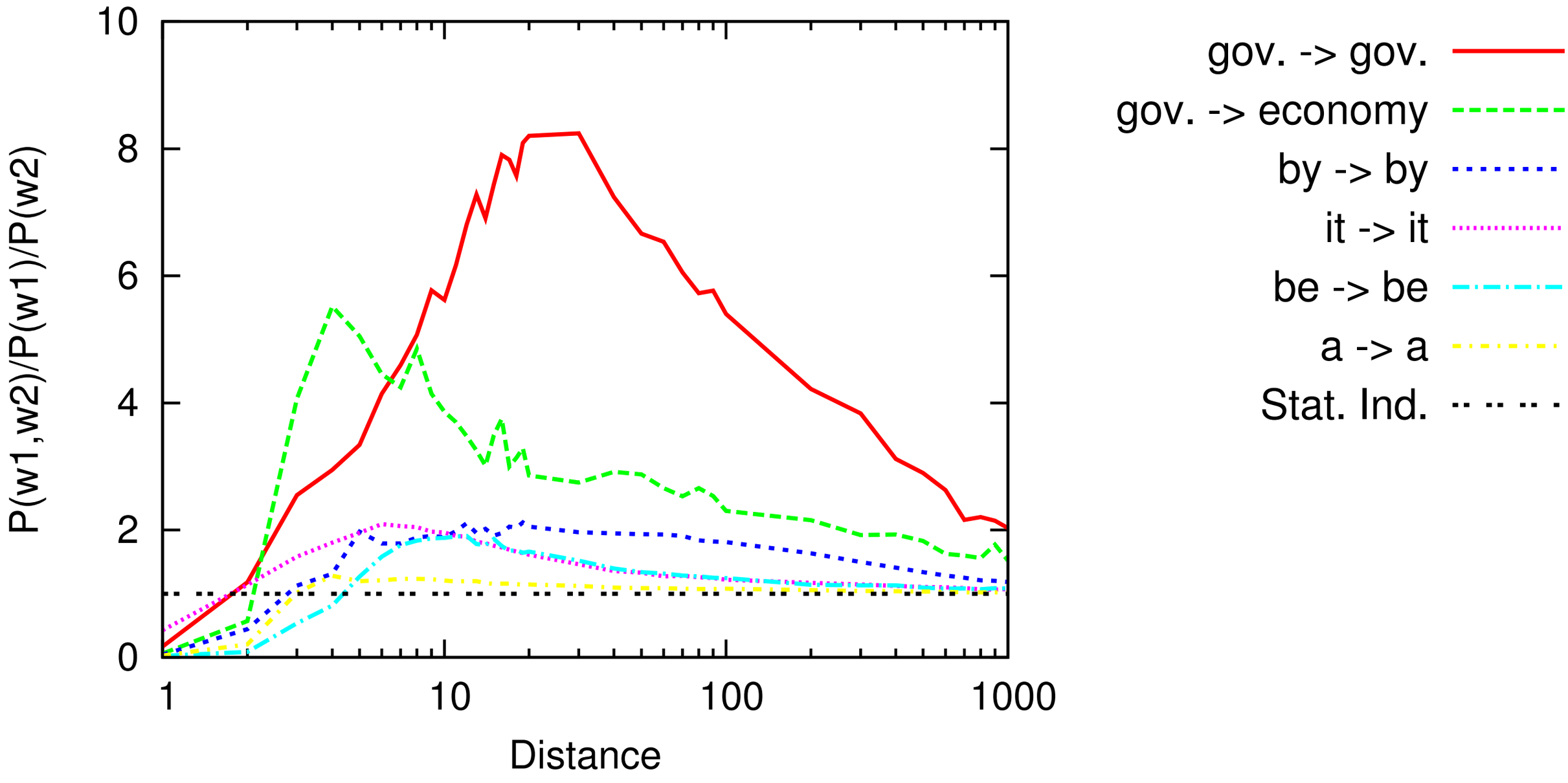
Sequence Modelling: Recap

- Sequence data
- Recurrent Neural Nets (RNN)
- Training and Design Patterns
- Bidirectional RNNs allowing to encode bidirectional sequence-related dependencies

Sequence Modelling: Recap

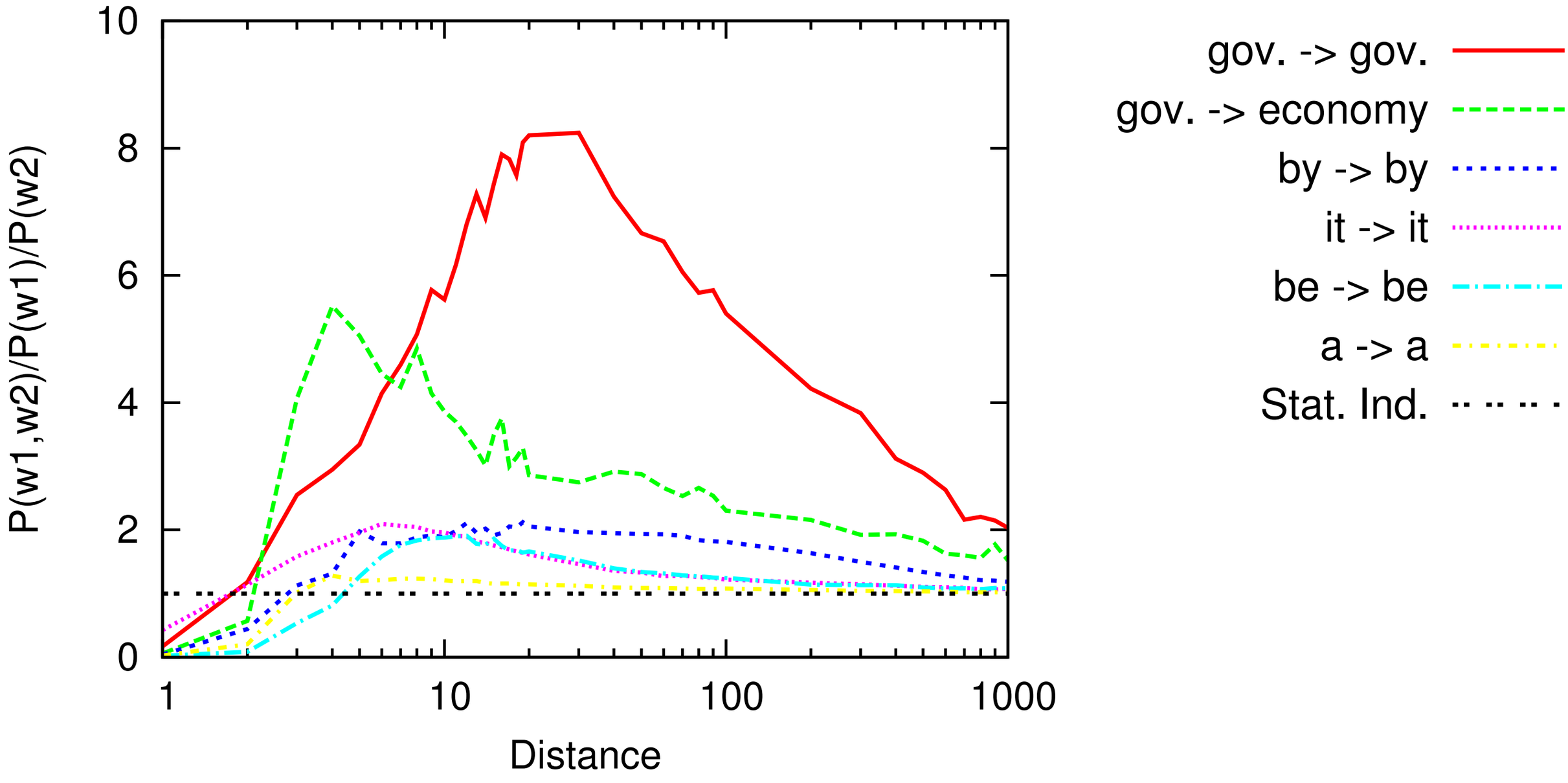
- Sequence data
- Recurrent Neural Nets (RNN)
- Training and Design Patterns
- Bidirectional RNNs allowing to encode bidirectional sequence-related dependencies
- Encoder-Decoder Architecture

Long-Term Dependencies



Long-Term Dependencies

There are long-term dependencies which can be beneficial to a predictive system

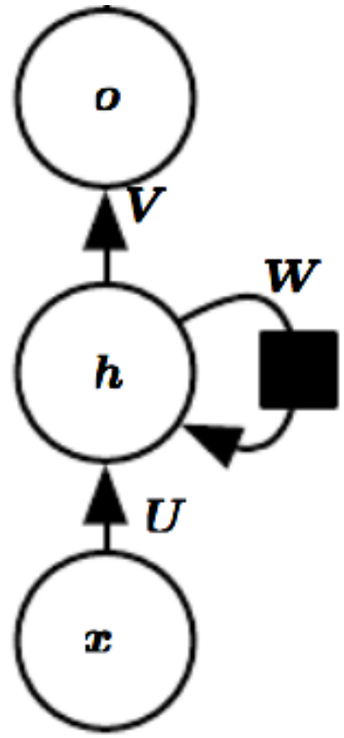


Sequence Modelling - II

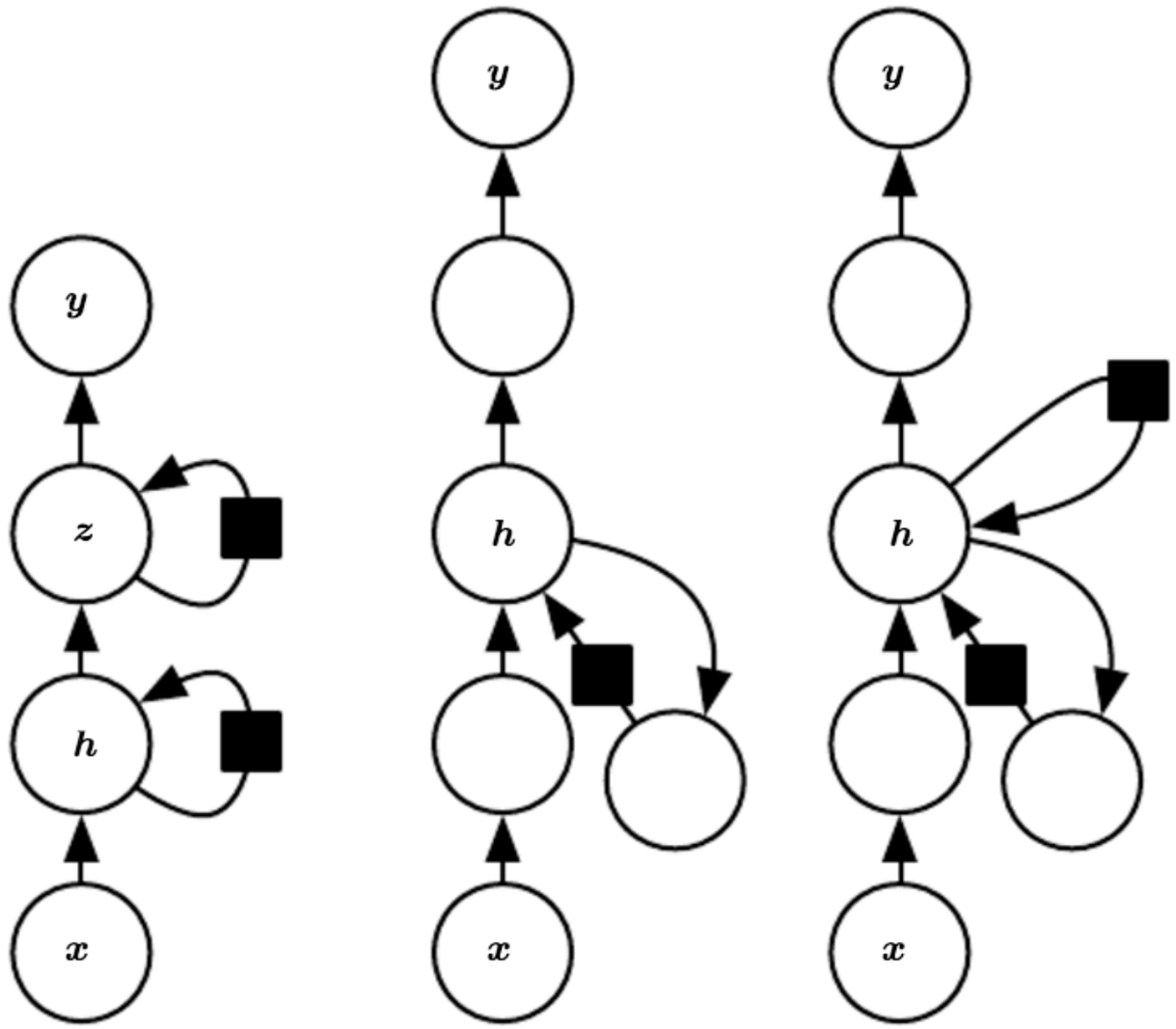
- Deep Recurrent Nets
- Recursive Neural Nets
    - SRNN
- Long-Term Dependencies
    - Challenges
    - LSTMs
    - LSRC

Deep Recurrent Nets

- Add layers from Input to Hidden State
- From previous hidden states to the next hidden state
- From hidden state to output



Deep Recurrent Nets

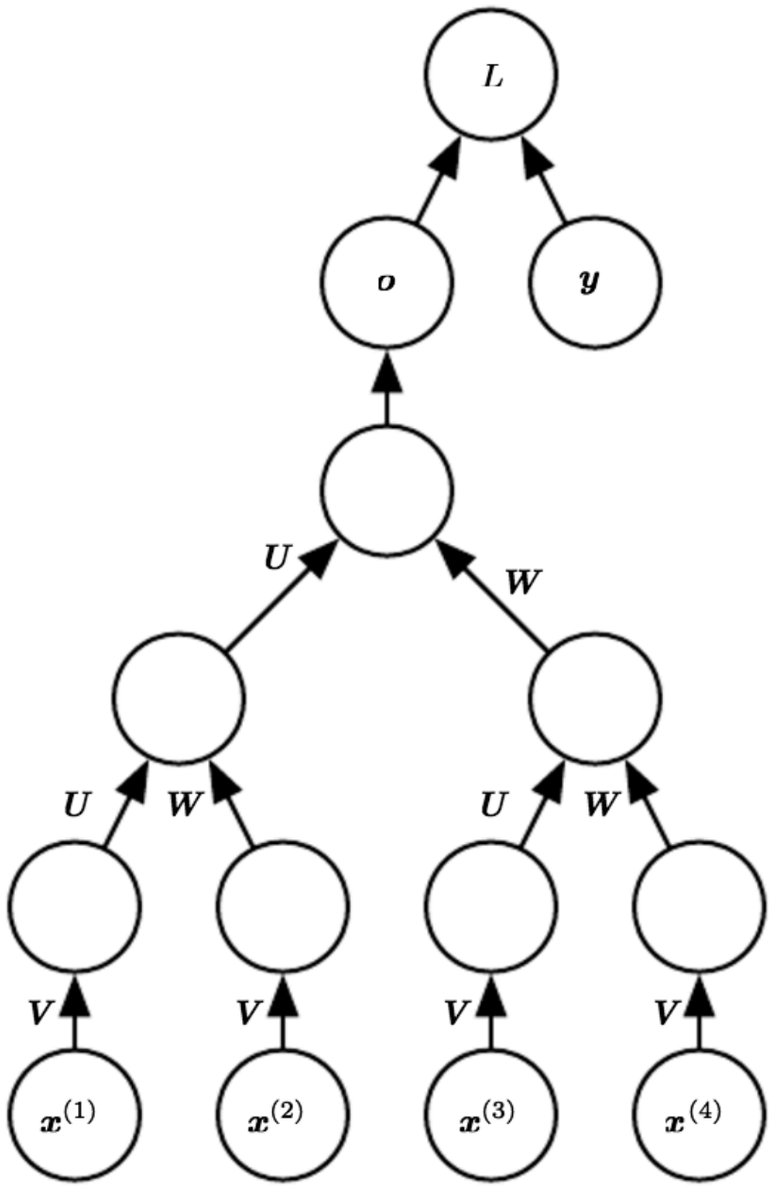


Deep Recurrent Nets

- The chapter mentions a few work which show evidence for deep RNNs performing well
  - An example from NLP, you can look at *Character-Aware Neural Network Language Model*
- Optimisation is more difficult
  - Skip connections

Recursive Neural Nets

- **Advantage** the depth of the network is reduced in comparison to Recurrent versions, which might help better deal with long-term dependencies



Recursive Nets: Factoid QA

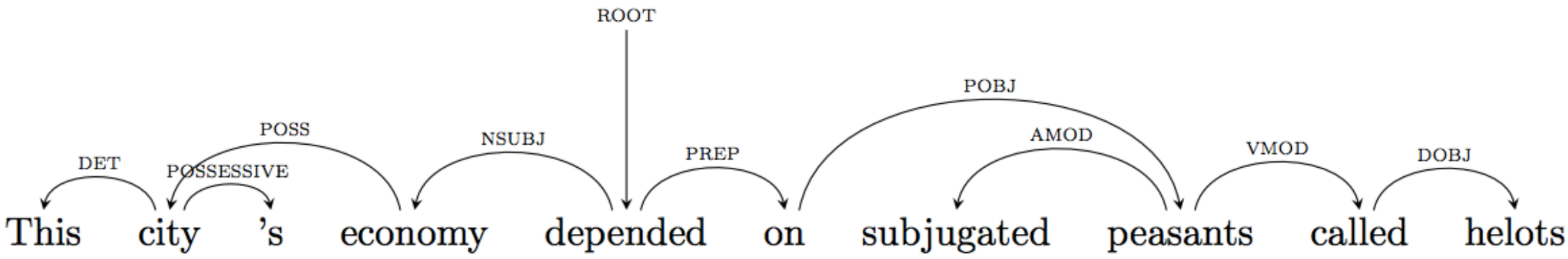
Later in its existence, this polity's leader was chosen by a group that included three bishops and six laymen, up from the seven who traditionally made the decision. Free imperial cities in this polity included Basel and Speyer. Dissolved in 1806, its key events included the Investiture Controversy and the Golden Bull of 1356. Led by Charles V, Frederick Barbarossa, and Otto I, for 10 points, name this polity, which ruled most of what is now Germany through the Middle Ages and rarely ruled its titular city.

Recursive Nets: Factoid QA

Later in its existence, this polity's leader was chosen by a group that included three bishops and six laymen, up from the seven who traditionally made the decision. Free imperial cities in this polity included Basel and Speyer. Dissolved in 1806, its key events included the Investiture Controversy and the Golden Bull of 1356. Led by Charles V, Frederick Barbarossa, and Otto I, for 10 points, name this polity, which ruled most of what is now Germany through the Middle Ages and rarely ruled its titular city.

Answer: Holy Roman Empire

Recursive Nets: Factoid QA

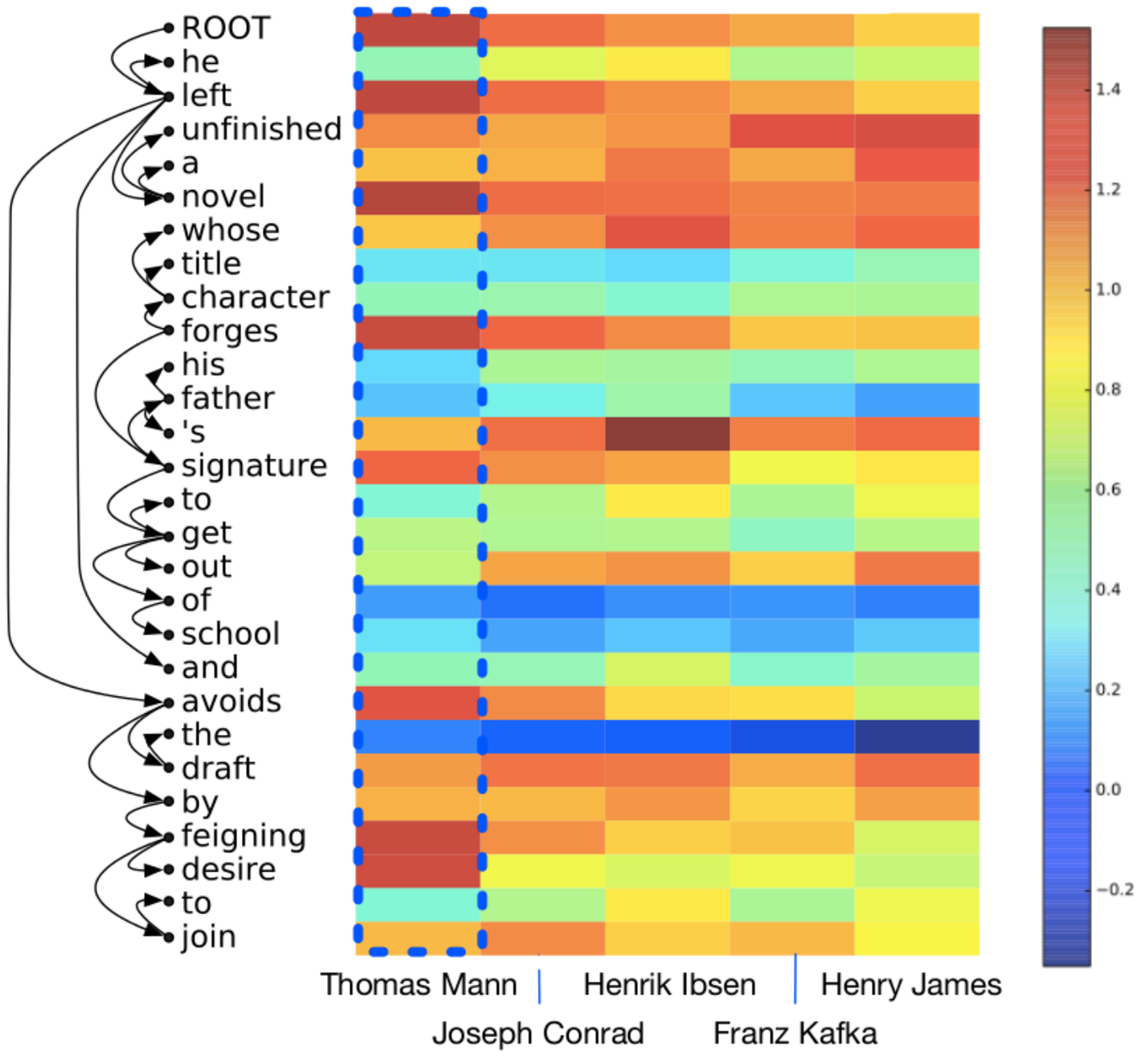


$$h_{\text{helots}} = f(W_v \cdot x_{\text{helots}} + b), \longrightarrow h_{\text{called}} = f(W_{\text{DOBJ}} \cdot h_{\text{helots}} + W_v \cdot x_{\text{called}} + b).$$

$$h_{\text{depended}} = f(W_{\text{NSUBJ}} \cdot h_{\text{economy}} + W_{\text{PREP}} \cdot h_{\text{on}} + W_v \cdot x_{\text{depended}} + b).$$

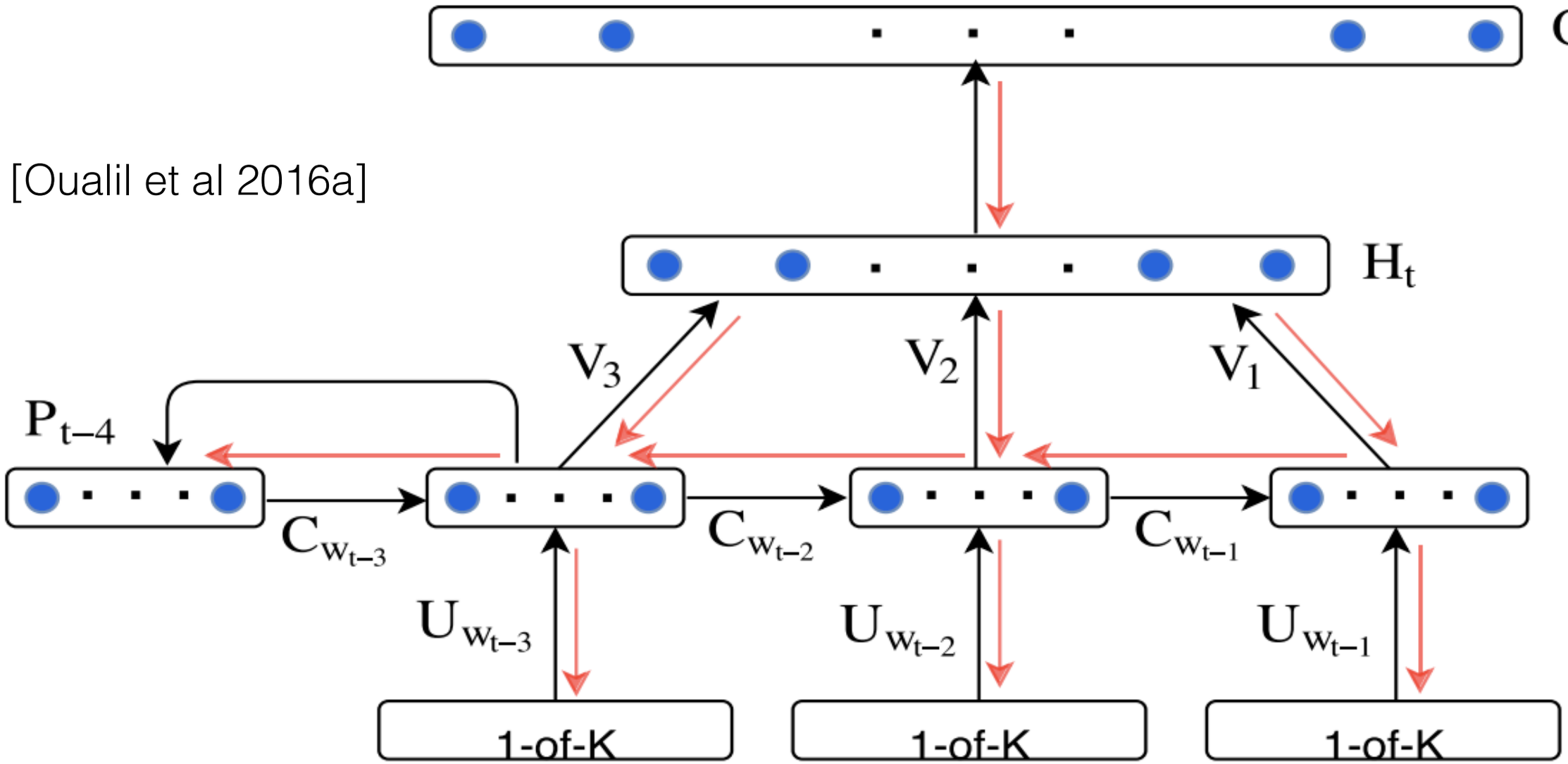
[Iyyer et al 2014]

Recursive Nets: In Action



SRNN: Lateral Connections

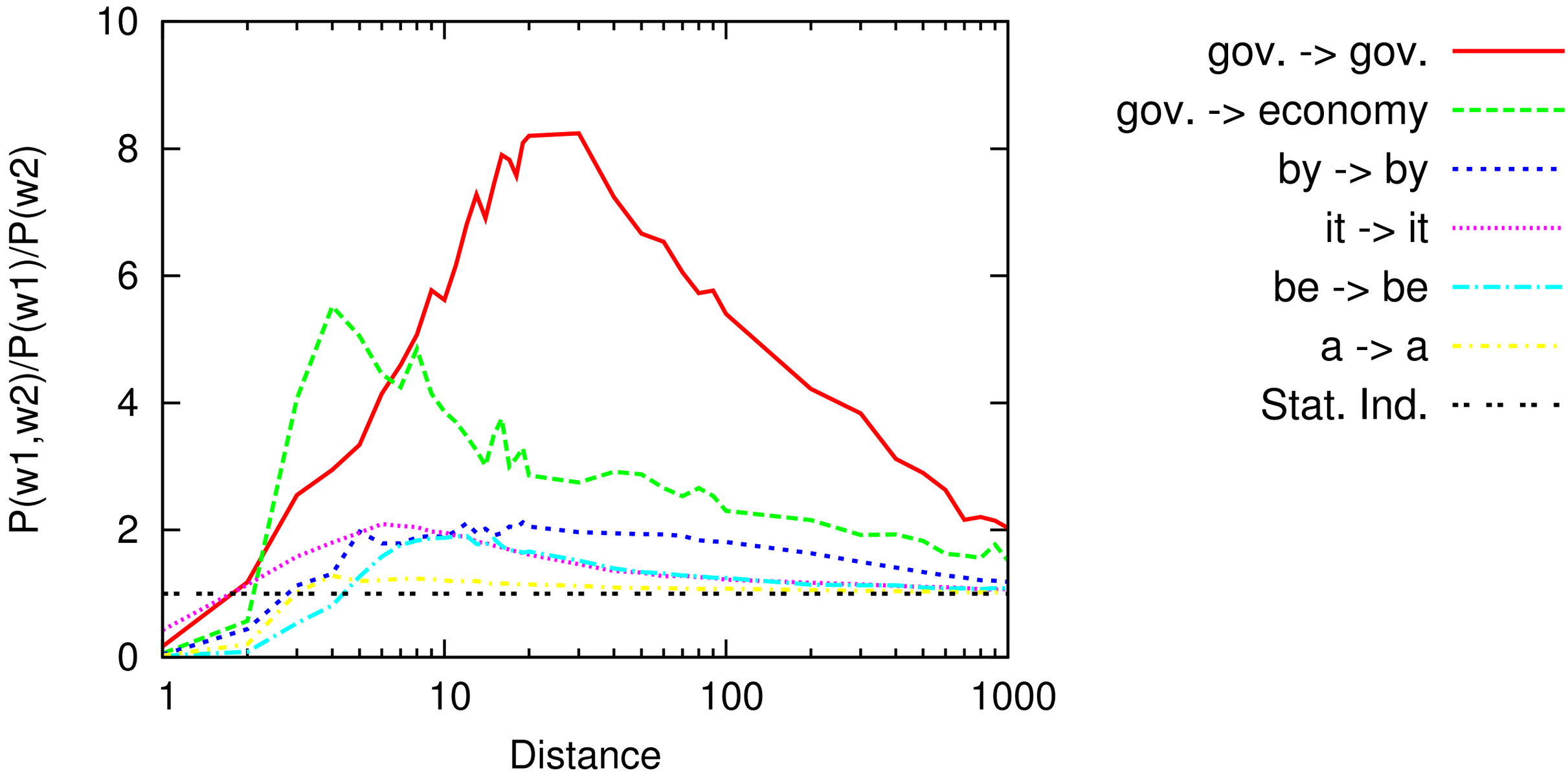
[Oualil et al 2016a]



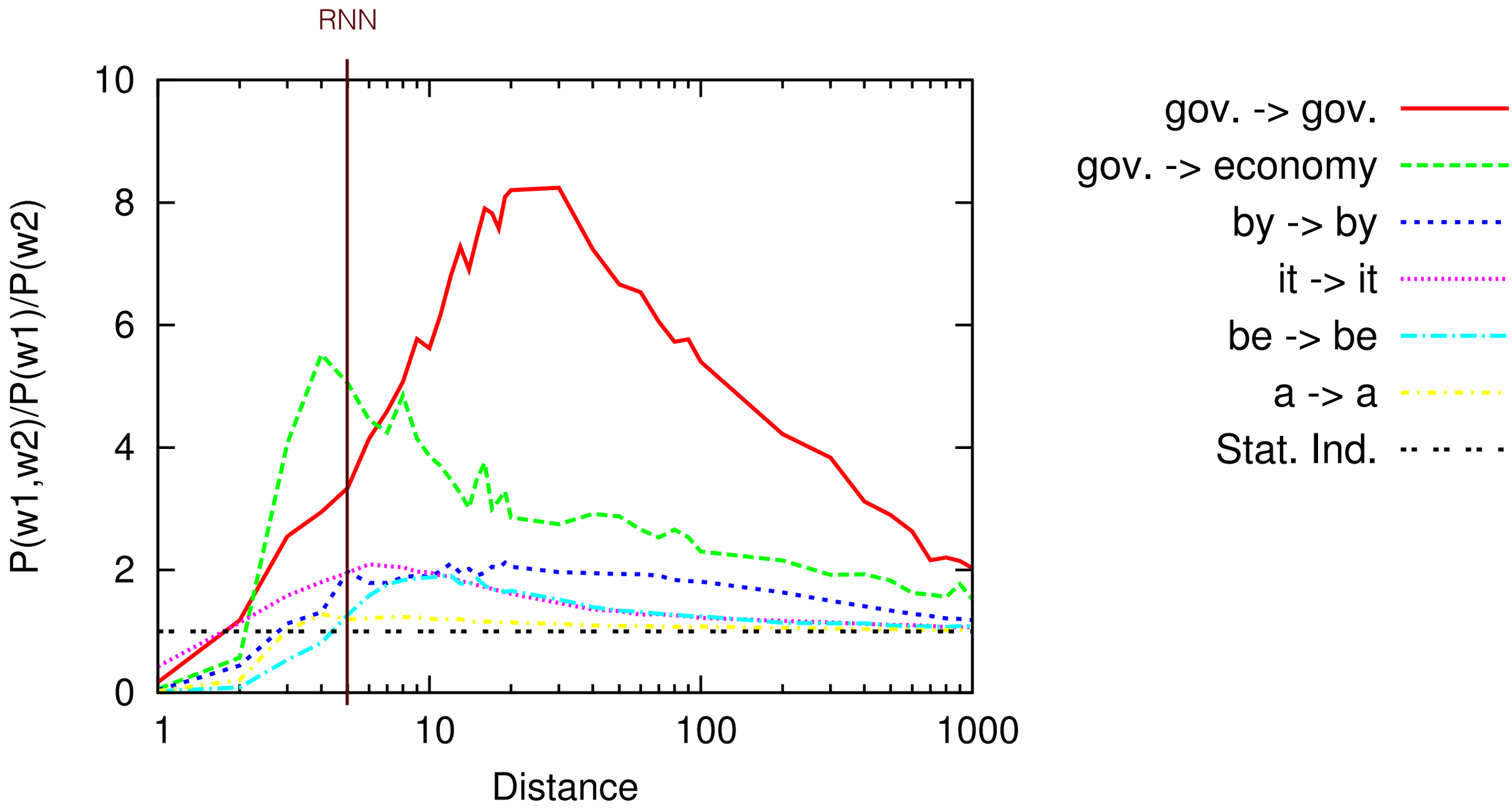
SRNN: Salient Points

- Performs better on sequence prediction tasks
- Requires ~50% fewer parameters than RNN
- RNN is equivalent to SRNN with context size one

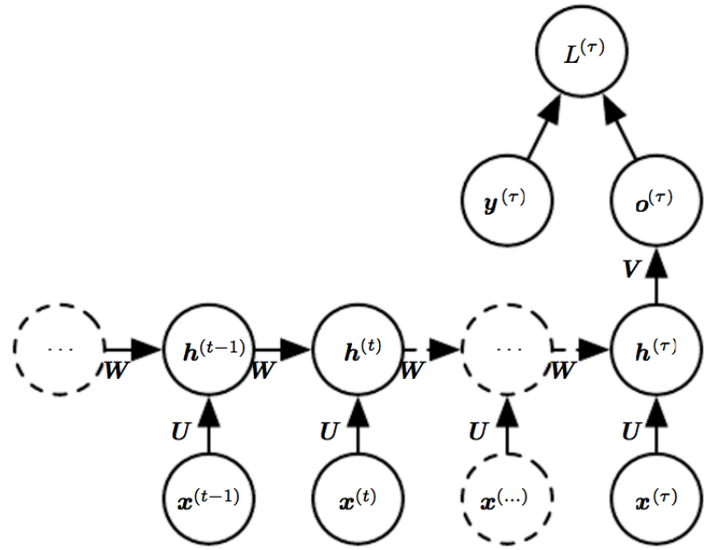
Long-Term Dependencies



Long-Term Dependencies

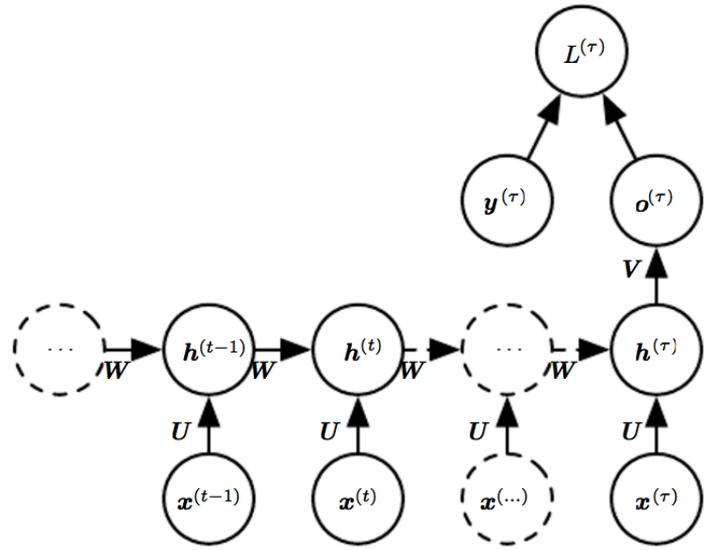


Challenge of Long-Term Dependencies



Challenge of Long-Term Dependencies

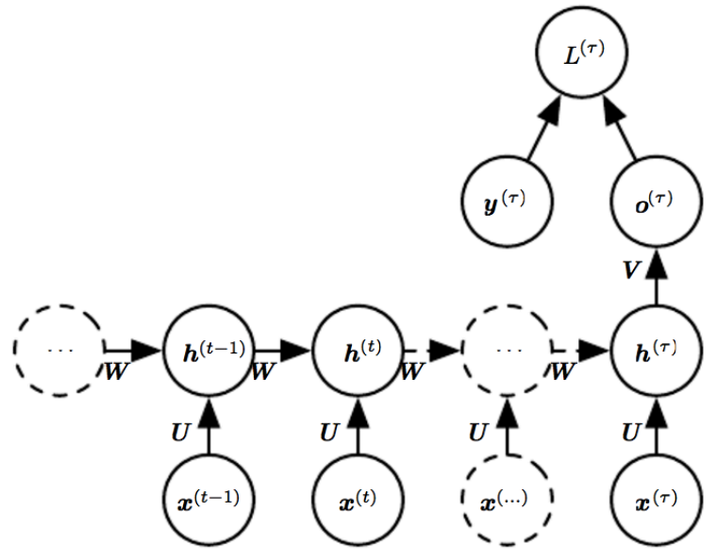
$h^{(t)} = W^T h^{(t-1)}$



Challenge of Long-Term Dependencies

$$h^{(t)} = W^T h^{(t-1)}$$

$$h^{(t)} = (W^t)^T h^{(0)}$$

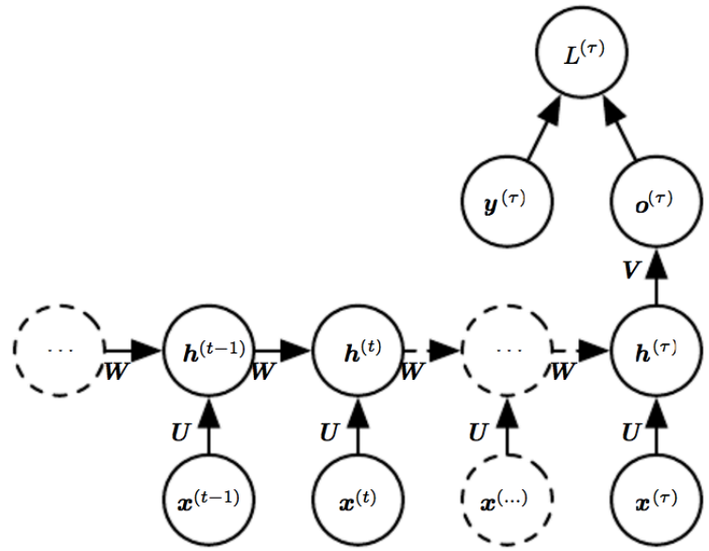


Challenge of Long-Term Dependencies

$$h^{(t)} = W^T h^{(t-1)}$$

$$h^{(t)} = (W^t)^T h^{(0)}$$

$$W = Q \Lambda Q^T$$



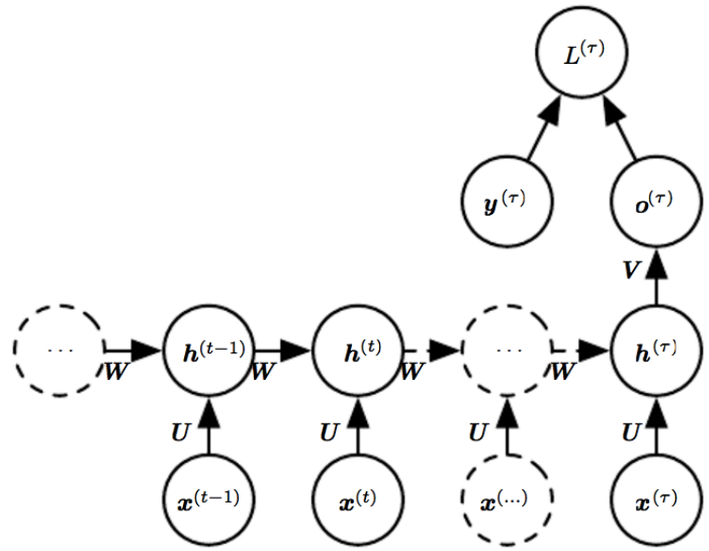
Challenge of Long-Term Dependencies

$$h^{(t)} = W^T h^{(t-1)}$$

$$h^{(t)} = (W^t)^T h^{(0)}$$

$$W = Q \Lambda Q^T$$

$$h^{(t)} = Q^T \Lambda^t Q h^{(0)}$$



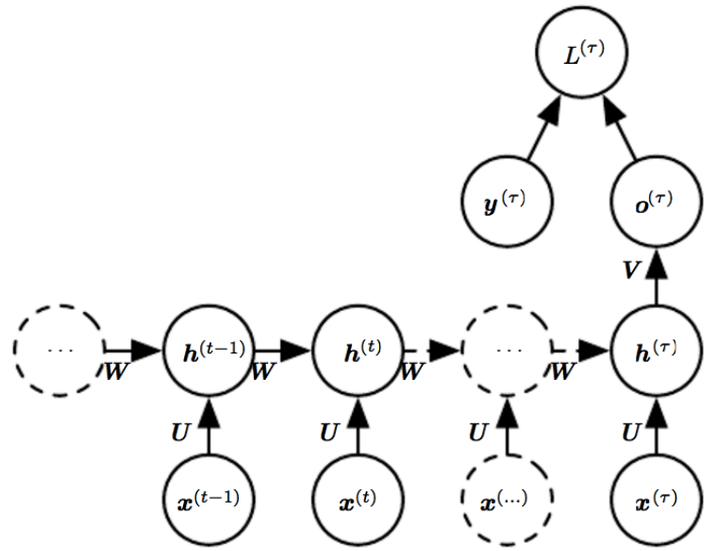
Challenge of Long-Term Dependencies

$$h^{(t)} = W^T h^{(t-1)}$$

$$h^{(t)} = (W^t)^T h^{(0)}$$

$$W = Q \Lambda Q^T$$

$$h^{(t)} = Q^T \Lambda^t Q h^{(0)}$$



- Eigenvalues < 1 decay to zero

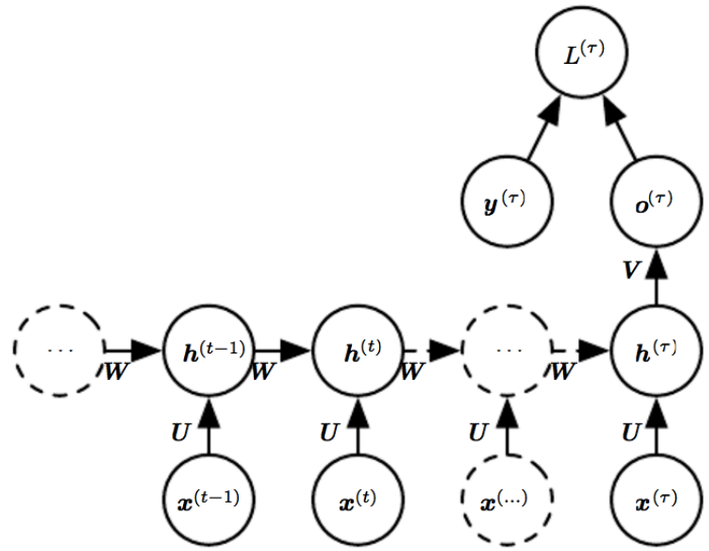
Challenge of Long-Term Dependencies

$$h^{(t)} = W^T h^{(t-1)}$$

$$h^{(t)} = (W^t)^T h^{(0)}$$

$$W = Q \Lambda Q^T$$

$$h^{(t)} = Q^T \Lambda^t Q h^{(0)}$$



- Eigenvalues < 1 decay to zero
- Components not aligned with larger eigenvalues are discarded

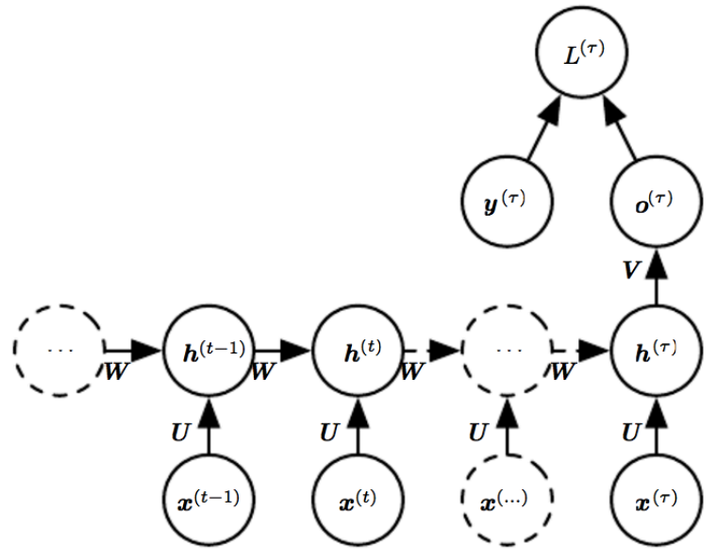
Challenge of Long-Term Dependencies

$$h^{(t)} = W^T h^{(t-1)}$$

$$h^{(t)} = (W^t)^T h^{(0)}$$

$$W = Q \Lambda Q^T$$

$$h^{(t)} = Q^T \Lambda^t Q h^{(0)}$$



- Eigenvalues < 1 decay to zero
- Components not aligned with larger eigenvalues are discarded
- Gradients vanish!

Handling Long-Term Dependencies

- Carefully chosen scaling can avoid vanishing gradients

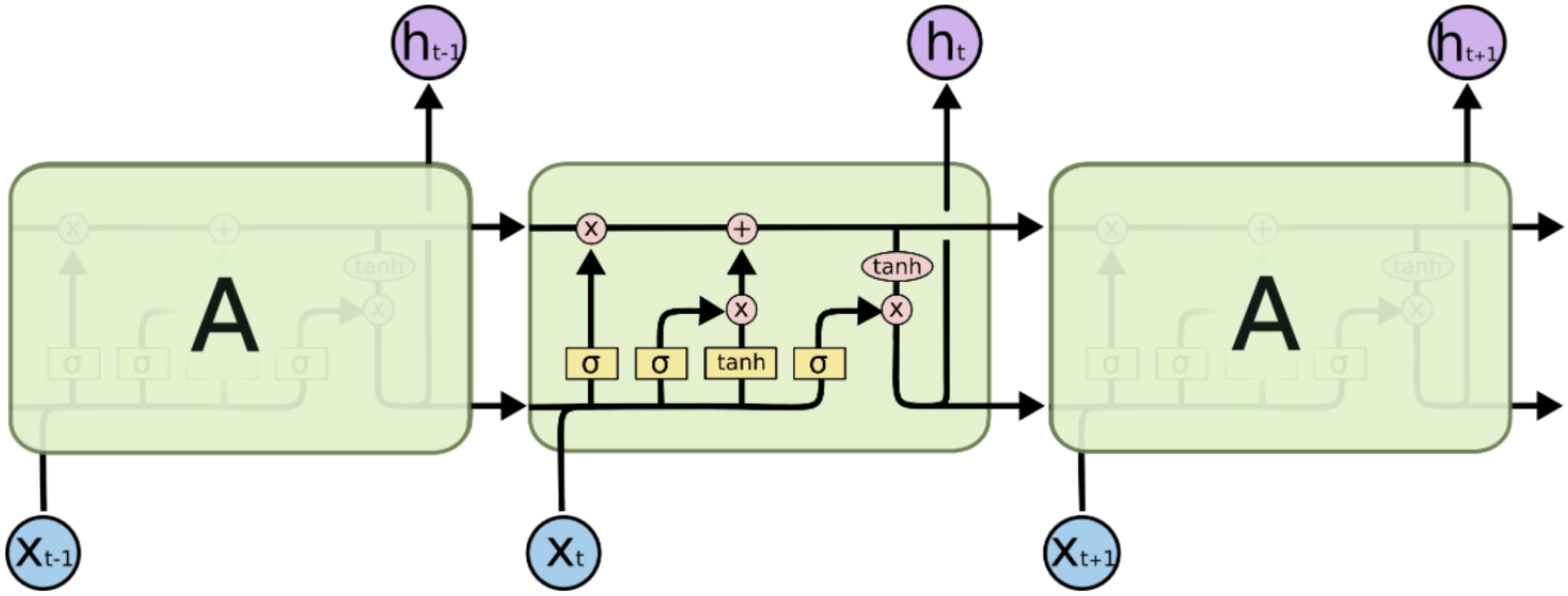
- Model long-term dependencies at different time scales: fine and coarse

- Skip connections

- Leaky self-connections

- Actively removing connections

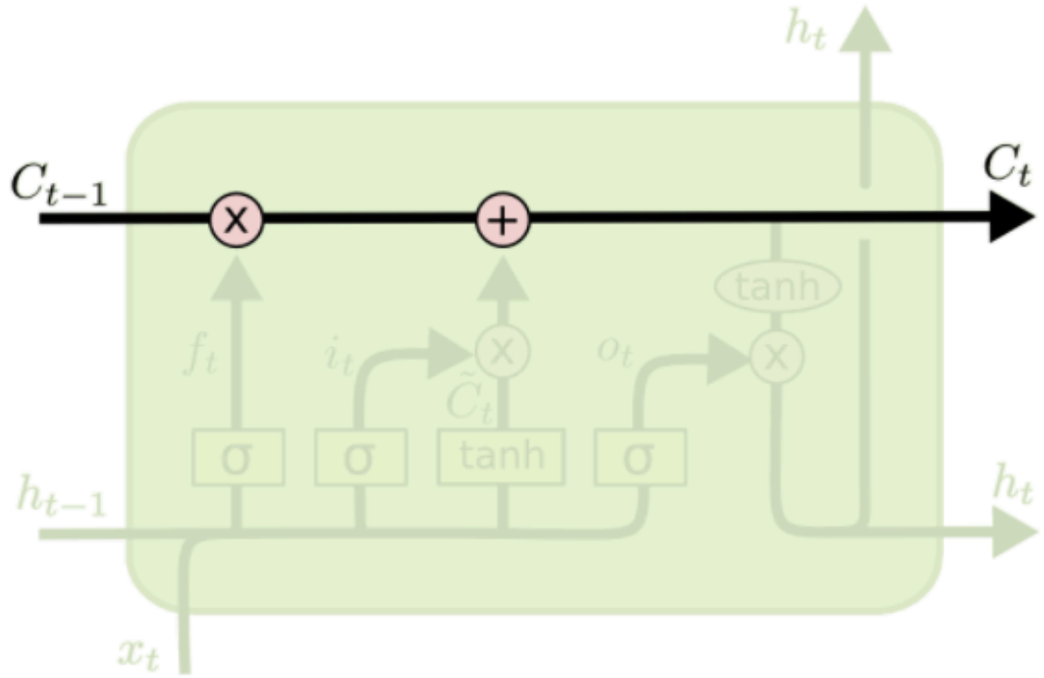
LSTMs: Leaky Connections



[Olah 2015]

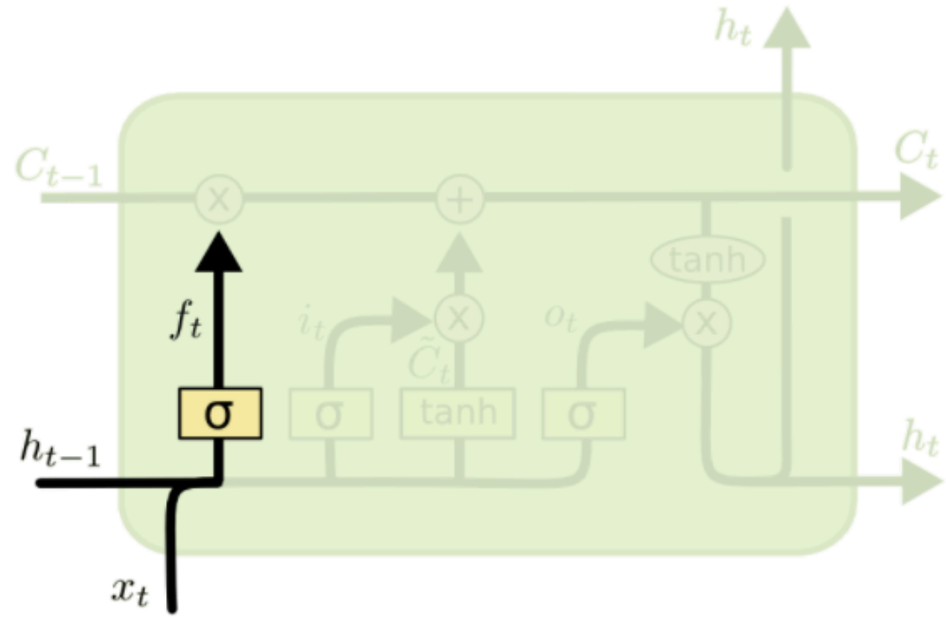
LSTM: Core Idea

- Cell state (C_t) carries the long-term information
- Minor linear interactions, mostly unchanged flow of information
- Can be changed by careful regulation using gates
- Gates outputs numbers between zero and one
- zero = let nothing through
- one = let everything through



LSTM: throwing away information from cell state

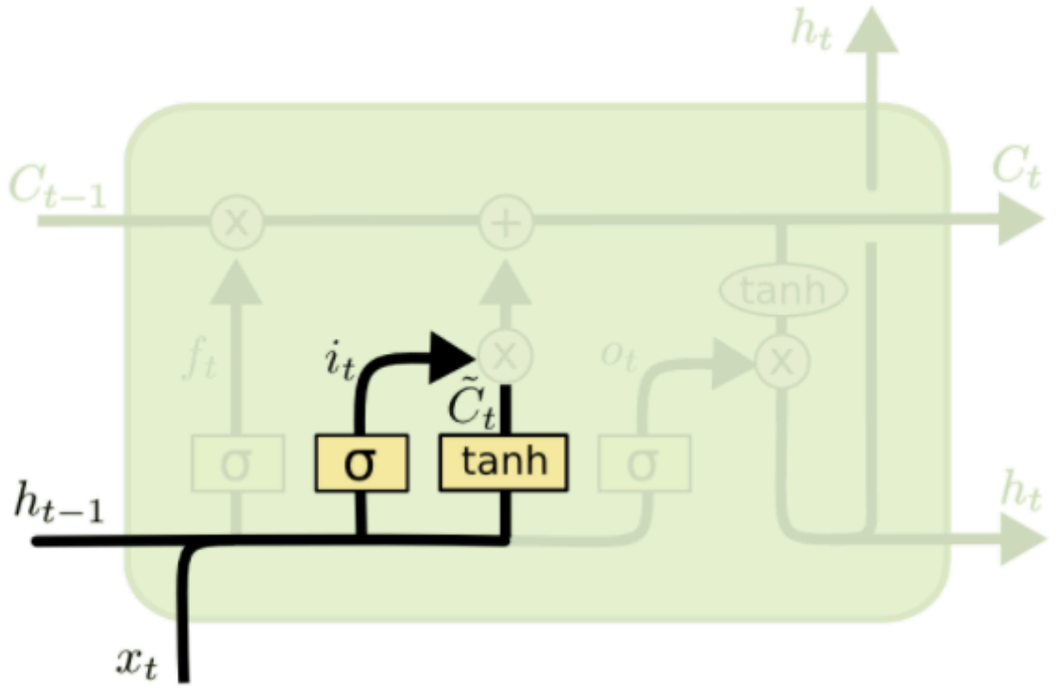
- Forget gate layer
- Forget specific parts of the previous cell state



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LSTM: Storing new info

- Updates the old cell state
- We already forgot things we wanted to forget
- We now add i_t * $\tilde{C}_t$ to the old cell state

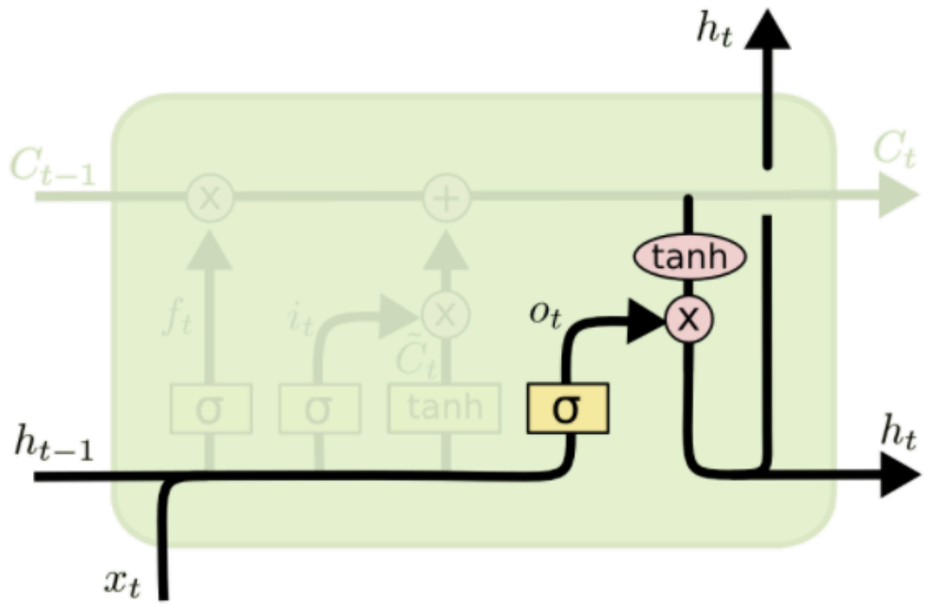


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

LSTM: Outputting new Cell State

- Output depends on the cell state
- Gate only outputs the part we decided to

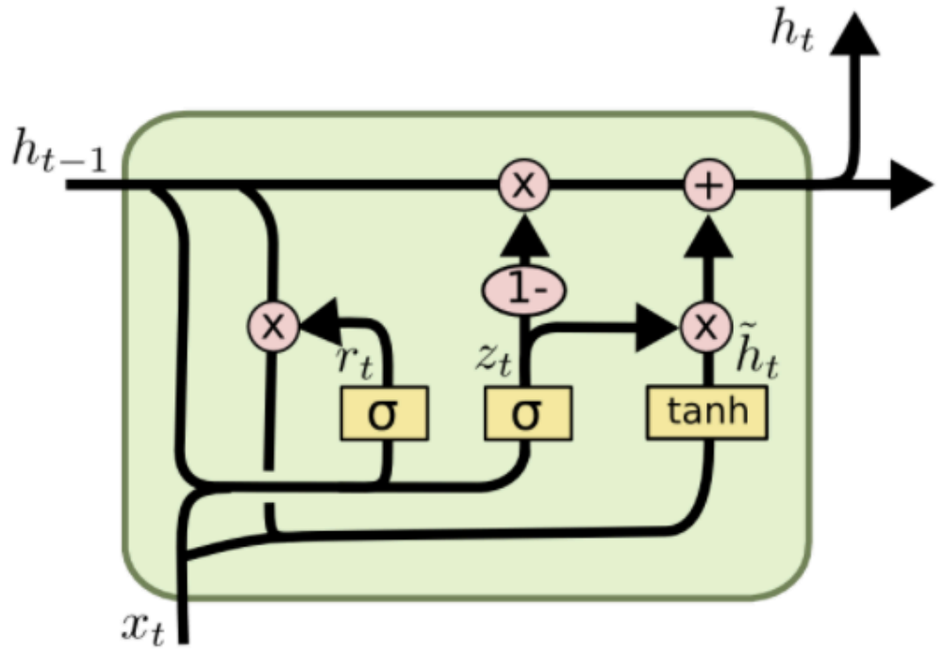


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Gated Recurrent Unit

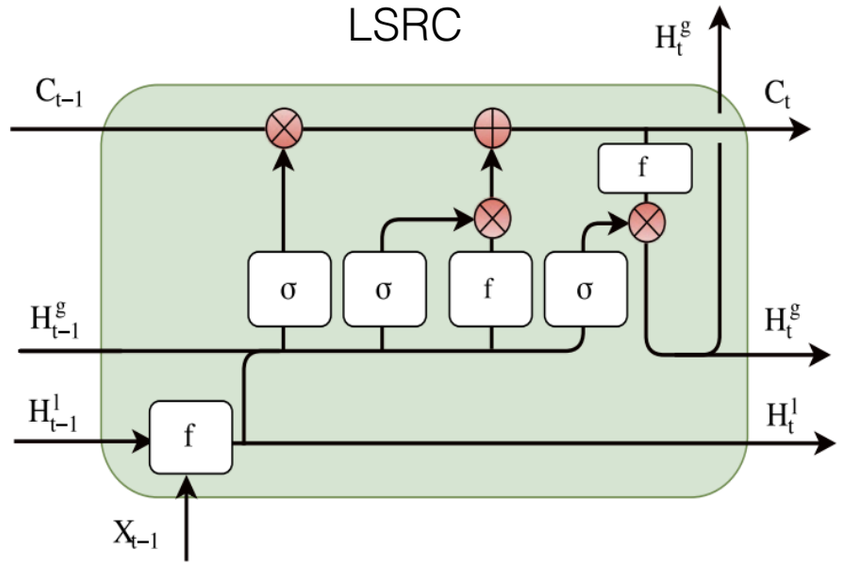
- Cell state is merged with hidden State
- Forget and input gates are combined into an “update” gate (z)

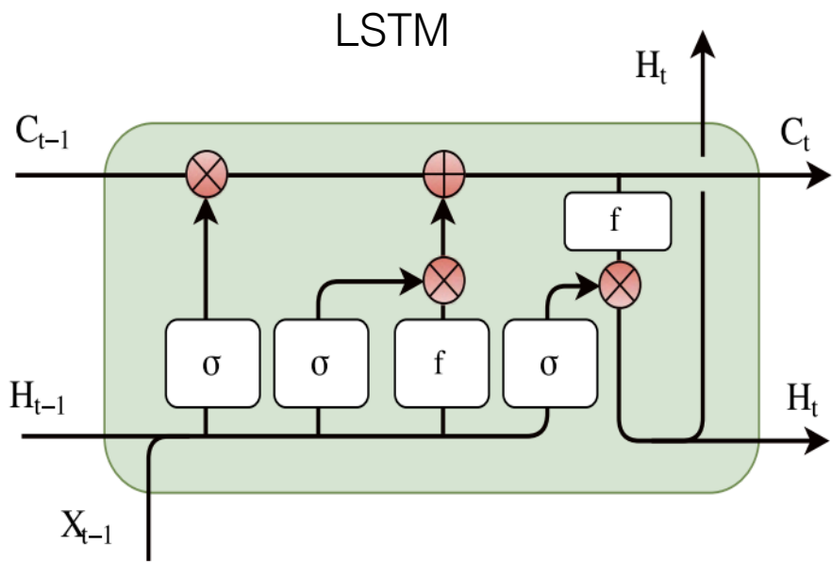


$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Long-Short Range Context Units

- Include variety of short-range information
- Divide the hidden state into two: a local hidden state and a global hidden state
- Use the local hidden state instead of input in LSTM





[Oualil et al 2016b]

Summary

- Recursive Neural Nets: Capturing long-term dependencies using a balanced tree structure

- Vanishing Gradient Problem

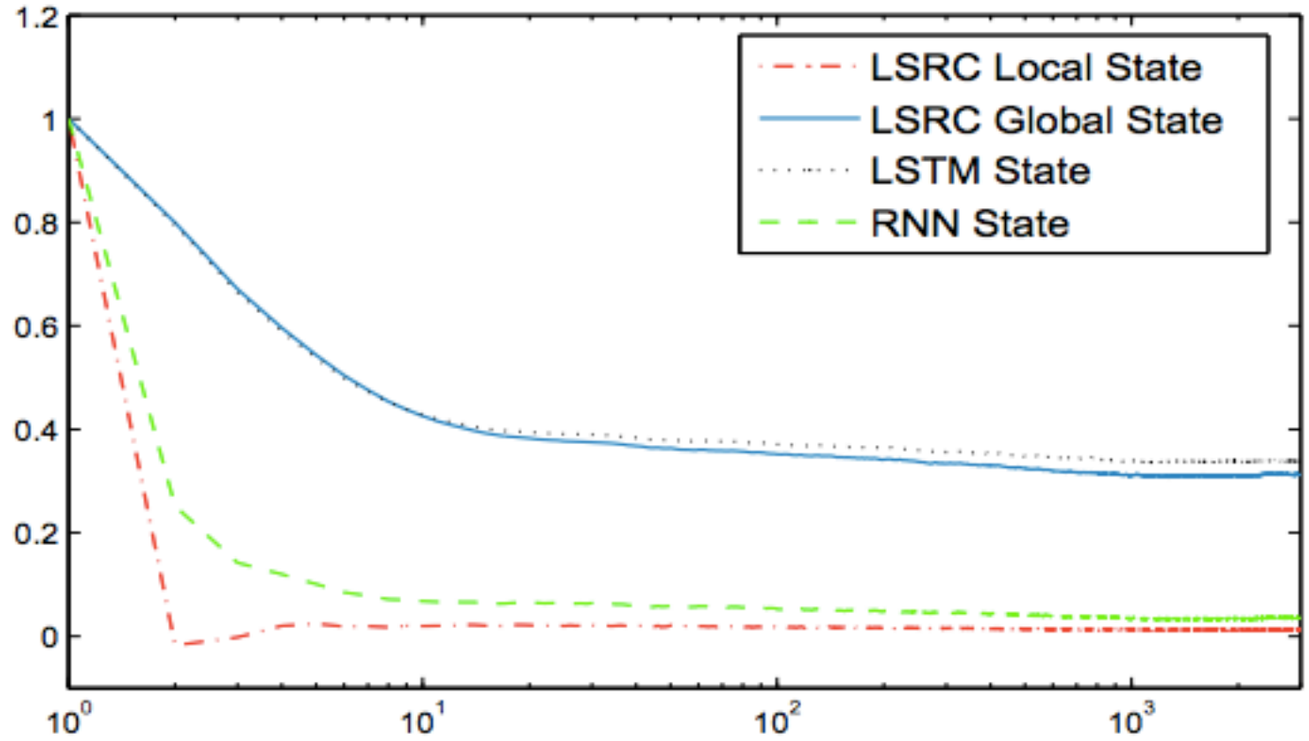
- Handling Vanishing Gradient Problem

- LSTMs and its structure

References

- **[Goodfellow et al 2016]** Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. http://www.deeplearningbook.org. MIT Press, 2016.
- **[Iyyer et al 2014]** Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal DaumÃ. In Proceedings of EMNLP, 2014
- **[Olah 2015]** Christopher Olah. Understanding LSTM Networks. Colah's Blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ , 2015
- **[Oualil et al 2016a]** Youssef Oualil, Mittul Singh, Clayton Greenberg and Dietrich Klakow. Sequential Recurrent Neural Networks for Language Modeling. In Proceedings of INTERSPEECH, 2016.
- **[Oualil et al 2016b]** Youssef Oualil, Mittul Singh, Clayton Greenberg, and Dietrich Klakow. Long-Short Range Context Neural Networks for Language Modeling. In Proceedings of EMNLP, 2016.

LSRC



Temporal correlation of LSRC in comparison to LSTM and RNN.

SRNN: Input to Hidden weights Histogram