Language modeling, RNNs

Feb 10, 2026

RIT Rochester Institute of Technology

Outline

RIT Rochester Institute of Technology

Outline

RIT Rochester Institute of Technology

Language modeling

- **Language modeling** is the task of predicting the next word in a sequence.
- Example:

  the students opened their ____

  {*books*, *laptops*, *exams*, *minds*}

- Formally, given a sequence $x_1, x_2, \ldots, x_t$, we estimate

$$P\left(x_{t+1} \mid x_1, x_2, \ldots, x_t\right).$$

RIT Rochester Institute of Technology

Language modeling

- A language model can also be viewed as a system that assigns a probability to an entire sequence of tokens.
- For a text $x_1, \ldots, x_T$, the joint probability is

$$
\begin{aligned}
&P\big(x_1, \ldots, x_T\big) \\
&= P\big(x_1\big)\, P\big(x_2 \mid x_1\big) \cdots P\big(x_T \mid x_1, \ldots, x_{T-1}\big) \\
&= \prod_{i=1}^{T} P\big(x_i \mid x_1, \ldots, x_{i-1}\big)
\end{aligned}
$$

RIT Rochester Institute of Technology

Example: Sequence probability

- Consider the sentence: *"I like apples"*.
- The joint probability is decomposed as:
  - $P(\text{“I”})$ = probability that “I” starts the sentence
  - $P(\text{“like”} \mid \text{“I”})$ = probability that “like” follows “I”
  - $P(\text{“apples”} \mid \text{“I like”})$ = probability that “apples” follows “I like”
- Multiplying these gives the overall probability of the sentence:

$$P(\text{“I like apples”}) = P(\text{“I”}) \cdot P(\text{“like”} \mid \text{“I”}) \cdot P(\text{“apples”} \mid \text{“I like”})$$

How do we build a language model?

- **Question:** We want to estimate

$$P(x_1, \ldots, x_T) = \prod_{i=1}^{T} P(x_i \mid x_1, \ldots, x_{i-1})$$

- **Very first idea:** Approximate by looking only at a few previous words.
- This leads us to n-gram language models.

Outline

RIT Rochester Institute of Technology

n-gram language models

$$P(x_{t+1} \mid x_t, \ldots, x_1) \approx P(x_{t+1} \mid x_t, \ldots, x_{t-n+2})$$

- t: position of the current token in the sequence
- n: size of the n-gram (the model looks back $n-1$ tokens)

Only the last $(n-1)$ words matter.

Conditional probability

- **Definition:**

$$P(A \mid B) = \frac{P(A, B)}{P(B)}.$$

- Apply to n-gram:

$$P(x_{t+1} \mid x_t, \ldots, x_{t-n+2}) = \frac{P(x_{t+1}, x_t, \ldots, x_{t-n+2})}{P(x_t, \ldots, x_{t-n+2})}.$$

Example: *Every morning, my neighbor yelled at the* _____

Example: *Every morning, my neighbor yelled at the* _____
(**4-gram**) Conditioning only on the **last three words**:

Example: *Every morning, my neighbor yelled at the* _____
(**4-gram**) Conditioning only on the **last three words**:
Every morning, my neighbor yelled at the _____

Example: *Every morning, my neighbor yelled at the* _____
(**4-gram**) Conditioning only on the **last three words**:
Every morning, my neighbor yelled at the _____

$$\hat{P}(w \mid \text{yelled at the}) = \frac{\text{count}(\text{yelled at the } w)}{\text{count}(\text{yelled at the})}.$$

Example: *Every morning, my neighbor yelled at the* _____
(**4-gram**) Conditioning only on the **last three words**:
Every morning, my neighbor yelled at the _____

$$\hat{P}(w \mid \text{yelled at the}) = \frac{\text{count}(\text{yelled at the } w)}{\text{count}(\text{yelled at the})}.$$

Suppose in the corpus:

- `yelled at the` occurs 600 times,
- `yelled at the dog` occurs 250 times, so

Example: *Every morning, my neighbor yelled at the* _____
(**4-gram**) Conditioning only on the **last three words**:
Every morning, my neighbor yelled at the _____

$$\hat{P}(w \mid \text{yelled at the}) = \frac{\text{count}(\text{yelled at the } w)}{\text{count}(\text{yelled at the})}.$$

Suppose in the corpus:

- `yelled at the` occurs 600 times,
- `yelled at the dog` occurs 250 times, so

$$P(\text{dog} \mid \text{yelled at the}) = 0.42,$$

- `yelled at the kids` occurs 180 times, so

Example: *Every morning, my neighbor yelled at the* _____
(**4-gram**) Conditioning only on the **last three words**:
Every morning, my neighbor yelled at the _____

$$\hat{P}(w \mid \text{yelled at the}) = \frac{\text{count}(\text{yelled at the } w)}{\text{count}(\text{yelled at the})}.$$

Suppose in the corpus:

- `yelled at the` occurs 600 times,
- `yelled at the dog` occurs 250 times, so

$$P(\text{dog} \mid \text{yelled at the}) = 0.42,$$

- `yelled at the kids` occurs 180 times, so

$$P(\text{kids} \mid \text{yelled at the}) = 0.30.$$

Progress

- As vocabulary and context grow, count-based n-grams suffer from s_____.
- n_____ language models address this by learning distributed representations within a context window.

Window-based neural language model

output distribution

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{U}\boldsymbol{h} + \boldsymbol{b}_2) \in \mathbb{R}^{|V|}$$

hidden layer

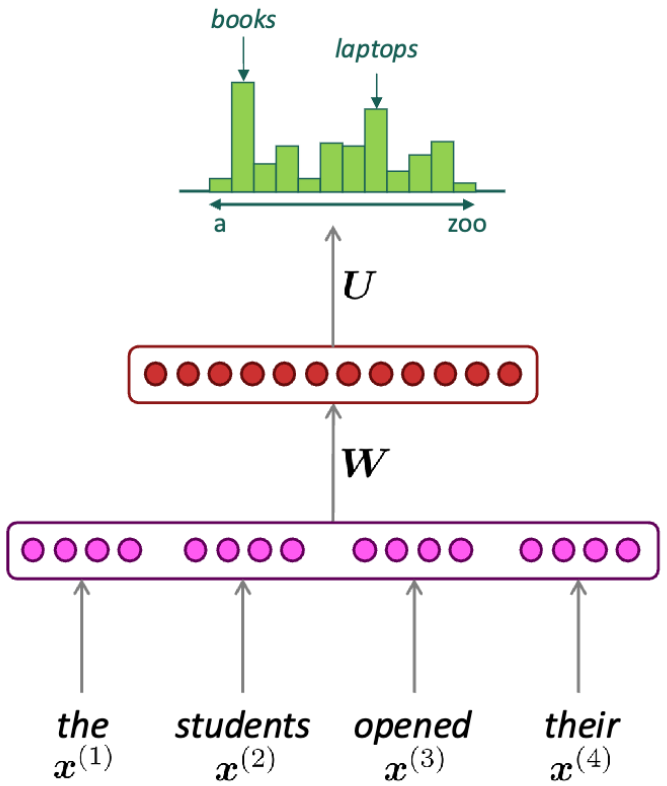
$$\boldsymbol{h} = f(\boldsymbol{W}\boldsymbol{e} + \boldsymbol{b}_1)$$

concatenated word embeddings

$$\boldsymbol{e} = [\boldsymbol{e}^{(1)}; \boldsymbol{e}^{(2)}; \boldsymbol{e}^{(3)}; \boldsymbol{e}^{(4)}]$$

words / one-hot vectors

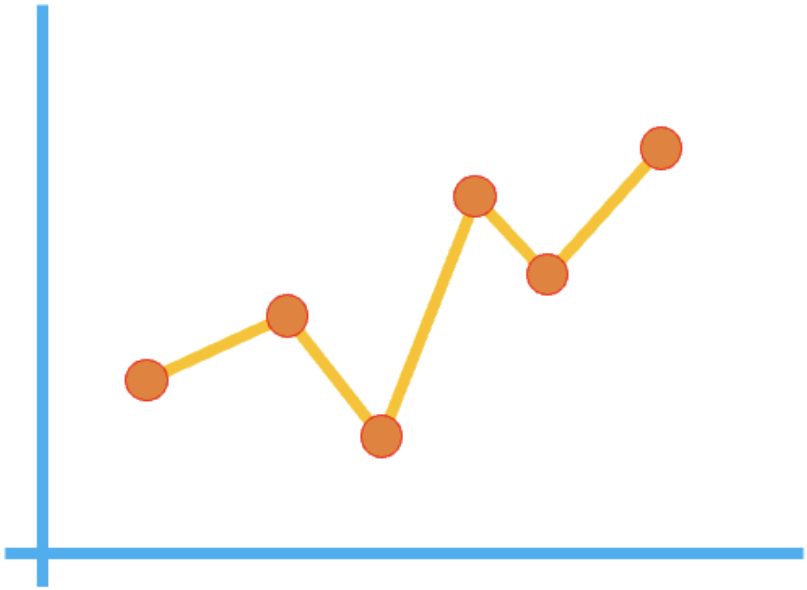
$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)}, \boldsymbol{x}^{(4)}$$



Outline

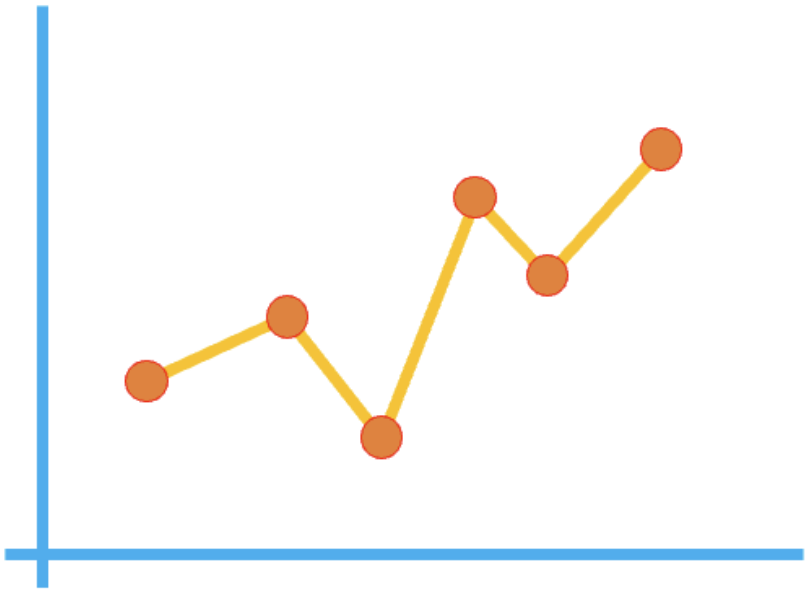
Intro

Need to process (time series) dataset (e.g., words in a sentence)



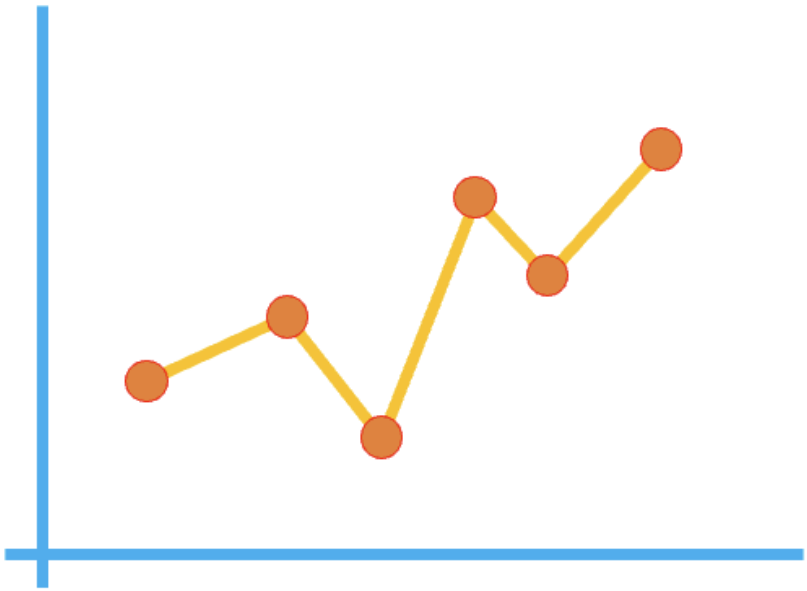
This is

Need to process (time series) dataset (e.g., words in a sentence)



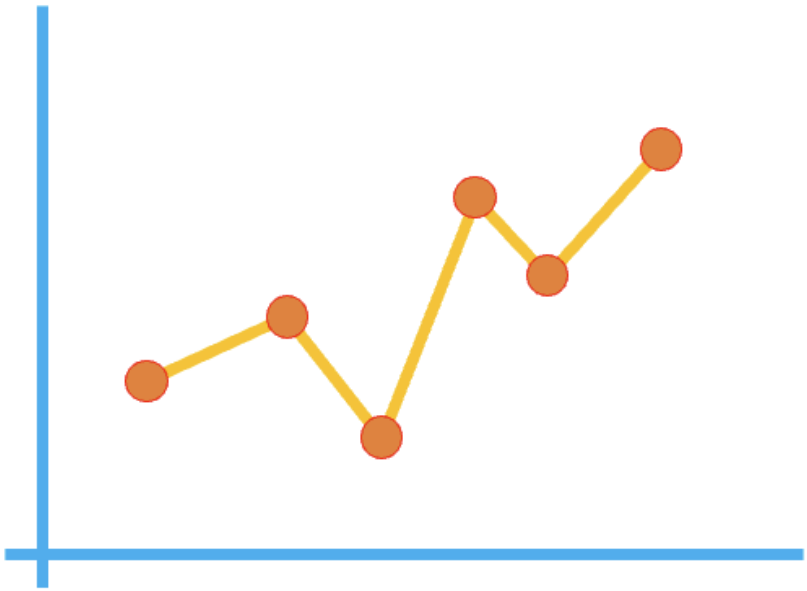
This is an

Need to process (time series) dataset (e.g., words in a sentence)



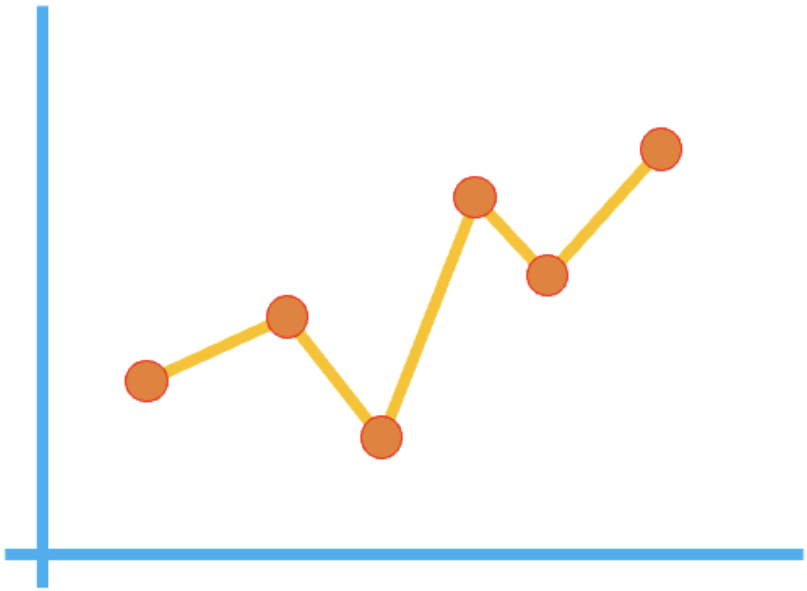
This is an awesome

Need to process (time series) dataset (e.g., words in a sentence)



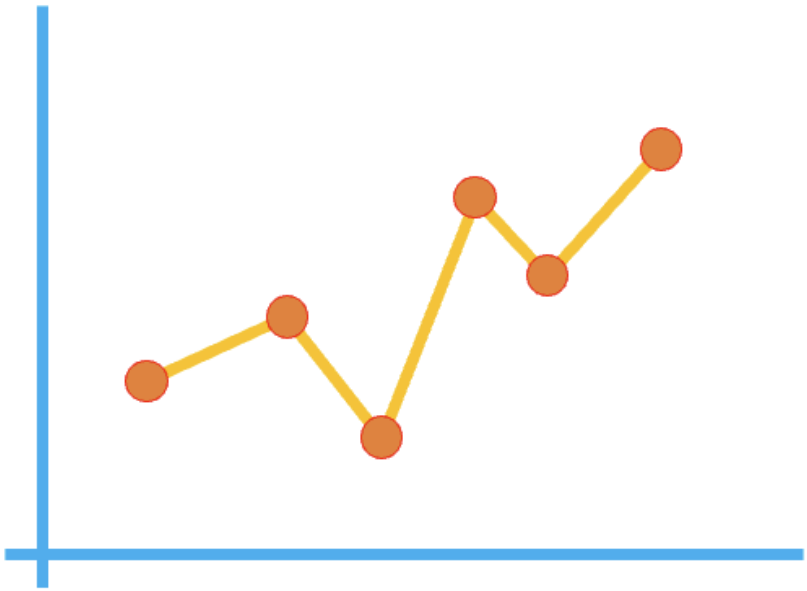
This is an awesome

Need to process (time series) dataset (e.g., words in a sentence)



This is an awesome
sentence

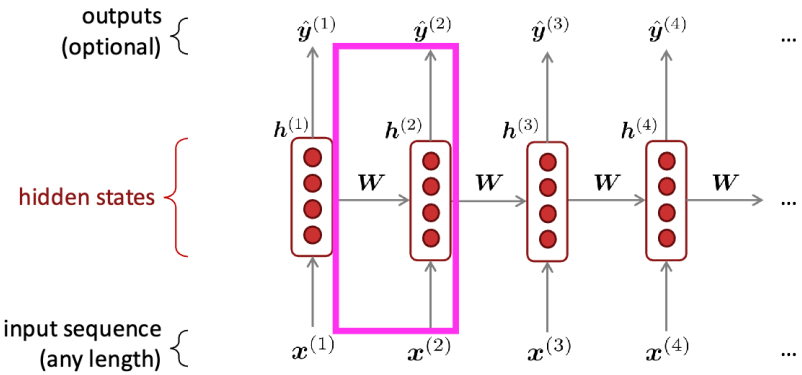
Need to process (time series) dataset (e.g., words in a sentence)



This is an awesome
sentence that

RNNs

- Idea: Repeatedly apply the **same weight** matrix W at each time step
- Maintain a hidden state over time, feeding it back into the network to capture temporal dependencies



1. Start with a corpus, represented as a sequence of words $w_1, \ldots, w_{T-1}, w_T$.
2. Feed this sequence into the RNN-based language model.
3. At each time step t, the model outputs a probability distribution $\hat{\mathbf{y}}_t$ over the vocabulary.
   - Internally, the RNN updates its hidden state $\mathbf{h}_t$, then applies a linear layer followed by softmax:

$$\hat{\mathbf{y}}_t = \mathrm{softmax}(W_o \mathbf{h}_t + b_o).$$

   - Each component of $\hat{\mathbf{y}}_t$ corresponds to

$$P\big(w_{t+1} = v_i \mid w_1, \ldots, w_t\big),$$

     i.e., the probability that the next word is v_i.
   - Put simply, at every step t, the model predicts the likelihood of each possible next word **given all preceding words**.

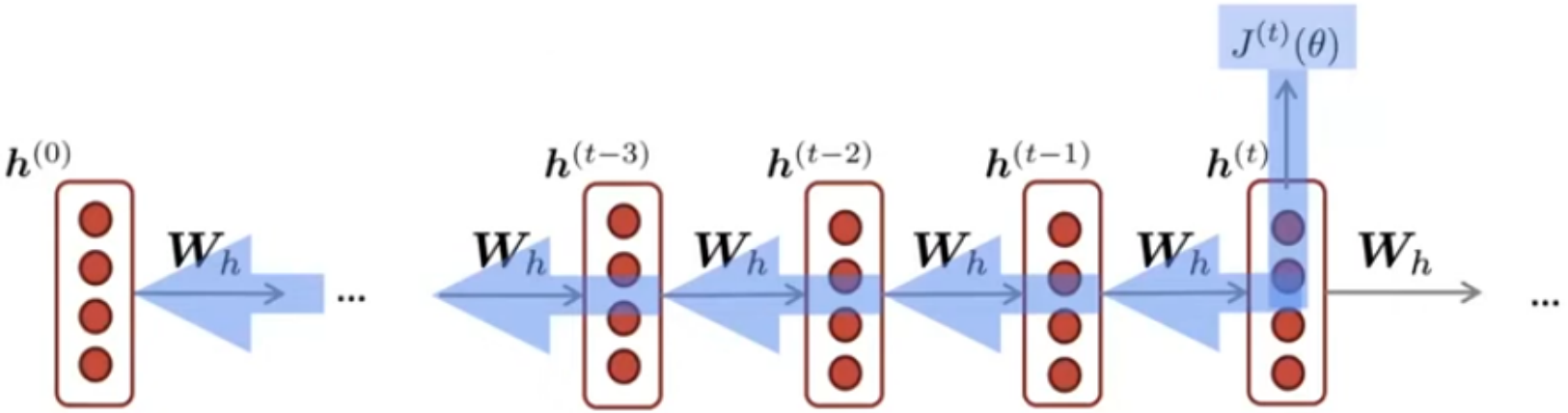
- **Loss** at step t:

$$\mathcal{J}^{(t)} = -\sum_{i=1}^{|V|} y_i^{(t)} \log \hat{y}_i^{(t)} = -\log \hat{y}_{w_{t+1}}^{(t)},$$

where:

- $y^{(t)}$: one-hot vector for the true next word w_{t+1}.
- $\hat{y}^{(t)}$: predicted probability distribution over the vocabulary from the softmax layer.
- This is the cross-entropy **loss** between the predicted distribution and the true label.

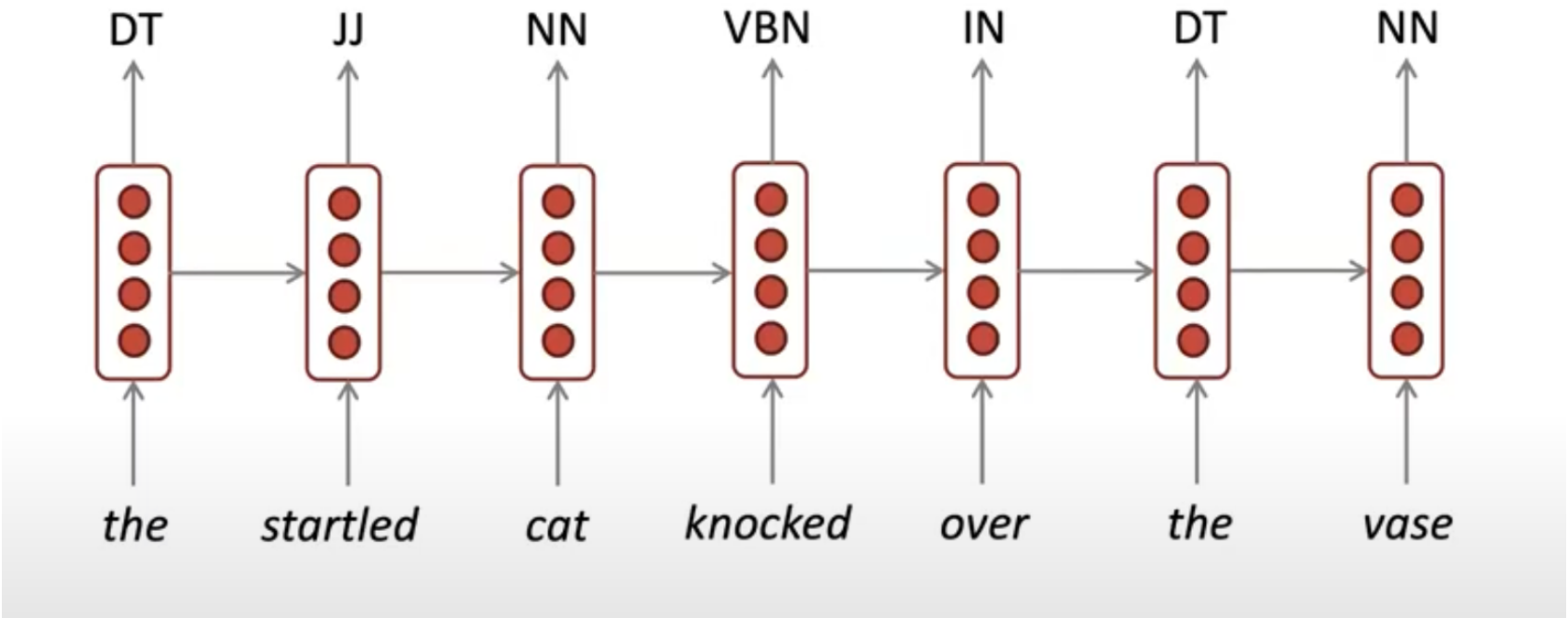
RNNs+Backpropagation



RIT Rochester Institute of Technology

NLP applications

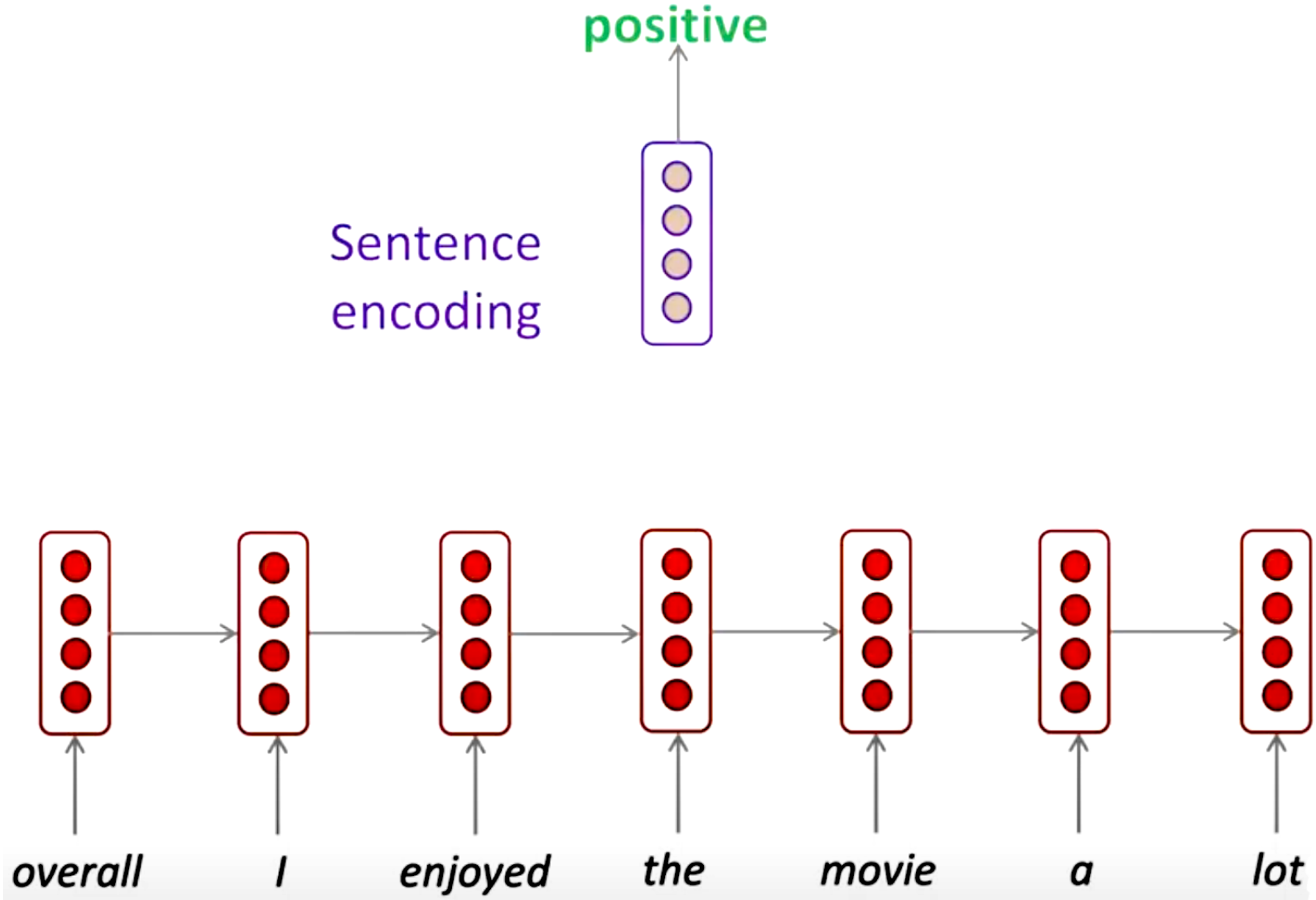
POS tagging



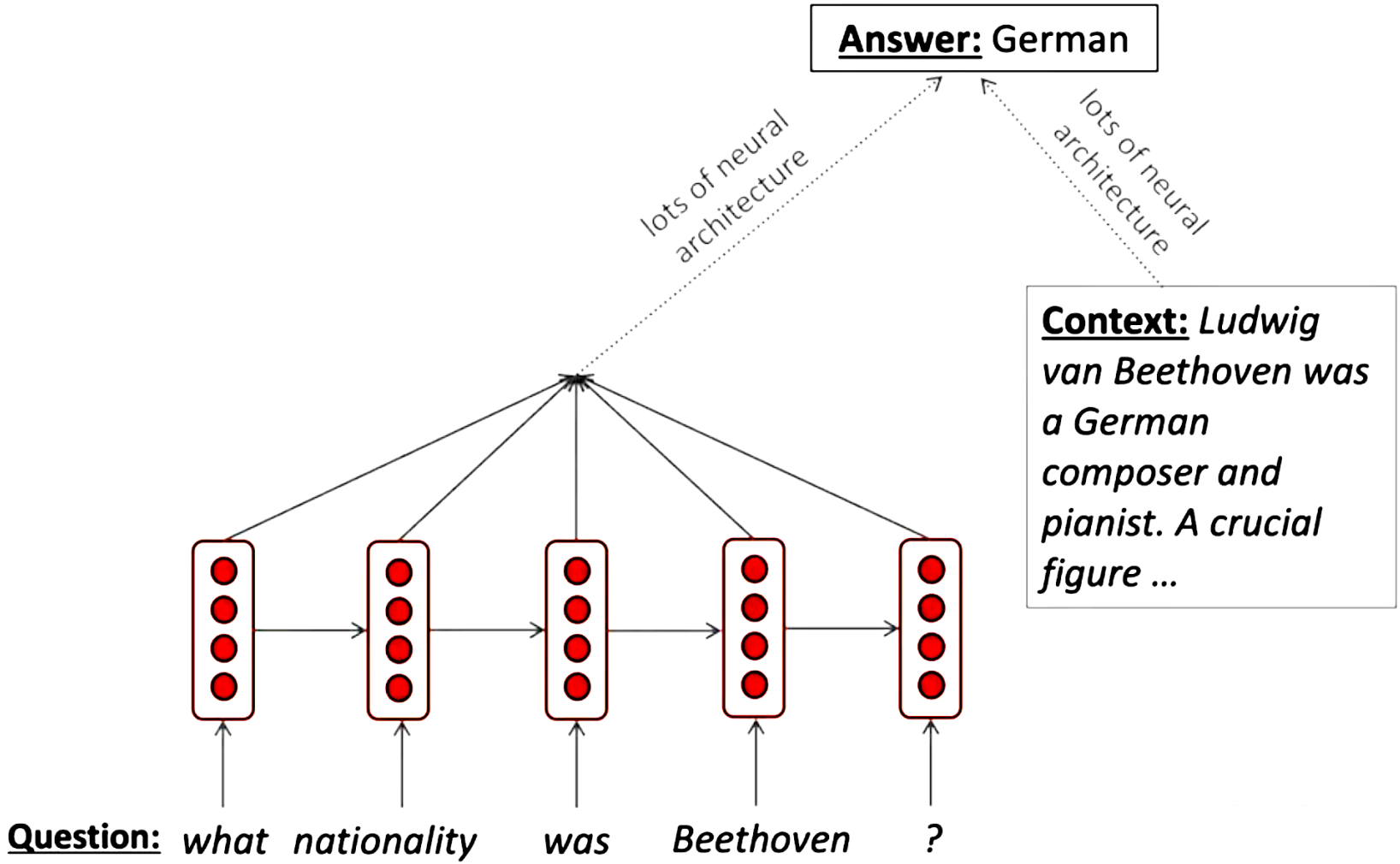
NER tagging

contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported ORG byF.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top F.B.I. GPE counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump PERSON were uncovered, was fired. CreditT.J. Kirkpatrick PERSON for The New York TimesBy Adam Goldman ORG and Michael S. SchmidtAug PERSON . 13 CARDINAL , 2018WASHINGTON CARDINAL — Peter Strzok PERSON , the F.B.I. GPE senior counterintelligence agent who disparaged President Trump PERSON in inflammatory text messages and helped oversee the Hillary Clinton PERSON email and Russia GPE investigations, has been fired for violating bureau policies, Mr. Strzok PERSON 's lawyer said Monday DATE .Mr. Trump and his allies seized on the texts — exchanged during the 2016 DATE campaign with a former F.B.I. GPE lawyer, Lisa Page — in PERSON assailing the Russia GPE investigation as an illegitimate "witch hunt." Mr. Strzok PERSON , who rose over 20 years DATE at the F.B.I. GPE to become one of its most experienced counterintelligence agents, was a key figure in the early months DATE of the inquiry.Along with writing the texts, Mr. Strzok PERSON was accused of sending a highly sensitive search warrant to his personal email account.The F.B.I. GPE had been under immense political pressure by Mr. Trump PERSON to dismiss Mr. Strzok PERSON , who was removed last summer DATE from the staff of the special counsel, Robert S. Mueller III PERSON . The president has repeatedly denounced Mr. Strzok PERSON in posts on

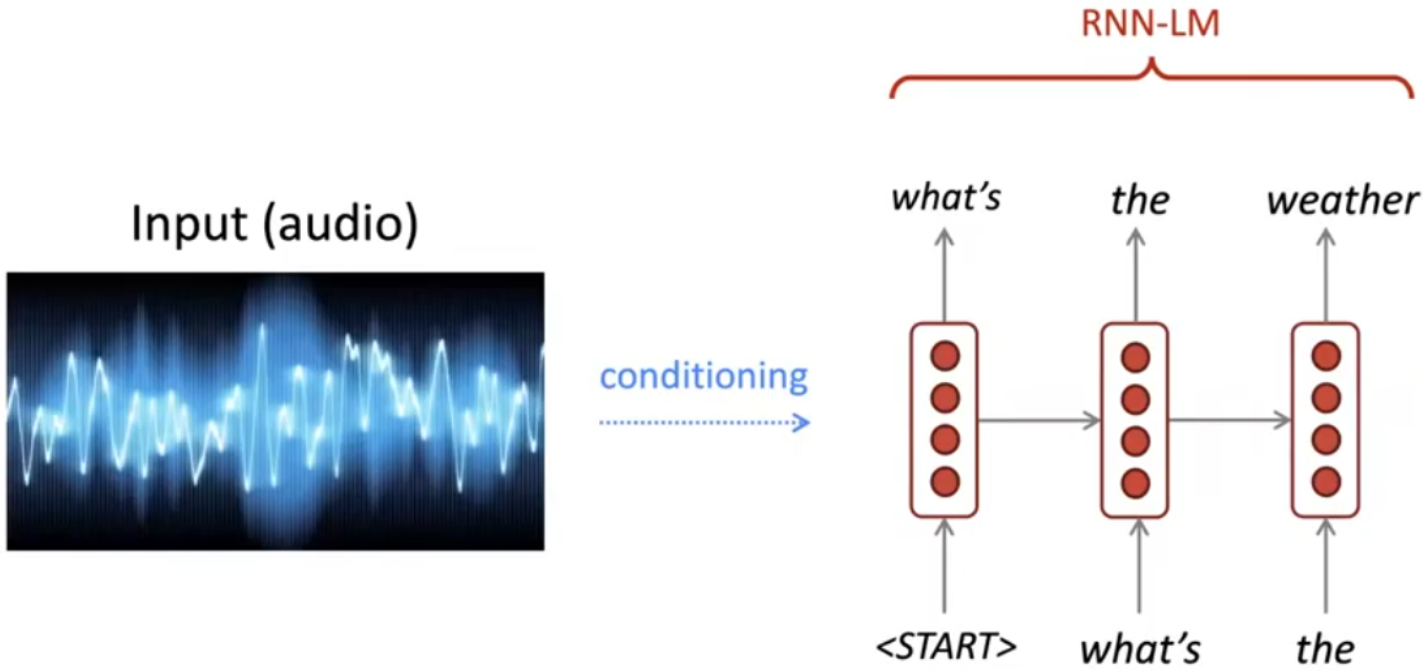
Sentiment classification



Question answering

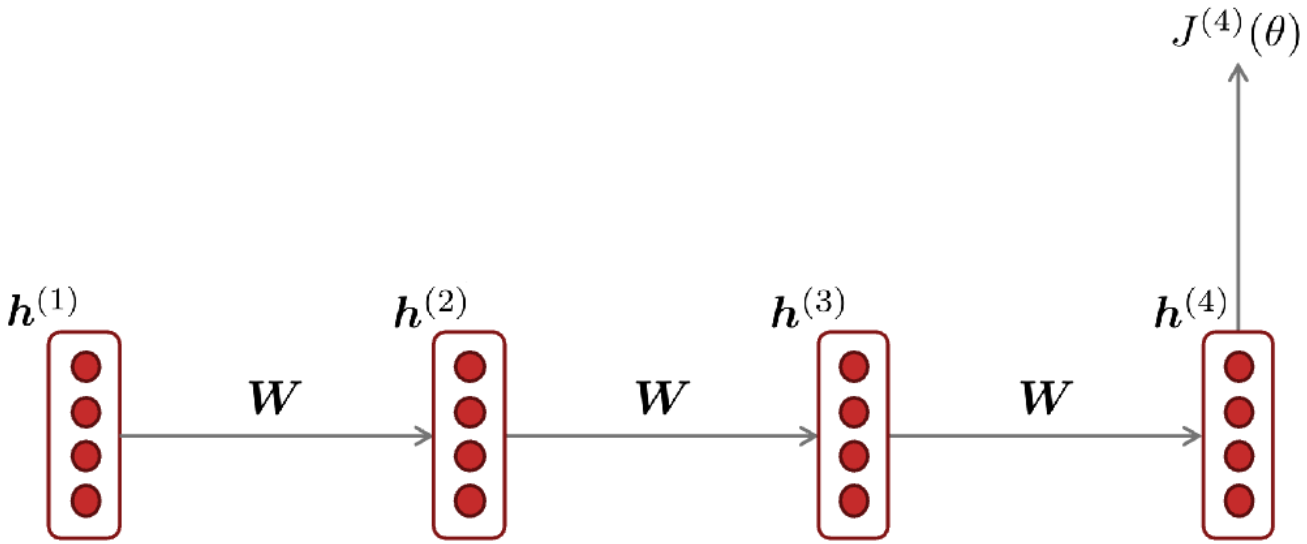


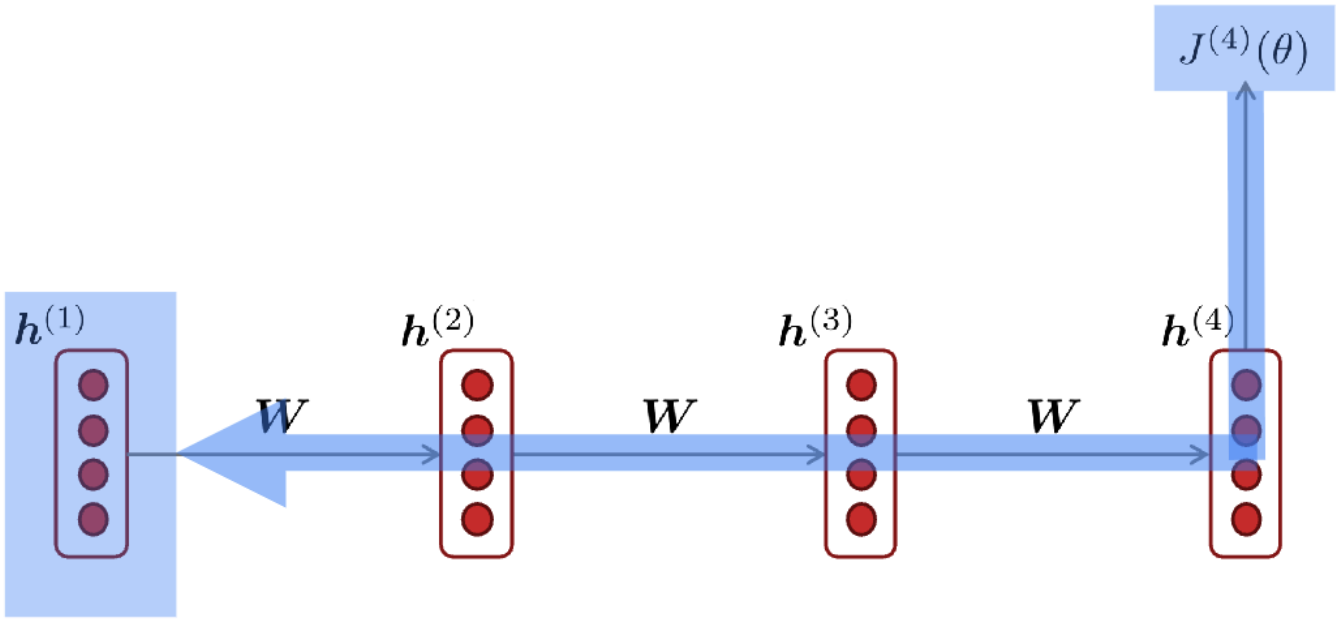
Speech recognition



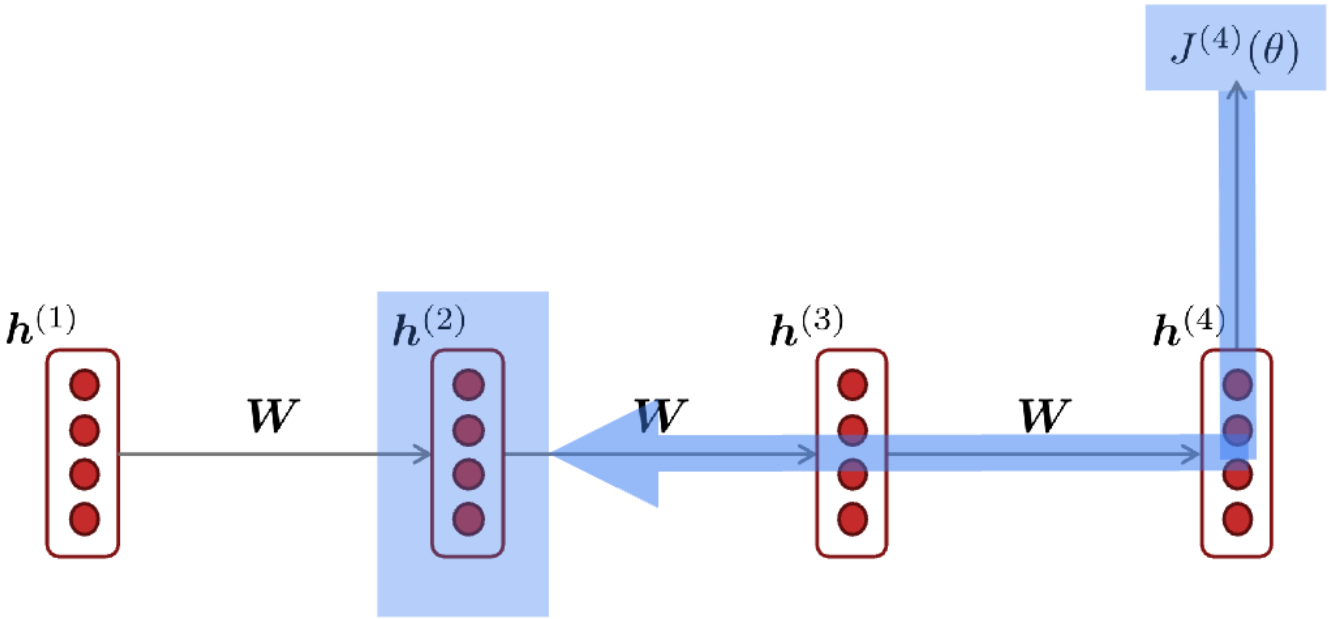
Outline

Problem with RNN 1: Vanishing gradient



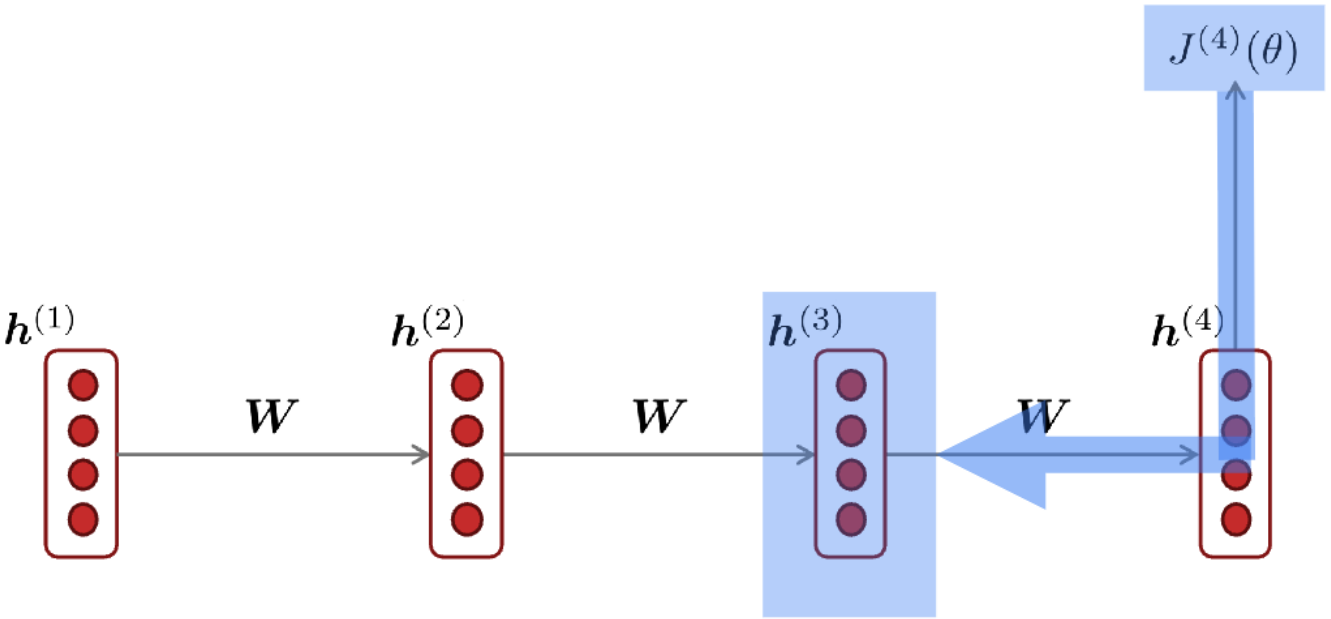


$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = \text{?}$$

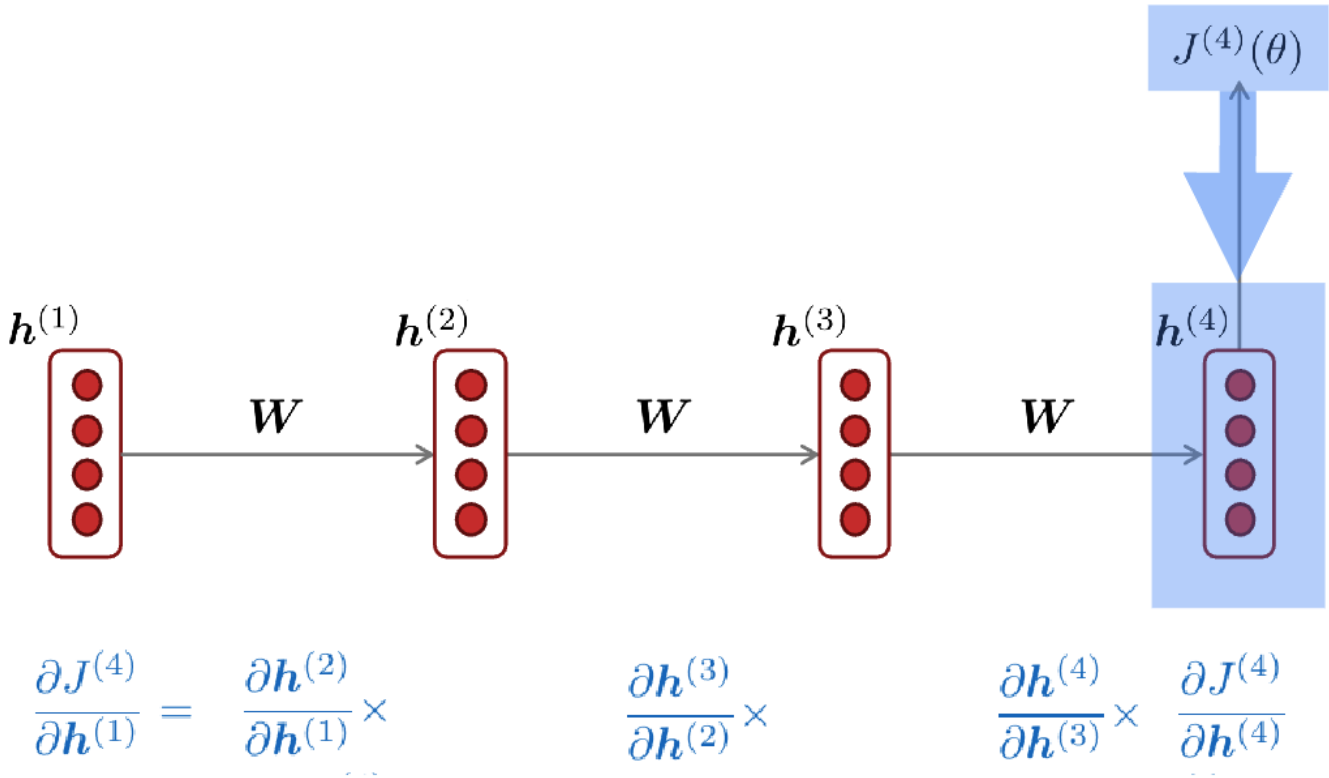


$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = \frac{\partial h^{(2)}}{\partial h^{(1)}} \times \frac{\partial J^{(4)}}{\partial h^{(2)}}$$

RIT Rochester Institute of Technology



$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = \frac{\partial h^{(2)}}{\partial h^{(1)}} \times \frac{\partial h^{(3)}}{\partial h^{(2)}} \times \frac{\partial J^{(4)}}{\partial h^{(3)}}$$



If each step's gradient is too small, multiplying across many steps makes it shrink exponentially.
The overall gradient $\to 0$, so the model cannot learn long-range dependencies.

Example:

LM task: *When she tried to print her tickets, she found that the printer was out of toner. She went to the stationery store to buy more toner. It was very overpriced. After installing the toner into the printer, she finally printed her _____*

Example:

LM task: *When she tried to print her tickets, she found that the printer was out of toner. She went to the stationery store to buy more toner. It was very overpriced. After installing the toner into the printer, she finally printed her* _____

- To learn from this training example, the LM needs to model the dependency between "tickets" on the 7th step and the target word "tickets" at the end.
- But if the gradient is small, the model can't learn this dependency
  - So, the model is unable to predict similar long-distance dependencies at test time

Problem with RNN 2: Exploding gradient

- When gradients become very large:
  - A single update step can overshoot the minimum
  - and destabilize or even blow up the model.

Vanishing problem: Solution

Solutions explored:

- Separate memory cell (e.g., **LSTM**) with gating mechanisms to add/erase information.
- Direct pass-through connections (attention, residual links) for better gradient flow.

Presentation

Shubh Sudan - Sak et al. (2014). LSTM Architectures for Acoustic Modeling.

Outline

RIT Rochester Institute of Technology

On Thursday

- Presentation: Willow - Du et al. (2024). Financial Sentiment Analysis.
- Work on Lab 5 - Sentiment analysis