STA 221: LECTURE 14

KRISHNA BALASUBRAMANIAN

(UNIVERSITY OF CALIFORNIA, DAVIS)

Recurrent Neural Network

Time Series/Sequence Data

Input: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$

- Each $\boldsymbol{x}_t$ is the feature at time step t
- Each $\boldsymbol{x}_t$ can be an d-dimensional vector

Output: $\{y_1, y_2, \cdots, y_T\}$

- Each y_t is the output at step t
- Multi-class output or Regression output:

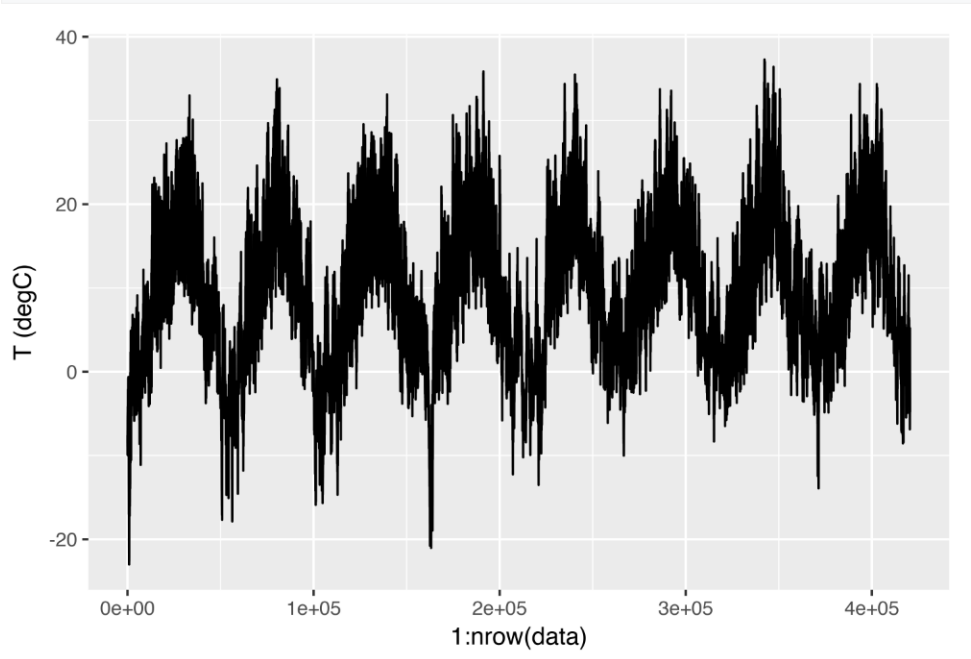
$$y_t \in \{1, 2, \cdots, L\} \quad \text{or} \quad y_t \in \mathbb{R}$$

Example: Time Series Prediction

Climate Data:

- $\boldsymbol{x}_t$: temperature at time t
- y_t: temperature (or temperature change) at time $t+1$

Stock Price: Predicting stock price

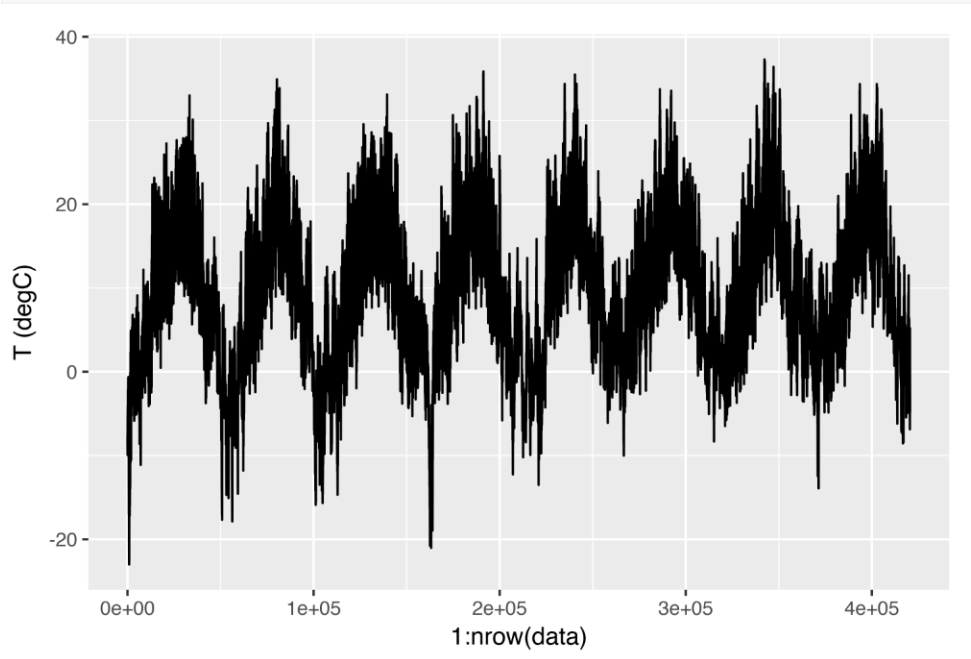


Example: Time Series Prediction

Climate Data:

- $\boldsymbol{x}_t$: temperature at time t
- y_t: temperature (or temperature change) at time $t+1$

Stock Price: Predicting stock price



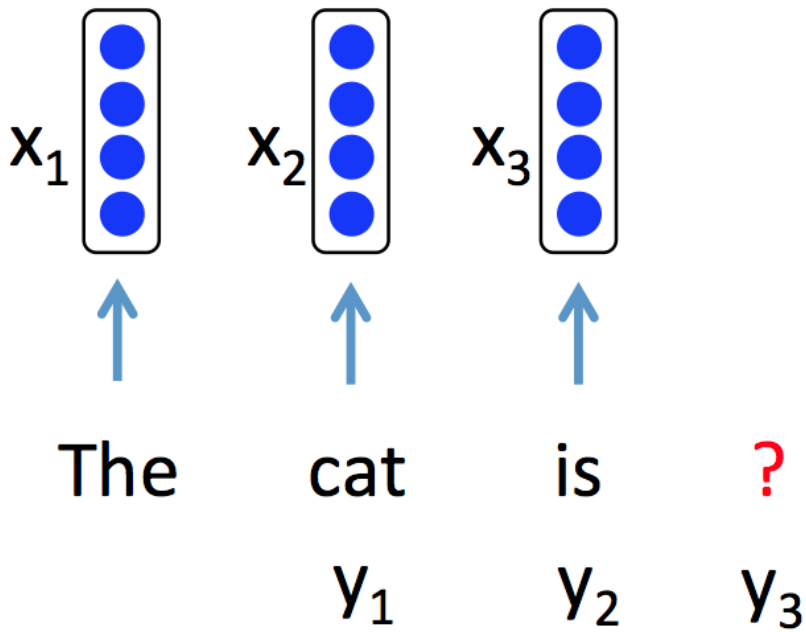
Example: Language Modeling

The cat is ?

x_t: one-hot encoding to represent the word at step t ($[0, \ldots, 0, 1, 0, \ldots, 0]$)

y_t: the next word

$y_t \in \{1, \cdots, V\}$ V: Vocabulary size



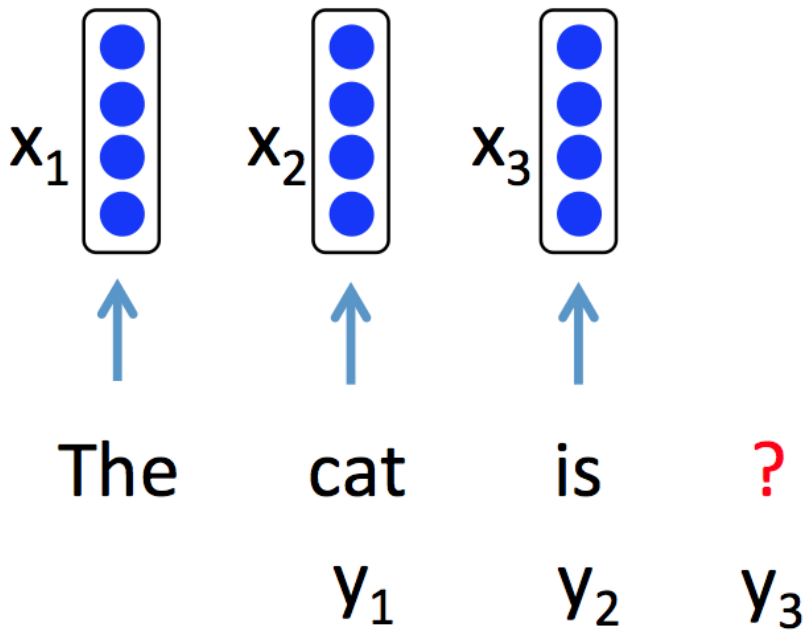
Example: Language Modeling

The cat is ?

$\boldsymbol{x}_t$: one-hot encoding to represent the word at step t ($[0, \ldots, 0, 1, 0, \ldots, 0]$)

y_t: the next word

$y_t \in \{1, \cdots, V\}$ V: Vocabulary size



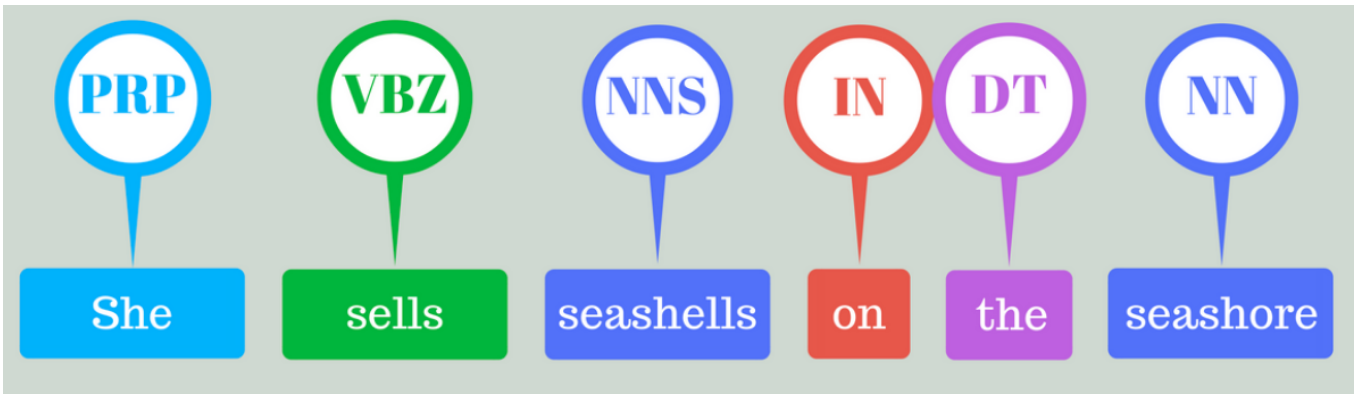
Example: POS Tagging

Part of Speech Tagging:

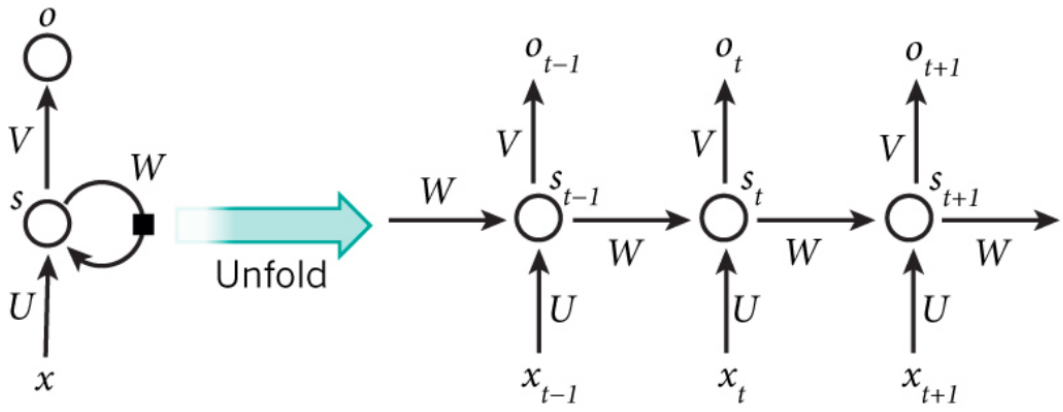
Labeling words with their Part-Of-Speech (Noun, Verb, Adjective, $\cdots$)

$\boldsymbol{x}_t$: a vector to represent the word at step t

y_t: label of word t



Recurrent Neural Network (RNN)



$\boldsymbol{x}_t$: t-th input

$\boldsymbol{s}_t$: hidden state at time t ("memory" of the network)

$$s_t = f(Ux_t + Ws_{t-1})$$

W: transition matrix s_0 usually set to be 0

Predicted output at time t:

$$o_t = \arg\max_i (V\boldsymbol{s}_t)_i$$

Training: Find U, W, V to minimize empirical loss:

Loss of a sequence:

$$\sum_{t=1}^{T} \text{loss}(V\boldsymbol{s}_t, y_t)$$

($\boldsymbol{s}_t$ is a function of U, W, V)

Loss on the whole dataset:

Average loss of all sequence

Solve by Stochastic Gradient Descent (SGD)

Training: Find U, W, V to minimize empirical loss:

Loss of a sequence:

$$\sum_{t=1}^{T} \text{loss}(V\boldsymbol{s}_t, y_t)$$

($\boldsymbol{s}_t$ is a function of U, W, V)

Loss on the whole dataset:

Average loss of all sequence

Solve by Stochastic Gradient Descent (SGD)

Training: Find U, W, V to minimize empirical loss:

Loss of a sequence:

$$\sum_{t=1}^{T} \text{loss}(V\boldsymbol{s}_t, y_t)$$

($\boldsymbol{s}_t$ is a function of U, W, V)

Loss on the whole dataset:

Average loss of all sequence

Solve by Stochastic Gradient Descent (SGD)

RNN: Text Classification

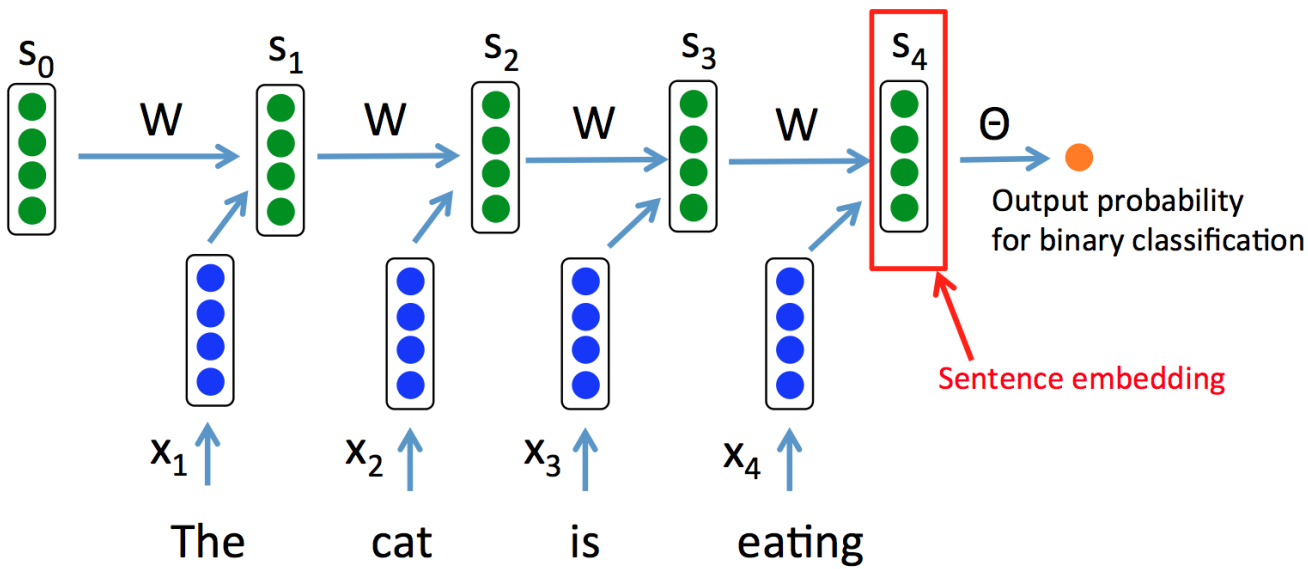
Not necessary to output at each step

Text Classification:

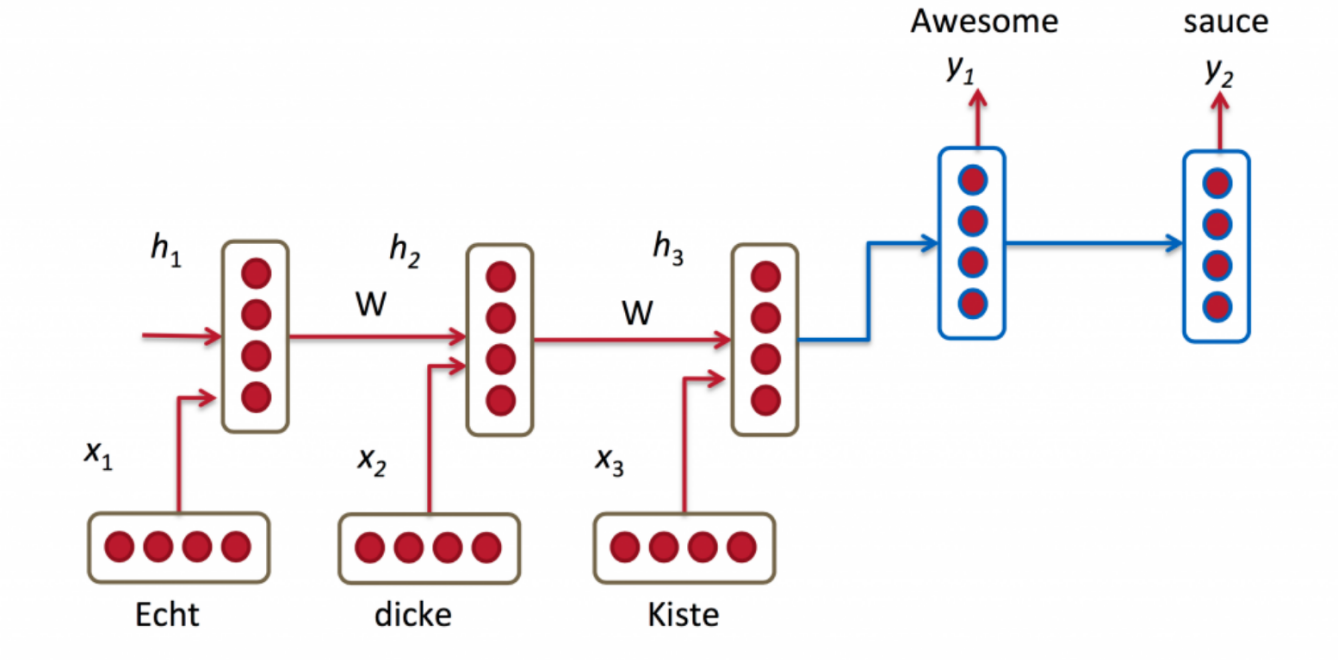
$$\text{Sentence} \rightarrow \text{category}$$

Output only at the final step

Model: add a fully connected network to the final embedding



RNN: Neural Machine Translation



Problems of Classical RNN

Hard to capture long-term dependencies

Hard to solve (vanishing gradient problem)

Solution:

- LSTM (Long Short Term Memory networks)
- GRU (Gated Recurrent Unit)
- ...