Recurrent Neural Network
Time Series/Sequence Data

- **Input:** \( \{x_1, x_2, \cdots, x_T\} \)
  - Each \( x_t \) is the feature at time step \( t \)
  - Each \( x_t \) 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:

- $x_t$: temperature at time $t$
- $y_t$: temperature (or temperature change) at time $t + 1$
Example: Time Series Prediction

- **Climate Data:**
  - $x_t$: 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 ?
Example: Language Modeling

The cat is ?

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

$y_t \in \{1, \ldots, V\}$  \( V: \text{Vocabulary size} \)
Example: POS Tagging

- Part of Speech Tagging:
  Labeling words with their Part-Of-Speech (Noun, Verb, Adjective, …)
- $x_t$: a vector to represent the word at step $t$
- $y_t$: label of word $t$
Recurrent Neural Network (RNN)

- $x_t$: $t$-th input
- $s_t$: 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 $t$:

\[ o_t = \arg \max_i (Vs_t)_i \]
Recurrent Neural Network (RNN)

- **Training**: Find $U, W, V$ to minimize empirical loss:

- **Loss of a sequence**:

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

($s_t$ is a function of $U, W, V$)
Recurrent Neural Network (RNN)

- Training: Find $U, W, V$ to minimize empirical loss:
- Loss of a sequence:
  \[
  \sum_{t=1}^{T} \text{loss}(V s_t, y_t)
  \]
  \(s_t\) is a function of $U, W, V$
- Loss on the whole dataset:
  Average loss of all sequence
Training: Find $U, W, V$ to minimize empirical loss:

Loss of a sequence:

$$\sum_{t=1}^{T} \text{loss}(Vs_t, y_t)$$

($s_t$ 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:
  
  Sentence $\rightarrow$ 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)
- ...
LSTM

RNN:

LSTM:
Conclusions

- Final project.

Questions?