Lecture 8: LSTM for Computer Vision

Areej Alasiry

CIS 6217 – Computer Vision for Data Representation

College of Computer Science, King Khalid University

Outline

1. Recurrent Neural Networks (RNNs)
2. Long Short Term Memory
3. Applications in Computer Vision
4. Language Modeling & Image Captioning
5. Vision and Language Integration
6. Attention Mechanism

Learning Outcomes

- **Explain** the principles of **recurrent neural networks (RNNs)** and their role in modeling sequential data in computer vision.
- **Describe** the internal architecture and gating mechanisms of **Long Short-Term Memory (LSTM)** networks.
- **Analyze** how LSTMs overcome the vanishing gradient problem in sequence learning.
- **Apply** CNN–LSTM architectures to model **temporal and contextual features** in videos and sequential visual data.
- **Construct** encoder–decoder models for **image captioning** and **video description** tasks.
- **Discuss** the role of **attention mechanisms** in enhancing visual–linguistic models.
- **Evaluate** recent advances from LSTM-based models to **transformer-based** vision–language architectures.

What is the motivation behind RNN?

RNN

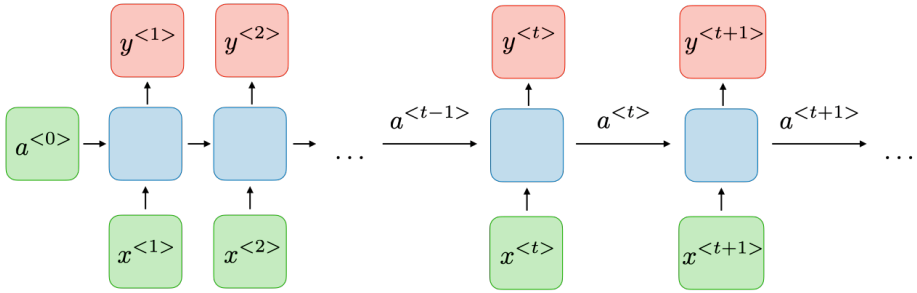
Recurrent Neural Network

RNN Overview

- Handles Sequential data
- Neural networks with feedback connections that retain information from previous inputs

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a$$

$$y^t = g_2(W_{ya}a^{<t>} + b_y)$$

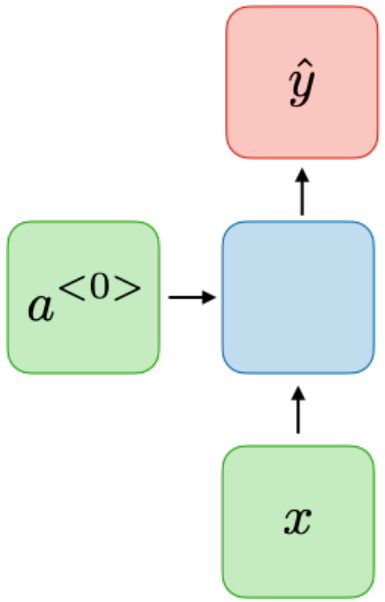


Recurrent neural networks cheatsheet star. CS 230 - Recurrent Neural Networks Cheatsheet. (n.d.).

RNN Architectures

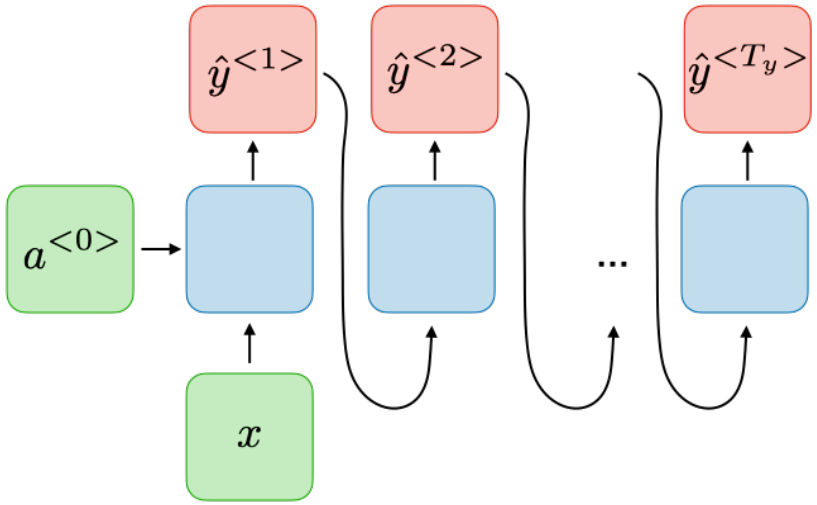
- One-to-One:

$$T_x = T_y = 1$$



- One-to-Many:

$$T_x = 1, T_y > 1$$

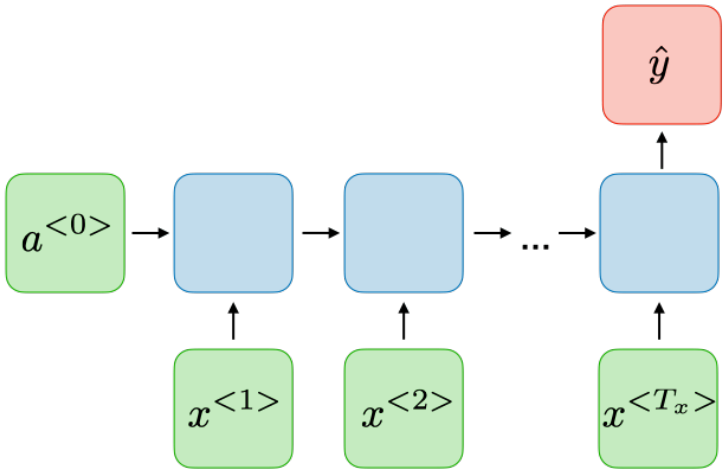


Recurrent neural networks cheatsheet star. CS 230 - Recurrent Neural Networks Cheatsheet. (n.d.).

RNN Architectures

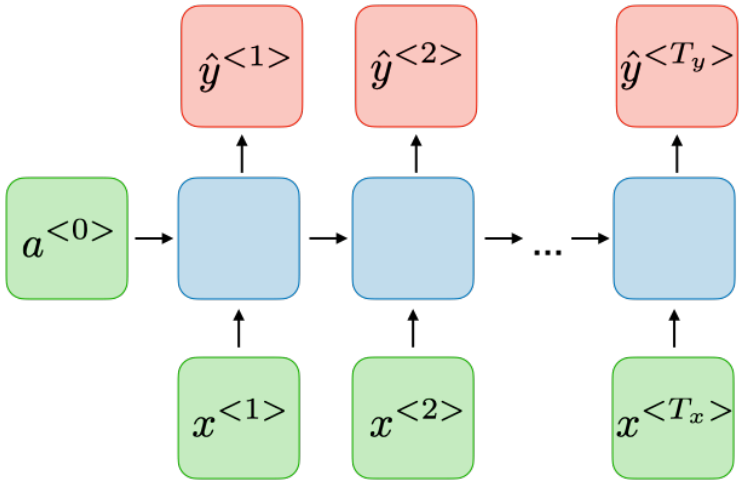
- Many-to-one

$$T_x > 1, T_y = 1$$



- Many-to-Many

$$T_x = T_y$$

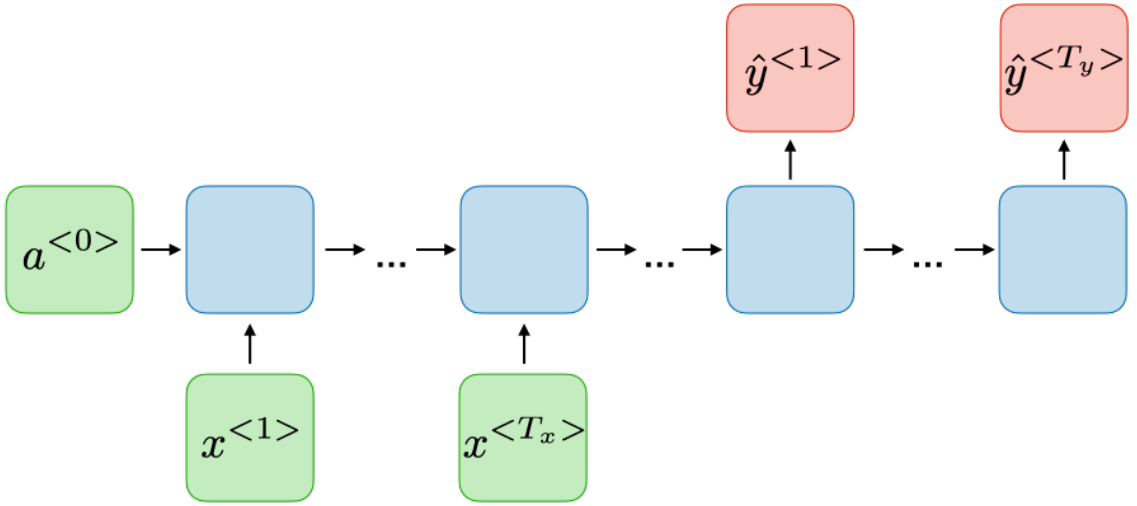


Recurrent neural networks cheatsheet star. CS 230 - Recurrent Neural Networks Cheatsheet. (n.d.).

RNN Architecture

- Many-to-Many

$$T_x \neq T_y$$



Recurrent neural networks cheatsheet star. CS 230 - Recurrent Neural Networks Cheatsheet. (n.d.).

Pros and Cons

- Possibility of processing input of any length
- Model size not increasing with size of input
- Computation takes into account historical information
- Weights are shared across time

- Computation being slow
- Difficulty of accessing information from a long time ago
- Cannot consider any future input for the current state

Challenges of RNN

- Vanishing Gradient
- Exploding Gradient

LSTM

Long Short Term Memory

LSTM Overview

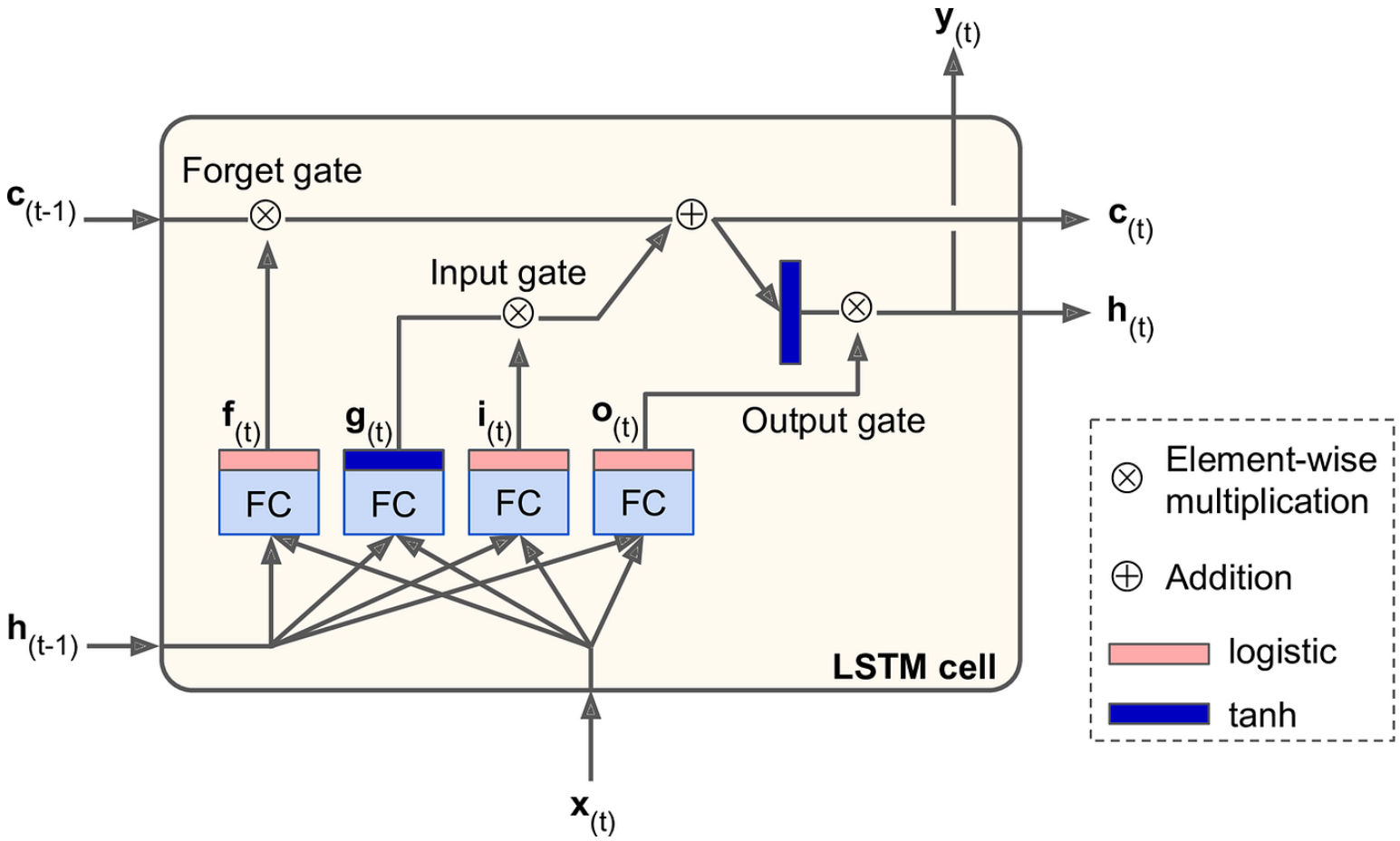
LSTMs are designed to handle the vanishing gradient problem by introducing a memory cell that can maintain information over long periods.

Components:

- **Cell State**: Stores long-term dependencies.
- **Gates**:
  - **Input Gate**: Controls how much new information flows into the cell.
  - **Forget Gate**: Decides what information to discard from the cell.
  - **Output Gate**: Determines what information to output.

Use-Case: Ideal for tasks requiring long-term memory, like essay writing or speech recognition.

LSTM Architecture



Source: Neural Networks and deep learning, **by Aurélien Géron** Oriely

Applications

Sequence Learning in Vision

- Video classification: sequence of frames → label (e.g., “playing guitar”).
- CNN extracts features per frame.
- LSTM models temporal evolution
  - $ht=LSTM(CNN(framet),ht-1)$
- Applications:
  - Human action recognition
  - Emotion detection
  - Object tracking over time

Image Captioning

- Goal: Generate a natural language description of an image.
- Pipeline:
  - CNN: extracts image features.
  - LSTM: generates sentence word-by-word.
  - Combined via feature vector → language decoder.

Vision + Language

- Combines visual and linguistic embeddings.
- Two networks:
  - Vision encoder (CNN, ViT)
  - Language encoder (RNN, LSTM, Transformer)
- Applications:
  - Visual Question Answering (VQA)
  - Image Retrieval by Text
  - Multimodal Translation

Attention Mechanism

Attention Mechanism

- LSTMs struggle with long sequences
- Core idea: focus on relevant parts of the input at each time step.
- Enables model to “look” at specific image regions when generating each word.

A cute teddy bear is reading Persian literature

A cute teddy bear is reading Persian literature

Encoder–Decoder Architecture

- Encoder (CNN): Encodes image into feature vector.
- Decoder (RNN/LSTM/Attention): Generates textual sequence.
- Key use: Image Captioning, VQA, Video Description.

References

- Guide to CNNs for CV – Khan et al. (2018)
- Deep Learning with Python – Chollet (2018)
- Deep Learning in Computer Vision – Awad & Hassaballah (2020)

- Deep Learning for Vision Systems by Mohamed Elgendy (2020)