Text Recognition với CRNN và CTC

CS331.P21 - Thị giác máy tính nâng cao

Trương Huỳnh Thúy An Hoàng Đức Chung Nguyễn Hải Đăng

Khoa Khoa học Máy tính
Trường Đại học Công nghệ Thông tin, ĐHQG-HCM

Thứ Ba, ngày 20 tháng 5, 2025

Overview

1. CNN & RNN
2. CRNN
3. CTC

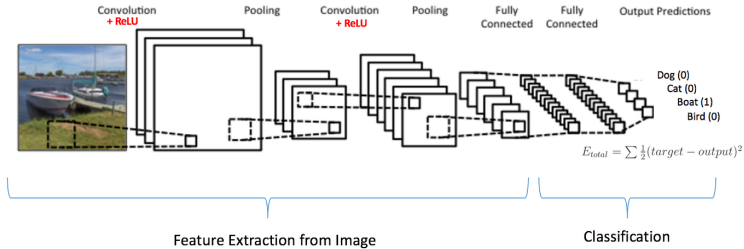
Overview

1. CNN & RNN
2. CRNN
3. CTC

CNN - Convolutional Neural Network

CNN là kiến trúc phổ biến để trích xuất đặc trưng từ ảnh.

- Sử dụng các lớp tích chập để phát hiện đặc trưng không gian.
- Giảm chiều và trích lọc ảnh, giữ đặc trưng quan trọng.
- Đầu ra là các Feature Maps, dùng làm đầu vào cho các bước xử lý sau.

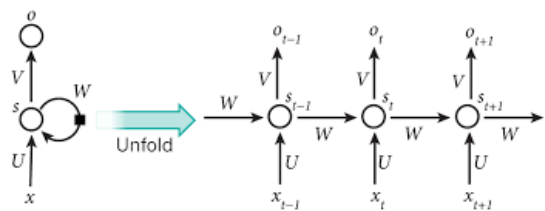


Ảnh: Mô phỏng quá trình trích xuất đặc trưng của CNN

RNN - Recurrent Neural Network

RNN là mạng nơ-ron lý tưởng cho dữ liệu tuần tự (chuỗi), như văn bản, âm thanh, chuỗi thời gian.

- Ghi nhớ trạng thái trước đó qua mỗi bước thời gian.
- Mỗi đầu ra phụ thuộc vào cả đầu vào hiện tại và trạng thái trước đó.
- Gặp khó khăn khi xử lý chuỗi dài do hiện tượng *vanishing gradient*.



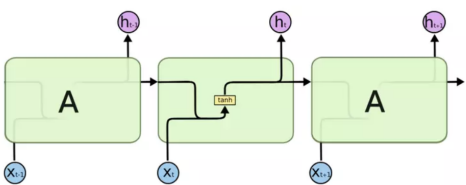
Ảnh: Kiến trúc cơ bản của RNN

Giải pháp: LSTM và các biến thể như GRU, BiLSTM.

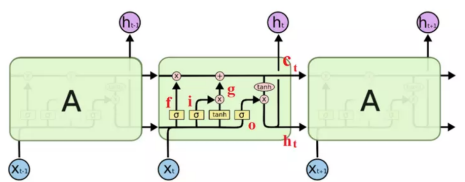
LSTM - Long Short-Term Memory

LSTM là một cải tiến quan trọng của RNN giúp mô hình ghi nhớ lâu hơn, khắc phục vấn đề *vanishing gradient*.

- Có cấu trúc gồm 3 "cửa" (gates): **forget gate**, **input gate**, **output gate**.
- Sử dụng *cell state* để truyền thông tin qua các bước thời gian mà không bị thay đổi nhiều.

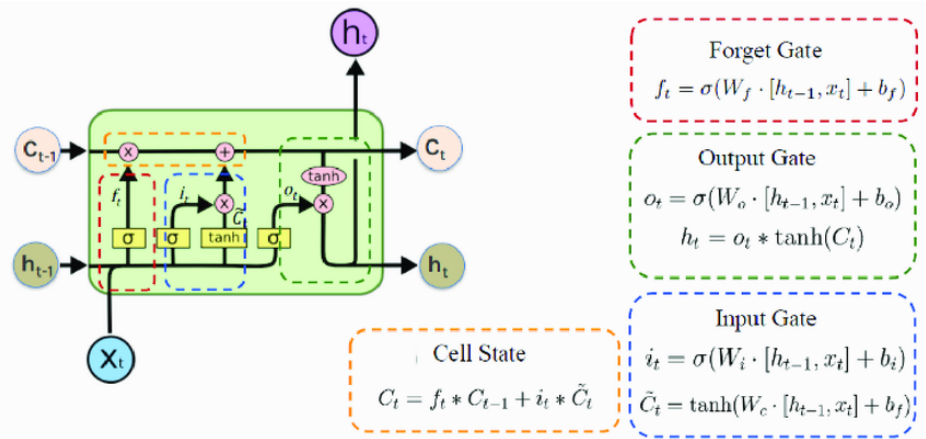


RNN: Đơn giản, dễ gặp vanishing gradient khi chuỗi dài



LSTM: Có bộ nhớ dài hạn, chống được vanishing gradient

LSTM - Kiến trúc



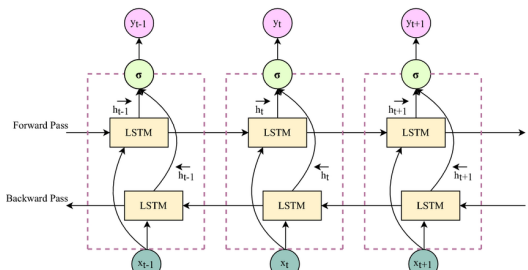
Ảnh: Kiến trúc một khối LSTM

Bidirectional LSTM (BiLSTM)

BiLSTM là biến thể của LSTM giúp khai thác ngữ cảnh cả quá khứ và tương lai.

- Bao gồm hai LSTM chạy theo hai chiều: tiến (forward) và lùi (backward).
- Kết quả đầu ra là sự kết hợp của cả hai chiều → giàu ngữ cảnh hơn.
- Rất hiệu quả trong các tác vụ như phân loại chuỗi, gán nhãn chuỗi,...

Nhược điểm: Tăng chi phí tính toán, không phù hợp với xử lý thời gian thực.



Ảnh: Kiến trúc BiLSTM

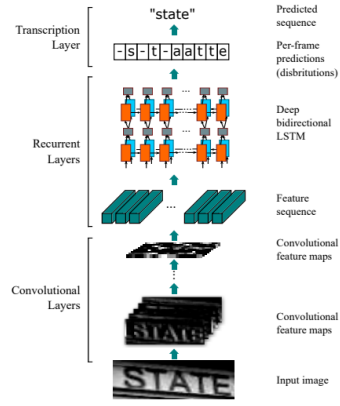
Overview

1. CNN & RNN
2. **CRNN**
3. CTC

CRNN - Convolutional Recurrent Neural Network

CRNN xuất hiện trong *An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition* [1], và đến nay vẫn là một trong những mô hình hiệu quả nhất cho bài toán OCR.

- Kết hợp giữa CNN và RNN.
- Không cần tách ký tự (character segmentation-free).
- Huấn luyện end-to-end với hàm mất mát CTC.



Ảnh: Kiến trúc CRNN cơ bản

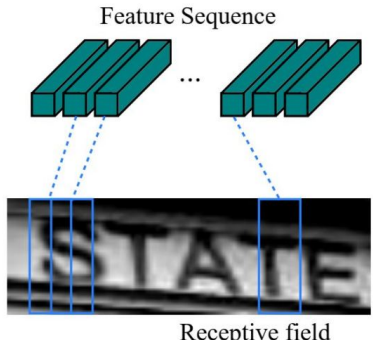
Convolutional Layers trong CRNN

Mục tiêu: Trích xuất đặc trưng không gian từ ảnh đầu vào.

Đầu vào: Ảnh văn bản (Text Image)

- Ảnh được đưa qua các lớp tích chập (CNN).
- Kết quả là các **Feature Maps** chứa thông tin không gian.
- Mỗi **cột** trong Feature Map ứng với một **Feature Vector**.

Đầu ra: Chuỗi đặc trưng theo chiều ngang ảnh (*Feature Sequence*)



Ảnh: Chuyển ảnh thành chuỗi đặc trưng qua CNN

Recurrent Layers trong CRNN

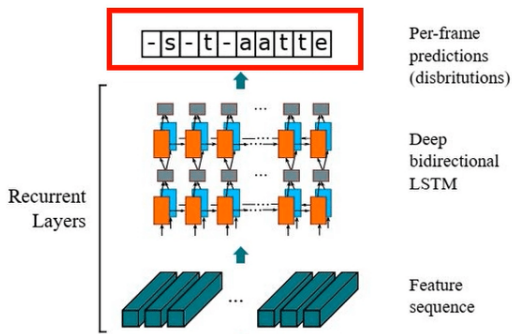
Mục tiêu: Xử lý chuỗi đặc trưng theo thứ tự để tạo chuỗi ký tự.

Đầu vào: Chuỗi đặc trưng từ CNN (*Feature Sequence*)

- Feature Sequence được đưa qua các lớp BiLSTM.
- Mỗi TimeStep tạo ra một vector xác suất trên không gian ký tự.

Đầu ra: Dãy ký tự dự đoán, nhưng:

- Có thể trùng lặp ký tự.
- Có thể thiếu ký tự.
- Có ký tự rỗng (blank).

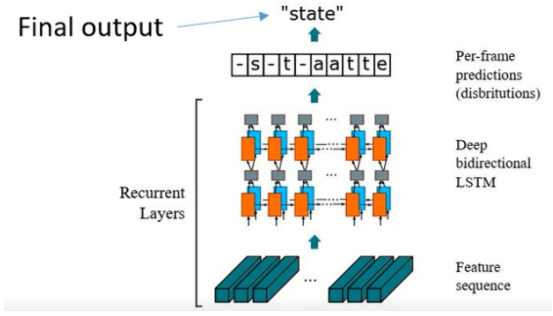


Ảnh: Dự đoán chuỗi ký tự từ Feature Sequence

Transcription Layers trong CRNN

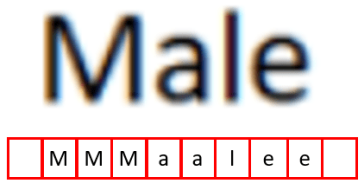
Mục tiêu: Chuyển đầu ra của RNN thành chuỗi văn bản hoàn chỉnh.

- RNN output: `-hh-ee-ll-ll-oo-`
- Transcription: `hello`



Transcription Layers - Một số hướng tiếp cận

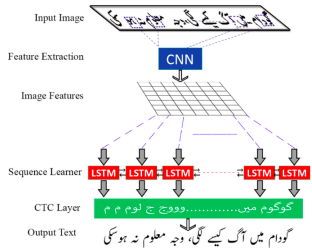
- **Cách 1 – Chia đều ký tự theo TimeStep:**
  - Ví dụ: "STATE" $\rightarrow$ [S, S, T, T, A, A, T, T, E, E]
  - Đơn giản, nhưng không linh hoạt với độ dài và font khác nhau.
- **Cách 2 – Gán nhãn thủ công từng TimeStep:**
  - Huấn luyện mô hình với nhãn cụ thể cho mỗi TimeStep.
  - Chính xác nhưng tốn thời gian tạo dữ liệu.



Ảnh: Gán nhãn ký tự cho từng TimeStep (Cách 2)

Transcription Layers - Một số hướng tiếp cận

- **Cách 3 – Connectionist Temporal Classification (CTC)**
  - Tự động ánh xạ chuỗi xác suất tại mỗi TimeStep thành chuỗi ký tự cuối cùng.
  - Cho phép trùng lặp ký tự và thêm ký tự rỗng (blank) để tăng tính linh hoạt.
  - Không cần gán nhãn theo từng TimeStep, phù hợp với chuỗi có độ dài biến đổi.



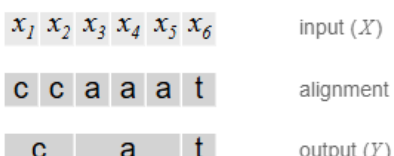
→ **CTC là giải pháp phổ biến nhất cho OCR và Speech Recognition.**

Overview

1. CNN & RNN
2. CRNN
3. **CTC**

CTC – Connectionist Temporal Classification

- CTC là một **Loss Function** dùng trong các mô hình Deep Learning như CRNN.
- Mục tiêu: tìm cách ánh xạ chuỗi đầu vào X (chuỗi đặc trưng ảnh) với chuỗi đầu ra Y (văn bản).
- Không yêu cầu dữ liệu **gán nhãn theo từng TimeStep**, chỉ cần ảnh và text tương ứng.



CTC cho phép ký tự lặp và ký tự rỗng để hỗ trợ việc ánh xạ từ TimeStep sang chuỗi ký tự.

CTC – Nguyên lý hoạt động

- CTC không biết rõ cách align giữa X và Y, nhưng nó tạo ra tất cả **khả năng hợp lệ** từ X để tạo thành Y.
- Mỗi khả năng đi kèm với một xác suất.
- Quá trình huấn luyện tối ưu xác suất sao cho **chuỗi đầu ra gần với nhãn Ground Truth nhất**.
- Đây là cách tiếp cận phổ biến cho **OCR** và **Speech Recognition**.

CTC hoạt động qua 3 bước chính: Encoding Text, Loss Calculation và Decoding Text.

Bước 1 – Encoding Text

- Mỗi TimeStep **tương ứng với một ký tự**, nhưng thực tế có thể lặp ký tự: `ttiien ssuu` ⇒ `tien su`.
- CTC sẽ **gộp các ký tự lặp** lại thành một.
- Với từ có ký tự lặp thật sự (`hello`, `coffee`...), CTC dùng ký tự đặc biệt **blank** (ký hiệu là "-") để phân tách.
- Ví dụ: `meet` ⇒ `mm-ee-ee-t` ⇒ giữ được cả hai ký tự e.

Ghi nhớ: blank là cách CTC phân biệt "ký tự lặp" thật sự với lỗi do TimeStep.

Bước 2 – Loss Calculation

Mục tiêu: Tính xác suất để chuỗi đầu ra của mô hình sinh đúng GT Text qua tất cả các cách align hợp lệ.

Quy trình:

- Mỗi sample là 1 cặp (ảnh, GT Text), ví dụ: "a", "ab".
- Mô hình CRNN tạo ra Score Matrix (ma trận xác suất tại mỗi TimeStep).
- Với mỗi GT Text, có nhiều cách align hợp lệ do CTC cho phép lặp lại và thêm ký tự blank.
- Loss được tính bằng công thức:

$$\text{Loss} = -\log_{10}(\text{Tổng xác suất các align sinh đúng GT Text})$$

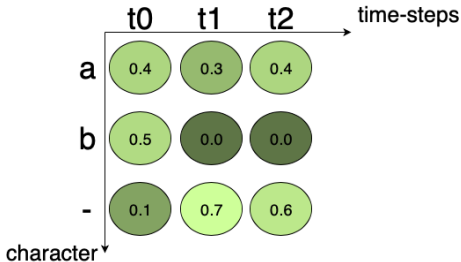
- Nếu có nhiều GT Text (nhiều chuỗi cần khớp), **Tổng Loss** sẽ bằng tổng các Loss của từng chuỗi:

$$\text{Tổng Loss} = \sum_{i=1}^{n} \text{Loss}_i$$

Loss sẽ được tối ưu qua Backpropagation và SGD trong quá trình huấn luyện.

Ví dụ minh họa tính Loss CTC

Cho: GT Text = "a", Score Matrix gồm 3 TimeSteps như hình:



Các alignment hợp lệ sinh ra "a" gồm:

- aaa: $0.4 \times 0.3 \times 0.4 = 0.048$ a–: $0.4 \times 0.7 \times 0.6 = 0.168$ a-: $0.4 \times 0.7 = 0.28$
- aa-: $0.4 \times 0.3 \times 0.6 = 0.072$ -aa: $0.1 \times 0.3 \times 0.4 = 0.012$ –a: $0.1 \times 0.7 \times 0.4 = 0.028$

Tổng xác suất:

$$0.048 + 0.168 + 0.28 + 0.072 + 0.012 + 0.028 = 0.608$$

Tính Loss: $-\log(0.608) \approx 0.216$

Ví dụ minh họa tính Loss CTC

Cho: GT Text = "b" và GT Text = blank, dùng cùng Score Matrix như slide trước.

Các alignment hợp lệ sinh "b" gồm:

- bbb: $0.5 \times 0.0 \times 0.0 = 0$ b–: $0.5 \times 0.7 \times 0.6 = 0.21$ b-: $0.5 \times 0.7 = 0.35$
- bb-: $0.5 \times 0.0 \times 0.6 = 0$ -bb: $0.1 \times 0.0 \times 0.0 = 0$ –b: $0.1 \times 0.7 \times 0.0 = 0$

Tổng xác suất cho "b":

$$0 + 0.21 + 0.35 + 0 + 0 + 0 = 0.56 \quad \Rightarrow \quad \text{Loss} = -\log(0.56) \approx 0.25$$

Các alignment hợp lệ sinh ra ký tự blank:

- –: $0.1 \times 0.7 = 0.07$ —: $0.1 \times 0.7 \times 0.6 = 0.042$

Tổng xác suất cho blank:

$$0.07 + 0.042 = 0.112 \quad \Rightarrow \quad \text{Loss} = -\log(0.112) \approx 0.95$$

Ví dụ minh họa tính Loss CTC

Giả sử ảnh có 3 GT Text: "a", "b", và ký tự blank.

- Loss với "a": $-\log(0.608) \approx 0.216$
- Loss với "b": $-\log(0.56) \approx 0.25$
- Loss với blank: $-\log(0.112) \approx 0.95$

Tổng Loss cho ảnh:

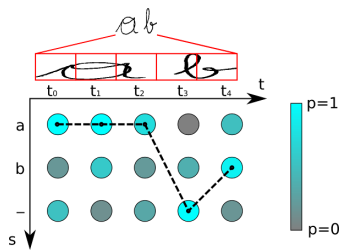
$$\text{Loss}_{\text{total}} = 0.216 + 0.25 + 0.95 = 1.416$$

Mỗi chuỗi GT Text được tính Loss riêng biệt, sau đó cộng lại để ra Loss tổng thể cho ảnh trong batch.

Bước 3 – Decoding Text

Mục tiêu: Dự đoán chuỗi văn bản từ ảnh chưa thấy (Unseen Image).

- Tìm đường đi có **xác suất cao nhất** trong Score Matrix.
- Loại bỏ các ký tự **trống** (blank) và **ký tự trùng nhau**.
- Ví dụ: Chuỗi `a a a - b` $\Rightarrow$ `ab`



Decoding giúp chuyển từ chuỗi xác suất sang văn bản đầu ra hợp lệ.

Demo

Tài liệu tham khảo

- B. Shi, X. Bai, và C. Yao, *An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition*, arXiv preprint arXiv:1507.05717. [Link bài báo]
- TheAILearner, *CTC – Problem Statement*. [Link bài viết]
- Siddhant, *Explanation of Connectionist Temporal Classification*. [Link bài viết]