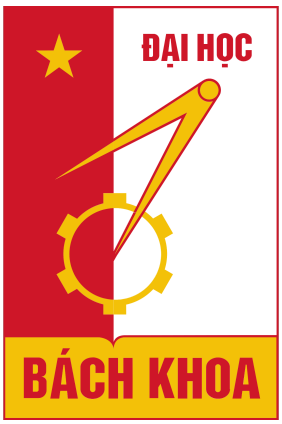
ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Báo cáo BTL Nhập môn Trí tuệ nhân tạo

Nhận diện chữ viết tay

Giảng viên hướng dẫn: Thầy Trần Thế Hùng

Phạm Hải Nam Anh
Trần Đức Hoàng Nam
Bùi Thị Khánh Huyền
Hà Tuấn Hoàng
Đào Sỹ Phúc

Ngày 8 tháng 6 năm 2024

1 Giới thiệu bài toán

Nhận diện chữ viết tay là một trong những lĩnh vực nghiên cứu quan trọng trong ngành trí tuệ nhân tạo và xử lý hình ảnh. Bài toán này nhằm mục đích chuyển đổi văn bản viết tay thành dạng văn bản số mà máy tính có thể hiểu và xử lý được. Đây là một bài toán thách thức do tính đa dạng và phức tạp của chữ viết tay, bao gồm các biến thể về hình dạng, kích thước, và phong cách viết của từng cá nhân.

Bài toán nhận diện chữ viết tay có ứng dụng rộng rãi trong nhiều lĩnh vực như tự động hóa nhập liệu, phân loại tài liệu, và hỗ trợ người khuyết tật. Với sự phát triển của công nghệ học sâu (Deep Learning), các mô hình mạng nơ-ron tích chập (Convolutional Neural Network - CNN), mạng nơ-ron hồi quy (Recurrent Neural Network - RNN), và đặc biệt là mạng LSTM (Long Short-Term Memory) đã cho thấy hiệu quả cao trong việc nhận diện chữ viết tay.

Trong báo cáo này, chúng em sẽ trình bày về phương pháp sử dụng mạng CRNN (Convolutional Recurrent Neural Network) kết hợp với hàm mất mát CTC (Connectionist Temporal Classification) và mạng LSTM để giải quyết bài toán nhận diện chữ viết tay. Sự kết hợp này giúp mô hình có thể học và nhận diện các chuỗi ký tự trong văn bản viết tay một cách chính xác và hiệu quả.

2 Tập dữ liệu sử dụng

2.1 Nguồn gốc dữ liệu

Tập dữ liệu sử dụng trong dự án này được lấy từ trang web Kaggle:

- Handwriting Recognition

2.2 Cấu trúc dữ liệu

Tập dữ liệu này bao gồm các thư mục chứa hình hình ảnh chữ viết tay và các tệp .csv (dạng bảng) tương ứng. Cụ thể như sau:

- **Hình ảnh**: Các hình ảnh được lưu trữ ở định dạng JPEG, đặt tên theo cú pháp thư mục_số thứ tự.jpg, chứa chữ viết tay là tên người.
- **File .csv**: Mỗi tệp gồm 2 cột là 'FILENAME' và 'IDENTITY'. Cột 'FILENAME' chứa tên hình ảnh và cột 'IDENTITY' chứa nhãn viết tay tương ứng có trong hình ảnh.

2.3 Số lượng và chất lượng dữ liệu

Tập dữ liệu này bao gồm hơn bốn trăm nghìn tên viết tay được thu thập thông qua các dự án từ thiện. Cụ thể:

- **Tổng số lượng dữ liệu**: Tập dữ liệu này có tổng cộng 413,823 hình ảnh chữ viết tay, bao gồm 206,799 tên riêng và 207,024 họ.

- **Phân chia dữ liệu**: Dữ liệu được chia thành ba tập chính:
  - Tập train: 331,059 ví dụ
  - Tập test: 41,382 ví dụ
  - Tập validate: 41,382 ví dụ

  Một ví dụ bao gồm cả hình ảnh và nhãn của nó

2.4 Tiền xử lý dữ liệu

Việc phân tích và tiền xử lý dữ liệu giúp thu được những đặc trưng của bộ dữ liệu, đảm bảo rằng mô hình được train trên một tập dữ liệu sạch và cân bằng, từ đó cải thiện độ chính xác và hiệu quả của hệ thống nhận diện chữ viết tay. Đầu tiên, phân tích tập dữ liệu, ta có những thống kê như sau:

- **Những ví dụ có nhãn trong cột "IDENTITY" mang giá trị rỗng**:
  - Tập Train: 565
  - Tập Validation: 78
- **Số giá trị riêng biệt của nhãn "IDENTITY" trong tập Train**: 100539
- **Số ví dụ không thể đọc (mang nhãn "UNREADABLE")**: 102
- **Số ví dụ rỗng (mang nhãn "EMPTY")**: 1796
- **Số ví dụ được đánh nhãn bằng chữ in thường**: 17

Sau đó, ta sẽ tiến hành loại bỏ những ví dụ có nhãn không thể đọc, rỗng, và sửa nhãn những ví dụ in thường thành in hoa. Việc quy về nhãn nhãn in thường giúp giảm kích cỡ "từ điển" những kí tự cần dự đoán.
Để đảm bảo các hình ảnh có kích thước cố định và được chuẩn hóa, giúp mô hình học sâu có thể tiếp nhận và xử lý dữ liệu một cách hiệu quả, ta tiến hành

- Chuyển đổi hình ảnh sang mảng NumPy giúp dễ dàng thao tác trên hình ảnh.
- Cắt hình ảnh nếu kích thước lớn hơn giới hạn đảm bảo tất cả các hình ảnh có kích thước phù hợp.
- Xoay hình ảnh để phù hợp với định dạng đầu vào của model.
- Chuyển đổi sang tensor PyTorch và chuẩn hóa giá trị pixel về [0, 1] để chuẩn bị hình ảnh nạp vào model.

Đồng thời, ta cần mã hóa nhãn để dễ dàng sử dụng các nhãn trong quá trình huấn luyện mô hình.

- **Từ điển mã hóa**: `alphabet = u"ABCDEFGHIJKLMNOPQRSTUVWXYZ-'' "`

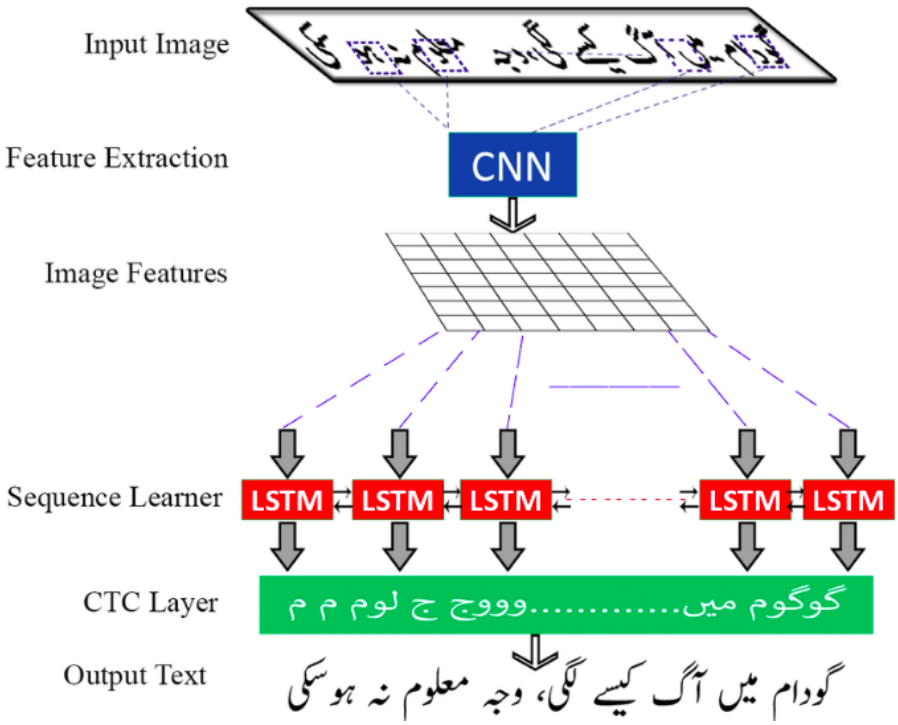
- **Thực hiện**: Chuyển đổi mỗi ký tự thành index của dãy alphabet

Với các chuỗi nhãn có độ dài khác nhau, để đảm bảo rằng các nhãn có cùng độ dài, ta tiến hành đệm (pad)

- Tìm độ dài tối đa của nhãn trong lô dữ liệu để xác định kích thước đệm cần thiết.
- Thêm các ký tự <BLANK> còn thiếu để các nhãn có cùng độ dài

Cuối cùng, ta sẽ tạo các lô dữ liệu một cách ngẫu nhiên và quản lý việc nạp dữ liệu vào mô hình trong quá trình huấn luyện.

3 Phương pháp



Hình 1: Kiến trúc CRNN + CTC

Để giải quyết bài toán Nhận Dạng Chữ Viết Tay, nhóm chúng em quyết định chọn kiến trúc CRNN (Convolutional Recurrent Neural Networt) kết hợp với CTC (Connectionist Temporal Classification). Sự kết hợp giữa CRNN và CTC

được sử dụng rộng rãi trong các bài toán nhận dạng viết tay do một số ưu điểm chính giúp giải quyết các thách thức cụ thể của bài toán này:

1. **Khả năng train từ đầu đến cuối:** CRNN + CTC cho phép train toàn diện từ đầu vào hình ảnh thô đến đầu ra văn bản cuối cùng. Điều này có nghĩa là toàn bộ mô hình có thể được train trong một lần mà không yêu cầu các bước trung gian hoặc các mô hình riêng biệt cho các giai đoạn khác nhau, đơn giản hóa quy trình train và giảm độ phức tạp tổng thể.

2. **Xử lý các chuỗi có độ dài thay đổi:** Nhận dạng văn bản viết tay bao gồm các chuỗi ký tự có thể có độ dài khác nhau rất nhiều. CRNN có thể xử lý các chuỗi có độ dài thay đổi này một cách hiệu quả bằng cách sử dụng các lớp chập để trích xuất bản đồ đặc trưng từ hình ảnh và các lớp lặp lại để xử lý tính chất tuần tự của dữ liệu. CTC tiếp tục căn chỉnh các trình tự này, làm cho mô hình trở nên mạnh mẽ với các độ dài đầu vào khác nhau.

3. **Không cần phân đoạn trước:** Các phương pháp OCR (Nhận dạng ký tự quang học) truyền thống thường yêu cầu phân đoạn trước văn bản thành các ký tự riêng lẻ, việc này vừa tốn công sức vừa dễ xảy ra lỗi. CTC loại bỏ yêu cầu này bằng cách căn chỉnh các chuỗi ký tự được dự đoán với các chuỗi hình ảnh đầu vào, cho phép mô hình học cách phân đoạn ngầm trong quá trình đào tạo.

4. **Tính linh hoạt và mạnh mẽ:** Kiến trúc của CRNN, với các lớp tích chập theo sau là các lớp lặp lại, nắm bắt cả các đặc điểm không gian và sự phụ thuộc theo thời gian. Điều này làm cho nó đặc biệt hiệu quả trong việc xử lý sự thay đổi trong chữ viết tay, chẳng hạn như các kiểu, độ nghiêng và khoảng cách khác nhau giữa các ký tự. CTC bổ sung cho điều này bằng cách căn chỉnh trình tự đầu ra mà không cần ánh xạ một-một chính xác, xử lý sự không nhất quán và bất thường trong chữ viết tay.

5. **Hiểu biết theo ngữ cảnh:** Các lớp hồi quy hai chiều trong CRNN cho phép mô hình sử dụng bối cảnh từ cả quá khứ và tương lai trong một chuỗi. Điều này rất quan trọng để hiểu và diễn giải chính xác văn bản viết tay trong đó bối cảnh do các ký tự xung quanh cung cấp có thể phân biệt các hình dạng khó hiểu.

6. **Chú thích dữ liệu đơn giản:** Với CTC, yêu cầu về chú thích chi tiết, chẳng hạn như phân đoạn cấp độ ký tự, sẽ bị loại bỏ. Dữ liệu huấn luyện chỉ cần nhãn cấp độ trình tự (văn bản thực tế trong ảnh), dễ dàng lấy hơn và giảm đáng kể công sức cũng như chi phí chuẩn bị dữ liệu.

7. **Độ chính xác và hiệu suất cao:** Sự kết hợp giữa CRNN và CTC đã được chứng minh là đạt được độ chính xác cao trong việc nhận dạng văn bản viết tay, vượt trội so với nhiều phương pháp truyền thống. Khả năng học hỏi và khái quát hóa của kiến trúc từ các phong cách viết tay và bối cảnh khác nhau góp phần mang lại hiệu suất vượt trội cho nó.

3.1 Kiến trúc mạng CRNN

Convolutional Recurrent Neural Network (CRNN) là một mô hình học sâu phức tạp, đặc biệt phù hợp cho các tác vụ liên quan đến dữ liệu tuần tự, chẳng hạn như nhận dạng văn bản viết tay. CRNN tận dụng các điểm mạnh của **Convolutional Neural Network (CNN)** và **Recurrent Neural Network (RNN)** để xử lý hiệu quả các phụ thuộc về không gian và thời gian vốn có trong các chuỗi viết tay. Kiến trúc bắt đầu bằng một loạt các lớp tích chập, có khả năng nắm bắt các mô hình cục bộ và phân cấp không gian trong dữ liệu hình ảnh một cách thành thạo. Các lớp này trích xuất các tính năng cấp cao từ hình ảnh đầu vào, chuyển đổi hiệu quả dữ liệu pixel thô thành cách trình bày trừu tượng và nhiều thông tin hơn. Theo sau các lớp tích chập, các tính năng được trích xuất sau đó sẽ được xử lý bởi các lớp lặp lại, thường sử dụng các đơn vị **Long Short-Term Memory (LSTM)**, vượt trội trong việc mô hình hóa các phụ thuộc tầm xa và trình tự thời gian. Quá trình xử lý tuần tự này cho phép CRNN duy trì và sử dụng thông tin theo ngữ cảnh trong toàn bộ chiều dài của chuỗi đầu vào, điều này rất quan trọng để diễn giải chính xác văn bản viết tay có nhiều kiểu dáng, khoảng cách và căn chỉnh khác nhau. Giai đoạn cuối cùng của kiến trúc CRNN thường bao gồm lớp phiên mã, chẳng hạn như **Connectionist Temporal Classification (CTC)**, giúp dịch đầu ra tuần tự của RNN thành văn bản có thể đọc được bằng cách căn chỉnh các chuỗi ký tự được dự đoán với hình ảnh đầu vào. Mô hình có thể huấn luyện từ đầu đến cuối này không chỉ nâng cao độ mạnh mẽ và độ chính xác của hệ thống nhận dạng mà còn đơn giản hóa quá trình huấn luyện bằng cách loại bỏ nhu cầu về các ký tự được phân đoạn trước. Về bản chất, CRNN thể hiện một cách tiếp cận mạnh mẽ và linh hoạt để nhận dạng văn bản viết tay, kết hợp khả năng trích xuất đặc điểm không gian của CNN với khả năng mô hình hóa trình tự của RNN, từ đó giải quyết hiệu quả sự phức tạp của các đầu vào viết tay có độ dài thay đổi và phụ thuộc vào ngữ cảnh. Sự tích hợp các thành phần tích chập và lặp lại trong một khuôn khổ thống nhất này khiến CRNN trở thành nền tảng trong lĩnh vực nhận dạng ký tự quang học (OCR), thúc đẩy những tiến bộ trong cả nghiên cứu học thuật và ứng dụng thực tế.

3.2 CTC Loss

Connectionist Temporal Classification (CTC) là một thuật toán được thiết kế để giải quyết vấn đề về sự không tương thích về độ dài giữa chuỗi đầu vào và chuỗi đầu ra trong các mô hình nhận diện chuỗi, đặc biệt hữu ích trong bài toán nhận diện chữ viết tay.

Trong bài toán này, chuỗi đầu vào là ảnh chụp chữ viết tay, thường có độ dài khác nhau và được biểu diễn dưới dạng ma trận pixel, trong khi chuỗi đầu ra là văn bản tương ứng, thường có độ dài ngắn hơn và có định dạng khác biệt so với ảnh đầu vào. Điều này dẫn đến vấn đề là mô hình có thể nhận diện sai và lặp lại các ký tự khi chiều dài của chúng trong ảnh chụp quá lớn.

CTC giải quyết vấn đề này bằng cách thêm các ký tự khoảng trống (blank) vào chuỗi đầu ra thu được từ mạng CRNN (Convolutional Recurrent Neural

Network). Cụ thể, trước khi dự đoán các ký tự đầu ra, hình ảnh đầu vào được chia thành nhiều khung nhỏ theo chiều ngang. Mỗi khung chứa một phần của hình ảnh và có thể bao gồm khoảng trắng hoặc một số ký tự.

Mạng RNN (Recurrent Neural Network) sau đó sẽ xử lý từng khung hình và đưa ra dự đoán xác suất của từng ký tự hoặc khoảng trắng cho mỗi khung. Chuỗi kết quả từ RNN là một chuỗi các khung hình với xác suất dự đoán cho mỗi ký tự mà nó chứa.

CTC hoạt động như một hàm mất mát (Loss function) để tính toán sự khác biệt giữa chuỗi ký tự đã dự đoán và chuỗi ký tự mục tiêu. Nó thực hiện điều này bằng cách xem xét tất cả các chuỗi khả dĩ có thể ánh xạ từ chuỗi đầu vào sang chuỗi đầu ra mục tiêu, và sau đó tối ưu hóa xác suất của chuỗi mục tiêu thực sự trong không gian tất cả các chuỗi khả dĩ.

Để cụ thể hơn, CTC sử dụng một kỹ thuật gọi là "labeling with blanks" (gán nhãn với khoảng trắng). Trong đó, các ký tự dự đoán có thể bao gồm các khoảng trắng ở giữa, giúp mô hình có khả năng bỏ qua những phần không liên quan trong chuỗi đầu vào. Ví dụ, nếu chuỗi mục tiêu là "HELLO" và mạng RNN dự đoán "H-EE-LL-O" (trong đó "-" là khoảng trắng), CTC sẽ gán nhãn cho chuỗi này và loại bỏ các khoảng trắng để so sánh với chuỗi mục tiêu.

Việc sử dụng CTC làm hàm mất mát giúp mô hình CRNN học cách căn chỉnh chính xác giữa đầu vào và đầu ra mà không cần phải có một sự căn chỉnh thủ công trước đó. Điều này đặc biệt quan trọng trong các bài toán nhận diện chữ viết tay, nơi mà mỗi mẫu chữ có thể có hình dạng và kích thước khác nhau.

3.3 Tham số khác

4 Đánh giá

Sau khi train model và test lại với bộ test, ta có kết quả như sau:

```
Correct characters predicted : 87.65%
Correct words predicted      : 75.11%
```

Hình 2: Kết quả độ chính xác

Kết quả test cho thấy model hiện tại đang hoạt động tương đối tốt. Đối với những test có độ dài quá lớn, model đang chưa thể hiện độ chính xác cao.

5 Kết luận

Trong báo cáo này, chúng em đã trình bày về bài toán nhận diện chữ viết tay và cách tiếp cận sử dụng mô hình Convolutional Recurrent Neural Network (CRNN) kết hợp với hàm mất mát Connectionist Temporal Classification

(CTC). Quá trình này bao gồm từ việc thu thập và tiền xử lý dữ liệu, thiết kế và huấn luyện mô hình, cho đến đánh giá kết quả đạt được.

Kết quả thử nghiệm cho thấy mô hình CRNN kết hợp với CTC đã đạt được hiệu suất cao trong việc nhận diện chữ viết tay, đặc biệt là trong việc xử lý các chuỗi ký tự có độ dài thay đổi và không yêu cầu phân đoạn trước các ký tự. Mô hình đã thể hiện khả năng mạnh mẽ trong việc nhận diện chính xác văn bản viết tay và có thể áp dụng trong nhiều ứng dụng thực tiễn như tự động hóa nhập liệu, phân loại tài liệu, và hỗ trợ người khuyết tật.

Tuy nhiên, cũng cần lưu ý rằng mô hình hiện tại vẫn chưa đạt độ chính xác cao đối với các chuỗi ký tự quá dài và phức tạp. Để cải thiện hiệu suất trong tương lai, có thể thực hiện một số biện pháp như sau:

1. **Tăng cường dữ liệu huấn luyện**: Thu thập thêm dữ liệu chữ viết tay đa dạng để mô hình có thể học và tổng quát hóa tốt hơn.

2. **Tối ưu hóa kiến trúc mô hình**: Thử nghiệm với các kiến trúc mạng khác nhau hoặc các kỹ thuật học sâu tiên tiến hơn để cải thiện hiệu suất.

3. **Điều chỉnh tham số**: Tinh chỉnh các tham số của mô hình để đạt được hiệu suất tối ưu.

4. **Sử dụng kỹ thuật học tăng cường**: Áp dụng các kỹ thuật như học tăng cường để cải thiện khả năng dự đoán của mô hình trong các tình huống khó.

Cuối cùng, nhận diện chữ viết tay là một lĩnh vực đầy thách thức và tiềm năng trong ngành trí tuệ nhân tạo và xử lý hình ảnh. Chúng em hy vọng rằng báo cáo này sẽ đóng góp vào việc phát triển và ứng dụng các công nghệ nhận diện chữ viết tay trong thực tiễn, giúp nâng cao hiệu quả và tự động hóa các quy trình xử lý văn bản.

Chúng em xin chân thành cảm ơn sự hướng dẫn và hỗ trợ từ các thầy trong quá trình thực hiện dự án này.