TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG TPHCM

UIT

Đồ án môn: Máy học

Chủ đề: Đồ án cuối kỳ

Link github: https://github.com/23520276/Hand-written-digit-classification

Giảng viên hướng dẫn:

Sinh viên thực hiện:

23520276 – Trương Hoàng Đạt

This page is intentionally left blank

Contents

Phần A: Handwritten Digits Classification

Chương 0: Những thay đổi so với vấn đáp:

Ở buổi vấn đáp, bài toán này em chỉ áp dụng model CNN cơ bản và xử lí ảnh đơn giản resize về 28x28, do đó lần này em đã thêm một số thay đổi sau:

- Thêm 2000 ảnh data cho mô hình huấn luyện – tổng 7800 tấm.
- Thêm minh họa một vài ảnh sau xử lí với một số model khi xử lí phức tạp.
- Thêm model random forest
- Thêm model SVM
- Thêm model resnet18 (best result)
- Ở mô hình Resnet, resize ảnh về kích cỡ lớn hơn 224x224 (thay vì 28x28 so với model khác) và chuẩn hóa ảnh RGB theo mean và giá trị lệch chuẩn.

Chương 1: Giới thiệu bài toán và các nhóm đã tham gia xây dựng dữ liệu:

- Giới thiệu bài toán: Bài toán nhận diện chữ số viết tay là một bài toán dạng Classification, nó là trong những bài toán kinh điển và cơ bản trong lĩnh vực Machine Learning. Bài toán yêu cầu máy tính phân loại một hình ảnh chứa một chữ số viết tay từ 0-9 thành một con số tương ứng.

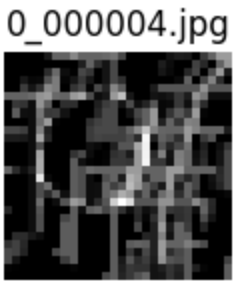
  + Input: là một ảnh chứa chữ số viết tay

  + Output: là nhãn tương ứng với chữ số viết tay đó.

- Dữ liệu được dùng trong huấn luyên chung: Toàn thể dữ liệu hình chụp chữ số viết tay được lấy từ Github của tất cả các bạn trong lớp CS114.P21 và 2000 tấm ảnh MNIST lấy từ thư viện pytorch.

Chương 2: Xử lí dữ liệu:

- Xử lý data cho model **Resnet**: Resize 224x224, normalize theo mean/std ImageNet, convert RGB
- Xử lý data cho model **CNN cơ bản**: Resize về 28x28, ảnh xám đảo màu nền đen chữ trắng để đồng bộ dữ liệu.
- Xử lý data cho model **Random Forest**: Ảnh xám, thực hiện resize về 100x100, sau đó tách chữ số riêng với ô ly rồi resize về 28x28. (Không hiệu quả, sau hậu xử lý ảnh nát như tương, chỉ có một số ảnh có thể nhìn thấy số nhưng có quá nhiều điểm nhiễu)



- Xử lý ảnh cho mô hình **SVM**: Ảnh xám, thực hiện resize về 100x100, sau đó tách chữ số riêng bằng cách làm mờ nền, chuẩn hóa, nhị phân hóa ảnh rồi đảo màu, sau đó tách chữ số ra và resize về 28x28. (Phương pháp này cũng không hiệu quả vì một số ảnh sau xử lý bị khuyết hoàn toàn vì nhầm lẫn chữ số viết tay là background nên bóc tách không thành công, còn ảnh bóc tách được số thì lại bóc tách dính nhiễu như ô ly)

Chương 3: Training.

Bảng thống kê:

Model	Tham số chính	Accuracy cuối	Số epoch	Training time
CNN cơ bản	Conv2D(32)-ReLU-MaxPool + Flatten + FC	66.10%	25	~30 phút
ResNet18	lr=0.0001, optimizer=Adam, batch=32	94.67%	10	~100 phút
Random Forest	n_estimators=100	59.57%	-	~1 phút
SVM	kernel='rbf', C=10, gamma= 0.01	55%	-	~1 phút

- **MÔ HÌNH CNN CƠ BẢN:** vì là mô hình đơn giản nên ta chọn 1 lớp tích chập 32 kernal 3x3, dụng hàm kích hoạt phi tuyến ReLU và giảm chiều, dễ chọn ra đặc trung nổi bật. Epoch=25 để cho mô hình hội tụ tốt hơn. Sau trainning model có accuracy train/test là 66.1% điều này cũng dễ đoán vì Ảnh đặc trưng sau xử lý quá nhỏ cũng như là có quá nhiều yếu tố nhiễu (thử nghiệm với 10 Epoch đã đưa ra con số 56% accuracy).
- **MÔ HÌNH RESNET18**: áp dụng learning rate 0.0001 để cập nhật trọng số chậm, đảm bảo mô hình sẽ có loss fuction tối ưu, dụng optimizer=Adam để hội tụ dễ hơn, cũng như là phù hợp khi xử lí dữ liệu ảnh thực, batch = 32 để tránh trand RAM. Sau trainning, mô hình đạt accuracy là 94.67%. Cho thấy mô hình học sâu và ổn định. Thời gian xấp xỉ 100 phút là một thời gian lâu vì mô hình phải đọc ảnh 224x224 là kích thước lớn hơn so với mô hình khác và cấu trúc của Resnet phức tạp.
- **MÔ HÌNH RANDOM FOREST:** n_estimators=100 giới hạn số cây trong rừng sao cho giảm sức nặng tính toán và giảm phương sai. Vì không định nghĩa max depth nên không giới hạn độ sâu giúp cho cây học đầy đủ các đặc trưng đầu vào. Sau training, cho ra 59.57%, con số này cho thấy model không học hiệu quả dựa trên ảnh bị biến dạng nhiều.
- **MÔ HÌNH SVM:** kernel='rbf' là Radial Basis Function kernel, còn gọi là Gaussian kernel. Với mục đích phân tách dữ liệu không tuyến tính (dữ liệu không thể tách bằng đường thẳng). Vì ảnh viết tay có độ cong, lệch, nhiễu nên rbf là lựa chọn phổ biến nhất. C=10C điều khiển mức độ nghiêm khắc khi xử lý lỗi phân loại. Càng lớn thì mô hình càng khó tha thứ cho lỗi, giúp tìm đường ranh phân chia sát dữ liệu nhất có thể. Chọn C=10 (cao hơn mặc định), nghĩa là ưu tiên độ chính xác trên tập huấn luyện, chấp nhận model phức tạp hơn. gamma=0.01 làm mượt ranh giới quyết định (decision boundary), tránh overfitting. Sau training, accuracy 55% cho thấy SVM hạn chế khi xử lí ảnh phức tạp.

Chương 4: Đánh giá và so sánh.

Bảng kết quả wecode: **Có thể theo dõi trên sheet "Thống kê submission" để nắm tên file trong github.*

Số thứ tự	Bài toán	Model	Score	ScoreScale%	Submit ID
1	Handwritten Classification	CNN cơ bản	2216/2939	75%	272946
2	Handwritten Classification (Large dataset)	CNN cơ bản	7410/9987	75%	272950
3	Handwritten Classification (Large dataset)	Random Forest	7776/9987	78%	274398

4	Handwritten Classification (Large dataset)	SVM	2216/2939	74%	272946
5	Handwritten Classification (Large dataset)	RESNET18	9370/9987	94%	272950
6	Handwritten Classification	RESNET18	2704/2939	92%	274398

- Qua kết quả trên, ta thấy được mô hình CNN cơ bản cũng như các mô hình khác như SVM hay RF đều không hiệu quả nếu ảnh đầu vào không đồng bộ, quá nhiều điểm nhiễu. Với tính đặc trưng của mô hình SVM và RF, nếu để size lớn hơn 28x28 sẽ gây thông tin nhiễu, dẫn đến mô hình học không hiệu quả. Còn về CNN, nếu là model cơ bản thì việc tăng size lớn hơn con số 28x28 cũng làm cho model không học hiệu quả mà có khi còn làm giảm hiệu quả, vì là mô hình đơn giản cơ bản nên không tận dụng tốt chi tiết thêm khi để ảnh quá lớn. Do mô hình nhóm sử dụng đều ở mức cơ bản (CNN, SVM, RF), nên ảnh được resize về 28x28 để tránh curse of dimensionality và phù hợp với năng lực mô hình. Chỉ có Resnet18 là deep CNN tới tận 18 layers khác với CNN cơ bản chỉ có 3 layers, do đó ảnh càng lớn thì mô hình này càng dễ dàng tận đụng các chi tiết thêm, do đó thời gian train thường khá lâu và ta không thể giảm size ảnh về 28x28 để model này train nhanh hơn được vì ảnh quá nhỏ khiến các tầng sau không có gì để học cũng như vì là model pretrained nên dẫn đến vi phạm input shape của model.

- Nhìn chung kết quả của từng model đều từ 74% trở lên, một phần là do dữ liệu thu thập dùng để train/test có thể trùng với dự liệu kiểm tra trên wecode, khiến cho accuracy (độ khoảng 60%) lại lệch trái hơn so với scorescale thực tế. Điều này chứng minh mô hình SVM, CNN cơ bản hay RF đều thật sự không ổn khi đưa ra ứng dụng thực tế khi con số khi train chỉ thể hiện từ trên dưới 60%. Duy chỉ có Resnet cho ra kết quả accuracy khi train/test so khớp với lại kết quả kiểm nghiệm trên wecode và đặc biệt nhận diện đúng đến 90% ảnh đầu vào.

- Từ bảng trên cũng cho thấy kết quả của RESNET18 cũng cho ra prediction trên 90%, đủ để cho thấy model này thật sự tốt trong việc nhận diện noise của ảnh, hiểu được những hình ảnh phức tạp, cộng với việc xử lí dữ liệu dễ dàng. Đây là lựa chọn sáng giá cho việc nhận diện chữ số viết tay.

Chương 5: Kết luận

Sau khi làm phần đồ án này, em đã nắm rõ hơn một pipeline của machine learning có thể được viết ra như thế nào, hiểu tốt hơn về cách xử lí dữ liệu đầu vào, nắm rõ khái niệm cũng như là yêu cầu của từng loại model khi train/test, độ hiệu quả của từng model khi được đem ra ứng dụng trên một bài toán thực tế. Qua đồ án này, em có thấy không có model nào là đúng tuyệt đối, chỉ có sai ít hoặc sai hết. Có lẽ khó nhất trong đồ án này là khâu chuẩn bị cũng như là xử lí dữ liệu, khi mà chỉ có 100 tấm

cũng đã mất 10 phút để chuẩn bị (labeled image) thì nói chi 10000 tấm, chưa kể những tấm ảnh này còn có yếu tố ngoại cảnh như: ánh sáng, orientation, độ phân giải, màu mực, giấy viết,... Khiến cho việc chuẩn hóa những tấm ảnh này đã khó càng thêm khó. Chưa kể nếu muốn model học tốt hơn thì 10000 tấm quả thật là không đủ. Lấy ví dụ như pretrained RESNET18, bản thân model tuy phức tạp nhưng đã được pre-train sẵn trên tập ImageNet 1,2 triệu tấm ảnh thật mới có thể đưa ra con số dự đoán đúng trên 90%, điều này đủ để khẳng định công đoạn chuẩn bị và xử lý dữ liệu là công đoạn khó nhất.

Phần B: Dự đoán điểm môn học từ kết quả nộp bài trên wecode:

Chương 0: Những thay đổi so với vấn đáp:

- Thêm đặc trưng (từ 5 lên 15)
- Thêm model Random-Forest.

Chương 1: Giới thiệu bài toán và giới thiệu dữ liệu:

- Giới thiệu bài toán: Bài toán dự đoán điểm môn học từ kết quả nộp bài trên wecode là bài toán thuộc dạng Regression, bài toán đưa vào các thông tin như tên của sinh viên và các thông tin liên quan đến kết quả nộp bài của sinh viên đó ví dụ như điểm, kết quả nộp, thời gian nộp,... và output là tên sinh viên đó với số điểm dự đoán tương ứng với sinh viên đó.

+ Input: Các đặc trưng của kết quả nộp bài.

+ Output: điểm sinh viên đó (làm tròn điểm 0,5)

- Giới thiệu dữ liệu: Dữ liệu gồm 4 file:
  + Annonimized.csv: Chứa thông tin của sinh viên đó (tên bị hash) và kết quả nộp bài tương ứng với sinh viên đó. Để miêu tả kết quả file gồm các cột: “id bài tập”, “id sinh viên”, “is_final” để xem bài mộp đó có phải là bài cuối cùng điểm cao nhất được nộp hay không, “status” trạng thái bài nộp, “coefficent, “id lớp”, “Thời gian bài tập được giao”, “Thời gian nộp”, “judgement” nêu đánh giá bài nộp.
  + th-public.csv: Chứa thông tin gần 800 sinh viên với tên bị hash được trích ra từ annonimized.csv và điểm thực hành tương ứng.
  + qt-public.csv và ck-public.csv suy ra tương tự như trên.

Chương 2: Xử lí dữ liệu:

- Đầu tiên là load file "annonimized.csv" bằng thư viện pandas, sau đó giữ lại một số cột quan trọng như "mã hash tên sinh viên", điểm, coeficient, is_final, thời gian, judgement và status để tính toán các đặc trưng.
- Các đặc trưng được tính toán bao gồm:
  + Điểm trung bình
  + Điểm cao nhất
  + Điểm thấp nhất
  + Độ lệch điểm
  + Trung vị điểm
  + Tổng số lần nộp bài
  + Số đề bài đã làm.
  + Coefficent's mean/max/median/std
  + Số lần nộp bị 0 điểm
  + Số lần nộp điểm 50 trên thang 100
  + Số lần nộp điểm 90 trên thang 100
  + Tỉ lệ điểm cao
  + Thời gian giải trung bình
  + Thời gian giải dài nhất
  + Số lần nộp bài có điểm
  + Số lần nộp muộn
- Sau đó join chung với file TH-public (qt-public hoặc ck-public) on hash và mã số sinh viên bị hash để lấy data train/test mô hình. Xử lý các cột không có dữ liệu bằng cách drop đi hoặc điền 0.
- Đối với giải quyết tính điểm trung bình do 3 file có chung tất cả các hash đều giống nhau nên hash join cả 3 file và tính điểm trung bình cả 3 file đó rồi xuất vào file "diem tb.csv"

Chương 3: Training

Bảng thống kê:

Model	**Tham số chính**	**R2 Score (lúc 5 features – lúc 15 features)**	**Thời gian huấn luyện**
Linear Regression	LinearRegression() của skit-learn	0.1986 - ~0.28	5 giây

Random Forest	n_estimators=100, max_depth=6	0.23 - 0.3695	5 giây

Chương 4: Đánh giá

Bảng thống kê: **Có thể theo dõi trên sheet "Thống kê submission" để nắm tên file trong github.*

Bài toán	Submit ID	Model	Score	ScoreScale%
Dự đoán điểm thực hành 1	273188	Linear Regression	1955	20%
Dự đoán điểm quá trình	273190	Linear Regression	195	2%
Dự đoán điểm cuối kì	274210	Linear Regression	905	9%
Dự đoán điểm trung bình	274211	Linear Regression	-19821	?
Dự đoán điểm thực hành 2	274209	Random Forest	1984	20%
Dự đoán điểm thực hành	274486	Thêm 10 đặc trưng + chạy Linear	2592	26%
Dự đoán điểm thực hành	274511	Thêm 10 đặc trưng + chạy Random Forest	2751	28%

Có thể nhận thấy vì dữ liệu điểm thực hành nó gắn liền với kết quả nộp thực hành nên kết quả dự đoán của điểm thực hành lại cao hơn so với dự đoán các cột điểm khác. Song xác suất dự đoán vẫn còn kém (cao nhất 26-28%), như vậy có thể thấy dữ liệu đầu vào không phụ thuộc tuyến tính với kết quả nộp thực hành. Khi thêm nhiều đặc trưng hơn, mô hình cũng dự đoán tốt hơn (LinearReg: từ 20% lên 26%) nhưng con số thay đổi không đáng kể. Còn RF, điểm R^2 tăng từ 0.23 lên 0.36 điểm, tuy nhiên khi điểm thay đổi khi nộp trên wecode cũng tăng từ 20% lên 28%, điều này càng khẳng định điểm dữ liệu phi tuyến tính.

Chương 5: Kết luận

Qua đồ án này, em rút ra đường các phương pháp trích xuất dữ liệu, tính đặc trưng dữ liệu, nắm rõ khái niệm cũng như yêu cầu của mỗi model khi train/test. Qua bài này cũng đủ thấy số lượng đặc trưng thực sự ảnh hưởng đến quá trình học của model. Cũng như là việc lựa chọn model sao cho phù hợp với bài toán.