TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
MÔN MÁY HỌC

BÁO CÁO ĐỒ ÁN:
PHÂN LỚP ẢNH CHỮ SỐ VIẾT TAY BẰNG SVM

Người thực hiện:
Huỳnh Hoàng Huy – 1612861
Trần Mạnh Thắng – 1612892

GVHD: Thầy Trần Trung Kiên

MỤC LỤC

I) TỔNG QUAN 3
a) Thông tin thành viên 3
b) Phân công công việc 3
II) CHI TIẾT 4
a) Mô tả bộ dữ liệu MNIST: 4
b) Cài đặt SVM: 4
c) Huấn luyện SVM: 4
c) Kết quả thực nghiệm trên bộ test 8
III) Tài liệu tham khảo 8

I) TỔNG QUAN

a) Thông tin thành viên

Tên sinh viên	Mã số sinh viên	Email
Huỳnh Hoàng Huy	1612861	huynhhoanghuy11111998@gmail.com
Trần Mạnh Thắng	1612892	thangblack11l21081998@gmail.com

b) Phân công công việc

Ngày	Công việc	Tên sinh viên	Hoàn thành
10/06 - 16/06	Lên kết hoạch	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành
	Xem video bài giảng: 14 – SVM, 15 – Kernel Methods, 16 – RBF	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành
	Tìm hiểu và trả lời các câu hỏi phần 1.1	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành
	Tìm hiểu các thư viện hỗ trợ: scikit-learn	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành
17/06 - 23/06	Viết khung sườn	Huỳnh Hoàng Huy	Đã hoàn thành
	Viết hàm run_model	Huỳnh Hoàng Huy	Đã hoàn thành
	Viết hàm SVM	Huỳnh Hoàng Huy	Đã hoàn thành
	Lưu và vẽ độ lỗi Viết hàm plot_rbf_kernel và plot_linear_kernel	Trần Mạnh Thắng	Đã hoàn thành
	Chạy thử các tham số C, gamma khác nhau trên Google colab và máy thật	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành
	Chỉnh một số lỗi cú pháp	Trần Mạnh Thắng	Đã hoàn thành

24/06 - 28/06	Đánh giá dựa trên kết quả tập validation, chọn tham số C và gamma phù hợp	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành
	Chạy thử tập Test và ghi nhận kết quả	Trần Mạnh Thắng	Đã hoàn thành
	Hoàn chỉnh source code và báo cáo	Huỳnh Hoàng Huy Trần Mạnh Thắng	Đã hoàn thành

II) CHI TIẾT

a) Mô tả bộ dữ liệu MNIST:

MNIST Handwritten Digits dataset: chứa hình ảnh viết tay của các chữ số {0, 1, …, 9} được lấy từ nhiều tài liệu được quét, được chuẩn hóa về kích thước 28x28 $pixel^2$.

Bộ dữ liệu chứa: 50.000 mẫu ở tập training, 10.000 mẫu ở tập validation và 10.000 mẫu ở tập test.

b) Cài đặt SVM:

Sử dụng các thư viện: scikit-learn, numpy, gzip, pickle, matplotlib, time.

Các hàm chính:

read_mnist: đọc file mnist và trả về các bộ dữ liệu: train_X, train_Y, val_X, val_Y, test_X, test_Y.

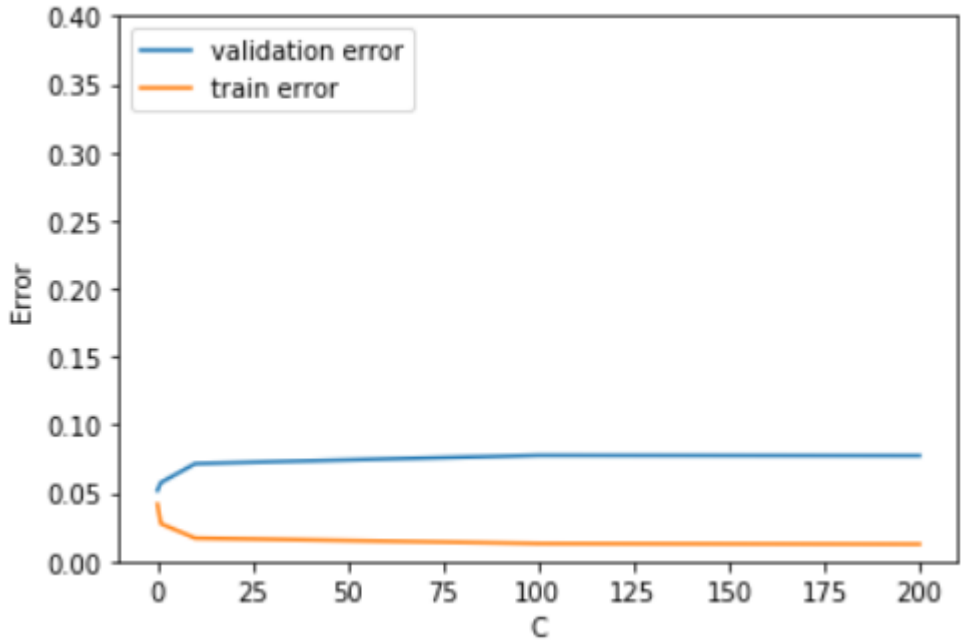
SVM: lần lượt chạy thuật toán svm trên kernel linear và rbf với các hệ số C và gamma được chọn. Trong quá trình chạy sẽ xuất ra độ lỗi, thời gian train và vẽ đồ thị biểu thị độ lỗi tương ứng của các cặp C và gamma.

c) Huấn luyện SVM:

- Dùng linear kernel:
  - Thử nghiệm với các tham số C khác nhau:

Tham số C	Độ lỗi training (%)	Độ lỗi validation (%)	Thời gian chạy (s)
C = 0.1	4.188	5.19	241
C = 1	2.754	5.77	293
C = 10	1.692	7.16	423
C = 100	1.3	7.77	1212
C = 200	1.256	7.75	2256

- Bình luận về kết quả:



Hình 2.1: Biểu đồ thể hiện sự biến thiên của độ lỗi trên tập validation và tập train ứng với mỗi giá trị C

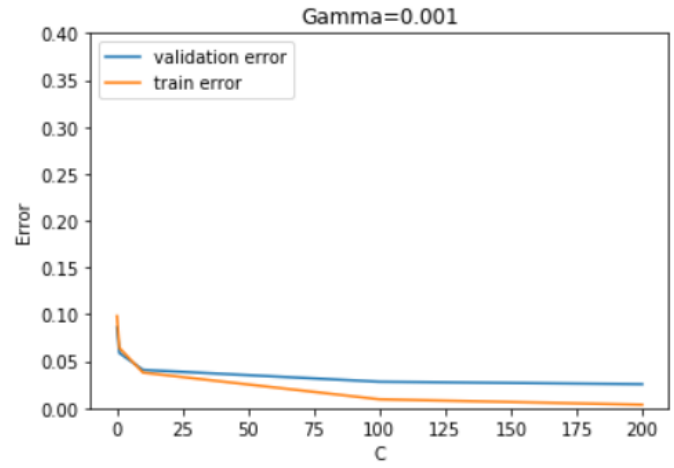
Khi tăng giá trị của C, độ lỗi trên tập train giảm dần, giống với lý thuyết. Tuy nhiên độ lỗi trên tập validation tăng lên, chứng tỏ mô hình với C quá lớn sẽ bị overfitting.

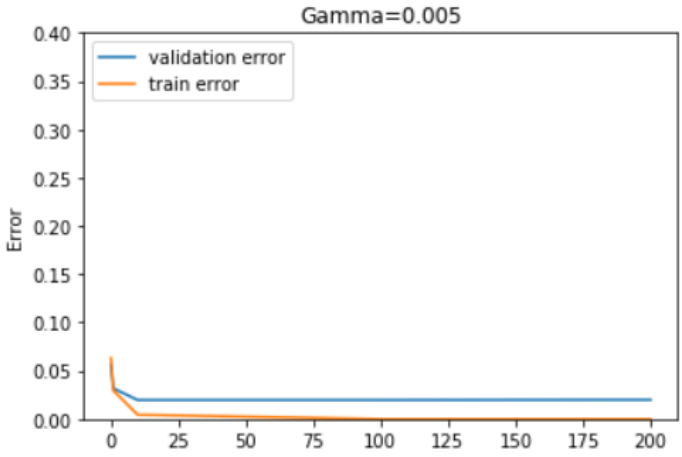
- Dùng Gaussian RBF kernel:
  - Thử nghiệm với các cặp tham số C và gamma khác nhau:

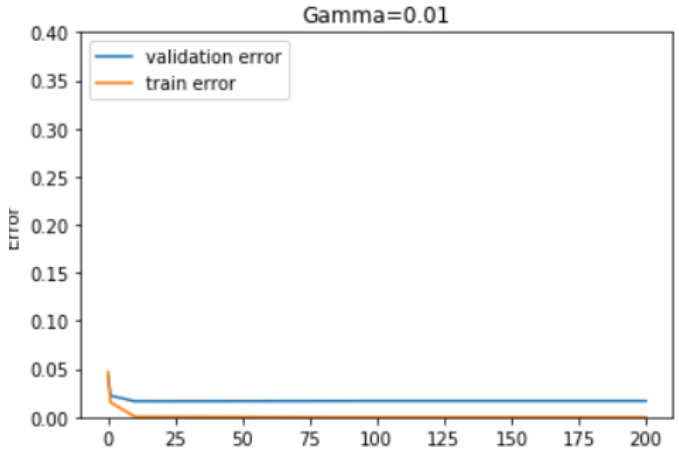
Các cặp tham số		Độ lỗi training (%)	Độ lỗi validation (%)	Thời gian chạy (s)
$\gamma = 0.001$	C = 0.1	9.824	8.61	1393
	C = 1	6.422	5.89	541
	C = 10	3.794	4.08	278
	C = 100	0.942	2.82	337
	C = 200	0.368	2.56	317
$\gamma = 0.005$	C = 0.1	6.32	5.61	1279
	C = 1	2.946	3.18	468
	C = 10	0.444	1.97	299
	C = 100	0	1.97	315
	C = 200	0	1.99	265
$\gamma = 0.01$	C = 0.1	4.702	4.22	896
	C = 1	1.526	2.23	373
	C = 10	0.058	1.65	292
	C = 100	0	1.68	277
	C = 200	0	1.68	337
$\gamma = 0.05$	C = 0.1	3.188	3.47	2249
	C = 1	0.084	1.79	1419
	C = 10	0	1.67	1245
	C = 100	0	1.67	1289
	C = 200	0	1.67	1302
$\gamma = 0.1$	C = 0.1	28.952	31.25	3702
	C = 1	0.004	4.48	5565
	C = 10	0	4.34	5749
	C = 100	0	4.34	5799
	C = 200	0	4.34	6270

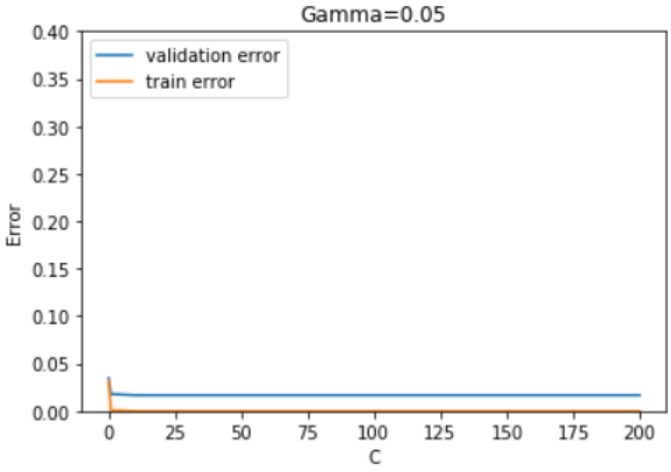
- Bình luận về kết quả:

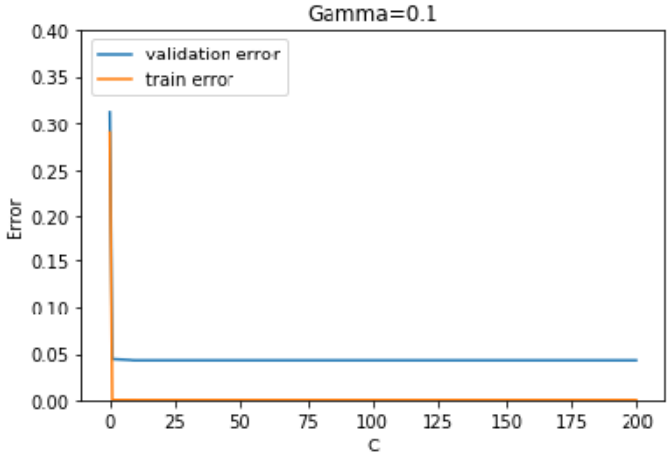
Dưới đây lần lượt là 5 biểu đồ biểu thị sự thay đổi độ lỗi trên tập validation và tập train ứng:











Với từng γ_i (với C_i vẫn giữ nguyên như ở trường hợp Linear kernel):

Cùng với một giá trị C, khi tăng giá trị γ độ lỗi trên tập train giảm dần. Tuy nhiên khi giá trị γ đến ngưỡng đủ lớn nhất định thì độ lỗi trên tập validation sẽ tăng, tức là mô hình bị overfitting, giống với lý thuyết.

Cùng một giá trị γ, khi ta tăng giá trị C, độ lỗi trên tập train giảm dần. Tuy nhiên đến một giá trị C đủ lớn nhất định độ lỗi trên tập validation tăng lên, chứng tỏ mô hình với C quá lớn sẽ bị overfitting, giống với lý thuyết.

c) Kết quả thực nghiệm trên bộ test

Dựa vào bảng số liệu thực nghiệm, ta thấy mô hình tốt nhất là:

kernel = "rbf", C = 10, $\gamma = 0.01$

```
Test with best model: kernel =  rbf , C =  10 , gamma =  0.01

Training

Running time:  344.5836730003357

Predicting

Calculate error

Test error rate:  1.7999999999999998 %
```

Ta thấy kết quả độ lỗi là: 1.8%, gần với giá trị trong document là 1.4%. Để đạt được giá trị tốt hơn ta có thể tăng giá trị C (trong khoảng từ 10-100) hoặc γ (trong khoảng từ 0.01-0.05) hoặc cả 2 để độ lỗi gần với giá trị 1.4% hơn.

III) Tài liệu tham khảo

[1] http://work.caltech.edu/lectures.html

[2] Machine Learning cơ bản site

[3] https://ongxuanhong.wordpress.com/

Hết