ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

MÁY HỌC THỐNG KÊ

Bài tập: Đồ án thực hành cuối kì

Cài đặt Neural Network

GVLT: Thầy Nguyễn Đình Thúc

GVTH Thầy Lương Việt Thắng

SV: Nguyễn Phan Mạnh Hùng 1312727

I/ Cài đặt mạng nơ-ron 1 lớp ẩn

1) Số lượng nút lần lượt trong các lớp là bao nhiêu? Vì sao?

- Do dữ liệu đầu vào là tập các ảnh grayscale có kích thước 28*28 = 784, và số lượng lớp là 10 (gồm các số từ 0-9), nên ta chọn kích thước lớp đầu và lớp cuối như sau (chưa bao gồm bias). Cụ thể: số lương nút từng lớp [784, x, 10]
- x càng lớn thì model sẽ học được nhiều features của input hơn. Tuy vậy, khi x quá lớn có thể dẫn tới tình trạng overfit và việc học đương nhiên cũng chậm hơn (do kích thước model lớn).

2) ***Mục tiêu của hàm softmax trong bài toán gán nhãn đa lớp***

- Hàm softmax là trường hợp tộng quát của hàm logistic khi số lớp lớn hơn 2
- Kết quả ứng với mỗi nút ouput của hàm softmax sẽ là xác suất mà một input thuộc lớp tương ứng (lưu ý: tổng các xác suất không nhất thiết bằng 1). Lớp được chọn sẽ là lớp có xác suất lớn nhất.

3) ***Thực nghiệm việc thay đổi số lượng nút của lớp hidden***

- l2 regularization: 0.0001
- Early stopping: 20 (nếu độ lỗi trên tập validation không được cải thiện sau 20 lần lặp thì sẽ dừng lại)
- Mini batch size: 10
- Learning rate (sigmoid & tanh): 0.1
- Learning rate (relu): 0.001

(Trong trường hợp sử dụng learning rate khi dùng hàm kích hoạt relu: nếu dùng quá lớn như sigmoid và tanh thì dẫn tới trường hợp bùng nổ gradient làm giảm độ chính xác của model. Lúc này có thể dẫn tới trường hợp xác suất mỗi lớp là như nhau và bằng 1. Để giải quyết trường hợp này, ta cần sử dụng một số phương pháp như batch normalization để scale input cho từng lớp về khoảng thích hợp)

Nhận xét: ở các bảng 1, 2, 3: ta nhận thấy khi số nút ở lớp ẩn tăng thì độ lỗi trên tập test cũng giảm tương ứng. Ban đầu thì độ lỗi giảm nhanh nhưng khi số nút ngày càng

lớn thì độ lỗi giảm chậm lại.

Tuy vậy, vẫn cần phải cẩn thận, bởi khi số lượng nút ẩn quá lớn có thể dẫn tới tình trạng overfit. Do đó cần phải sử dụng các kĩ thuật normalization như weight decay, dropout, ...

4) ***Thực hiện thay đổi các hàm kích hoạt***

Dựa vào bảng 1, ta thấy với x = 100 thì hàm tanh cho kết quả tốt nhất. Hàm relu cho kết quả tồi nhất và quan sát thấy độ lỗi trên cả 3 tập (train, validation, test) đều giảm chậm hơn so với 2 hàm còn lại.

Đặc điểm các hàm:

- Hàm tanh: đạo hàm lớn hơn hàm sigmoid (4 lần - 1-t^2(x) = 4*s(x)*(1-s(x)) > s(x)*(1-s(x)), do t(x) = 2*s(x) - 1). Điều này dẫn đếm hàm tanh hội tụ nhanh hơn so với sigmoid nếu có cùng các siêu tham số.
- Hàm relu: đạo hàm 0/1 - khá ổn định. Do đó cần cẩn thận khi thiết lập các tham số như learning rate (nên để nhỏ) để tránh việc tràn số do gradient quá lớn. Có thể thấy dù chỉnh learning rate nhỏ nhưng hàm relu hội tụ cũng khá nhanh. Điều đó cũng dẫn tới việc khó đạt tới cực tiểu do bước nhảy lớn dẫn tới quá trình học không ổn định. Cần phải có các biện pháp phụ trợ để normalize lại giá trị như batch normalization.

x	**1**	**2**	**5**	**15**	**20**	**30**	**40**	**50**	**100**
Sigmoid	65.63	34.58	10.22	4.35	4.56	2.95	2.77	2.81	2.46
Tanh	72.84	57.43	12.04	5.07	4.64	3.7	2.82	2.47	2.34
Relu	67.31	67.78	12.03	8.37	6.21	6.07	5.55	4.82	4.59

1 Bảng độ độ lỗi trên tập test

x	**1**	**2**	**5**	**15**	**20**	**30**	**40**	**50**	**100**
Sigmoid	64.23	33.94	9.69	4.38	3.91	2.91	2.64	2.49	2.25
Tanh	72.21	57.57	11.89	4.55	4.07	3.12	2.86	2.6	1.91
Relu	67.7	68.02	11.26	7.9	5.85	5.46	4.75	4.63	3.98

2 Bảng độ lỗi trên tập validation

x	1	2	5	15	20	30	40	50	100
Sigmoid	65.006	34.506	10.034	3.82	3.444	2.114	1.824	1.584	1.43
Tanh	72.688	57.578	11.164	3.682	2.972	1.87	0.996	0.752	0.424
Relu	67.47	67.998	12.598	8.656	6.216	6.002	5.256	4.862	4.146

3 Bảng độ lỗi trên tập train

II/Cài đặt mạng nơ-ron đa lớp.

Mạng càng nhiều lớp thì ta càng có khả năng biểu diễn nhiều tính chất của dữ liệu nhưng ngược lại việc học sẽ chậm, khó, và mô hình dễ bị overfit.

Sau đây là 1 số yếu tố gây cản trở cho việc học:

- Gradient vanishing: hiện tượng này xảy ra làm cho các lớp càng gần input học càng chậm, do đạo hàm tính được khi lan truyền ngược tới đây ngày càng nhỏ dần.
- Gradient exploding: trong vài trường hợp, thậm chí gradint này sẽ rất lớn dẫn tới quá trình học mất ổn định gây underfit.

Chính bởi tốc độ học rất khác của mỗi lớp khi xây dựng mạng đa lớp làm cho việc học trở nên khó khăn hơn khi mạng nhiều lớp nếu chỉ dùng các phương pháp học thông thường.

Layer	Test error
[784, 50, 30, 10]	2.63 %
[784, 50, 30, 20, 10]	2.78 %
[784, 50, 30, 20, 20, 10]	2.64 %

4 Sigmoid activation

Quan sát bảng 4 nhận thấy, dù độ lỗi khá thấp nhưng kết quả của các mô hình này vẫn chưa tốt như mô hình 3 lớp (1 lớp ẩn)