Computing the Shattering coefficient

CHEN Hang

Idea1:

From the architecture of the given question, we need more than one hyperplane to resolve it. So at the base of the code provided during the first two classes, we add a loop whose loop time is the number needed hyperplanes to get the result of each hyperplane and then combine it into a final result.

For the Hidden1 Layer(R=7, hyperplane=9), we have:

Sample size	Number of different classifications
1	428
2	3973
3	9000
4	16000
5	25000
...	...

For the Hidden2 Layer(R=9, hyperplane=5), we have:

Sample size	Number of different classifications
1	32
2	975
3	7204
4	15674
5	24905
6	35985
7	48999
8	64000
...	...

For the Output Layer(R=5, hyperplane=3), we have:

Sample size	Number of different classifications
1	8
2	64
3	499
4	3347
5	9435
6	22549
7	37816
8	57144
9	75467
10	91154
...	...

So from the result table, we can know that with the sample size growing, the result has more and more loss of precision. What's more, the running time is too long to get enough data to compute the regression function and there will be a little big error for the final result.

Idea2:

The second method is that we can get the result of one hyperplane to estimate the result with multi-hyperplanes. For example, if we want to compute the number of different classifiers with 7 dimensions input and 9 hyperplanes, firstly we compute the result of 7 dimensions input and one single hyperplane, then making the result pow 9, the final result is what we want.

For the Hidden1 Layer(R=7, hyperplane=9), we have:

When R=7 for one hyperplane:

Sample size	Number of different classifications
1	2
2	4
3	8
4	16
5	32
6	64
7	120
8	241
9	433
10	741
11	1393
12	2154
13	3384
14	4487
15	6455
16	11287
17	14034
18	17555
19	22830
20	31559
…	…

x^7	x^6	x^5	x^4	x^3
2.935799e-03	-2.062594e-01	5.764196e+00	-8.122488e+01	6.122427e+02
x^2	**x**	**Intercept**		
-2.390422e+03	4.274969e+03	-2.516417e+03		

So the final result is that the value above power 9.

For the Hidden2 Layer(R=9, hyperplane=5), we have

When R=9 for one hyperplane:

Sample size	Number of different classifications
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	244
9	458
10	904
11	1591
12	2932
13	4506
14	5598
15	10150
16	15446
17	23860
18	35204
19	41764
20	51008
...	...

x^9	x^8	x^7	x^6	x^5
2.205742e-04	-2.032948e-02	7.855647e-01	-1.656123e+01	2.077486e+02
x^4	**x^3**	**x^2**	**x**	**Intercept**
-1.586478e+03	7.249466e+03	-1.868711e+04	2.393603e+04	-1.115884e+04

So the final result is that the value above power 5.

For the Output Layer(R=5, hyperplane=3), we have:

When R=5 for one hyperplane:

Sample size	**Number of different classifications**
1	2
2	4
3	8
4	16
5	32
6	63
7	110
8	212
9	341
10	572
11	930
12	1218
13	2018
14	2510
15	3609
16	4239
17	6230
18	8121
19	10304
20	12787
...	...

x^5	x^4	x^3	x^2
-0.001288655	0.223546380	-3.577046224	29.186067194
x	**Intercept**		
-96.110864756	91.850257998		

So the final result is that the value above power 3.